Journal of Technology Informatics and Engineering
Vol.1, No.1 April 2022
e-ISSN: 2961-8215; p-ISSN: 2961-9068, Hal 50-79

MACHINE LEARNING TECHNIQUE FOR CREDIT CARD SCAM DETECTION

Fujiama Diapoldo Silalahi[1], Toni Wijanarko Adi Putra[2], Edy Siswanto[3]
Universita Sains dan Teknologi Komputer

Abstract

Credit Card (CC) scam In financial markets is a growing nuisance. CC scams increasing rapidly and causing large amounts of financial losses for organizations, governments, and public institutions, especially now that all payment methods for e-commerce shopping can be done much more easily through digital payment methods. For this reason, the purpose of this study is to detect scam CC transactions from a given dataset by performing a predictive investigation on the CC transaction dataset using machine learning techniques. The method used is a predictive model approach, namely logistic regression models (LR-M), random forests (RF), and XGBoost combined along particular resampling techniques that have been practiced to anticipate scams and the authenticity of CC transactions. Model performance was calculated grounded Re-call Curve (RC), precision, f1-score, PR, and ROC.

The experimental results show that the random forest in combination with the hybrid resampling approach of SMOTE and removal of Tomek Links works better than other models. The random forest model and XGBoost accomplished are preferred over the LR-M as long as their global f1 score is without re-sampling. This demonstrates the strength of one technique that can provide greater achievement alike in the existence of class inequality dilemmas. Each approach, at the same time when used with Ran-Under, will give a great memory score but fails cursedly in the language of accuracy. Compared to the coordinate model sine re-sampling, the accuracy and RS are not repaired in cases where Tomek linker displacement was used. RF and xgboost perform quite well in terms of f1-S when Ran-Over is used. SMOTE increases the random forest draw score and xgboost but the precision score (PS) decreases slightly.

Completely, during a hybrid solution of Tomek delinker and SMOTE was practiced with random forest, it gave equitable attention and RS in the PR-AUC. XGboost failed to increase the PS even though the same re-sampling technique was used. For future research, a fee-delicate study method can be applied as long as fee misclassifications. So for future research, it is very necessary to consider this behavior change and it is also very important to develop predictive models. In addition to this, much larger data is needed so that detailed studies on handling non-stationary properties in CC scam detection can be carried out better.

Keywords: *Scam Detection, Credit Card, Financial Scam, Machine Learning*

Received Maret 05, 2022; Revised Maret 20, 2022; April 13, 2022

INTRODUCTION

In addition to consumptive users, for most companies and government agencies or public institutions, e-commerce is an important means of increasing the productivity of global trade. The use of digital payment methods that are easier and more automated makes e-commerce users divert some of their shopping payment methods to digital payment methods via debit i-banking or m-banking, digital money, or Credit cards. "From all the surveys, one of the supporters of the success of e-commerce is the ease of transactions or payments using online methods, especially credit cards" Lebichot et al, (2016). "From here financial scam must also be considered financial scam is a deliberate harm where the scammers take advantage for themself by advising the victim's rights or by obtaining financial benefits" Awoyemi et al, (2017). Since CC transactions are the most familiar payment method, besides being so easy to use that only requires one click for approval, this has resulted in a rapid increase in scam activity.

Financial losses due to scam activities are a major problem faced by companies and public institutions. According to a report from the "Nelson Report" losses reached IDR 445K billion in CC scams in 2020, or IDR 2000 per IDR 1,5 million of purchases (see figure 2), and are expected to exceed a more significant number in years to come. next year (See figure 3). Scam solutions can be designed into avoidance, which includes prevention of scams at their cause, and detection, which is a plan appropriate since the case was developed. "Technologies such as the 'Address Verification System' (AVS) and 'Card Verification System' (CVM) are commonly used to prevent scams, rule-based filter methods and data mining are used for prevention" Juszczak et al, (2008). When fraud cannot be prevented it must be detected as soon as possible, and measures must be decided and implemented for security reasons. Furthermore, various transactions must be detected as much as possible to anticipate fraud and fake transactions. (Maes, 1993). An automated FDS is necessary especially given that the traffic of transaction data is huge, and it is impossible for humans to manually check each transaction individually whether it is a scam or not. This research was conducted based on an automated FDS using machine learning techniques.

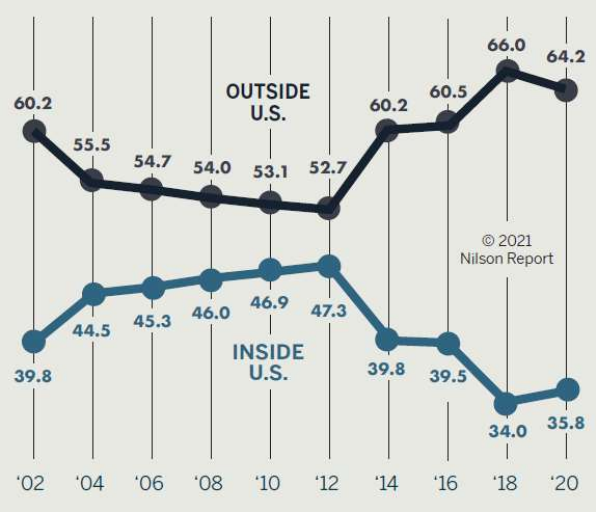


Figure 1 Nilson Report Scam Inside vs. Outside the US 2002 — 2020

Detection process (Scam)

As shown in Figure 4, the transaction will be checked at the terminal point first to ensure the validity of the transaction being carried out. At the point terminal, certain important information will appear, such as balance amounts, PIN information, and so on, which will later be validated and then filtered so that the accuracy of the information that has been validated is truly valid so that there are no errors in transactions. Furthermore, to identify the authenticity or falsity of transactions, all valid transactions will be assessed using a predictive model. To improve model performance, Investigators should investigate any security alerts or fraud alerts and provide

feedback to the predictive model as soon as possible (Pozzolo, 2015). This research is limited to predictions because basically, this research will only focus on description problems.

Challenges

"To build a scam detection system, practitioners need to determine learning strategies, algorithms, features, and strategies to deal with class inequality dilemmas" Pozzolo, (2015). Overlapping of real and deceptive classes because of defined instructions about purchase archives is another dilemma in task analysis (Holte et al, 1989), and "most machine learning algorithms perform poorly under this scenario" Japkowicz et al, (2002). In real-life scenarios, scam detection models predict class properties and provide investigators with alerts for the most suspicious transactions. The investigator then conducts further investigations and provides evaluation to the FDS to develop the performance. Nonetheless, this process can be an overhead for the investigator because only a few transactions are authorized on time by the investigator. In such cases, "little feedback is provided to the predictive model, which generally results in a less accurate model" Pozzolo, (2015). Finally, "financial institutions very rarely disclose customer data to the public due to confidentiality issues, making real financial data sets very difficult to find. This is one of the main challenges in scam detection research work" Pozzolo, (2014).

Figure 2 Scam by Type of Card (Nilson Report 2021)

The purpose of this study is to implement anticipating investigation on CC transaction datasets using machine learning techniques and expose fake transactions from a given dataset, the focus of this study is to analyze various transactions that fall into the normal class by using predictive models. To overcome the problem of class imbalance, an unbalanced set of ML algorithms (such as RL, RF and xgboost will be applied to the data set, and the results will be reported at end of this study) different sampling techniques at this study will be applied.

Literature Review

Many financial institutions have lost fantastic amounts of money due to CC scams, and this has become a very big problem in the financial market, institutions put together a team of experts to establish a scam detection system. Many institutes have been intently employed to modify the dilemma encountered during building "scam detection systems" (FDS) like class imbalance, class overlap, scam behavior changes, and so on. Pozzolo et al, (2014) focus on 2 diverse secluded to detect scams: the passive approach where the disclosure model is competent seasonally, and the networked model where the model is modified along it its recent activity data arrives. A specified networked learning model is preferred because scam behavior changes from time to time. Pozzolo, (2014) "suggested that Average Precision (AP), Area Under Curve (AUC), and Precision Rank (PR) is the best quantify for scam detection tasks".

YEAR	Total Volume (TRIL.)	Fraud (BIL.)	Cents per $100 VOLUME
2020	$41.962	$28.58	6.81
2021	$47.229	$32.20	6.82
2022	$50.868	$34.36	6.75
2023	$54.061	$36.13	6.68
2024	$57.323	$38.07	6.64
2025	$60.583	$39.89	6.58
2026	$64.038	$41.73	6.52
2027	$67.570	$43.76	6.48
2028	$71.221	$45.54	6.39
2029	$75.111	$47.50	6.32
2030	$79.140	$49.32	6.23

© 2021 Nilson Report

Figure 3 Projected Card Scam to 2030 (Nilson Report 2021)

Awoyemi, (2017) analyzed K-nearest (K-N) neighbor data, logistic regression, and Naive Bayes when applied to CC activity data, which were then re-sampled using the SMOTE or *"Synthetic Minority Over-sampling Technique"*. The results show that the K-N acquaintance exceeds others, the conduct was deliberate in details of recollection, attention, equitable allocation level, a specific city, and Mathews interaction coordinate. Another study by Lebich, (2016) suggested another approach based on graphics to develop a scam tracking down the system. In this model, cumulative assumption algorithms are used to transmit the scam effect in the system by using some scam activity data. Lebich, (2016) formed many upgrades in the current scam disclosure structure, its call "Anomaly Prevention using Advanced Transaction Exploration" to reach its goals.

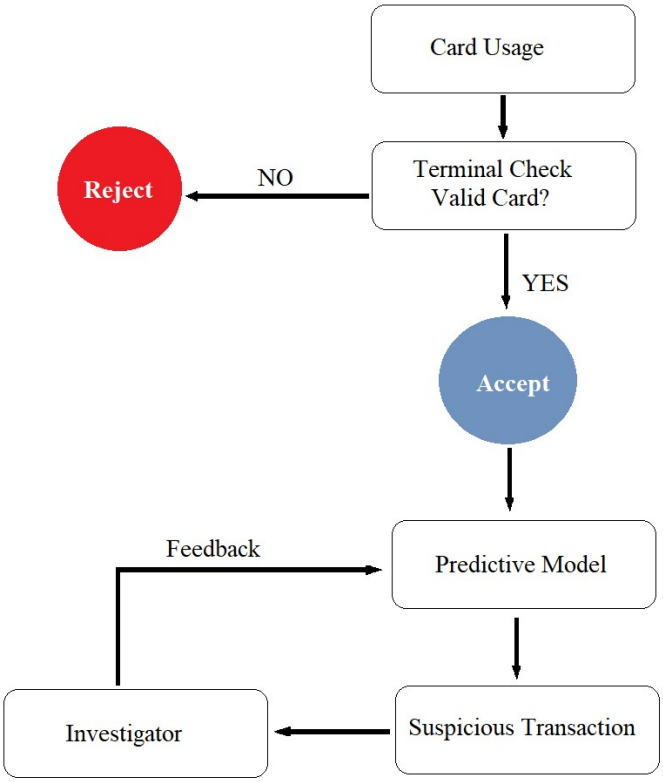


Figure 4 Scam detection process

Domingos, (1999) suggested a fee-effective model called "MetaCost". A specific study by Domingos (1999) specified costs of all misclassifications are different, so the approach to encapsulate a cost-minimizing measure in analysis with placed into account the costs of misclassification is proposed". From these experiments, Domingos definite that was an analytical downturn in the number of costs when using "MetaCost" correlated to other insensitive classifiers. In different study other by Aleskerov, (1997) suggested a data mining structure used neural network for scam discover. Srivastava, (2008) suggested a "Hidden Markov Model" for customers shopping habits are analyzed to detect scam behavior. Wheeler & Aitken, (2000) investigated several algorithms for detecting scams, the outcome shows a flexible model can screen and manage scam cases which can resolve and slash the total of scam inspections.

Although there are many research studies conducted on various conditions of scam disclosure equally hunting cardholder performance, increasing scam detection clarification time, overcoming fee classification misallocation, and assorted data mining procedures, very insufficient study that conducted in this method for addressing the class inequality dilemma in a scam detection task. Accordingly, this study's objective is to execute a predictive investigation on the CC scam detection task which primarily spotlights different approaches of equally re-sampling, altogether, and combination models to address the class inequality dilemma.

RESEARCH METHODOLOGY

The dataset used in this study is the dataset used for the research project Pozollo et al, (2015) which contains 284.8K transactions, with a total of 492 scam data. The dataset is deeply unequal as the +class details only 0.172 percent of the total transactions. Look at figure 5, in this figure 5, there is a bar chart that visualizes unreliable class circulation.

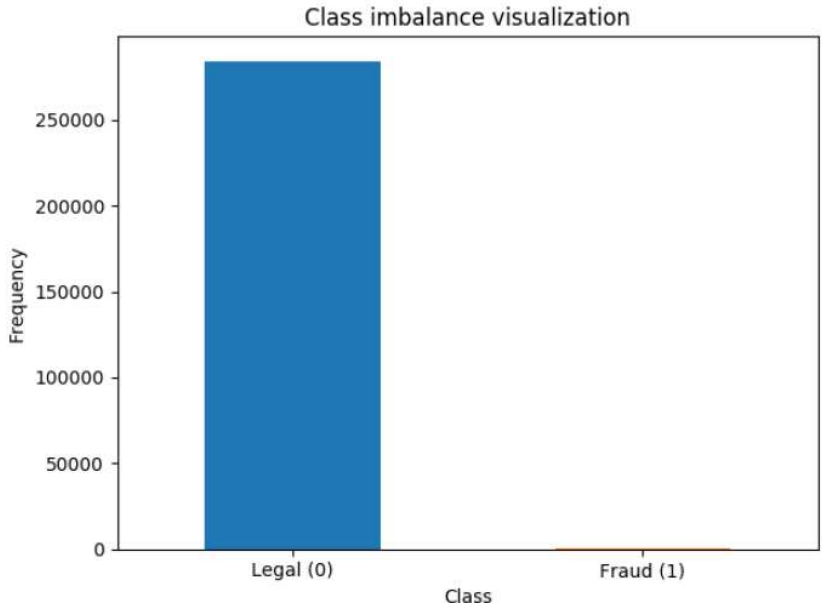


Figure 5 Resolve of a very irregular class circulation

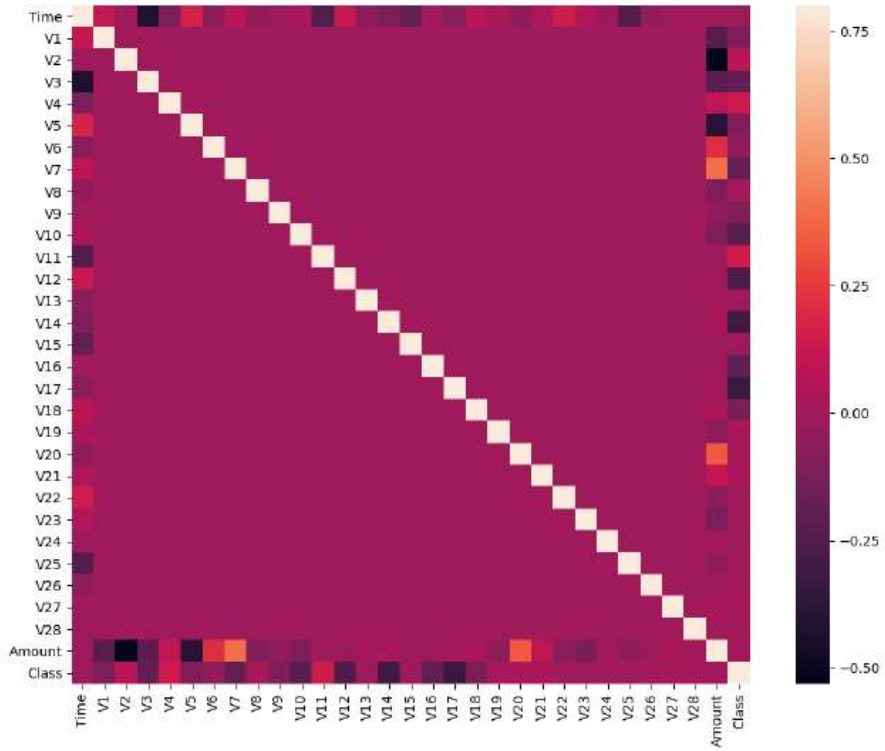


Figure 6 interaction of cast presenting the interaction in the data

The dataset consists of numeric ideals resulting from the primary fundamental investigation revolution. However, due to confidentiality reasons, the original features are not disclosed, in which a total of 30 features are owned, and a total of 28 features are generated from principal component analysis, then two appearances have not been converted into the main belly namely "amount" and "time". Figure 6 serves as the interaction cast that contributes an interpretation of the interaction of the pair enclosed by a particular variable. Interaction cast given shows that no one of the main components V1 to V-end is correlated with each other. But, on close inspection, the feedback variable “class” has different models of (+) and (-) correlation with the main elements, but not with “time” and “amount”.

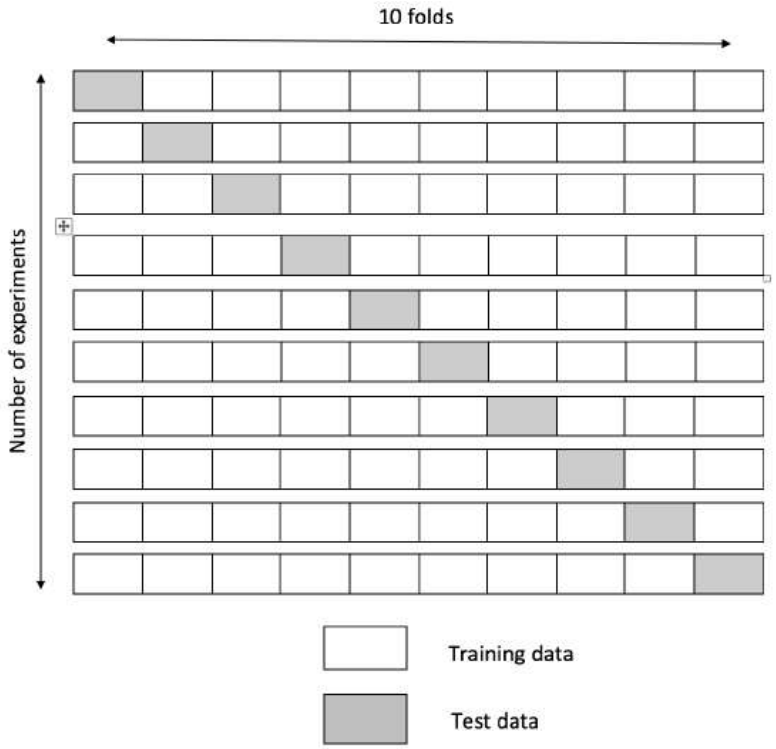


Figure 7 10-bend-cross-validation (BCV)

Standardization

Standardization of components assigns to scaling components to prevent all of that aim the setting of a basic typical circulation along a mean of 0 and a basic diversion of 1. This is a general specification for various ML approaches that appearance must be regulated previously accepting ML approach. In case regulation is uncarried out, it can influence the achievement of the model. Similarly the 'Sum' component is used “Standard-Scalar” in the sci-kit-learn library was carried out in this research and achieved using the following equation:

$$z = \frac{x - \mu}{\sigma}$$

$$Mean(\mu) = \frac{1}{N}\sum_{i=1}^{N}(x_i)$$

where

$$Standard\ Deviation\ (\sigma) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2}$$

Data separation

For each experiment, all data sets were divided into 2 test parts, namely 70 percent and 30 percent. The tester set is used for re-sampling, setting hyperconfines, and practicing the miniature, while the test set is used to test the achievement of the model being trained. When separating data, specifying a random grain is chosen to establish the same file is separated at any moment when the program is run.

Re-sampling of data

Because the dataset is unequal in size, where appropriate transactions exceed the total of cheat transactions, several resampling techniques ("random undersampling" in this study was short as "Ran-Under" "random oversampling" be "Ran-Over), "SMOTE", "Tomek unlink", "combined SMOTE" and "Tomek unlink") are used here. To make the dataset balanced, these techniques are applied to the data separately.

Hyper-criterion tuning uses 10-bend CV

CV techniques are used to tune hyperconfines. The cross-validation for the K-bend value was set as 10. In 10-bend CV, the training dataset was divided into 10 bends, and for each bend, in this study, the current bend was selected as the test set and the stopping bends as the tester set. Then the model is entered into the tester set and the data testing set is evaluated (see figure 7). In the search for the best hyperconfines, the scikit learn grid search function is used together with cross-validation techniques.

Hyper-criterion search:

Logistic regression (LR)

In LRM, the standardization specification "C" is an essential hyper-criterion that demands to be agreeable precisely. The amount of C precisely interests the generalizability of the model. In the basic account of accepted C characters, the "GridSearchCV" technique was carried out on a re-sampled exercise file to acquire the perfect C specification for LR-M.

Random Forest (RF)

In the training dataset in this study, a grid search was performed using a 10-bend CV approach on the re-sampled instruction dataset to catch the perfect hyperparameter using the n-estimator.

Xgboost (XGb)

In this study, a grid search was performed using a 10-bend CV approach on the re-sampled tester dataset to acquisition the values of the subsequent hyperconfines:

- **"n** estimator" determines the amount of trees or the amount of additional rounds.
- **"minimum (min)** child weight" determines the minimum sum of weights from all investigations needed on a child.
- **"max-depth"** is the maximum depth of the tree. Usually, the value lies in the range of 3-10.
- **"gamma"** is the minimum loss reduction required to perform the split. Splitting within a node is performed only when the resulting split gives a positive reduction in the loss function.
- **"subsample"** determines the fraction of observations that will be randomly sampled for each tree. Typically, the value lies in the range of 0.5-1.
- **"colsample bytree"** indicates the fraction of columns that will be randomly sampled for each tree.
- **"alpha"** denotes a regularization parameter.

Rounds of Testing-Tester

Below setting the hyperconfines for several prognostic approaches, the hyperconfines are applied to different approaches then the set is continued with re-sample training for every model as tester data. Thus, the model learns each arrangement in the re-sampling of practicing data. Next, the sets of approval that were split earlier when shattering the unified dataset are passed down to approve the typical work.

Evaluate the performance of the selected model

In this study, performance evaluation is used for models that use recall curves, precision, f1-scores, and areas of under-precision recall. In the analysis issues, the ROC curve is the main choice to calculate the performance of the model at different thresholds. However, if there is an unequal class distribution where the negative examples dwarf the number of positive examples, a precision-withdrawal curve will be more useful.

RESEARCH RESULT

LR - ***No re-sampling*** - *Best C parameter is 0.1*

Note: LR accomplished definitely in analyzing valid cases alike after re-sampling along precision scores (PS) is 0,9991, recall scores (RS) is 0.9995, and f1 scores (f1-S) is 0.9993, appropriately served in Table 2. The performance of this model was unsatisfactory when dealing with the scam class, where the PS gain 0,6774 and 0,5250 appropriately. Additionally, the PR AUC and ROC AUC values are also not good, it is shown appropriately in fig-9 and fig-10. The distraction cast of the LR-M is shown in fig-8, when no re-sampling method is used.

Class	Precision	Recall	F1 score	Support
0	0.9991	0.9995	0.9993	85283
1	0.6774	0.5250	0.5915	160
avg/total	0.9985	0.9986	0.9986	85443

Table 2 Evaluation metrics-LR-no resampling

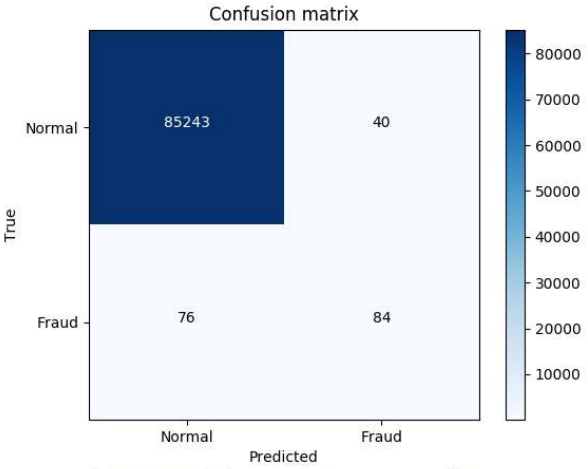


Figure 8 Confusion matrix-LR-no resampling

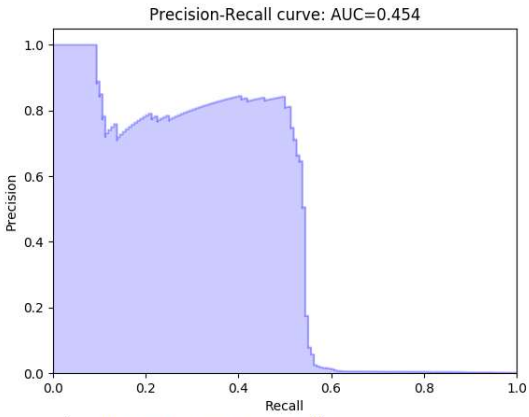


Figure 9 PR curve-LR-no resampling

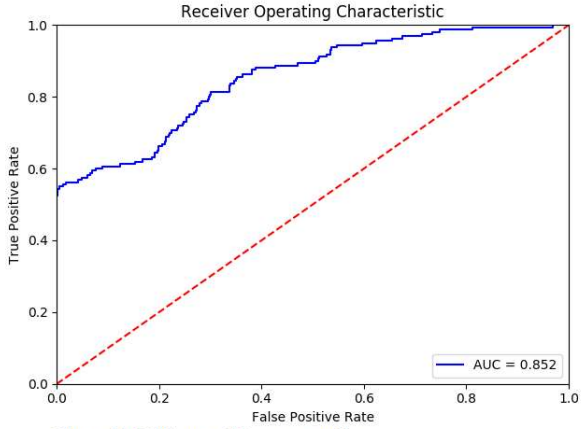


Figure 10 ROC curve-LR-no resampling

Ran-Under - *Best C-Parameter is 0.1*

Notes: As forecasted, this result is shown in table 3, LR performs definitely in characterizing the negative class, when ran-under is used. Nonetheless, in the case of the +class, the correctness is very poor, its decreases to 0.0857. While the recovery is a very good score which is 0.9062, its precision cannot be neglected. In addition, Figure 13 shows the ROC curve where ROC AUC very good this is 0.973 but in Figure 12 the PR curve present various description in which the precision is lowest during the RS is less than 0,90. The distraction cast by the LR-M during the ran-under approach is used shown in figure 11.

Class	Precision	Recall	F1 score	Support
0	0.9998	0.9819	0.9908	85283
1	0.0857	0.9062	0.1566	160
avg/total	0.9981	0.9817	0.9892	85443

Table 3 Evaluation metrics-LR-random undersampling

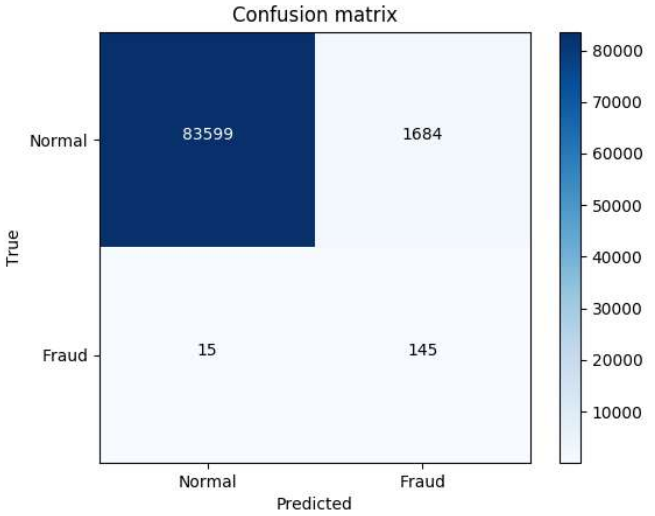


Figure11 Confusion matrix-LR-random undersampling

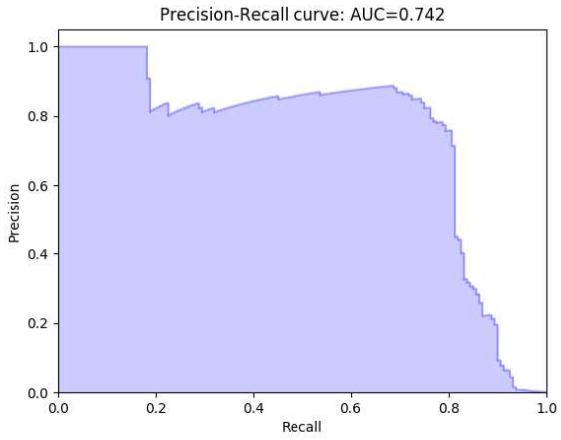


Figure 12 PR curve-LR-random undersampling

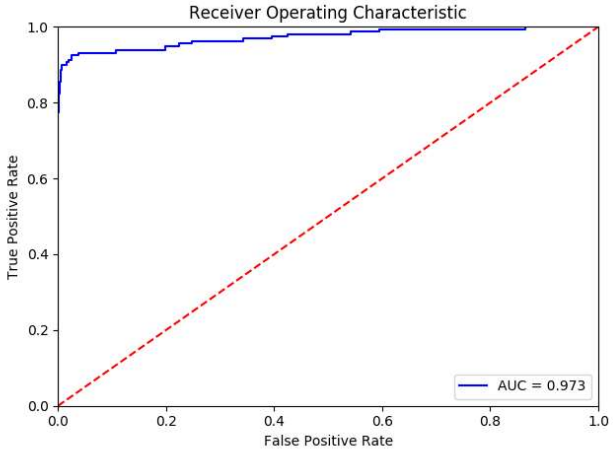


Figure 13 ROC curve-LR-random undersampling

Tomek unlinked - *Best C-Parameter is 0.1*

Notes: LR evaluation metrics are shown in table 4 when the tomek delink method was used. In addition, Figure 14, Figure 15, and Figure 16 show the distraction cast, ROC-curve, and PR-curve appropriately. Just like the two previous models, LR with Tomek link deletion works definitely in analyzing the adverse class. In comparison to ran-under, the precision increases considerably but the gain score decreases by a large margin.

Class	Precision	Recall	F1 score	Support
0	0.9991	0.9995	0.9993	85283
1	0.6746	0.5312	0.5944	160
avg/total	0.9985	0.9986	0.9986	85443

Table 4 Evaluation metrics-LR-tomek links removal

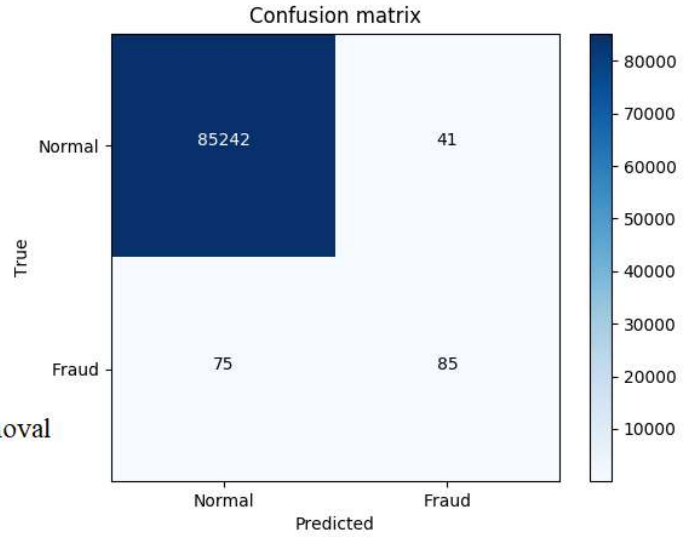


Figure 14 Confusion matrix-LR-tomek links removal

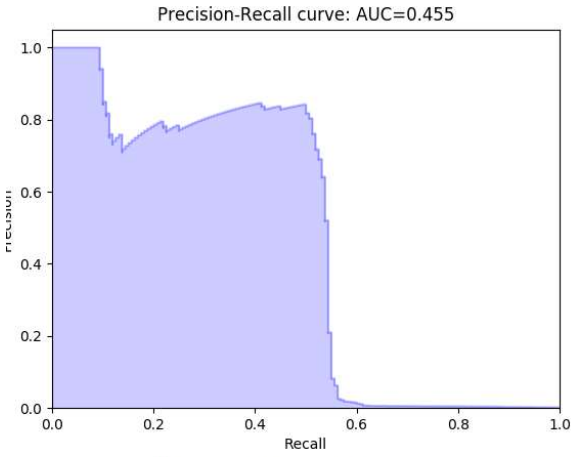


Figure 15 PR curve-LR-tomek links removal

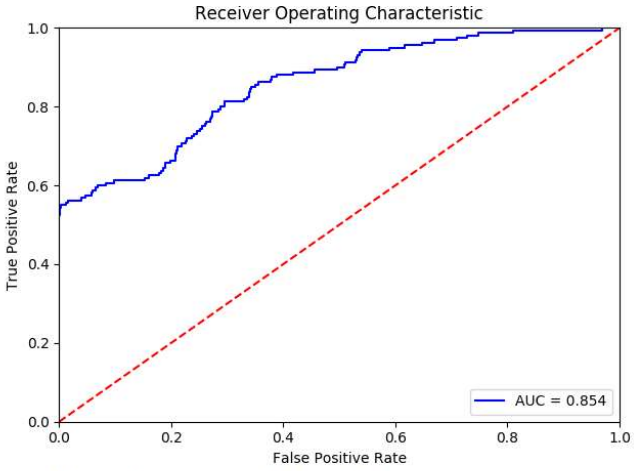


Figure 16 ROC curve-LR-tomek links removal

Ran-Over - *Best C-Parameter is 0.001*

Notes: The LR calculation metrics when the Ran-Over method is used are shown in table 5 above. In addition, cast, ROC-curve, and PR-curve respectively, are the distraction shown in Figure 17, Figure 18, and Figure 19. LR with Ran-Over executes definitely in items of anamnesis but performs the lowest in items of correctness, offering a poor f1-S of 0,1340.

Class	Precision	Recall	F1 score	Support
0	0.9998	0.9784	0.9890	85283
1	0.0724	0.9000	0.1340	160
avg/total	0.9981	0.9782	0.9874	85443

Table 5 Evaluation metrics-LR-random oversampling

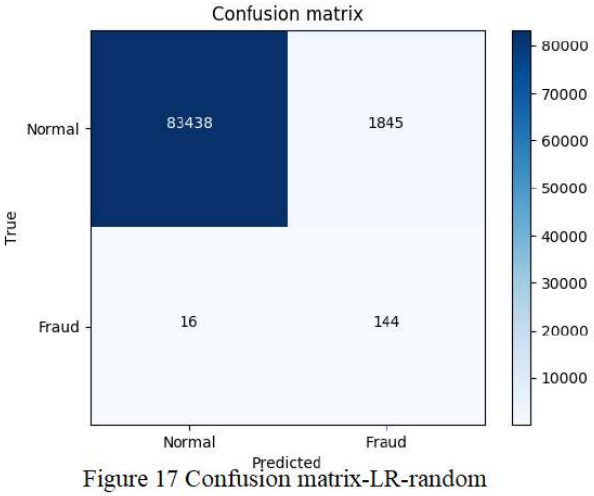


Figure 17 Confusion matrix-LR-random oversampling

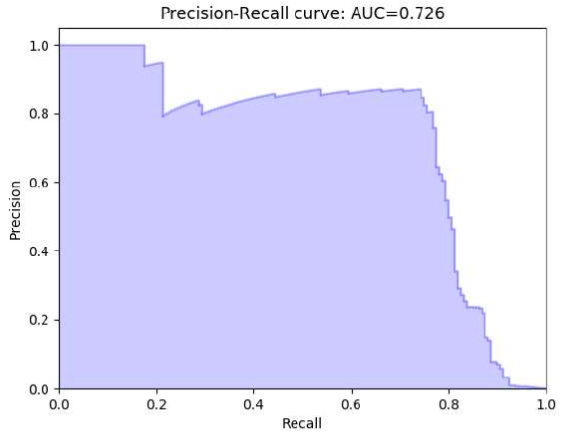


Figure 18 PR curve-LR-random oversampling

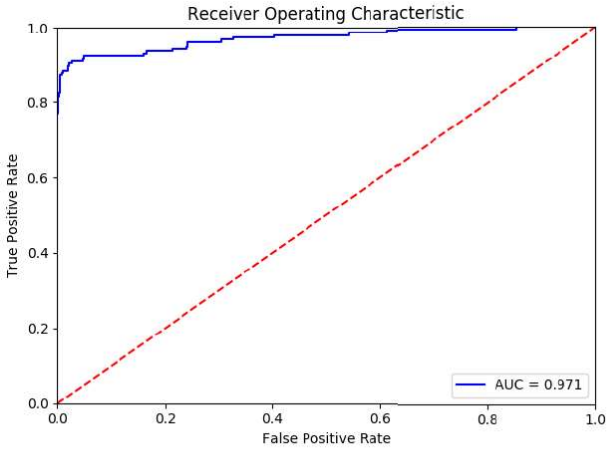


Figure 19 ROC curve-LR-random oversampling

SMOTE - *Best C-Parameter is 0.01*

Notes: Table 6 shows the LR evaluation metrics when the "Synthetic Minority Oversampling Technique" was used. Similarly, Figure 20 Figure 21, and Figure 22 show the confusion cast, PR curve, and ROC curve respectively. Using the synthetic minority over-sampling technique with LR also does not help improve performance because the f1 score is only 0.1695.

Class	Precision	Recall	F1 score	Support
0	0.9998	0.9840	0.9918	85283
1	0.0938	0.8812	0.1695	160
avg/total	0.9981	0.9838	0.9903	85443

Table 6 Evaluation metrics-LR-SMOTE

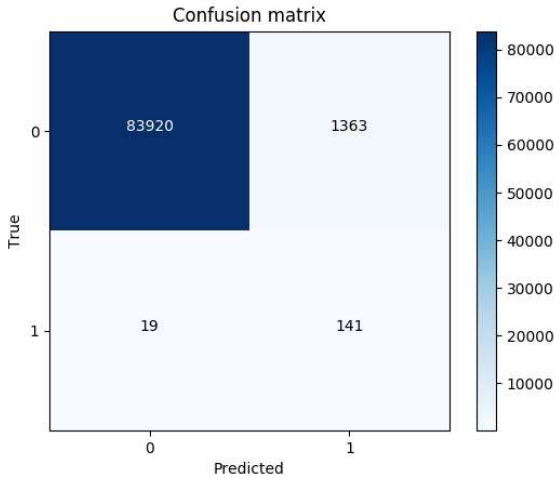


Figure 20 Confusion matrix-LR-SMOTE

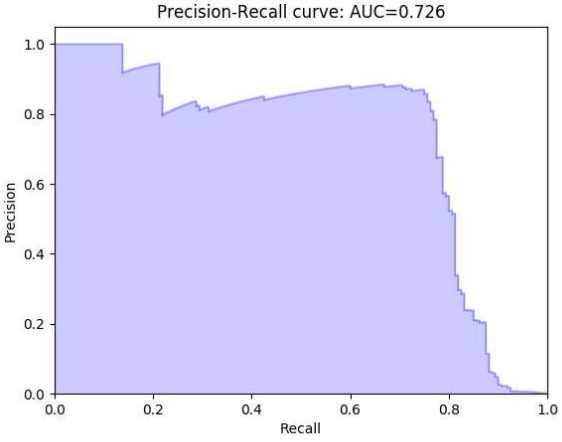


Figure 21 PR curve-LR-SMOTE

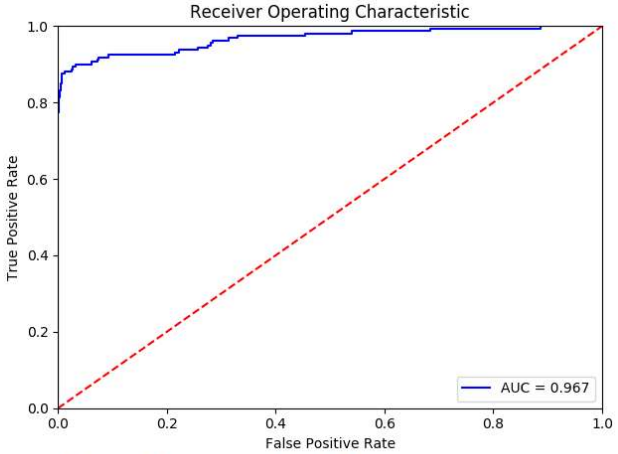


Figure 22 ROC curve-LR-SMOTE

SMOTE & Tomek unlink – with the *best C-Parameter is in 0.01*

Notes: Table 7 shows the LR evaluation metrics although a solution of the "Synthetic Minority Over-sampling Technique" and Tomek linker deletion was used. Figure 23 Figure 24 and Figure 25 show the confusion cast, PR curve, and ROC curve, respectively. Similarly, using a combination of SMOTE and Tomek unlink also doesn't improve the performance of the model at all, since the f1 score is only 0.15.

Class	Precision	Recall	F1 score	Support
0	0.9998	0.9803	0.9899	85283
1	0.0793	0.9062	0.1458	160
avg/total	0.9981	0.9801	0.9884	85443

Table 7Evaluation metrics-LR-SMOTE % tomek link removal

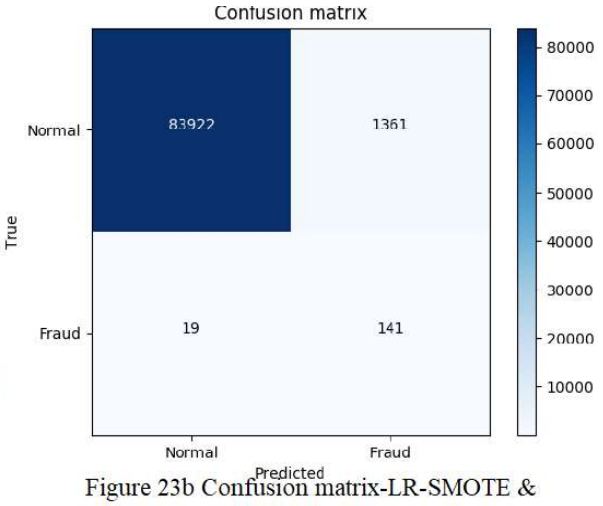


Figure 23b Confusion matrix-LR-SMOTE & tomek links removal

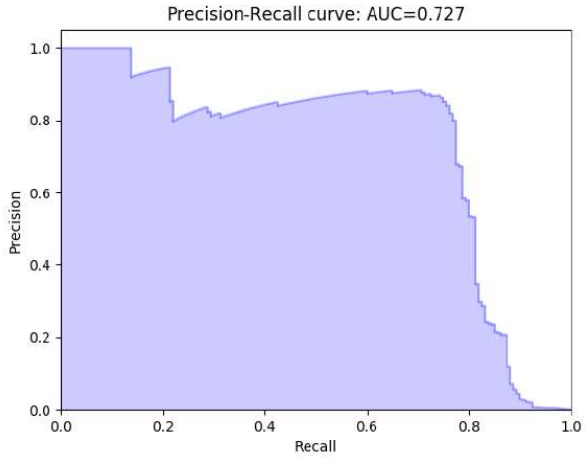


Figure 24 PR curve-LR-SMOTE & tomek links removal

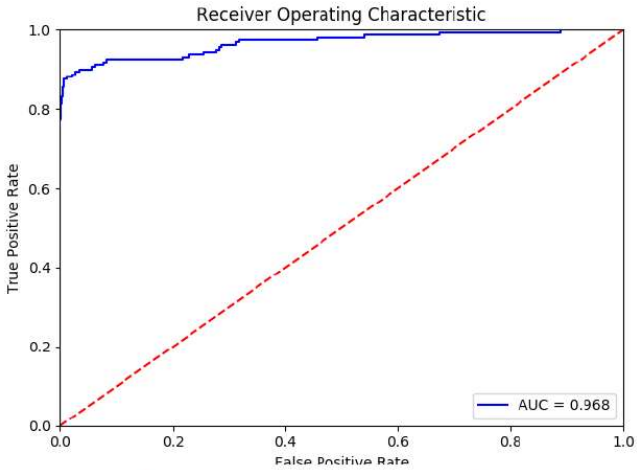


Figure 25 ROC curve-LR-SMOTE & tomek links removal

Random forest (RF) ***with un-resampling and the best hyperparameter is***

- *"Amount-of-trees"* = four hundred
- *"Max. features"* = automatic

Notes: Table 8 shows the random forest evaluation metrics when no resampling method is used. Figure 26, Figure 27, and Figure 28 show the distraction cast, PR curve, and ROC curve, respectively. The random forest performed very well in classifying the negative class and the positive class with an overall f1 score of 0,87 even without resampling. This demonstrates the power of the ensemble technique when used on unbalanced data sets. Nevertheless, here still would like to have a slightly higher draw score of 0,7937.

Class	Precision	Recall	F1 score	Support
0	0.9996	0.9999	0.9998	85283
1	0.9621	0.7937	0.8699	160
avg/total	0.9995	0.9996	0.9995	85443

Table 8 Evaluation metrics-RF-no resampling

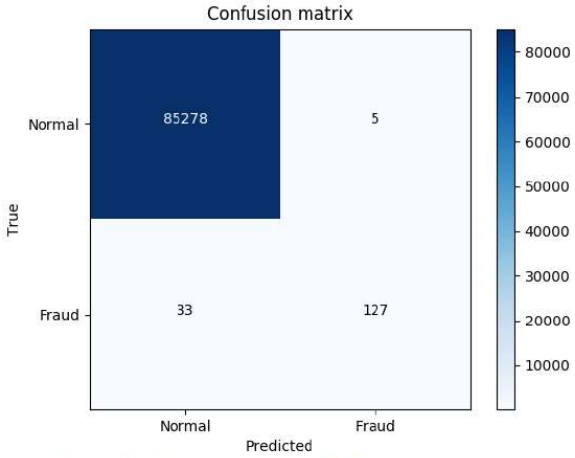


Figure 26 Confusion matrix-RF-no resampling

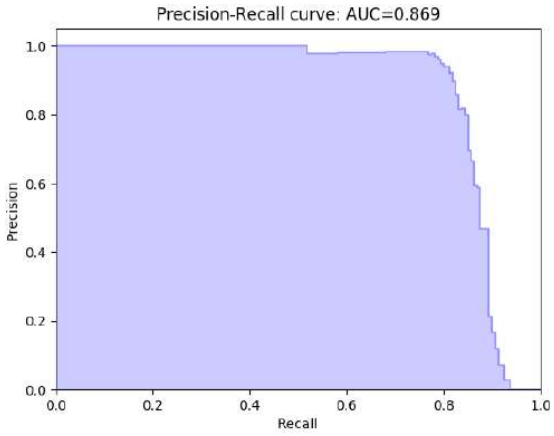


Figure 27 PR curve-RF-no resampling

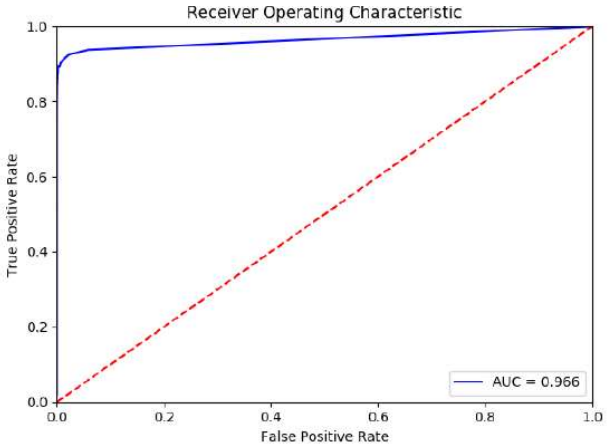


Figure 28 ROC curve-RF-no resampling

Ran-Under – With the best hyperparameter is:

- "*Amount-of-trees*" = four hundred
- "*Max.features*" = automatic

Notes: Table 9 shows the evaluation metrics of the random forest when Ran-Under is used. Figure 29, Figure 30, and Figure 31 show the confusion cast, PR curve, and ROC curve, respectively. The Ran-Under approach, when used with logistic regression, performs worst in terms of precision but does the contradictory in items of recall. In the same case, it happens in the case of RF too which terminates it as a bad classifier.

Class	Precision	Recall	F1 score	Support
0	0.9998	0.9637	0.9814	85283
1	0.0453	0.9187	0.0864	160
avg/total	0.9981	0.9636	0.9798	85443

Table 9 Evaluation metrics-RF-random undersampling

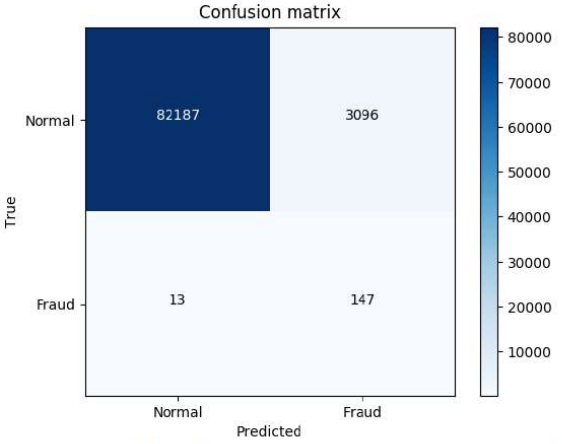


Figure 29 Confusion matrix-RF-random undersampling

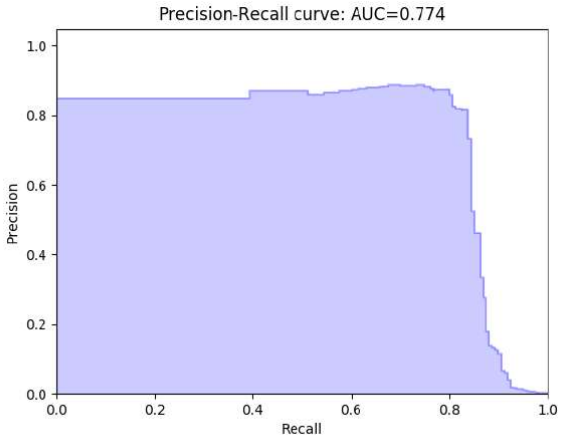


Figure 30 PR curve-RF-random undersampling

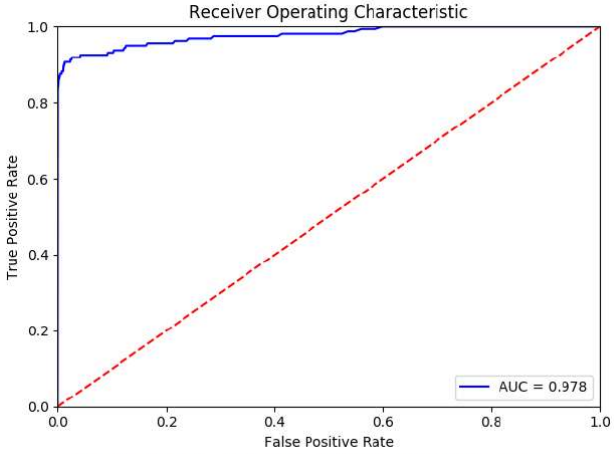


Figure 31 ROC curve-RF-random undersampling

Tomek unlinked - *With perfect hyperconfines is:*

- *"Total-of-trees"* = six hundred
- *"Max.features"* = automatic

Notes: Table 10 shows the random forest evaluation metrics when the tomek delink method was used. Figure 32, Figure 33, and Figure 34 show the distraction cast, PR curve, and ROC curve, respectively. The performance of the random forest when used with the tomek unlinking approach is very similar to the performance without resampling. The precision and recall are pretty high, but still, we'd like the memory score to be a bit higher than that. The area under the score curve for the PR curve and the operating characteristics of the receiver is also quite high.

Class	Precision	Recall	F1 score	Support
0	0.9996	0.9999	0.9997	85283
1	0.9185	0.7750	0.8407	160
avg/total	0.9994	0.9994	0.9994	85443

Table 10 Evaluation metrics-RF-tomek links removal

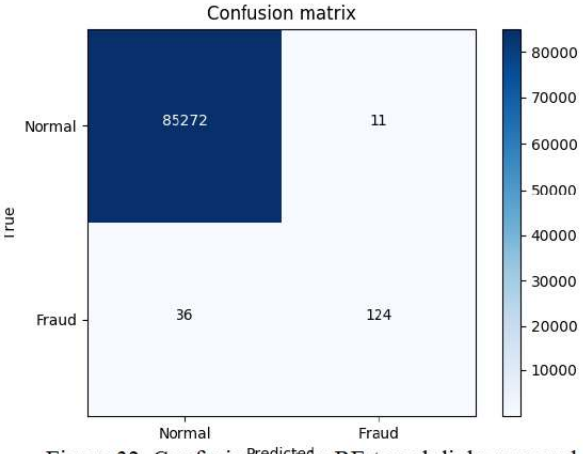


Figure 32 Confusion matrix-RF-tomek links removal

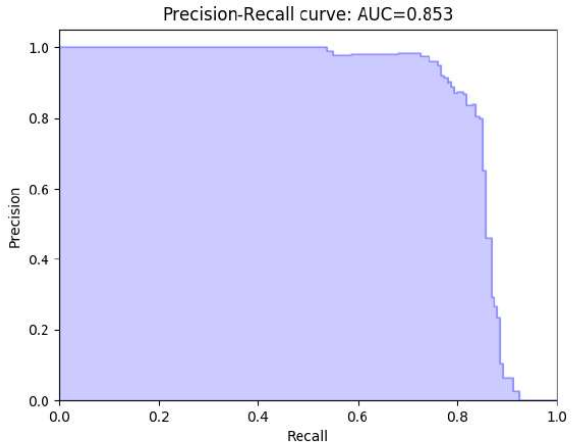


Figure 33 PR curve-RF-tomek links removal

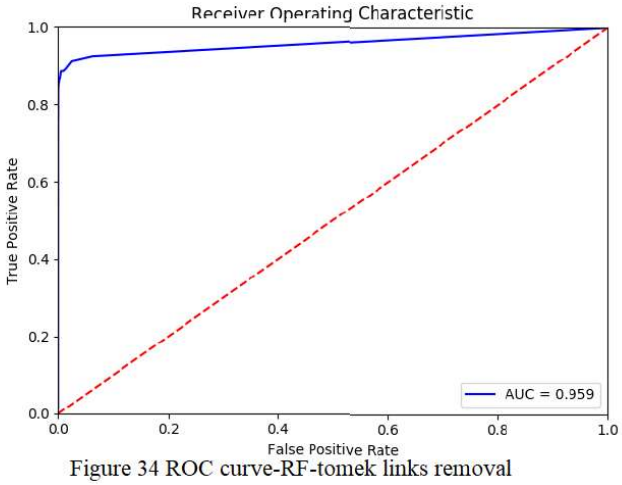


Figure 34 ROC curve-RF-tomek links removal

Ran-Over – *With Best hyperconfines are:*

- *"Number of trees"* = 800
- *"Max features"* = auto

Notes: Table 11 shows the random forest evaluation metrics when Ran-Over is used. Figure 35, Figure 36, and Figure 37 show the distraction cast, PR curve, and ROC curve, respectively. Similar to Tomek link removal, the PS is high and the RS is decent. But we want the memory score to be slightly higher than that.

Class	Precision	Recall	F1 score	Support
0	0.9996	1.0000	0.9998	85283
1	0.9685	0.7688	0.8571	160
avg/total	0.9995	0.9995	0.9995	85443

Table 11 Evaluation metrics-RF-random oversampling

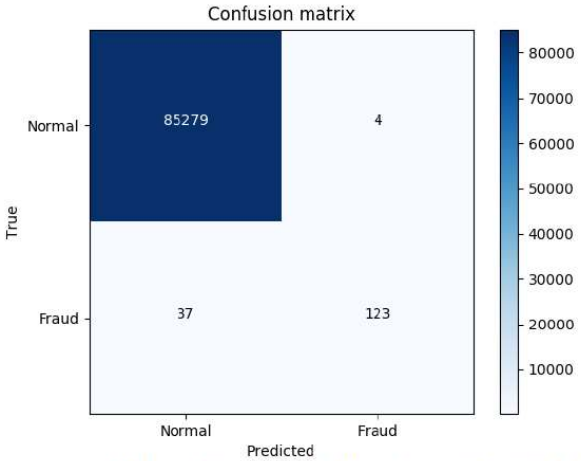


Figure 35 Confusion matrix-RF-random oversampling

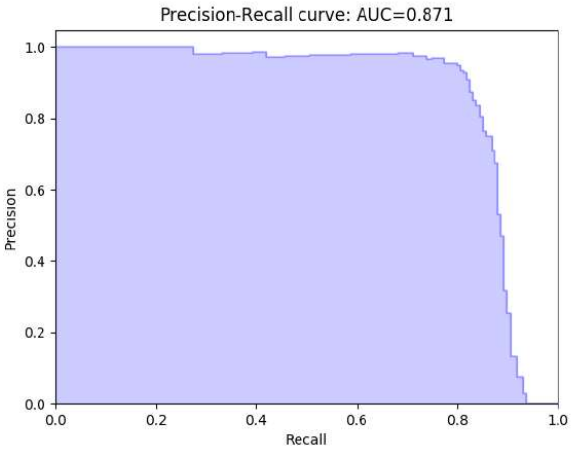


Figure 36 PR curve-RF-random oversampling

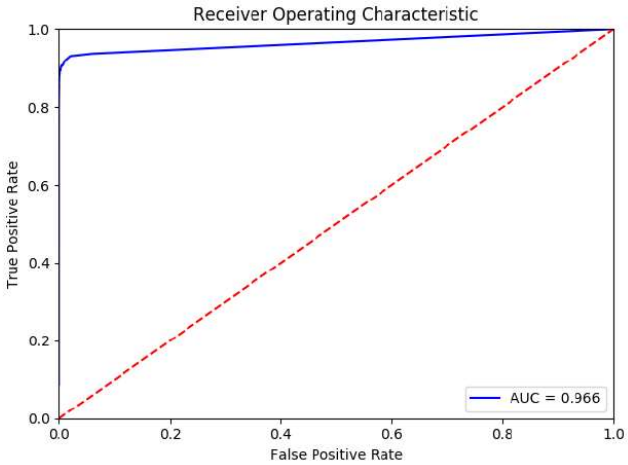


Figure 37 ROC curve-RF-random oversampling

SMOTE – *with the perfect hyperconfines:*

- *"Amount of trees"* = 600
- *"Max features"* = auto

Notes: Table 12 shows RF evaluation metrics when SMOTE was used. Figure 38, Figure 39, and Figure 40 show the confusion cast, PR curve, and ROC curve, respectively. This time we got a higher memory score, which is good, however, the PS decreased by a considerable amount. We want to have a balance between recall and precision.

Class	Precision	Recall	F1 score	Support
0	0.9998	0.9992	0.9995	85283
1	0.6814	0.8688	0.7637	160
avg/total	0.9992	0.9990	0.9991	85443

Table 12 Evaluation metrics-RF-SMOTE

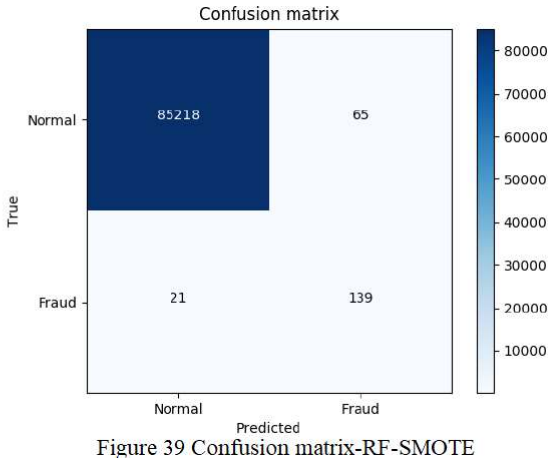


Figure 39 Confusion matrix-RF-SMOTE

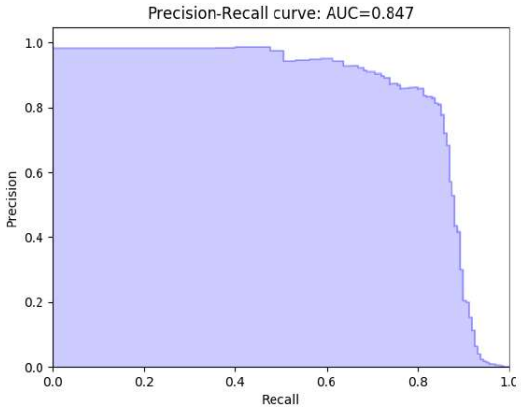


Figure 40 PR curve-RF-SMOTE

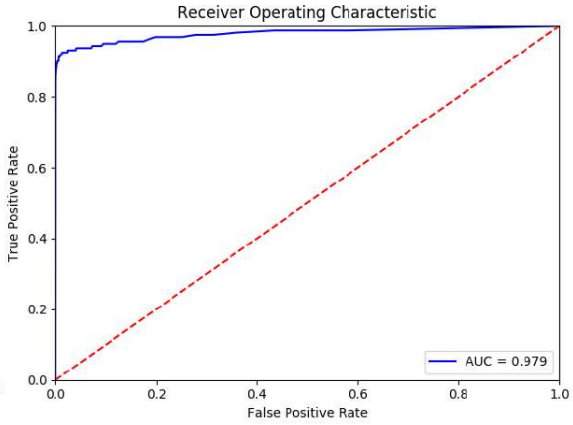


Figure 41 ROC curve-RF-SMOTE

SMOTE & Tomek unlink – *With the Best hyperconfines are:*

- *"Amount of trees"* = 400
- *"Maximum features"* = auto

Notes: Table 13 shows the RF evaluation metrics when a combination of SMOTE and Tomek unlinking was used. Figure 41, Figure 42, and Figure 43 show the confusion cast, PR curve, and ROC curve, respectively. Random forest, when used with the combination of SMOTE and Tomek unlinking, performed very well given the 0.84 PS and 0,84 RS, which can also be seen from the higher PR AUC and ROC AUC scores. Random forest works very well when used with any resampling technique compared to logistic regression. This shows that ensemble techniques such as random forest are very effective when used to classify unbalanced data.

Class	Precision	Recall	F1 score	Support
0	0.9997	0.9997	0.9997	85283
1	0.8438	0.8438	0.8438	160
avg/total	0.9994	0.9994	0.9994	85443

Table 13 Evaluation metrics-RF-SMOTE & tomek link removal

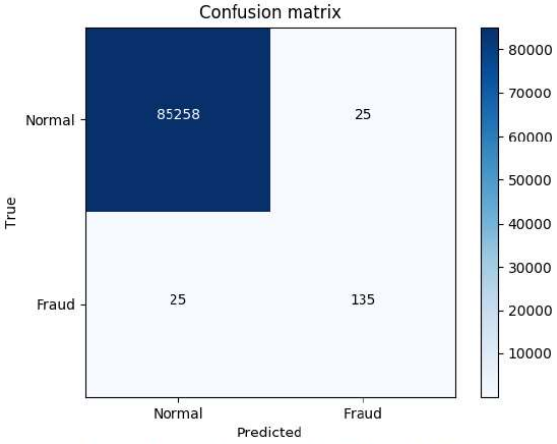


Figure 41Confusion matrix-RF-SMOTE & tomek links removal

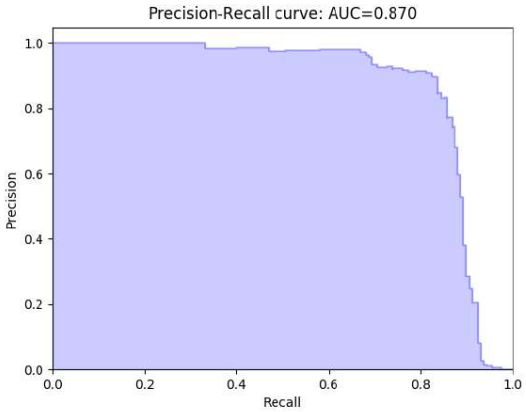


Figure 42 PR curve-RF-SMOTE & tomek links removal

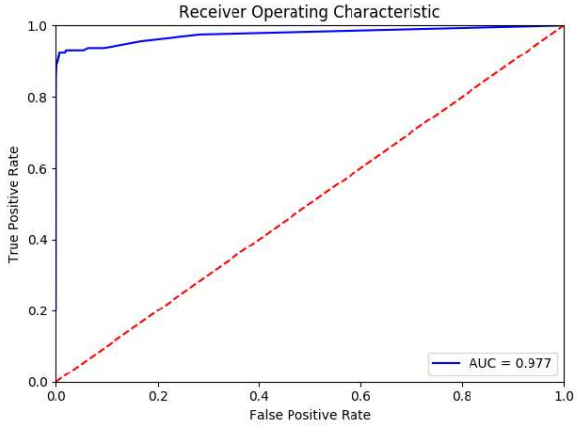


Figure 43 ROC curve-RF-SMOTE & tomek links removal

XGBoost (XGB) – *With no resampling and the best hyperparameter is:*

- *"Learning rate"* = 0,1
- *"Amount of estimator"* = 1000
- *"Max.depth"* = 6
- *"Min.child-weight"* = 6
- *"Sub-sample"* = 0.85
- *"Colsample bytree"* = 0.75
- *"Alpha"* = 1e-5
- *"Gamma"* = 0.0

Note: XGBoost calculation metrics when nothing re-sampling method used shown in Table 14. Figure 43, Figure 44, and Figure 45 show the distraction cast, PR curve, and ROC curve, appropriately. XGBoost performs very well in classifying the positive class in terms of precision even without resampling. The gain score is not very high compared to the random forest and can be improved.

Class	Precision	Recall	F1 score	Support
0	0.9995	0.9999	0.9997	85296
1	0.9291	0.7375	0.8223	147
avg/total	0.9994	0.9994	0.9994	85443

Table 14 Evaluation metrics-XGB-no resampling

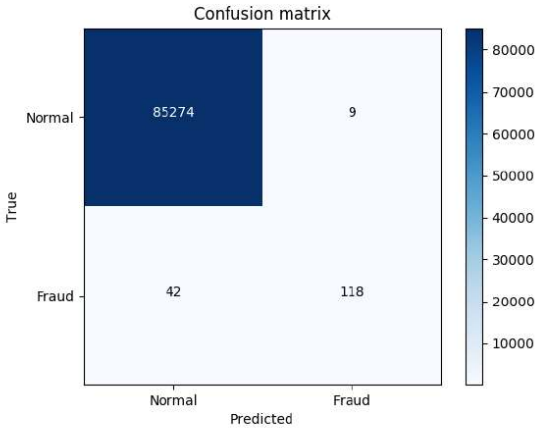


Figure 43 Confusion matrix-XGB-no resampling

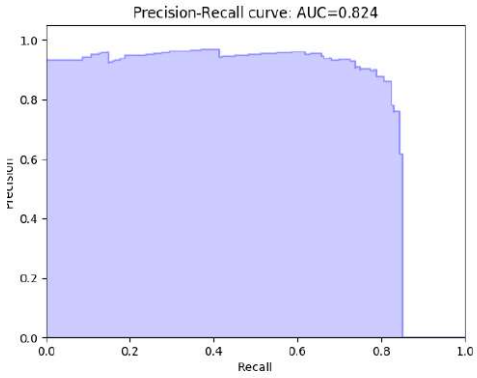


Figure 44 PR curve-XGB-no resampling

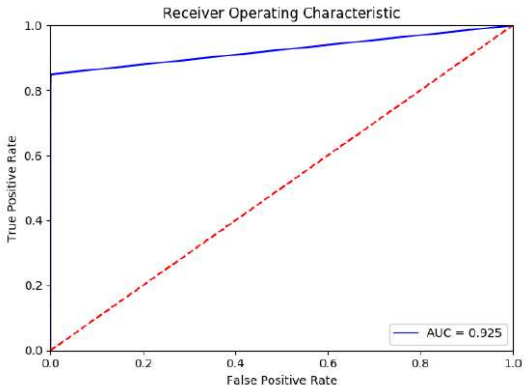


Figure 45 ROC curve-XGB-no resampling

Ran-Under – *with the perfect hyperconfines are:*

- *"Learning rate"* = 0,1
- *"Number of estimators"* = 50
- *"Max.dept"* = 5
- *"Min.child-weight"* = 4
- *"Subsample"* = 0.85
- *"Colsample bytree"* = 0.8
- *"Alpha"* = 1
- *"Gamma"* = 0.1

Notes: The XGBoost calculation metrics when the Ran-Under method is used are shown in Table 15. Figure 46, Figure 47, and Figure 48 show the distraction cast, PR curve, and ROC curve, appropriately. XGBoost, when used with Ran-Under, performed the worst in terms of precision despite very good gain scores. So far all classifiers have shown similar results when Ran-Under is used to address the class inequality dilemma.

Class	Precision	Recall	F1 score	Support
0	0.9998	0.9594	0.9792	85283
1	0.0397	0.8938	0.0760	160
avg/total	0.9980	0.9593	0.9775	85443

Table 15 Evaluation metrics-XGB-random undersampling

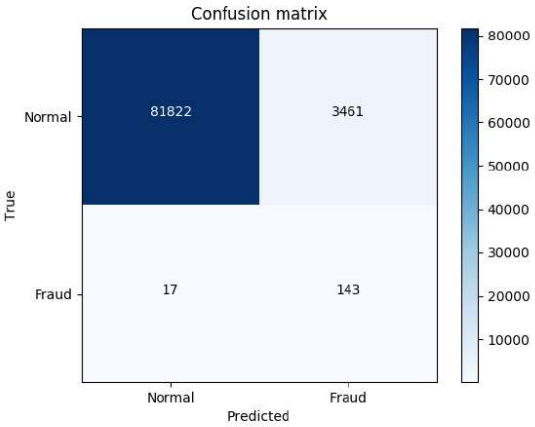


Figure 46 Confusion matrix-XGB-random undersampling

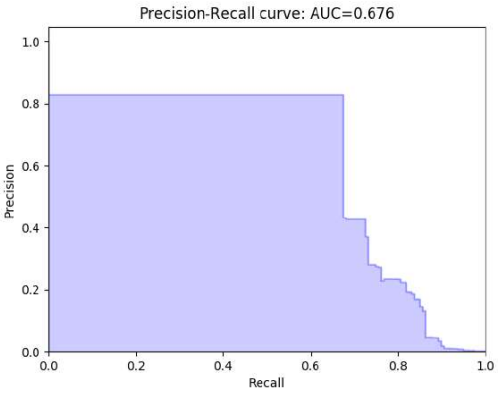


Figure 47 PR curve-XGB-random undersampling

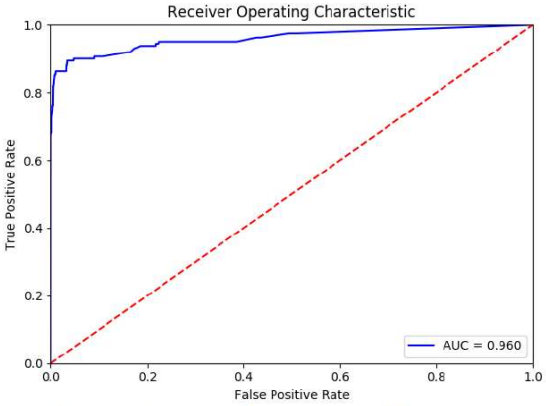


Figure 48 ROC curve-XGB-random undersampling

Tomek unlinked – *with the best hyperparameter:*

- *"Learning rate"* = 0.1
- *"Number-of-estimators"* = 115
- *"Max.depth"* = 4
- *"Min.child-weigh"* = 5
- *"Subsample"* = 0.65
- *"Colsample bytree"* = 0.6
- *"Alpha"* = 1e-5
- *"Gamma"* = 0.2

Notes: The XGBoost calculation metrics when the tomek unlink method is used are shown in Table 16. Figure 49, Figure 50, and Figure 51 show the distraction cast, PR curve, and ROC curve, appropriately. There is a large increase in precision when XGBoost is used with the tomek unlinking approach, compared to the Ran-Under technique. However, recalls have decreased quite a lot.

Class	Precision	Recall	F1 score	Support
0	0.9995	0.9998	0.9996	85283
1	0.8702	0.7125	0.7835	160
avg/total	0.9992	0.9993	0.9992	85443

Table 16 Evaluation metrics-XGB-tomek links removal

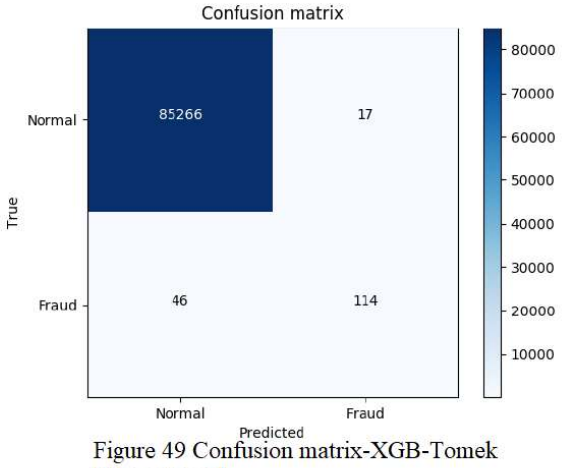


Figure 49 Confusion matrix-XGB-Tomek links removal

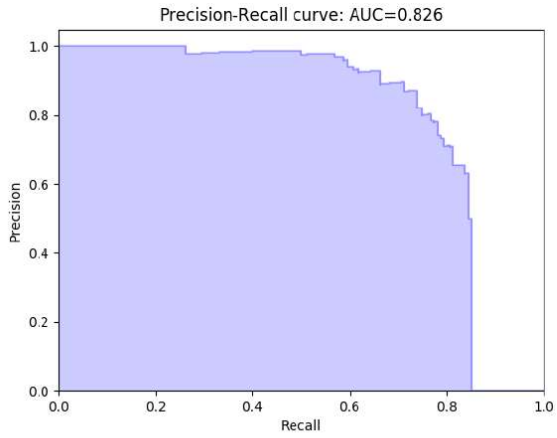


Figure 50 PR curve-XGB-tomek links removal

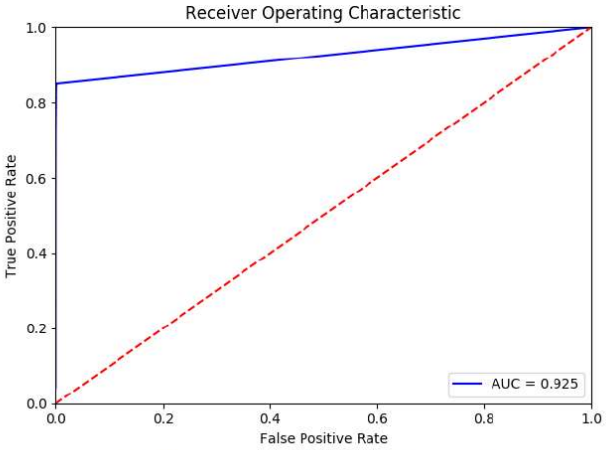


Figure 51 ROC curve-XGB-tomek links removal

Ran-Over – *With the perfect hyperconfines:*

- *"Learning rate"* = 0.1
- *"Number of estimators"* = 572
- *"Max.depth"* = 5
- *"Min.child-weight"* = 6
- *"Subsample"* = 0.75
- *"Colsample bytree"* = 0.65
- *"Alpha"* = 1e-5
- *"Gamma"q* = 0.1

Note: The XGBoost calculation metrics when the Ran-Over method is used are shown in Table 17. Figure 52, Figure 53, and Figure 54 show the distraction cast, PR curve, and ROC curve, appropriately. XGBoost with Ran-Over does quite well in terms of score precision and gain.

Class	Precision	Recall	F1 score	Support
0	0.9997	0.9996	0.9997	85283
1	0.8024	0.8375	0.8196	160
avg/total	0.9993	0.9993	0.9993	85443

Table 17 Evaluation metrics-XGB-random oversampling

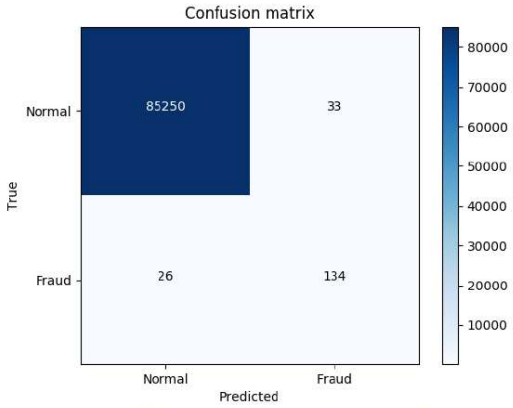


Figure 52 Confusion matrix-XGB-random oversampling

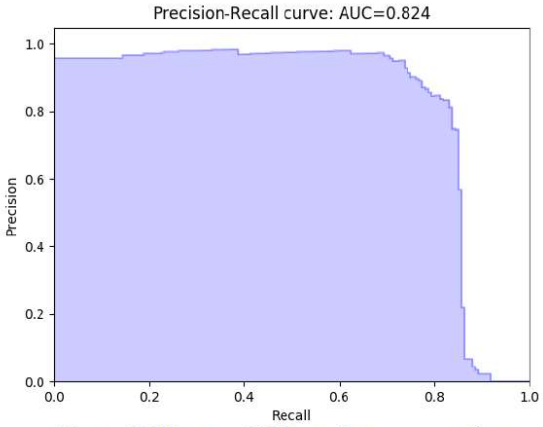


Figure 53 PR curve-XGB-random oversampling

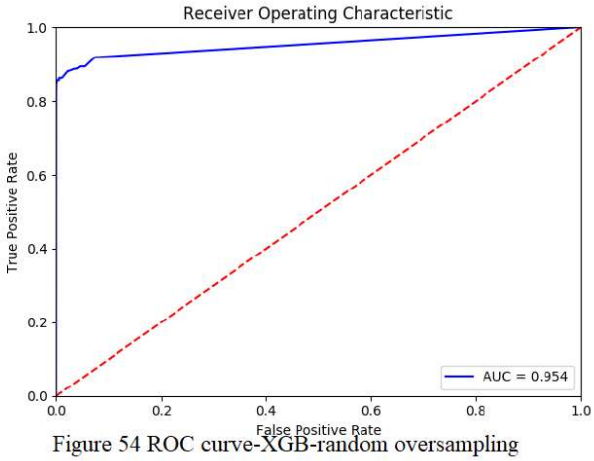


Figure 54 ROC curve-XGB-random oversampling

SMOTE – with the *Best hyperparameter:*

- *"Learning rate"* = 0.1
- *"Total estimator"* = 870
- *"Max.depth"* = 10
- *"Min.child-weight"* = 6
- *"Subsample"* = 0.7
- *"Colsample bytree"* = 0.95
- *"Alpha"* = 1e-5
- *"Gamma"* = 0.1

Notes: the XGBoost calculation metrics when the Synthetic Minority Oversampling approach is used as shown in table 18. The distraction of cast, ROC curve, and, PR curve respectively can be seen in Figure 55, Figure 56, and Figure 57. There is a significant drop in precision when using SMOTE with XGBoost. But the memory score is very high.

Class	Precision	Recall	F1 score	Support
0	0.9998	0.9985	0.9992	85283
1	0.5303	0.8750	0.6604	160
avg/total	0.9989	0.9983	0.9985	85443

Table 18 Evaluation metrics-XGB-SMOTE

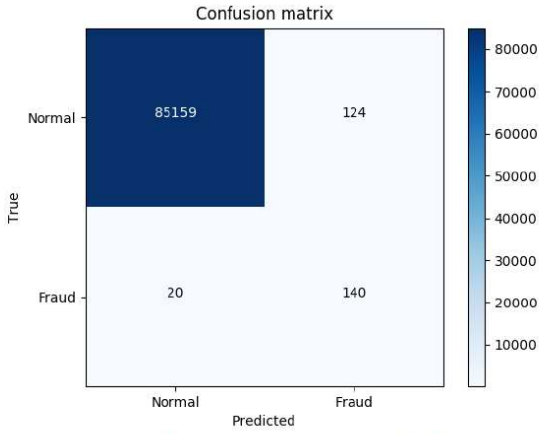


Figure 55 Confusion matrix-XGB-SMOTE

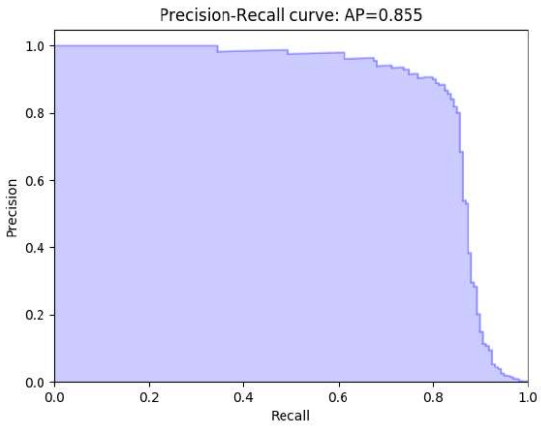


Figure 56 PR curve-XGB-SMOTE

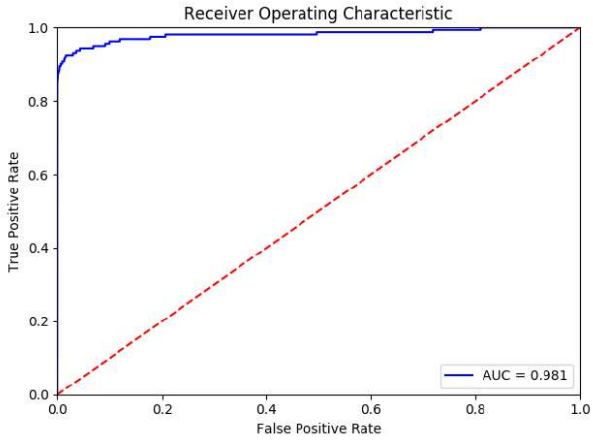


Figure 57 ROC curve-XGB-SMOTE

SMOTE & Tomek unlink – With the *Best hyperparameter:*

- *"Learning rate"* = 0.1
- *"Total estimator"* = 600
- *"Max.depth"* = 7
- *"Min.child-weight"* = 6
- *"Subsample"* = 0.7
- *"Colsample bytree"* = 0.85
- *"Alpha"* = 1e-5
- *"Gamma"* = 0.1

Notes: The calculation metrics of XGBoost when the combination of SMOTE and Tomek unlink is used are shown in Table 19. Figure 58, Figure 59, and Figure 60 show the distraction cast, PR curve, and ROC curve, appropriately. XGBoost with the combination of SMOTE link and Tomek does quite well in terms of memory. However, the PS is very low.

Class	Precision	Recall	F1 score	Support
0	0.9998	0.9934	0.9966	85283
1	0.2023	0.8875	0.3295	160
avg/total	0.9983	0.9932	0.9954	85443

Table 19 Evaluation metrics-XGB-SMOTE & tomek link removal

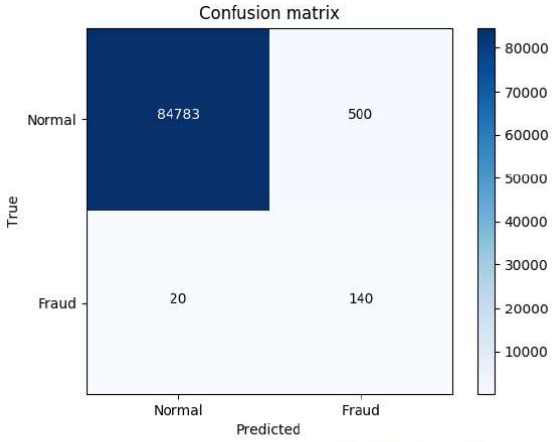
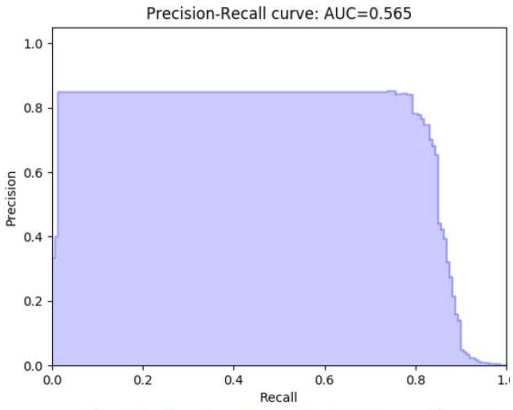
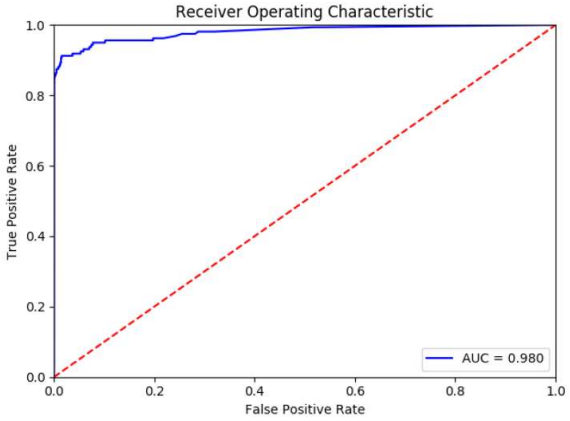


Figure 58 Confusion matrix-XGB-SMOTE & tomek links removal



Gambar 59 PR curve-XGB-SMOTE & penghapusan tautan tomek



Gambar 60 Penghapusan tautan ROC curve-XGB-SMOTE & tomek

Results summary

Table 20 serves that RF and XGBoost even on the outside of re-sampling perform better than the LR-M as long as their comprehensive f1-S. This determines the durabilities of altogether techniques that could bring greater achievement alike in the appearance of class inequality dilemmas. Each model, when used with Ran-Under, gives a good memory score but fails miserably in terms of precision. Compared to the ensemble model without resampling, the precision and RS did not improve in cases where Tomek linker removal was used. Random forest and xgboost perform quite well in terms of f1-S when Ran-Over is used. SMOTE increases the random forest draw score and xgboost but the PS decreases slightly. When a hybrid consolidation of SMOTE and Tomek delinker was tested with random forest, it gave a fair correctness and RS of 0.84 and PR-AUC of 0,870. Even though in case the same re-sampling approach is used, xgboost fall to establish the PS. Furthermore, here still can see that there are only slight differences in the ROC-AUC in the middle of the models.

Classifier	Resampling method	Precision	Recall	F1 Score	PR	ROC
Logistic regression	No resampling	0.68	0.53	0.59	0.454	0.852
	Random undersampling	0.09	0.91	0.16	0.729	0.973
	Tomek links removal	0.67	0.53	0.59	0.455	0.854
	Random oversampling	0.07	0.90	0.13	0.726	0.971
	SMOTE	0.09	0.88	0.17	0.726	0.967
	SMOTE + Tomek links removal	0.08	0.91	0.15	0.727	0.973
Random forest	No resampling	0.96	0.79	0.87	0.869	0.966
	Random undersampling	0.05	0.92	0.09	0.774	0.978
	Tomek links removal	0.92	0.78	0.84	0.853	0.959
	Random oversampling	0.97	0.77	0.86	0.871	0.966
	SMOTE	0.68	0.87	0.76	0.847	0.979
	SMOTE + Tomek links removal	0.84	0.84	0.84	0.870	0.977
XGBoost	No resampling	0.93	0.74	0.82	0.824	0.925
	Random undersampling	0.04	0.89	0.08	0.676	0.960
	Tomek links removal	0.87	0.71	0.78	0.826	0.925
	Random oversampling	0.80	0.84	0.82	0.824	0.954
	SMOTE	0.53	0.88	0.66	0.855	0.981
	SMOTE + Tomek links removal	0.22	0.88	0.35	0.565	0.980

Table 20 Summary of results

CONCLUSIONS AND FUTURE WORK

ML Techniques adopted to forecast scams of CC transactions. The data set used in this study was 284.8K and only 492 outgoing transactions were scammed so that dataset is very unstable as long as the specific class is donated 0.172 percent only of the overall transactions. Although inputting deeply unequal class circulation data into a prognostic approach. The approach tends to be partial toward the greater number of samples. As the result, it tends to mistake scam transactions for genuine transactions. To overcome this issues, a data-level approach that includes various resampling techniques namely, random undersampling, Tomek unlinking, random over-sampling, SMOTE, and a hybrid method of “SMOTE resampling” and “Tomek delinking”. Moreover, in this study, an algorithmic approach similar to sacking and additions is applied to overcome the class inequality dilemma. Because of this, the random forest approach was chosen as the sacking approach and XGBoost was chosen as the addition method. In addition to these models, the LR-M was also chosen to be compared with other models. Next, the model is analyzed using re-sampling and without re-sampling approach. The correlation outcome disclosed that the random forest in consolidation with the SMOTE blend re-sampling models and Tomek linker deletion accomplished better than the other models.

For the next research in the future, a fee-susceptible research method can be applied seeing cost misclassifications. The fee to misclassify a crooked class as a valid class corresponding to the amount of scam is much larger than the fee to misclassify a valid class as a crooked class according

to the fee associated with identification transactions and calling cardholders. Formerly, this sample of literature is concerned with allocating examples that have minimal normal costs. CC scam is relevant to the non-static quality of the distribution of transactions where scam perpetrators usually always come up with new ways to try these scam activities. So for future research, it is very necessary to consider this behavior change and it is also very important to develop predictive models. In addition to this, conducting future research requires much larger data so that detailed studies on handling non-static properties in the detection of CC scam can be carried out better.

REFERENCE

Abhinav Srivastava, Amlan Kundu, Shamik Sural, and Arun K. Majumdar. CC scam detection using the hidden Markov model. 2011 World Congress on Information and Communication Technologies, pages 1062-1066, 2008.

Andrea Dal Pozzolo, Giacomo Boracchi, Olivier Caelen, Cesare Alippi, and Gianluca Bontempi. CC scam detection and concept-drift adaptation with delayed supervised information. 2015 International Joint Conference on Neural Networks (IJCNN), pages 1-8, 2015.

Andrea Dal Pozzolo, Olivier Caelen, Reid A. Johnson, and Gianluca Bontempi. Calibrating probability with undersampling for unbalanced classification. 2015 IEEE Symposium Series on

Andrea Dal Pozzolo, Olivier Caelen, Yann-Ael Le Borgne, Serge Waterschoot, and Gianluca Bontempi. Learned lessons in CC scam detection from a practitioner's perspective. Expert System. Appl., 41:4915-4928, 2014.

Andrea Dal Pozzolo. Adaptive Machine Learning for CC Scam Detection. Ph.D. thesis, 2015.

Andrew P. Bradley. The use of the area under the roc curve in the evaluation of machine learning algorithms. Pattern Recognition, 30:1145-1159, 1997.

Bertrand Lebichot, Fabian Braun, Olivier Caelen, and Marco Saerens. A graph-based, semi-supervised, CC scam detection system. in COMPLEX NETWORKS, 2016.

Emin Aleskerov, Bernd Freisleben, and R. Bharat Rao. Card watch: a neural network-based database mining system for CC scam detection. in CIFEr, 1997.

Gustavo EAPA Batista, Ronaldo C. Prati, and Maria Carolina Monard. A study of the behavior of several methods for balancing machine learning training data. SIGKDD Explorations, 6:20-29, 2004.

John O. Awoyemi, Adebayo Olusola Adetunmbi, and Samuel Adebayo Oluwadare. CC scam detection using machine learning techniques: A comparative analysis. 2017 International Conference on Computing Networking and Informatics (ICCNI), pages 1-9, 2017.

Kevin W. Bowyer, Nitesh V. Chawla, Lawrence O. Hall, and W. Philip Kegelmeyer. Smote: Synthetic minority over-sampling technique. J. Artif. Intell. Res., 16:321-357, 2002.

Mock, TJ, Srivastava, RP, & Wright, AM (2017). Scam Risk Assessment Using the Scam Risk Model as a Decision Aid. Mock, TJ, Srivastava, RP & Wright, AM 2017, 'Scam Risk Assessment Using the Scam Risk Model as a Decision Aid', Journal of Emerging Technologies in Accounting,

Vol. 14, No. 1, pp. 37-56. Https://Doi.Org/10.2308/Jeta-51724 . https://cris.maastrichtuniversity.nl/en/publications/0c1747fa-e8e8-43cc-8f36-9c57a05c5963

Nathalie Japkowicz and Shaju Stephen. The class inequality dilemma: A systematic study. Intelligent Data Analysis, pages 429-449, 2002.

Neil Liberty. Decision trees and random forests towards data science, Jan 2017.

Pedro M. Domingos. Metacost: A general method for making classifiers cost-sensitive. in KDD, 1999.

Piotr Juszczak, Niall M. Adams, David J. Hand, Christopher Whitrow, and David John Weston. Off-the-peg and bespoke classifiers for scam detection. Computational Statistics Data Analysis, 52:4521-4532, 2008.

Richard Wheeler and J. Stuart Aitken. Multiple algorithms for scam detection. Knowledge-Based System, 13:93-99, 2000.

Robert C. Holte, L. Acker, and B. Porter. Concept learning and the problem of small disjuncts. In Proceedings of the Eleventh International Joint Conference on Artificial Intelligence (IJCAI-89), pages 813-818, Detroit, MI, 1989.

Sam Maes, Karl Tuyls, Bram Vanschoenwinkel, and Bernard Manderick. CC scam detection using bayesian and neural networks. In In: Maciunas RJ, editor. Interactive image-guided neurosurgery. American Association of Neurological Surgeons, pages 261-270, 1993.

SeattleDataGuy. Additioning and sacking: How to develop a robust machine learning algorithm, Nov 2017.

Thomas G. Dietrich. Multiple classifier systems. In Lecture Remarks in Computer Science, 2000.

Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree additioning system. in KDD, 2016.