Check for updates

RESEARCH ARTICLE

Mitigating unbalanced and overlapped classes in credit card fraud data with enhanced stacking classifiers system [version 1; peer review: 1 approved]

Nur Amirah Ishak ¹, Keng-Hoong Ng ¹, Gee-Kok Tong¹, Suraya Nurain Kalid¹, Kok-Chin Khor ²

¹Faculty of Computing and Informatics, Multimedia University, Cyberjaya, Malaysia

²Lee Kong Chian Faculty of Engineering and Science, Universiti Tunku Abdul Rahman, Sungai Long, Malaysia

V1 First published: 21 Jan 2022, 11:71
<https://doi.org/10.12688/f1000research.73359.1>

Latest published: 21 Jan 2022, 11:71
<https://doi.org/10.12688/f1000research.73359.1>

Abstract

Background: Credit cards remain the preferred payment method by many people nowadays. If not handled carefully, people may face severe consequences such as credit card frauds. Credit card frauds involve the illegal use of credit cards without the owner's knowledge. Credit card fraud was estimated to exceed a \$35.5 billion loss globally in 2020, and results in direct or indirect financial loss to the owners. Hence, a detection system capable of analysing and identifying fraudulent behaviour in credit card activities is highly desirable. Credit card data are not easy to handle due to their inherited problems: (i) unbalanced class distributions and (ii) overlapping classes. General learning algorithms may not be able to address and handle the problems well.

Methods: This study addresses these problems using an Enhanced Stacking Classifiers System (ESCS) that comprises two sequential levels. The first level is an excellent classifier for detecting normal credit card transactions (the majority class), while the second level contains stacking classifiers that distinguish credit card frauds (the minority class). The ESCS can improve the fraud detection via the second level, which contains sensitive classifiers to identify the misclassified fraud transactions as normal transactions from the first classifier. The meta-classifier then combines the decisions of the base classifiers from the levels to produce the final detections.

Results: We evaluated the ESCS using the benchmark credit card fraud dataset (CCFD) that exhibits the two problems. The highest true positive rate (TPR) for detecting credit card frauds was 0.8841, which outperformed the single classifiers, bagging, boosting, and other researchers' works.

Conclusions: This study proves that the ESCS, with an additional level added to the stacking classifiers, can improve fraud detection on credit card data.

Open Peer Review

Approval Status

1

version 1

21 Jan 2022

[view](#)

1. **Yen Lung Lai**, Monash University Malaysia,
Subang Jaya, Malaysia

Any reports and responses or comments on the article can be found at the end of the article.

Keywords

Enhanced Stacking Classifiers System, Unbalanced Class Distribution, Overlapping Classes, Credit Card Frauds

This article is included in the **Research Synergy** Foundation gateway.

Corresponding author: Nur Amirah Ishak (n.amirahishak@gmail.com)

Author roles: **Ishak NA:** Conceptualization, Data Curation, Formal Analysis, Investigation, Methodology, Resources, Software, Validation, Visualization, Writing – Original Draft Preparation, Writing – Review & Editing; **Ng KH:** Software, Supervision, Validation, Writing – Review & Editing; **Tong GK:** Software, Supervision, Validation, Writing – Review & Editing; **Kalid SN:** Supervision, Validation, Writing – Review & Editing; **Khor KC:** Software, Supervision, Validation, Writing – Review & Editing

Competing interests: No competing interests were disclosed.

Grant information: This work was supported by the Ministry of Higher Education (MOHE) of Malaysia under Grant FRGS/1/2019/SS01/MMU/03/11.

The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Copyright: © 2022 Ishak NA *et al.* This is an open access article distributed under the terms of the **Creative Commons Attribution License**, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

How to cite this article: Ishak NA, Ng KH, Tong GK *et al.* **Mitigating unbalanced and overlapped classes in credit card fraud data with enhanced stacking classifiers system [version 1; peer review: 1 approved]** F1000Research 2022, 11:71 <https://doi.org/10.12688/f1000research.73359.1>

First published: 21 Jan 2022, 11:71 <https://doi.org/10.12688/f1000research.73359.1>

Introduction

Credit cards were first introduced in the USA in the early 20th century, and in Malaysia in the mid-1970s.¹ Its usage has increased, and it is now widely used in financial transactions around the world. This growth, however, has led to an increase in the number of cases of fraudulent transactions using these cards.

Credit card fraud can be defined as the unlawful use of any system or criminal activity involving a physical card or card information, without the cardholder's knowledge.² Based on the study by Ref. 3, credit card fraud detection relies on the automatic analysis of recorded transactions to detect fraudulent behaviour. When a credit card is used, transaction data consisting of several attributes (e.g. credit card identifier, transaction date, recipient, transaction amount) are stored in a service provider's database.

According to the Nilson report,⁴ between 2015 and 2020, card fraud worldwide was expected to lead to a total loss of \$183.29 billion. In 2020, global card fraud was estimated to exceed \$35.54 billion. Credit card frauds have thus become a major issue in society.⁵

Numerous fraud detection studies have consistently proposed seamless approaches to overcome this issue. However, credit card data sets are not easy to handle as they usually present two challenging characteristics, i.e., (i) unbalanced class distributions and (ii) overlapping classes. Such characteristics are difficult for general classification algorithms to learn and detect credit card frauds.

Unbalanced in class distributions

According to Refs. 6–8, an unbalanced class distribution is said to happen when some classes in a data set have a much greater number of samples than the other classes (Figure 1). Classes with more samples are called the majority class, while on the other hand, classes with a few samples are called the minority class. In a credit card data set, legitimate transactions are the majority class, whereas frauds are the minority class. Fraudulent transactions happen infrequently compared to legitimate transactions, and the percentage of fraudulent transactions is typically low.

Having a few instances of one class means that the general learning algorithms are often unable to generalise their behaviour. Consequently, the algorithms tend to misrepresent a fraudulent transaction as a legitimate transaction.⁹

Furthermore, most general learning algorithms maximise their effectiveness based on classification accuracy, which is not a good metric for evaluating their performance in classifying unbalanced data sets. Learning algorithms usually assume an even distribution of samples for both classes.^{10,11} It has caused the general learning algorithms to be overwhelmed by the majority class, hence, perform poorly on the minority class.

Figure 1. Unbalanced class distribution in a credit card data set.

Overlapping of classes

Overlapping classes in data sets occur when samples in a minority class are overlapped with samples in a majority class (Figure 2), as the samples share common regions in feature space. When overlapping occurs, it causes difficulties for the general learning algorithms to identify the small class samples.^{12–14} Overlapping classes also occur when minority class samples are located near the decision boundary of a majority class. Thus, the decision boundary of a minority class and a majority class may overlap.^{15–17} A decision boundary is a borderline that separates the regions of different classes in a data set. When the overlapping scenario is combined with the unbalanced class distribution problem, it gives rise to even more difficulties for general learning algorithms in classifying the samples.

Related works

Husejinovic¹⁸ performed a study on credit card fraud detection using the single classifiers Naïve Bayes and C4.5, and the ensemble classifier Bagging. Bagging consists of a group of “weak learners” to form “strong learners” and uses majority voting to identify the predicted class by selecting the class with the highest vote assigned by the base learners.^{19–21} The researcher conducted experiments on the credit card fraud dataset (CCFD) and investigated the performance of these classifiers through recall, precision, and precision-recall curve (PRC) area rates; PRC was chosen as the main indicator for the study. PRC measures the overall ability to distinguish between binary classes to predict whether a transaction is normal or fraudulent. A higher PRC indicates that the model was performing better. We observed that the fraud detection rates were 0.8 for single classifiers and bagging, and still have room for improvements.

Divakar and Chitharanjan²² also experimented with the CCFD to study the role of boosting classifiers. Boosting is a classification method where each classifier tries to correct the previous classifiers by adding more weights to the previously misclassified sample, and these weighted samples are then given more attention when classified by the next classifiers.²³ Three classifiers, namely, AdaBoost, Gradient Boost, and XGBoost, were selected. The researchers managed to achieve a fraud detection rate for AdaBoost (0.69), Gradient boosting (0.72), and XGBoost (0.83), with model accuracies of 99.9%, 99.9%, and 100%, respectively. We can see that the classifiers performed averagely based on the fraud detection rates. The researchers also used the model accuracy as the metric for their performance evaluation. Accuracy is not a suitable performance metric when using unbalanced datasets, as classifiers maximise accuracy and are more biased towards the majority class. Kalid *et al.*²⁴ used a multiple classifier System (MCS), which utilised a cascading decision combination strategy to detect frauds. The MCS was tested on the credit card fraud dataset. Using this technique, the output of the first classifier was an input of the subsequent classifier, where the samples were classified several times. The classifiers used were C4.5 (which is good at classifying the majority class) for the first level, and Naive Bayes (for classifying the minority class) for the second level. The fraud detection rate achieved by the researchers was 0.872. This result is good, but there is still room for improvement. Sailusha *et al.*²⁵ also classified transactions in the CCFD using

Figure 2. Overlapping classes that occur in a credit card data set.

Random Forest and AdaBoost. The fraud detection rate achieved by Random Forest was 0.77 and 0.64 for AdaBoost. The results were average.

As single classifier and ensemble classifier cannot perform well in detecting credit card frauds, we proposed designing the enhanced stacking classifiers system (ESCS) to solve the two main characteristics presented by the credit card data mentioned above. ESCS is a multiple classifiers system that consists of two sequential levels. We integrated a single classifier on the first level, with stacking classifiers on the second level. Wolper first proposed stacking,²⁶ a learning technique that combines multiple classifications through a meta-classifier.²⁷ The meta-classifier then combines all base classifiers' decisions to produce a final detection. We evaluated the proposed ESCS using the credit card fraud dataset (CCFD), which exhibits unbalanced class distributions and overlapping classes, as mentioned earlier. We describe the detail of ESCS in the following section.

Ethical considerations

This work has been granted ethical approval (Approval Number: EA0532021) by the Research Ethics Committee (REC) of Multimedia University.

Methods

Credit card fraud dataset (CCFD)

In this study, we used a publicly available CCFD released by Ref. 9, which was collected and analysed during a research collaboration between Worldline and the Machine Learning Group of Université Libre de Bruxelles (ULB) on big data mining and fraud detection. The dataset comprises 31 numerical variables, as shown in Table 1. Variables V1 to V28 are the transformation resulting from a principal components analysis (PCA). The original variables and more background information cannot be provided due to confidentiality concerns. The only variables which have not been transformed using PCA are 'Time' and 'Amount'. 'Time' refers to the time elapsed between each transaction and the first transaction in the dataset in seconds. 'Amount' is the transaction amount. 'Class' is the target variable, and it indicates whether the case is a fraud, marked as '1', or normal, marked as '0'.

The CCFD contains credit card transactions made by European credit cardholders over two days in September 2013. It is highly unbalanced: out of 284,807 transactions, 492 were frauds, and the remaining 284,315 were labelled as legitimate transactions. Figures 3 and 4 depict the unbalanced class distributions and overlapped classes of the dataset, which are the main issues to be tackled in this study.

Class '0' in blue represents the majority class (normal transactions), and Class '1' in orange is the minority class (fraudulent transactions). As shown in Figures 3 and 4, the attributes involved overlapped with each other, and the samples in the majority class outnumbered the minority class. These characteristics also cause difficulties for general learning algorithms to detect credit card frauds effectively.

Enhanced stacking classifiers system

An enhanced stacking classifiers system (ESCS) is proposed to address the class distributions and overlapping issues. It was strategically designed by separating the classes and tackling the data individually at different levels to improve fraud detection rates. ESCS incorporates two sequential levels of multiple-classifier system. The first level contains a classifier that is excellent at detecting normal credit card transactions (the majority class), while the second level consists of single-level stacking classifiers that are good at distinguishing credit card fraud (the minority class). The fraudulent data that were misclassified as normal data, sorted by the classifier on the first level, were filtered out and passed to the second level for re-classification. The re-classification in the second level was performed using two base classifiers stacked with a meta-classifier. These base classifiers are more sensitive classifiers for identifying the misclassified frauds that passed the first level. The meta-classifier was used to combine the base classifier's decisions to produce the final detection. The framework of ESCS is shown in Figure 5.

Table 1. Attributes of the credit card fraud dataset (CCFD).

Variables	Descriptions
V1 to V28 (numerical input variables)	Not being disclosed as it contains sensitive data
Time	Time elapsed between each transaction and the first transaction in the dataset in seconds
Amount	Transaction amount
Class	Target variable, value '1' refers to a fraud, value '0' to a normal transaction

Figure 3. Visualisation of CCFD variables V10 & V26, showing unbalanced class distributions and overlapping classes.

Figure 4. Visualisation of CCFD features V5 & V17, showing unbalanced class distributions and overlapping classes.

Figure 5. The enhanced stacking classifiers system (ESCS) framework.

Pseudocode 1. Algorithm for ESCS fraud detection

```

Input: credit card fraud dataset, ccfd
Output: true positive rate for minority & majority class
1. //create a single level stacking classifier called SSC
2. //SSC ←two base classifiers, C2 and C3 and a meta- //classifier MC
3. //create a multiple classifiers system and named it as
   //Enhanced Stacking Classifiers System (ESCS)
4. //ESCS ←classifier C1 + SSC
   //ESCS is a model with the combination of a classifier
   //C1 and a stacking classifier SSC
5. divide ccfd into five partitions with equal distribution of normal and fraud data
6. label the five partitions with  $K_1, K_2, \dots, K_5$ 
   //5-fold cross validation
7. for  $i \leftarrow 1$  to 5 do
8.   set  $K_i$  as Test_ccfd //test set
9.   set remaining four partitions as Training_ccfd
   //training set
10.  train classifier C1 with Training_ccfd
    //C1 is classifier strong in classifying normal data
    //(majority class)
11.  classify Test_ccfd with trained classifier C1
12.  for each transaction,  $x$  in the Test_ccfd do
13.    if class( $x$ ) is equal to 1 //fraud data
14.      append  $x$  to ccfd(1)
        //ccfd(1) is a dataset of '1' /fraud data
15.    else {class( $x$ ) is 0} //normal data
16.      append  $x$  to ccfd(0)
        //ccfd(0) is a dataset of '0' data/predicted normal
17.    end if
18.  end for
19. end for
20. divide ccfd(0) into five partitions with equal distribution of normal and fraud data
21. label five partitions with  $P_1, P_2, \dots, P_5$ 
   //5-fold cross validation
22. for  $j \leftarrow 1$  to 5 do
23.   set  $P_j$  as Test_ccfd(0)
24.   set remaining four partitions as Training_ccfd(0)
25.   train SSC with Training_ccfd(0)
    //C2, C3 are classifiers strong in classifying
    //minority class
26.   classify Test_ccfd(0) with trained SSC
27.   for each transaction,  $y$  in the Test_ccfd(0) do
28.     if class( $y$ ) is equal to 1 //fraud data
29.       append  $y$  to ccfd(1)
        //ccfd(1) is a dataset of '1' /fraud data
30.     remove  $y$  from ccfd(0)
31.     end if
32.   end for
33. end for
34. combine ccfd(1), ccfd(0) to ccfdFinal
    //ccfdFinal is a combination of dataset ccfd(1) and
    //ccfd(0)
35. class ← Retrieve only 'class' column from ccfdFinal
36. predicted ← Retrieve only 'predicted' column from ccfdFinal
37. calculate confusion matrix (class, predicted)
38. calculate TPR(1) //TPR for minority class

```

```

39. calculate  $TPR(0)$  //TPR for majority class
40. return  $TPR(1)$ ,  $TPR(0)$ 

```

In this study, five-fold cross-validation was conducted on the CCFD, *ccfd*. The dataset was divided into five partitions with equal distribution of normal and fraud data (Line 5, Pseudocode 1). A single partition was reserved at each validation step as the test set, *Test_ccfd* (Line 8, Pseudocode 1), while the remaining four partitions were used as the training data, *Training_ccfd* (Line 9, Pseudocode 1). This process was then repeated five times until every partition was used for training and testing. On the first level, classifier *C1* was trained with *Training_ccfd* (Line 10, Pseudocode 1) and classified *Test_ccfd* with it (Line 11, Pseudocode 1). Classifier *C1* is a strong classifier of the majority class (normal data).

During classification, if the samples were classified as '1', then they were appended to *ccfd(1)*, which stores all the fraud data (Line 14, Pseudocode 1). If the samples were classified as '0', they were appended to *ccfd(0)* (Line 16, Pseudo. 1). The *ccfd(0)* dataset stores all data predicted as normal and is passed to the second level to re-classify the data.

On the second level, we conducted the five-fold cross-validation on *ccfd(0)* again, divided into five partitions with equal distribution of normal and fraud data, and labelled them P_1 to P_5 (Line 21, Pseudocode 1). At each validation step, a single partition was reserved as the test set, *Test_ccfd(0)* (Line 23, Pseudocode 1), while the remaining four partitions were employed as the training data, *Training_ccfd(0)* (Line 24, Pseudocode 1). A single level stacking classifier, which consisted of classifiers *C2*, *C3* and the meta-classifier, was trained with *Training_ccfd(0)* (Line 25, Pseudocode 1) and classified *Test_ccfd(0)* (Line 26, Pseudo. 1) with it. Classifiers *C2* and *C3* are strong at classifying the minority class.

During re-classification, if the samples were classified as '1', then the samples were appended to data set *ccfd(1)* (Line 29, Pseudocode 1), and the same samples were deleted in the *ccfd(0)* to avoid any redundancy in both data (Line 30, Pseudocode 1). If the samples were classified as '0', then they were still stored in *ccfd(0)*.

ccfd(1) and *ccfd(0)* were then combined and saved as *ccfdFinal* (Line 34, Pseudocode 1). From *ccfdFinal*, only the 'Class' column (Line 35, Pseudo. 1) and the 'Predicted' column (Line 36, Pseudocode 1) were retrieved to form the confusion matrix (Line 37, Pseudo. 1) as in Table 2. Lastly, the final true positive rate (TPR) score for the minority and majority classes was calculated (Lines 38-39, Pseudocode 1) as in Equation (1) and Equation (2).

$$TPR \text{ for minority} = \frac{TP}{\text{Actual positive}} = \frac{TP}{TP + FN} \quad (1)$$

$$TPR \text{ for majority} = \frac{TN}{\text{Actual negative}} = \frac{TN}{TN + FP} \quad (2)$$

Results and discussion

We conducted three experiments on the CCFD; single classifiers, bagging and boosting classifiers, and the proposed ESCS model. Their TPR, area under the *receiver operating characteristic* (ROC AUC) and accuracy, were calculated and are presented in the following tables.

For the single classifier experiment, seven classifiers were used, namely, naïve Bayes (NB), ID3, logistic regression (LR), random forest (RF), multi-layer perceptron (MLP), K-nearest neighbour (KNN) and CART. Overall, we observed good accuracy, where the algorithms achieved scores above 0.99, except KNN (0.4226), CART (0.7995) and RF (0.8030) (Table 3).

The TPR results were good for the majority class (class 0). This experiment yielded scores over 0.99 for most classifiers, except KNN (0.4227), CART (0.7996) and RF (0.8030). It was found that the best achievable TPR for the minority class

Table 2. Representation of confusion matrix.

	Predicted: 0	Predicted: 1
Actual: 0	True negative (TN)	False positive (FP)
Actual: 1	False negative (FN)	True positive (TP)

Table 3. Result of single classifiers. Generally, the algorithms could not perform well on the minority class. TPR (1) = true positive rate for minority class; TPR (0) = true positive rate for majority class; ROC AUC = area under the receiver operating characteristic.

Single classifiers				
	TPR (1)	TPR (0)	ROC AUC	Accuracy
Naïve Bayes	0.6585	0.9930	0.8257	0.9924
ID3	0.7520	0.9996	0.8758	0.9992
Logistic Regression	0.6402	0.9964	0.8183	0.9958
Random Forest	0.7947	0.8030	0.8632	0.8030
Multi-Layer Perceptron	0.7927	0.9958	0.7989	0.9954
K-Nearest Neighbour	0.3638	0.4227	0.3933	0.4226
CART	0.7398	0.7996	0.7697	0.7995

(class 1) was only 0.7947 (RF), followed by 0.7927 (MLP), with a slight difference of 0.002. Then, it was followed by ID3 (0.7520), CART (0.7398), NB (0.6585), LR (0.6402) and KNN (0.3638).

We then tried to improve the detection rate using bagging and boosting (ensemble classifiers) since the single classifiers did not perform well in detecting frauds. This experiment involved one bagging classifier and five boosting classifiers: AdaBoost, gradient boosting, extreme gradient boosting (XGBoost), light gradient boosting machine (LightGBM) and CatBoost.

As shown in Table 4, we can achieve a good overall accuracy rate and TPR for the majority class. The highest accuracy recorded was CatBoost by 0.9993, followed by AdaBoost (0.9979), LightGBM (0.9944), XGBoost (0.9939), bagging (0.8028) and Gradient boosting (0.8004). Similarly, for TPR for the majority class (class 0), CatBoost achieved the highest accuracy by 0.9996, followed by AdaBoost (0.9984), LightGBM (0.9953), XGBoost (0.9942), bagging (0.8028) and gradient boosting (0.8009). However, the overall TPR for the minority class (class 1) was not promising, being an average value. The highest fraud detection rate was 0.7846 by XGBoost and CatBoost. The second best was performed by the bagging classifier, with a value of 0.7744. Finally, it was followed by AdaBoost (0.6931), Gradient Boosting (0.5264) and LightGBM (0.4593).

We then designed an ESCS, comprising two sequential levels, to alleviate the two inherited problems of the credit card fraud data (Figure 6). On the first level, we used ID3, which is a strong classifier of the majority class (refer to Table 3). Then, the fraud data that misclassified as normal data were filtered and passed through the second level. On the second level, we used MLP and RF (refer to Table 3), which efficiently classify the minority class, and stacked them with a meta-classifier. These classifiers are more sensitive and identify the misclassified fraud detection from the ID3 at the first level. The meta-classifier was used to combine the decisions of the base classifiers to produce the final detection. We evaluated

Table 4. Result of bagging and boosting classifiers. Fraud detection rates achieved by the classifiers were still not performing at their best. TPR (1) = true positive rate for minority class; TPR (0) = true positive rate for majority class; ROC AUC = area under the receiver operating characteristic.

Bagging				
	TPR (1)	TPR (0)	ROC AUC	Accuracy
Bagging classifier	0.7744	0.8028	0.7886	0.8028
Boosting				
	TPR (1)	TPR (0)	ROC AUC	Accuracy
AdaBoost	0.6931	0.9984	0.8458	0.9979
gradient boosting	0.5264	0.8009	0.6636	0.8004
XGBoost	0.7846	0.9942	0.8894	0.9939
LightGBM	0.4593	0.9953	0.7273	0.9944
CatBoost	0.7846	0.9996	0.8921	0.9993

Figure 6. The enhanced stacking classifiers system.

five different classifiers for the meta-classifier, namely, ID3, RF, LR, NB and MLP. All the classifiers were chosen based on their performance on the CCFD. The ESCS can improve the fraud detection rate through the second level as it contains stacking classifiers that are effective at distinguishing credit card frauds. The ESCS framework is shown in Figure 6 below, and its performance is shown in Table 5.

We observed that NB was the best meta-classifier combining all the base classifiers' decisions to produce the final decision. It attained a 0.8841 fraud detection rate overall. It showed a good non-fraud detection rate of 0.9839, a ROC AUC score of 0.9340 and an accuracy of 0.9837. We could also achieve comparable accuracy rates and non-fraud detection rates for ESCS 1, ESCS 2, ESCS 3 and ESCS 5 when compared with single classifiers and ensemble classifiers. The second-best result was ESCS 2 with the meta-classifier RF, for which the fraud detection rate was 0.8028, followed by ESCS 3 with the meta-classifier LR at 0.7785, ESCS 1 with the meta-classifier ID3 at 0.7622 and ESCS 5 with the meta-classifier MLP at 0.7520.

In conjunction with this experiment, ESCS was compared to other researchers' works. The comparisons are shown in Table 6.

Table 5. Results of the enhanced stacking classifiers system. There were improvements in the fraud detection rate compared to single classifiers, bagging and boosting. TPR (1) = true positive rate for minority class; TPR (0) = true positive rate for majority class; ROC AUC = area under the receiver operating characteristic.

Enhanced stacking classifiers system				
	TPR (1)	TPR (0)	ROC AUC	Accuracy
ESCS 1; with meta-classifier ID3	0.7622	0.9996	0.8809	0.9992
ESCS 2; with meta-classifier RF	0.8028	0.9995	0.9012	0.9991
ESCS 3; with meta-classifier LR	0.7785	0.9996	0.8890	0.9992
ESCS 4; with meta-classifier NB	0.8841	0.9839	0.9340	0.9837
ESCS 5; with meta-classifier MLP	0.7520	0.9996	0.8758	0.9992

Table 6. Comparison between the enhanced stacking classifiers system (ESCS) and other researchers' work on the credit card fraud dataset (CCFD). ESCS outperformed the rest. TPR (1) = true positive rate for minority class; TPR (0) = true positive rate for majority class.

Credit card fraud dataset					
Researchers' works	Technique	Classifiers	TPR (1)	TPR (0)	Accuracy
ESCS	Enhanced Stacking Classifiers	ID3+MLP+RF Meta-classifier: NB	0.8841	0.9839	0.9837
Kalid <i>et al.</i> (2020) ²⁴	Cascading of Multiple Classifiers	C4.5+NB	0.8720	1.000	0.9990
Husejinović (2020) ¹⁸	Bagging	Bagging	0.7970	0.9160	-
Divakar and Chitharanjan (2019) ²²	Boosting	XGBoost	0.8300	0.9400	1.000

ESCS was able to achieve the highest TPR (0.8841) for the minority class, and outperformed the other researchers' models. ESCS also gave a good accuracy of 0.9837 and a TPR of 0.9839 for the majority class.

ESCS with NB as the meta-classifier showed great performance, and proved that ESCS could improve the fraud detection rate as it can effectively identify misclassified fraud transactions.

Conclusions

Nowadays, credit cards are the most common payment method because of the conveniences they provide. If credit card usage is not well-managed, it may lead to undesirable events, such as credit card frauds. Credit card frauds involve the illegal use of credit cards without the owner's consent and cause them to suffer a financial loss.

Utilising credit card transaction data is now a necessity to detect frauds. However, it would be challenging to handle credit card data because of their (i) unbalanced class distributions and (ii) overlapping classes. These characteristics also cause difficulties for general learning algorithms to detect frauds effectively.

This study proposes to address these two issues using an ESCS, strategically separating the classes and tackling the data individually at different levels to improve fraud detection rates. We compared the performance of ESCS with single bagging and boosting classifiers. The highest TPR for the minority class (frauds) was 0.8841 using ESCS with NB as the meta-classifier, which outperformed other combinations. We also compared our ESCS with previous research. The results showed that our ESCS outperformed other researchers' works. This study proves that ESCS can improve the fraud detection rate on credit card data.

Data availability

Underlying data

Figshare: CCFD_dataset, <https://doi.org/10.6084/m9.figshare.16695616.v3>.²⁸

This project contains the following underlying data:

- CCFD_dataset.xlsx (extracted credit card dataset from the original Kaggle dataset)

Data are available under the terms of the [Creative Commons Zero "No rights reserved" data waiver](#) (CC0 1.0 Public domain dedication).

Extended data

Analysis code available from: <https://github.com/nuramirahishak/ESCS/tree/escs>

Archived analysis code as at time of publication: <https://doi.org/10.5281/zenodo.5647747>.²⁹

License: [OSI 3.0](#)

References

1. Ahmed ZU, Ismail I, Sadiq Sohail M, *et al.*: **Malaysian consumers' credit card usage behavior**. *Asia Pac. J. Mark. Logist.* 2010; **22**(4): 528–544.
[Publisher Full Text](#)
2. Raju O: **A Survey on Machine Learning Algorithms in Credit Card Fraud Detection**. 2021; vol. **25**(no. 1): pp. 712–727.
[Reference Source](#)
3. Pozzolo AD, Caelan O, Borgne YL, *et al.*: **Learned Lessons in Credit Card Fraud Detection from A Practitioner Perspective**. *Expert Syst. Appl.* 2014; **41**: 4915–4928.
[Publisher Full Text](#)
4. Fulmer L: **Global Card Fraud Losses Reach \$16.31 Billion-Will Exceed \$35 Billion in 2020 According to The Nilson Report**. 2015. [2019, October 19].
[Reference Source](#)
5. Van Vlasselaer V, Bravo C, Caelen O, *et al.*: **APATE: A novel approach for automated credit card transaction fraud detection using network-based extensions**. *Decis. Support. Syst.* 2015; **75**: 38–48.
[Publisher Full Text](#)
6. de Sá AGC, Pereira ACM, Pappa GL: **A customised classification algorithm for credit card fraud detection**. *Eng. Appl. Artif. Intell.* 2018; vol. **72**: pp. 21–29. 0952-1976.
[Publisher Full Text](#)
7. Haixiang G, Yijing L, Shang J, *et al.*: **Learning from class-imbalanced data: Review of methods and applications**. *Expert Syst. Appl.* 2016; vol. **73**: pp. 220–239. 09574174
[Publisher Full Text](#)
8. Bekkar M, Djemaa HK, Alitouche TA: **Evaluation Measures for Models Assessment over Imbalanced Data Sets**. *Journal of*

- Information Engineering and Applications 2013; vol. 3(no.10): pp. 27–38.2225-0506.
9. Pozzolo AD, Caelan O, Johnson RA, *et al.*: **Calibrating Probability with Undersampling for Unbalanced Classification**. 2015 IEEE Symposium Series on Computational Intelligence 2015; pp. 159–166.
[Publisher Full Text](#)
 10. Akosa J: **Predictive Accuracy: A Misleading Performance Measure for Highly Imbalanced Data**. *Proc. SAS Global Forum* 2017; 1–12.
 11. Krawczyk B: **Learning from imbalanced data: open challenges and future directions**. *Prog. Artif. Intell.* 2016; 5: 221–232.
[Publisher Full Text](#)
 12. Lee H, Kim S: **An overlap-sensitive margin classifier for imbalanced and overlapping data**. *Expert Syst. Appl.* 2018; 98: 72–83.
[Publisher Full Text](#)
 13. Das B, Krishnan NC, Cook DJ: **Handling Class Overlap and Imbalance to Detect Prompt Situations in Smart Homes**. *IEEE 13th International Conference on Data Mining Workshops*. 2013.
[Publisher Full Text](#)
 14. Qu Y, Su H, Guo L, *et al.*: **A novel SVM modeling approach for highly imbalanced and overlapping classification**. *Intell. Data Anal.* 2011; 15: 319–341.
[Publisher Full Text](#)
 15. Napierala K, Stefanowski J: **Types of minority class examples and their influence on learning classifiers from imbalanced data**. *J. Intell. Inf. Syst.* 2016; 46: 563–597.
[Publisher Full Text](#)
 16. Blaszczyński J, Stefanowski J: **Neighbourhood sampling in bagging for imbalanced data**. *Neurocomputing* 2015; 150: 529–542.
[Publisher Full Text](#)
 17. Lango M, Stefanowski J: **Multi-class and feature selection extensions of Roughly Balanced Bagging for imbalanced data**. *J. Intell. Inf. Syst.* 2018; 50: 97–127.
[Publisher Full Text](#)
 18. Husejinovic A: **Credit card fraud detection using naive Bayesian and C4.5 decision tree classifiers**. *Periodicals of Engineering and Natural Sciences* 2020; 8(1): 1–5.
[Reference Source](#)
 19. Breiman L: **Bagging Predictors**. *Mach. Learn.* 1996; 24: 123–140.
[Publisher Full Text](#)
 20. Rajora S, Li D, Jha C, *et al.*: **A Comparative Study of Machine Learning Techniques for Credit Card Fraud Detection Based on Time Variance**. *IEEE Symposium Series on Computational Intelligence (SSCI)* 2018; 1958–1963.
[Publisher Full Text](#)
 21. Breiman L: **Random forests**. *Mach. Learn.* 2001; 45: 5–32.
[Publisher Full Text](#)
 22. Divakar K, Chitharanjan K: **Performance Evaluation of Credit Card Fraud Transactions using Boosting Algorithms**. *International Journal of Electronics Communication and Computer Engineering* 2019; 10: 262–270.
 23. Barahim A, Alhajri A, Alasaibia N, *et al.*: **Enhancing the Credit Card Fraud Detection Through Ensemble Techniques**. *J. Comput. Theor. Nanosci.* 2019; 16: 4461–4468.
[Publisher Full Text](#)
 24. Kalid SN, Ng KH, Tong GK, *et al.*: **A Multiple Classifiers System for Anomaly Detection in Credit Card Data with Unbalanced and Overlapped Classes**. *IEEE Access* 2020; 8: 28210–28221.
[Publisher Full Text](#)
 25. Sailusha R, Gnaneswar V, Ramesh R, *et al.*: **Credit Card Fraud Detection Using Machine Learning**. 4th International Conference on Intelligent Computing and Control Systems (ICICCS). 2020.
[Publisher Full Text](#)
 26. Wolpert DH: **Stacked generalisation**. *Neural Netw.* 1992; 5(2): 241–259.
[Publisher Full Text](#)
 27. Prabhakara E, Kumarb MN, Ponnarab K, *et al.*: **Credit card fraud detection using boosted stacking**. *South Asian J. Eng. Technol* 2019; 8(S1): 149–153.
 28. Ishak NA, Ng K-H, Tong G-K, *et al.*: **CCFD_dataset**. *figshare. Dataset*. 2021.
[Publisher Full Text](#)
 29. nuramirahishak: **nuramirahishak/ESCS: ESCS python scripts (escs)**. *Zenodo*. 2021.
[Publisher Full Text](#)

Open Peer Review

Current Peer Review Status:

Version 1

Reviewer Report 22 July 2022

<https://doi.org/10.5256/f1000research.77004.r122464>

© 2022 Lai Y. This is an open access peer review report distributed under the terms of the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Yen Lung Lai

School of Information Technology, Monash University Malaysia, Subang Jaya, Malaysia

This manuscript studies the classification problem over the credit card transactions. In particular, in a credit card transactions data set, legitimate transactions are the majority class, whereas frauds are the minority class.

The authors proposed the enhanced stacking classifier system (ESCS) to address class distributions and overlapping issues. It was strategically designed by separating the classes and tackling the data individually at different levels to improve fraud detection rates.

Overall, this manuscript is comprehensive and scientifically sound. I have some comments wish to improve the quality of it:

- It would be good if ROC curves can be provided for different classifiers/boosting/ESCS/ for better visualization of the system performance.
- It is noted that the TPR for the proposed ESCS is achievable at 0.8841 (Table 6). However, the TPR and the resultant accuracy doesn't outperform the other researchers' works as highlighted in Table 6. The author should elaborate more on such result for fair comparison. Maybe more explanation on why the TPR and overall accuracy are lower? Is there any tradeoff?
- Moreover, it seems the proposed ESCS framework can be considered as one type of the MSC (with multiple classifiers in parallel combination), proposed by Kalid et al., which only consists of two classifiers in level 1.

Is the work clearly and accurately presented and does it cite the current literature?

Yes

Is the study design appropriate and is the work technically sound?

Partly

Are sufficient details of methods and analysis provided to allow replication by others?

Yes

If applicable, is the statistical analysis and its interpretation appropriate?

Partly

Are all the source data underlying the results available to ensure full reproducibility?

Yes

Are the conclusions drawn adequately supported by the results?

Yes

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: information security

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard.

The benefits of publishing with F1000Research:

- Your article is published within days, with no editorial bias
- You can publish traditional articles, null/negative results, case reports, data notes and more
- The peer review process is transparent and collaborative
- Your article is indexed in PubMed after passing peer review
- Dedicated customer support at every stage

For pre-submission enquiries, contact research@f1000.com

F1000Research