Credit Card Fraud Detection Using Machine learning

Prem Pednekar(PE-46)
Department of Computer Engineering and Technology
1032201777

Satyam Singh (PB-26)
Department of Computer Engineering and Technology
1032201422

Mohit Bharadwaj (PE-20)
Department of Computer Engineering and Technology
1032201202

Sujal Dhanavade(PE-33)
Department of Computer Engineering and Technology
1032201628

Shubham Jadhav(PE-17)
Department of Computer Engineering and Technology
1032201143

Abstract

As we know credit cards are getting progressively more famous in the economic world, meanwhile on the other hand frauds like phishing, skimming, cloning etc. are also increasing the same. It has become a serious menace. Machine Learning (ML) is beneficial for building a rational model to detect fraudulent transactions. While dealing with the high-dimensional and imbalanced dataset which grows into hindrance to the real world applications like credit card fraud detection. For this end, it is obligatory for financial institutions to continuously improve their fraud detection system to reduce the huge losses. To overcome this we require certain pre-processing techniques to be adopted considering the classification performance and computational efficiency. The purpose of this paper is to develop a novel system for credit card fraud detection, first is to balance the highly imbalanced dataset using Adaptive Synthetic (ADASYN) algorithm furthermore the machine learning approach Neural Networks (Keras). This involves pattern classification in an unbalanced dataset to determine the fraudulent transactions. Then another approach based on balancing data using synthetic minority oversampling technique, after that sequential modelling of data, using attention mechanism and LSTM deep recurrent neural networks.
The experimentations of our model give strong results in terms of efficiency and effectiveness.

Key words: Deep Learning, neural Networks, fraud detection, Sequence learning, ADASYN, LTSM

Introduction

A transaction is a completed agreement between a buyer and a seller to exchange goods, services, or financial assets in return for money. The term is also commonly used in corporate accounting.
A credit card is a thin handy plastic card that contains identification information such as signature or picture and authorises the person named on it to charge purchases or service to his account - charges for which he/she will be billed periodically. Today, the information on the card is read by automated teller machines (ATMs), store readers, bank, scanning machines and is also used in online internet banking systems.

Every card holder has a unique card number which is of utmost priority. Its security relies on the physical security of the plastic card as well as the privacy of the credit card number.
Credit card fraud is a form of identity theft in which criminals make purchases or obtain cash advances using a credit card account assigned to you.
With the rapid growth in the number of credit card transactions which has also led to substantial rise in fraudulent transactions. This type of fraud is a wide ranging fraud for theft and fraud committed using a credit card as fraudulent source of funds in a particular transaction. The detection of credit card fraud has recently spread due to increased fraud that can be described as a deliberate tactic move played to achieve a kind of gain, usually based on monetary gain. We know that it is an unfair practice that progressively increases day after day. Internet fraud may include spam, scams, spyware, identity theft, phishing or internet banking fraud. Most of the credit card fraud detection systems are based on artificial intelligence (AI), meta learning and pattern matching. The genetic algorithms are evolutionary algorithms which aim to obtain better solutions in eliminating fraud. We should give high importance to developing an efficient and secure electronic payment system to detect whether a transaction is fraudulent or not.

A genetic algorithm generates better solutions as time progresses. The complete emphasis is given on developing efficient and secure electronic payment systems for detecting the fraudulent.

Data Collection

The dataset has been collected and analysed during a research collaboration of Worldline and the Machine learning Group (https://mlg.ulb.ac.be) of ULB on Big Data mining and fraud detection.
We have collected this dataset from kaggle. (https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud). The dataset contains transactions made by credit cards in September 2013 by European cardholders. This dataset presents the transactions that occurred in two days, where we have 492 frauds out of 284,807 transactions. The dataset is highly unbalanced, the positive class (frauds) account for 0.172% of all transactions.

It contains only numeric input variables which are the result of a PCA transformation. Unfortunately, due to confidentiality issues, they didn't provide the original features and more background information about the data. Features from V1 - V8 are the principal features obtained with PCA, the only features which have been transformed with PCA are 'Time' and 'Amount'. Feature time contains seconds elapsed between each transaction and the first transaction in the dataset. Feature amount contains the transaction amount. Feature class is the response variable and it takes value 1 in case of fraud and 0 otherwise

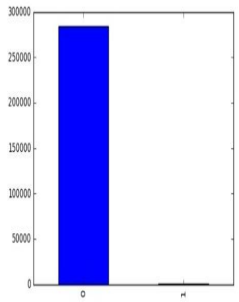
PCA Transformation

Principal Component Analysis (PCA) is a statistical procedure that uses an orthogonal transformation that converts a set of correlated variables to a set of uncorrelated variables. PCA is the most widely used tool in exploratory data analysis EDA and in machine learning for predictive models. It is a technique to draw strong patterns from the given dataset by reducing the variances

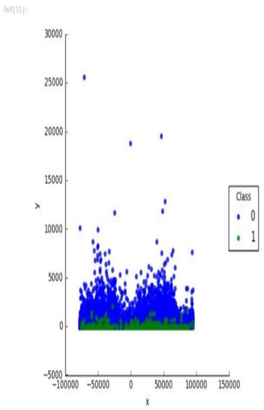
Module required for PCA are:
Pandas, numpy, matplotlib, seaborn

PCA generally tries to find the lower-dimensional surface to project the high-dimensional data. It works by considering the variance of each attribute because the high attribute shows the good split between the classes and hence it reduces dimensionality. It is based on some mathematical concepts like variance-covariance and eigenvalues-eigen factors.

Some properties of these principal components are
The principal component must be the linear combination of the original features.
These components are orthogonal
The importance of each component decreases when going 1 to n, which says that 1 principal component has most importance and n PC will have the least importance.



This gives a visual representation of the class imbalance. The below plot after PCA gives a better visualization of the imbalnce in the datasets.



Literature Survey

Sujal

Paper Name	Dataset	Algorithm	Evaluation Index
Detecting Credit Card Fraud using Machine Learning	European dataset, Australian dataset, German dataset	CNN, AE, LSTM, and AE&LSTM	AUC value
A two-phase feature selection technique using mutual information and XGBRFE for credit card fraud detection	IEEE-CIS fraud detection dataset	XGBoost, GBM, CatBoost, LGBM	AUC value
Enhanced credit card fraud detection based on attention mechanism and LSTM deep model	European cardholders from Kaggle, BankSim dataset	Attention mechanism and LTSM deep recurrent networks	Precision and Recall

Mohit

Paper Name	Dataset	Algorithm	Evaluation Index	Evaluation index value
Predictive Modelling For Credit Card Fraud Detection Using Data Analytics	Benchmark Dataset	Logistic regression, Decision tree,Random forest decision tree	Confusion matrix-Accuracy	Logistic regression-72%, Decision tree-72%, Random forest-76%
Credit card transaction fraud from a real world example	Random Unbalanced	Logistic model (regression) Support vector machines Random forests	Accuracy	96.6-99.4% 95.5-99.6% 97.8-99.6%
Financial statement fraud with managerial statements for US companies	Random Unbalanced	Text mining and decision tree hybrid Text mining and Bayesian belief network hybrid Text mining and support vector machine hybrid	Accuracy	67.3% 67.3% 65.8%

Satyam

Paper Name	Dataset	Algorithm	Evaluation Index	Evaluation index value
Ensemble Techniques for Credit Card Fraud Detection	European dataset	Logistic Regression, Naïve Bayes, Support Vector Machine (SVM), K-Nearest Neighbour (KNN), and Decision Tree,XGBOOST	Confusion Matrix - Precision - Recall - F1-score - AUC score	➢ For Logistic Regressi on Classifier 0.95653 0.9818 6 0.93029 0.98279 0.955 38 0.956 54 ➢ For XGBOO ST Classifier 0.99983 0.9996 6 1 0.99966 0.999 83 0.999 83
Credit Card Fraud Detection with Autoencoder and Probabilistic Random Forest	Data generated by European cardholders within 2 days	Autocender And Random Forest	Accuracy (ACC), the true positive rate (TPR), the true negative rate (TNR), and the false positive rate (FPR)	➢ The AUC of the ROC curve is 0.962, which is better than 0.960 of the method proposed Earlier ➢ AE-PRF for θ = 0.25, which yields the best average MCC
Emerging Approach for Detection of Financial Frauds Using Machine Learning	➢ Data Description of Credit Card Fraud Detection Model ➢ Data Description of Default Loan Prediction Model ➢ Data Description of Bankruptcy Detection Model	Random forest, Decision Tree, Logistic Regression and Gaussian Naive Bayes mode	Accuracy , F1 Score And MSE	Random Forest -- 0.999052 0.874699 0.000948 Decision Tree 0.999192 0.874510 0.000808

Shubham

Paper Name	Dataset	Algorithm	Evaluation Index	Evaluation index value
Fraud Detection using Machine Learning and Deep Learning	European dataset, Australian dataset, German dataset	KNN,SVM	AUC value	KNN: 88.87% SVM:90.07%
Detecting Financial Statement Fraud with Interpretable Machine Learning	Teddy database	XGBoost	AUC value	83.61%
Fraud Detection in Payments Transactions: Overview of Existing Approaches and Usage for Instant Payments	Credit card fraud detection	Banksealer and Random forest	Precision and Recall	Random forest: 0.987 Banksealer: 0.979

Prem

Paper Name	Data Set	Algorithm	Evaluation Index	Evaluation Index Value
Credit Card Fraud Detection using Machine Learning	European dataset	Decision Tree, XGboost, random forest	Accuracy score, classification report, F1-score, confusion matrix	ACCURACY SCORE Accuracy score of the Decision Tree model is 0.9993679997191109 Accuracy score of the KNN model is 0.9983146659176293 Accuracy score of the Logistic Regression model is 0.9989993328885 Accuracy score of the Random Forest Tree model is 0.99933288859239
Credit Card Fraud Detection System Using Machine Learning	Random Imbalanced	*Decision tree model, Training Support Vector Machine (SVM) , Artificial Neural Network (ANN)*	Accuracy, F Score ,Recall,Specificity , Precision, Misclassification Rate	HCCFD Accuracy 93.3% Misclassification rate 6.7% Recall 89.3% Specificity 98.0% Precision 98.1% F score 93.5%
Credit Card Fraud Detection using Machine Learning and Data Science	Random Imbalanced	Local Outlier Factor , Isolation Forest Algorithm	Accuracy score, F1-score,Macro avg, Precision.	Isolation Forest Number of Errors: 71 Accuracy Score: 0.99750711000316 precision recall f1-score support 0 1.00 1.00 1.00 28432 1 0.28 0.29 0.28 49 accuracy 1.00 28481 macro avg 0.64 0.64 0.64 28481 weighted avg 1.00 1.00 1.00 28481 Local Outlier Factor Number of Errors: 97 Accuracy Score: 0.9965942207085425 precision recall f1-score support 0 1.00 1.00 1.00 28432 1 0.02 0.02 0.02 49 accuracy 1.00 28481 macro avg 0.51 0.51 0.51 28481 weighted avg 1.00 1.00 1.00 28481

ADASYN

It's an improved version of Smote. What it does is the same as SMOTE just with a minor improvement. After creating those samples it adds random small values to the points thus making it more realistic.

ADASYN is based on the idea of adaptively generating minority data samples according to their distributions: more synthetic data is generated for minority class samples that are harder to learn compared to those minority samples that are easier to learn. The ADASYN method can not only reduce the learning bias introduced by the original imbalance data distribution, but can also adaptively shift the decision boundary to focus on those difficult to learn samples.

The key idea of ADASYN algorithm is to use a density distribution rˆi as a criterion to automatically decide the number of synthetic samples that need to be generated for each minority data example. Physically, rˆi is a measurement of the distribution of weights for different minority class examples according to their level of difficulty in learning. The resulting dataset post ADASYN will not only provide a balanced representation of the data distribution (according to the desired balance level defined by the β coefficient), but it will also force the learning algorithm to focus on those difficult to learn examples. This is a major difference compared to the SMOTE algorithm, in which equal numbers of synthetic samples are generated for each minority data example. Our objective here is similar to those in SMOTEBoost and DataBoost-IM algorithms: providing different weights for different minority examples to compensate for the skewed distributions. However, the approach used in ADASYN is more efficient since both SMOTEBoost and DataBoost-IM rely on the evaluation of hypothesis performance to update the distribution function, whereas our algorithm adaptively updates the distribution based on the data distribution characteristics. Hence, there is no hypothesis evaluation required for generating synthetic data samples in our algorithm. Fig. 1 shows the classification error performance for different β coefficients for an artificial two-class imbalanced data set. The training data set includes 50 minority class examples and 200 majority class examples, and the testing data set includes 200 examples. All data examples are generated by multidimensional Gaussian distributions with different mean and covariance matrix parameters. These results are based on the average of 100 runs with a decision tree as the base classifier. In Fig. 1, $\beta = 0$ corresponds to the classification error based on the original imbalanced data set, while $\beta = 1$ represents a fully balanced data set generated by the ADASYN

Conclusion of ADASYN

In this paper, we propose a novel adaptive learning algorithm ADASYN for imbalanced data classification problems. Based on the original data distribution, ADASYN can adaptively generate synthetic data samples for the minority class to reduce the bias introduced by the imbalanced data distribution. Furthermore, ADASYN can also autonomously shift the classifier decision boundary to be more focused on those difficult to learn examples, therefore improving learning performance. These two objectives are accomplished by a dynamic adjustment of weights and an adaptive learning procedure according to data distributions. Simulation results on five data sets based on various evaluation metrics show the effectiveness of this method. Imbalanced learning is a challenging and active research topic in artificial intelligence, machine learning, data mining and many related areas. We are currently investigating various issues, such as multiple classes imbalanced learning and incremental imbalanced learning. Motivated by the results in this paper, we believe that ADASYN may provide a powerful method in this domain.

Before Sampling (Unbalanced Data)

```
[ ] #checking the percentage of each class in the dataset
    (creditcard_data.Class.value_counts())/(creditcard_data.Class.count())

    0    0.998273
    1    0.001727
    Name: Class, dtype: float64
```

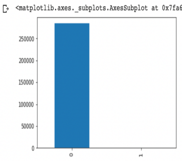
```
[ ] #number of instances per class
    creditcard_data.Class.value_counts()

    0    284315
    1       492
    Name: Class, dtype: int64
```

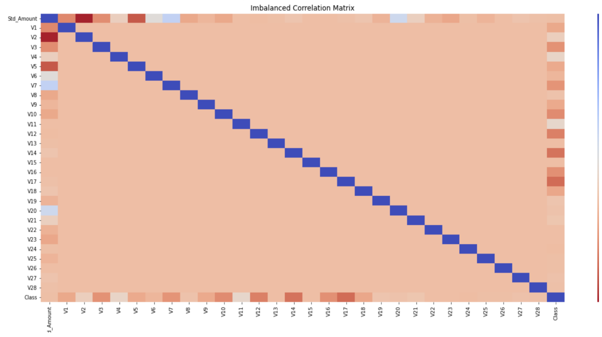
This shows a complete imbalance of classes. There are 284315 'Genuine' (0) instances and only 492 'Fraudulent' (1) instances.

```
#visual representation of instances per class
creditcard_data.Class.value_counts().plot.bar()

<matplotlib.axes._subplots.AxesSubplot at 0x7fa6b2818450>
```




```
Text(0.5, 1.0, 'Imbalanced Correlation Matrix')
```



After Sampling (Using ADASYN)

```
#Using ADASYN for Oversampling
ada = ADASYN(sampling_strategy='minority', random_state=42)

#Oversampling is applied only on the training set
X_adasampled, Y_adasampled = ada.fit_resample(Xtrain_final, Ytrain_final)
print('Resampled dataset shape %s' % Counter(Y_adasampled))
print('Shape of X_adasampled: {}'.format(X_adasampled.shape))
print('Shape of Y_adasampled: {}'.format(Y_adasampled.shape))
```

```
Resampled dataset shape Counter({1: 170555, 0: 170554})
Shape of X_adasampled: (341109, 29)
Shape of Y_adasampled: (341109,)
```

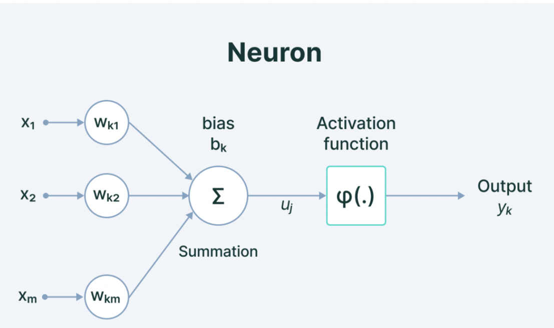
```
#check the disribution of both the labels
train_label, train_count = np.unique(Y_adasampled, return_counts=True)
print('Label Distributions: \n')
print(train_count/ len(Y_adasampled))
```

```
Label Distributions:

[0.49999853 0.50000147]
```

Neural Networks

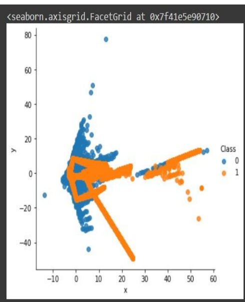
A neural network is a method in artificial intelligence that teaches computers to process data in a way that is inspired by the human brain. It is a type of machine learning process, called deep learning, that uses interconnected nodes or neurons in a layered structure that resembles the human brain. It creates an adaptive system that computers use to learn from their mistakes and improve continuously. Thus, artificial neural networks attempt to solve complicated problems, like summarising documents or recognizing faces, with greater accuracy.



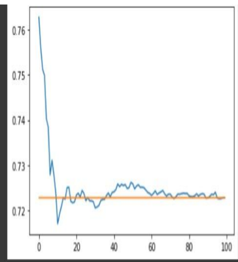
Neural networks can help computers make intelligent decisions with limited human assistance. This is because they can learn and model the relationships between input and output data that are nonlinear and complex. For instance, they can do the following tasks.

Neural network training is the process of teaching a neural network to perform a task. Neural networks learn by initially processing several large sets of labelled or unlabeled data. By using these examples, they can then process unknown inputs more accurately.

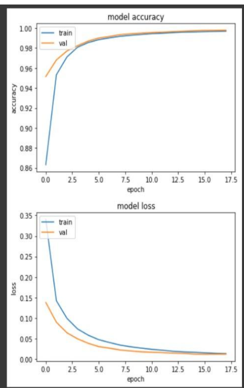
```
<seaborn.axisgrid.FacetGrid at 0x7f41e5e90710>
```



The above plot gives a visual representation of the balanced dataset. As seen, the data is not linearly separable. Hence, a Multi-Layer Perceptron is trained on this data, in order to predict the data as 'fraud' or 'genuine'. For the training of the Multi-layer perceptron, many parameters need to be tuned, thus a Grid-Search is used for this purpose. The Grid-Search is performed on the subset of the data, as it takes a lot of time to run. The following blocks create a subset of the dataset using random undersampling used only for this purpose.



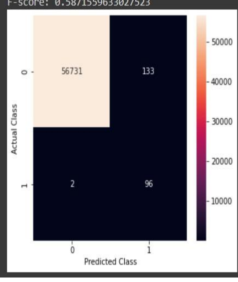
As seen in the plot above, the average of F-score is getting stabilized after 60 iterations. Thus, the sample size sufficiency is taken as 60, which means that the model will be trained for 60 repetitions and the overall average of the score from these iterations will determine the skill of the model.

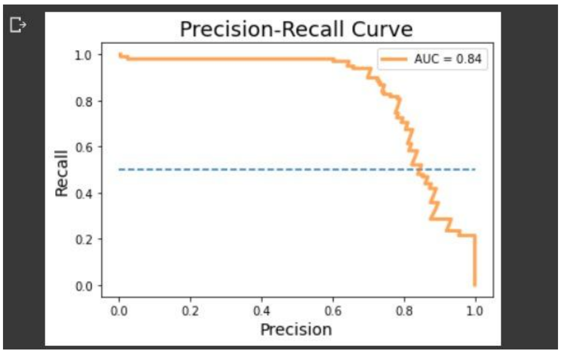


As observed, with the addition of Dropout, the validation dataset performs better. And the train and vaidation set performance is comparable.

```
Y_pred_cls = (Model1_drop.predict(Xtest_arr)>0.5).astype("int32")
print('Accuracy Model1 (Dropout): '+ str(Model1_drop.evaluate(Xtest_arr,Ytest_arr)[1]))
print('Recall_score: ' + str(recall_score(Ytest_arr,Y_pred_cls)))
print('Precision_score: ' + str(precision_score(Ytest_arr, Y_pred_cls)))
print('F-score: ' + str(f1_score(Ytest_arr,Y_pred_cls)))
conf_matrix(Y_pred_cls)
```

```
1781/1781 [==============================] - 2s 1ms/step
1781/1781 [==============================] - 2s 1ms/step - loss: 0.0095 - accuracy: 0.9976
Accuracy Model1 (Dropout): 0.9976300001144409
Recall_score: 0.9795918367346939
Precision_score: 0.4192139737991266
F-score: 0.5871559633027523
```





Supervised learning

In supervised learning, data scientists give artificial neural networks labelled datasets that provide the right answer in advance. For example, a deep learning network training in facial recognition initially processes hundreds of thousands of images of human faces, with various terms related to ethnic origin, country, or emotion describing each image. The neural network slowly builds knowledge from these datasets, which provide the right answer in advance. After the network has been trained, it starts making guesses about the ethnic origin or emotion of a new image of a human face that it has never processed before.

Confusion Matrix

Algorithm	True Positive	False Positive	True Negative	False Negative
Neural Networks	56731	133	96	2

Comparison Table

Algorithm	Accuracy	Precision	Recall
Neural Networks	99.7%	56%	94%
LSTM	96.7%	98.85%	91.9%

In the credit card fraud domain, fraud detection systems try to reduce the false positive and false negative rate, knowing that the latter (FN) has severe costs on financial institutions as well as a decrease in customer satisfaction. To assess the performance of our proposed fraud detection system with more accuracy, we use the confusion matrix

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN},$$

$$Sensitivity = \frac{TP}{TP + FN},$$

$$Specificity = \frac{TN}{FP + TN},$$

$$Precision = \frac{TP}{TP + FP}.$$

Accuracy

It's the ratio of the correctly labeled subjects to the whole pool of subjects.
Accuracy is the most intuitive one.
Accuracy answers the following question: How many students did we correctly label out of all the students?
Accuracy = (TP+TN)/(TP+FP+FN+TN)
numerator: all correctly labeled subject (All trues)
denominator: all subjects

Precision

Precision is the ratio of the correctly +ve labeled by our program to all +ve labeled.
Precision answers the following: How many of those who we labeled as fraud are actually fraud?
Precision = TP/(TP+FP)
numerator: +ve labeled fraud transactions.
denominator: all +ve labeled by our program.

Recall

Recall is the ratio of the correctly +ve labeled by our program to all fraud transactions in reality.
Recall answers the following question: Of all the transactions which are fraudlent, how many of those we correctly predict?
Recall = TP/(TP+FN)
numerator: +ve labeled fraud transactions.
denominator: all fraud transactions

F1-score

F1 Score considers both precision and recall.
It is the harmonic mean(average) of the precision and recall.
F1 Score is best if there is some sort of balance between precision (p) & recall (r) in the system.

Oppositely F1 Score isn't so high if one measure is improved at the expense of the other.
For example, if P is 1 & R is 0, F1 score is 0.
F1 Score = 2*(Recall * Precision) / (Recall + Precision)

True positives (TP) are cases classified as positive which are actually positive. True negative (TN) are cases classified rightly as negative. False positive (FP) are cases classified as positive but are negative cases. False negative (FN) are cases classified as negative but are truly positive. Specificity gives the accuracy on negative (legitimate) cases classification. Precision gives the accuracy in cases classified as fraud (positive) and sensitivity (Recall) gives the accuracy on positive (fraud) cases classification.

Conclusion

In this paper, we aimed to improve the prediction efficiency during the identification of fraudulent transactions, by combining the strength of different Machine Learning techniques. Method to reduce the dataset dimensionality, the ADASYN to overcome the problem of imbalanced data. Thus, our proposed model is capable of catching useful patterns within consumer behaviour which helps to distinguish effectively fraudulent transactions from the normal ones. To compare our results, we performed two different models,one on LSTM and other on Neural Networks. It shows its ability to deliver a high sensitivity performance during the detection of fraudulent instances that are of great interest in this domain. Furthermore, in terms of comparison with recent works, our model provides a very good performance.

Future Work

Future work will also include implementing the system by using neural networks to train the system for increasing efficiency. Having a data set with non-anonymized features would make this particularly interesting as outputting the feature importance would enable one to see what specific factors are most important for detecting fraudulent transactions.

References

1.ACFE. Report to the nations 2018 global study on occupational fraud and abuse. 2019. https://doi.org/10.1002/ 9781118929773.oth1.
2.Carcillo F, Le Borgne Y-A, Caelen O, Bontempi G. Streaming active learning strategies for real-life credit card fraud detection: assessment and visualization. Int J Data Sci Anal. 2018. https://doi.org/10.1007/s41060-018-0116-z.
3.Chandola V, Banerjee A, Kumar V. Anomaly detection for discrete sequences: a survey. IEEE Trans Knowl Data Eng. 2012;24:823–39.
4.Popat RR, Chaudhary J. A survey on credit card fraud detection using machine learning. In: Proceedings of the 2nd international conference on trends in electronics and informatics, ICOEI 2018; 2018. https://doi.org/10. 1109/ICOEI.2018.8553963.
5.Zafar A, Sirshar M. A survey on application of Data Mining techniques; it's profciency in fraud detection of credit card. Res Rev J Eng Technol. 2018;7:15–23.
6.Kültür Y, Çaglayan MU. Hybrid approaches for detecting credit card fraud. Expert Syst. 2017. https://doi.org/10. 1111/exsy.12191.
7.Mohammed E, Far B. Supervised machine learning algorithms for credit card fraudulent transaction detection: a comparative study. In: IEEE annals of the history of computing. IEEE; 2018. https://doi.org/10.1109/IRI.2018. 00025.
8.Carcillo F, Le Borgne Y-A, Caelen O, et al. Combining unsupervised and supervised learning in credit card fraud detection. Inf Sci. 2019. https://doi.org/10.1016/j.ins.2019.05.042.
9.Abdallah A, Maarof AM, Zainal A. Fraud detection system: a survey. J Netw Comput Appl. 2016;68:90–113.
10.Bhattacharyya S, Jha S, Tharakunnel K, Westland JC. Data mining for credit card fraud: a comparative study. Decis Support Syst. 2011;50(3):602–13.
11.Dhok SS, Bamnote GR. Credit card fraud detection using hidden Markov model. Int J Adv Res Comput Sci. 2012;3(3):816–20.