Accuracy Analysis on Credit Card Fraud Detection using Stacking Ensemble in Machine Learning

Jerome E. Lacieras
College of Engineering, Architecture and Fine Arts Department
Bachelor of Science in Computer Engineering
Batangas, Philippines
jerome.lacieras@g.batstate-u.edu.ph

***Abstract*—Credit card fraud is one of the growing banking transaction problems that is faced everyday by cardholder. It is a type of identity theft that involves the unlawful use of another person's credit card information to make transactions or withdraw monies from their account. There are fraud detection solutions have been proposed** [1]**,** [2] **and** [3] **that helps to distinguish the if the transaction was legit or fraud. However, the accuracy of classifying the transaction is one case that needs to be improve. This paper aims to develop a model using Stacking Ensemble that will help to analyze the accuracy of the Credit Card Fraud Detection. This study will utilize the use of algorithm such as K-Nearest neighbors and Decision Tree algorithm as a base classifier and a stacked model. It uses Grid Search for Hyperparameter tuning. The proponent hope that by the end of the paper, the reader will have better understanding in analyzing the accuracy of Credit card fraud detection using stacking ensemble.**

Keywords—credit card, fraud detection, fraud, transaction, decision tree, K-nearest neighbors, Grid Search

I. INTRODUCTION

The rapid evolution in terms of internet finance, electronic transfer and rapid expansion of credit card business, has bought the use of credit cards in mainstream. However, there are risk associated with the use of credit cards and also to the card holder [4]. Credit card fraud is one of the growing banking transaction problems that is faced everyday by cardholder. It is a type of identity theft that involves the unlawful use of another person's credit card information to make transactions or withdraw monies from their account. It happens in either online and offline transactions without the actual cardholder being aware of it [1][5].

S. K. Saddam Hussain, E. Sai Charan Reddy, K. G. Akshay and T. Akanksha [1] has made a study on fraud detection using SVM (support vector machine) and Random Forest algorithms in which the techniques' performance is judged based on precision sensitivity, & accuracy. It is proposed to combine the advantages of both methods to overcome their disadvantages and provide a more exact and accurate result for detecting fraud in the given dataset. It uses averaging to determine the exact number of fraudulent transactions in the dataset. M. R. Dileep, A. V. Navaneeth and M. Abhishek [2] also made a study which focuses on using Decision Tree and Ranform Forest Algorithm for credit card fraud detection. The relevance of the approaches utilized in the study is that the first way builds a tree against the user's activities, and frauds will be suspected using this tree. In the second way, a user activity-based forest will be built, and an attempt will be made to identify the suspect using this forest. The findings clearly suggest that the standard optional technique achieves reasonable precision levels in detecting credit card fraud situations. A. S. Rathore, A. Kumar, D. Tomar, V. Goyal, K. Sarda and D. Vij [3] has made a study in which they compared the performance Decision Tree, Random Forest, K-nearest neighbors, and Logistic regression on highly imbalanced data. Although this paper provided good results, the problem about imbalanced data should be addressed for more accurate and less biased result.

Although there are fraud detection solutions have been proposed [1], [2] and [3]. The accuracy of classifying the whether the transaction is legitimate or fraud is need to be improve. It is also important to test the data in balanced manner in order to prevent bias. Therefore, this paper aims to develop a model using Stacking Ensemble that will help to analyze the accuracy of the Credit Card Fraud Detection. This study will utilize the use of algorithm such as K-Nearest neighbors and Decision Tree algorithm as a base classifier and a stacked model. It uses Grid Search for Hyperparameter tuning. The proponent hope that by the end of the paper, the reader will have better understanding in analyzing the accuracy of Credit card fraud detection using stacking ensemble..

II. METHODOLOGY

A. Dataset

The dataset used in the study was obtained from Kaggle, an online community platform for data scientists and machine learning enthusiasts with a variety of publicly publish datasets to choose from [6]. Credit Card Fraud Detection dataset was uploaded by Machine Learning Group - ULB. It contains data from transactions made by credit cards in September 2013 by European cardholders. It presents transactions that occurred in two days where a total of 492 frauds was found out of 284,807 transactions.

The dataset contains only numerical input variables that have undergone a PCA transformation. It consists of 31 Features namely Time, Amount, V1, V2, ... V28, and Class. Features V1, V2, ... V28 are the components obtained with PCA. Feature 'Time' holds the seconds elapsed between each transaction and the first transaction in the dataset. The feature 'Amount' is the transaction Amount. Lastly, Feature 'Class' is the response variable and it uses 1 if the case was fraud and 0 otherwise[7]. Figure 1 shows the descriptive statistics that summarize the central tendency, dispersion and shape of a dataset's distribution.

	count	mean	std	min	25%	50%	75%	max
Time	284807.0	9.481386e+04	47488.145955	0.000000	54201.500000	84692.000000	139320.500000	172792.000000
V1	284807.0	3.918649e-15	1.958696	-56.407510	-0.920373	0.018109	1.315642	2.454930
V2	284807.0	5.682686e-16	1.651309	-72.715728	-0.598550	0.065486	0.803724	22.057729
V3	284807.0	-8.761736e-15	1.516255	-48.325589	-0.890365	0.179846	1.027196	9.382558
V4	284807.0	2.811118e-15	1.415869	-5.683171	-0.848640	-0.019847	0.743341	16.875344
V5	284807.0	-1.552103e-15	1.380247	-113.743307	-0.691597	-0.054336	0.611926	34.801666
V6	284807.0	2.040130e-15	1.332271	-26.160506	-0.768296	-0.274187	0.398565	73.301626
V7	284807.0	-1.698953e-15	1.237094	-43.557242	-0.554076	0.040103	0.570436	120.589494
V8	284807.0	-1.893285e-16	1.194353	-73.216718	-0.208630	0.022358	0.327346	20.007208
V9	284807.0	-3.147640e-15	1.098632	-13.434066	-0.643098	-0.051429	0.597139	15.594995
V10	284807.0	1.772925e-15	1.088850	-24.588262	-0.535426	-0.092917	0.453923	23.745136
V11	284807.0	9.289524e-16	1.020713	-4.797473	-0.762494	-0.032757	0.739593	12.018913
V12	284807.0	-1.803266e-15	0.999201	-18.683715	-0.405571	0.140033	0.618238	7.848392
V13	284807.0	1.674888e-15	0.995274	-5.791881	-0.648539	-0.013568	0.662505	7.126883
V14	284807.0	1.475621e-15	0.958596	-19.214325	-0.425574	0.050601	0.493150	10.526766
V15	284807.0	3.501098e-15	0.915316	-4.498945	-0.582884	0.048072	0.648821	8.877742
V16	284807.0	1.392460e-15	0.876253	-14.129855	-0.468037	0.066413	0.523296	17.315112
V17	284807.0	-7.466538e-16	0.849337	-25.162799	-0.483748	-0.065676	0.399675	9.253526
V18	284807.0	4.258754e-16	0.838176	-9.498746	-0.498850	-0.003636	0.500807	5.041069
V19	284807.0	9.019919e-16	0.814041	-7.213527	-0.456299	0.003735	0.458949	5.591971
V20	284807.0	5.126845e-16	0.770925	-54.497720	-0.211721	-0.062481	0.133041	39.420904
V21	284807.0	1.473120e-16	0.734524	-34.830382	-0.228395	-0.029450	0.186377	27.202839
V22	284807.0	8.042109e-16	0.725702	-10.933144	-0.542350	0.006782	0.528554	10.503090
V23	284807.0	5.282512e-16	0.624460	-44.807735	-0.161846	-0.011193	0.147642	22.528412
V24	284807.0	4.456271e-15	0.605647	-2.836627	-0.354586	0.040976	0.439527	4.584549
V25	284807.0	1.426896e-15	0.521278	-10.295397	-0.317145	0.016594	0.350716	7.519589
V26	284807.0	1.701640e-15	0.482227	-2.604551	-0.326984	-0.052139	0.240952	3.517346
V27	284807.0	-3.662252e-16	0.403632	-22.565679	-0.070840	0.001342	0.091045	31.612198
V28	284807.0	-1.217809e-16	0.330083	-15.430084	-0.052960	0.011244	0.078280	33.847808
Amount	284807.0	8.834962e+01	250.120109	0.000000	5.600000	22.000000	77.165000	25691.160000
Class	284807.0	1.727486e-03	0.041527	0.000000	0.000000	0.000000	0.000000	1.000000

Fig. 1. Descriptive Statistics of the Credit Card Fraud Detection Dataset

B. *Exploratory Data Analysis*

Exploratory Data Analysis is the critical process of using summary statistics and graphical representations to do initial investigations on data in order to uncover patterns, spot anomalies, test hypotheses, and verify assumptions [8]

1) Nullity Matrix

A graph that allows us to observed how data is distributed across all columns in the whole dataset. The sparkline in the matrix determines the nullity in rows in the dataset [9]. Results of the nullity matrix as shown in figure 2 shows that the dataset used has no missing value.

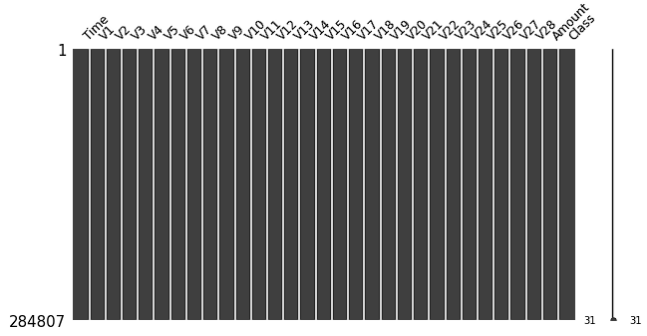

Fig. 2. Nullity Matrix

2) Nullity Correlation Heatmap

Nullity heatmap is useful for discovering data completeness correlations between variable pairs, but it has poor explanatory power for broader associations and offers no specific support for extremely large datasets [9]. Results of the nullity heatmap shown in figure 3 shows that the dataset used has no missing value.

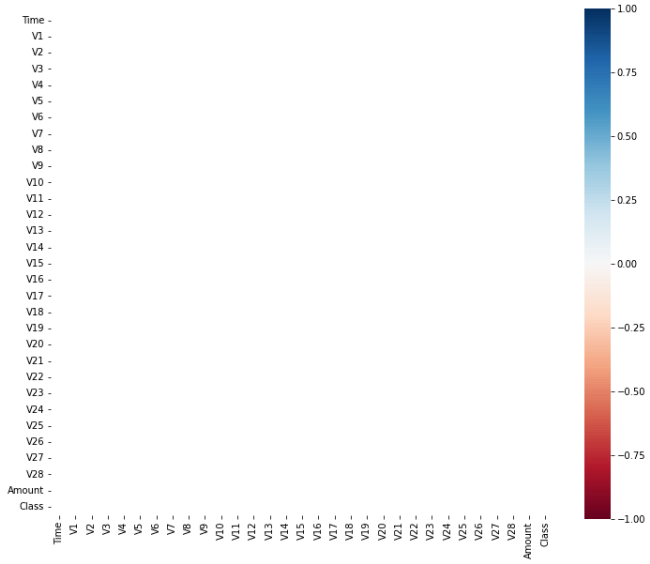

Fig. 3. Nullity Correlation Heatmap

3) Data Pre-processing

As presented in the figure 4, the target feature "Class" displays that there is a large gap in the number of legit and fraud transactions which results to the dataset to be highly imbalance. The proponent uses undersampling to even the number of the normal transactions and Fraudulent Transactions as shown in the code legit_sample = legit.sample (n=492). After building the sample dataset, next is to concatenate the two datasets. Now we have a new dataset which has a shape of 984, 31.

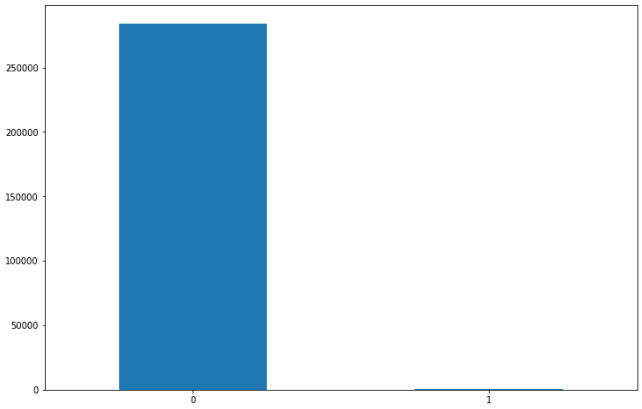

Fig. 4. Feature “Class” Graph

C. *Artificial Intelligence Modeling*

Artificial Intelligence Modeling is the creation, training, and deployment of machine learning algorithms that emulate logical decision-making based on available data [10].

1) Divide the Cleaned Data Set into Training and Validation

The proponent assigned all the independent variable in x while the dependent variable is assigned into y. It is important to split the two sets into their respective x and y to ensure that the model and process is correct. Figure show the code on how to execute the division of the cleaned data set.

Fig. 5. Dividing Data Set Codes

2) *K-Nearest Neighbors Classifier as first base model*

The k-nearest neighbors algorithm, often known as KNN or k-NN, is a non-parametric, supervised learning classifier that makes classifications or predictions about the grouping of individual data points based on closeness. It may be used for both regression and classification issues, however it is most commonly employed as a classification technique, based on the idea that comparable points can be discovered close together [11]. In most detection systems, the K-nearest neighbor technique is employed. It has also been demonstrated that KNN performs exceptionally well in credit card fraud detection systems that employ supervised learning approaches. KNN achieves a high performance rate without relying on previous distribution assumptions [3].

3) *Decision Tree Classifier as second base model*

Decision Tree is a decision-making technique that depicts a relationship between qualities and value. A test on a given characteristic is represented by each node of the tree structure. Each node makes a choice depending on the information gained. The model can be simply comprehended since Decision Tree may directly represent the properties of the data [12][13]. One of the most significant advantages of this algorithm is that it requires the study of all possible decision outcomes, keeps track of each path to a conclusion, and generates a full analysis of the consequences [2].

Use Case: A situation where a user makes transactions. A decision tree is constructed to forecast the possibility of fraud based on the transaction made as showed in Figure 8 [14].

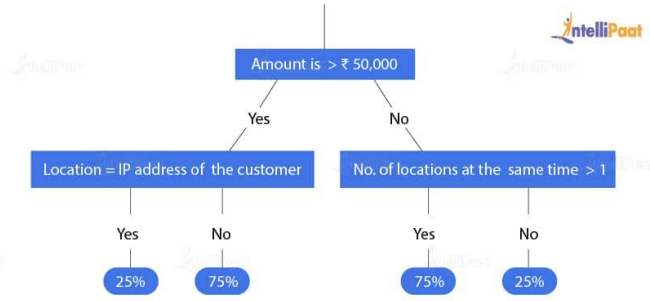

Fig. 6. Decision Tree Use Case

4) *GridSearchCV for Hyperparameter Optimization*

GridSearchCV is a technique for finding the optimal parameter values from a given set of parameters in a grid. It's essentially a cross-validation technique. The model as well as the parameters must be entered. After extracting the best parameter values, predictions are made [15].

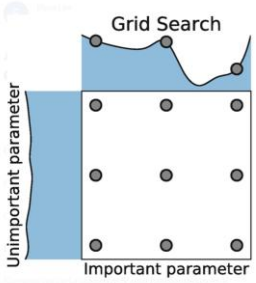

Fig. 7. Grid Search [16]

5) *Logistic Regression as Final Estimator for Stacked Model*

In order to run the Stacked Model developed, the proponent used Logistic Regression as the final estimator. Logistic regression is mostly used for binary classification problems. It predicts a categorical target variable based on a set of independent variables. The outcome will be categorical or discrete. True or false, yes or no, 0 or 1, and so forth. This algorithm calculates the likelihood of an element belonging to a specific class. The probability threshold has been established. When a data point crosses the threshold, it is assigned to the appropriate group. The sigmoid function is used to calculate the likelihood of membership [17].

III. Results and Discussions

A. *Confusion Matrix*

The confusion matrix in this paper generated a true negative that has 359 correctly predicted data points while True Positive has 384 correctly predicted data points. Meanwhile, False Negative is 8 which means that the recall value is high with 98.0% and False Positive is 17 which means that it has a high precision with 95.8%.

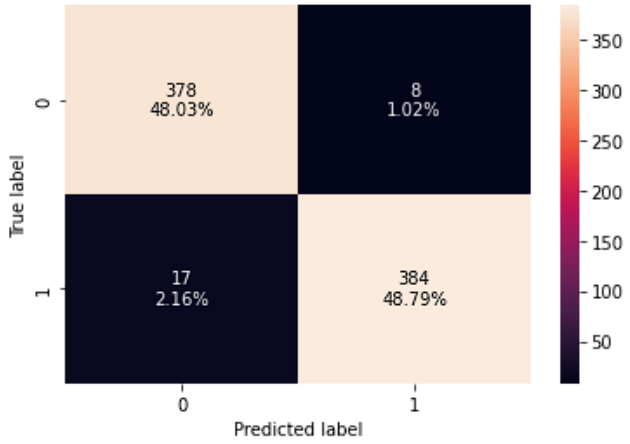

Fig. 8. Confusion Matrix

B. *Curve Plot*

The proponent plots the result using ROC Curve. The Receiver Operator Characteristic (ROC) curve is a binary classification issue evaluation metric. It's a probability curve that displays the TPR against the FPR at different threshold levels, thereby separating the 'signal' from the 'noise' [18].

KNN got an AUC of 0.923, while the DT got an AUC of 0.931 and lastly the Stacked Model got an AUC of 0.935

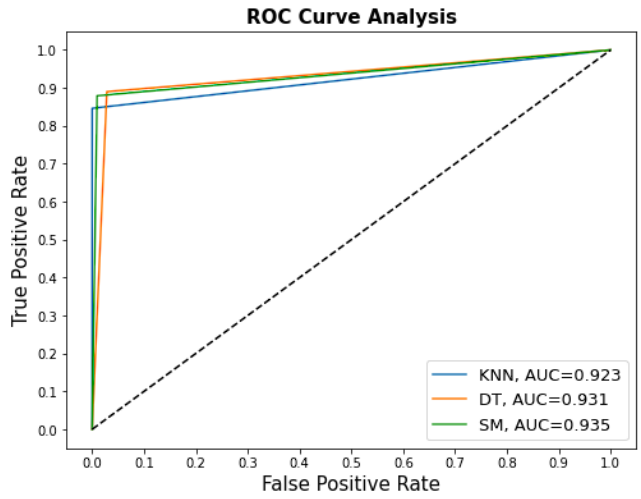

Fig. 9. ROC Curve Analysis

C. *Test Data Results*

Figure 10 shows the test data results of the algorithm in terms of accuracy, recall, precision and F1. K-Nearest Neighbor produced a result of 0.928934, 0.846154, 1.000000, and 0.916667 respectively. Decision Tree produced a result of 0.934010, 0.890110, 0.964286, and 0.925714 respectively. Lastly, the Stacked Model produced a result of 0.939086, 0.879121, 0.987654, 0.930233.

	K-Nearest Neighbor	Decision Tree	Stacked Model
Accuracy	0.928934	0.934010	0.939086
Recall	0.846154	0.890110	0.879121
Precision	1.000000	0.964286	0.987654
F1	0.916667	0.925714	0.930233

Fig. 10. Test Data Result Table

Figure 11 displays the test data results in bar graph. Blue bar refers to the result of K-Nearest Neighbor while orange bar refers to Decision Tree and lastly the green bar refers to the result of Stacked Model.

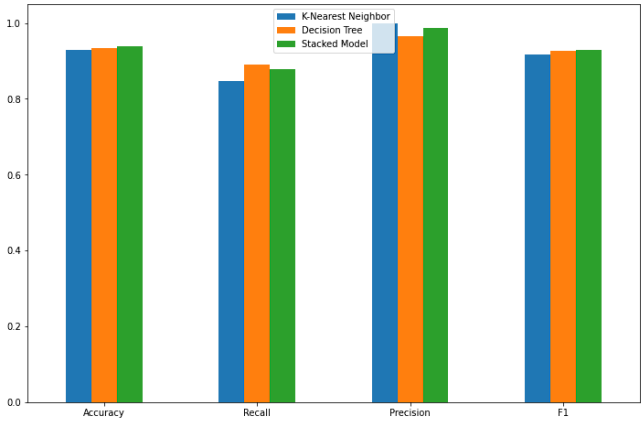

Fig. 11. Test Data Result Bar Graph

IV. Conclusions

Credit Card Fraud Detection Accuracy analysis using Stacking ensemble is presented in this paper. K-Nearest neighbors and Decision tree are used as base classifier for the model while Logical Regression is used as the final estimator for the Stacking Model. The proponent also used GridSearch CV to find the optimal parameters of the model. To avoid the problems caused by an unbalanced dataset and to smoothen the process of analysis, the proponents have treated the data to under sampling in order to have a balanced the dataset for the training and testing. The produced results showed that the Stacking Ensemble Model has 93.9% of Accuracy, 87.9% of Recall, 98.8% of Precision, and 93.0% of F1 which is higher than a single classifier. Result shows that the model a achieved a decent accuracy on the dataset.

V. Recommendations

For the future works, the proponents suggests to optimized the accuracy produced by the proposed model. Even if the Stacking Ensemble model has attained an accuracy of 93.9%, there is a chance that in future work, this accuracy can go higher. The future researchers can also try other balancing techniques for the dataset that may change the accuracy of the said model. They can also try implementing RandomSearchCV for the Hyperparameter Optimization. Lastly, it is best to try and optimize different other algorithms and classification models to compare it with one another.

Acknowledgement

The completion of this paper will not be possible without the presence and assistance of the people throughout the writing. Likewise, the author would like to express his sincerest gratitude to Elective 2, 2nd Semester, class of 2022 and most especially to Engr. Helcy Alon, professor, for her support, guidance and knowledge that helps the proponent to finish this paper.

[1] S. K. Saddam Hussain, E. Sai Charan Reddy, K. Gangadhar Akshay, and T. Akanksha, "Fraud Detection in Credit Card Transactions Using SVM and Random Forest Algorithms," *Proc. 5th Int. Conf. I-SMAC (IoT Soc. Mobile, Anal. Cloud), I-SMAC 2021*, pp. 1013–1017, 2021, doi: 10.1109/I-SMAC52330.2021.9640631.

[2] M. R. Dileep, A. V. Navaneeth, and M. Abhishek, "A novel approach for credit card fraud detection using decision tree and random forest algorithms," *Proc. 3rd Int. Conf. Intell. Commun. Technol. Virtual Mob. Networks, ICICV 2021*, no. Icicv, pp. 1025–1028, 2021, doi: 10.1109/ICICV50876.2021.9388431.

[3] N. Malini and M. Pushpa, "Analysis on credit card fraud identification techniques based on KNN and outlier detection," *Proc. 3rd IEEE Int. Conf. Adv. Electr. Electron. Information, Commun. Bio-Informatics, AEEICB 2017*, pp. 255–258, 2017, doi: 10.1109/AEEICB.2017.7972424.

[4] C. Wang, Y. Wang, Z. Ye, L. Yan, W. Cai, and S. Pan, "Credit card fraud detection based on whale algorithm optimized BP neural network," *13th Int. Conf. Comput. Sci. Educ. ICCSE 2018*, no. Iccse, pp. 614–617, 2018, doi: 10.1109/ICCSE.2018.8468855.

[5] "Credit Card Fraud | Wex | US Law | LII / Legal Information Institute." https://www.law.cornell.edu/wex/credit_card_fraud (accessed Jun. 06, 2022).

[6] "What is Kaggle? | DataCamp." https://www.datacamp.com/blog/what-is-kaggle (accessed Jun. 06, 2022).

[7] "Credit Card Fraud Detection | Kaggle." https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud (accessed Jun. 06, 2022).

[8] "What is Exploratory Data Analysis? | by Prasad Patil | Towards Data Science." https://towardsdatascience.com/exploratory-data-analysis-8fc1cb20fd15 (accessed Jun. 06, 2022).

[9] "Easy Way of Finding and Visualizing Missing Data in Python | by Mala Deep | DataDrivenInvestor." https://medium.datadriveninvestor.com/easy-way-of-finding-and-visualizing-missing-data-in-python-bf5e3f622dc5 (accessed Jun. 06, 2022).

[10] "What Is AI Modeling - Intel." https://www.intel.com/content/www/us/en/analytics/data-modeling.html (accessed Jun. 06, 2022).

[11] "What is the k-nearest neighbors algorithm? | IBM." https://www.ibm.com/ph-en/topics/knn (accessed Jun. 07, 2022).

[12] Y. Sahin, S. Bulkan, and E. Duman, "A cost-sensitive decision tree approach for fraud detection," *Expert Syst. Appl.*, vol. 40, no. 15, pp. 5916–5923, 2013, doi: 10.1016/j.eswa.2013.05.021.

[13] T. Wang and Y. Zhao, "Credit Card Fraud Detection using Logistic Regression," pp. 301–305, 2022, doi: 10.1109/bdicn55575.2022.00064.

[14] "Fraud Detection Algorithms | Fraud Detection using Machine Learning." https://intellipaat.com/blog/fraud-detection-machine-learning-algorithms/ (accessed Jun. 07, 2022).

[15] "DaskGridSearchCV - A competitor for GridSearchCV - GeeksforGeeks." https://www.geeksforgeeks.org/daskgridsearchcv-a-competitor-for-gridsearchcv/ (accessed Jun. 07, 2022).

[16] "An Intro to Hyper-parameter Optimization using Grid Search and Random Search | by Elyse Lee | Medium." https://medium.com/@cjl2fv/an-intro-to-hyper-parameter-optimization-using-grid-search-and-random-search-d73b9834ca0a (accessed Jun. 07, 2022).

[17] G. Kumar, S. Kumar, and A. A. Prakash, "Credit Card Fraud Detection using Machine Learning," *Int. J. Eng. Adv. Technol.*, vol. 10, no. 4, pp. 124–126, 2021, doi: 10.35940/ijeat.d2344.0410421.

[18] "AUC-ROC Curve in Machine Learning Clearly Explained - Analytics Vidhya." https://www.analyticsvidhya.com/blog/2020/06/auc-roc-curve-machine-learning/ (accessed Jun. 07, 2022).