Comparison of FLAML Machine Learning Algorithms for Fraud Detection

Kashish Gujral, Ben Jochem, Jie Zhu
{kmg6272, bmj5307, jzz5460}@psu.edu

I. Abstract

THIS project investigates the efficacy of various machine learning algorithms at detecting fraud in three different areas of financial activity: Credit Card Transactions, Ethereum Network Transactions, and Insurance Claims. We use the FLAML (A Fast Library for Automated Machine Learning Tuning) python package for training, tuning, and evaluating the performance of various different classes of machine learning models. We find that the XGBoost and LGBM FLAML techniques consistently outperform other methods across all three areas of financial activity. Our results offer direction for future, more rigorous investigations into machine learning techniques in different transactional contexts and benefit financial firms, individuals, and researchers that have an interest in machine learning as it applies to transactional data.

II. Introduction

The goals of the project can be broken down into two categories: (1) Team Goals and (2) Individual Domain Specific Goals. Our primary Team Goal is to determine whether there is a machine learning algorithm that performs best in all three areas or whether algorithm performance is dependent on the area of financial activity. Individual Domain Specific Goals vary between the three team members. Each focuses on one of the three areas of financial activity and conducts an independent analysis on their own datasets. Figure 1 below is a diagram that gives an overview of our project framework. In what follows we briefly describe each team member's area of financial activity and Domain Specific Goals, summarize previous works related to our goals, describe our methodology, detail our experiments, and discuss our results and conclusions.

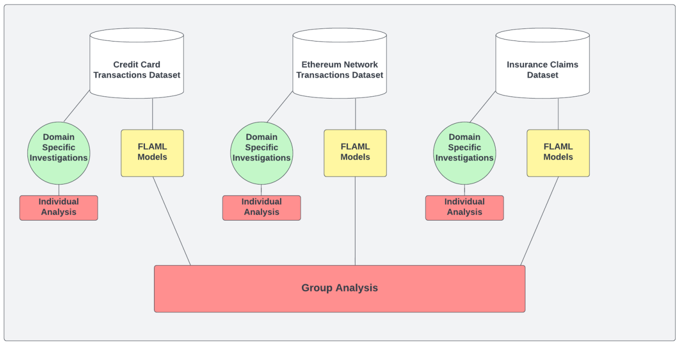

Credit Card Transactions Dataset
Ethereum Network Transactions Dataset
Insurance Claims Dataset
Domain Specific Investigations
FLAML Models
Domain Specific Investigations
FLAML Models
Domain Specific Investigations
FLAML Models
Individual Analysis
Individual Analysis
Individual Analysis
Group Analysis

Fig. 1. Project Framework

A. Areas of Financial Activity

*1) **Credit Card Transactions**:* Due to the importance and scale of the credit card transactions in real life, Jie decides to do the Fraud detection about Credit Cards. The common fraud of credit cards is misusing others credit cards to pay intentionally or occasionally. Jie finds that it is easy to find an effective database about this topic. Jie studied some algorithms during the previous semester (e.g. random forest, knn, decision tree). Therefore, she has an idea to implement these methods in this project. She searched dataset on Kaggle and found that there is a dataset about european credit cards accounts and can train the existing data to build a model and do testing data to predict the future transaction. Based on the common knowledge, Jie understands that the normal transaction usually happens during the day time, so it is worthwhile to notice that if there are some transactions during the night. Following this logic, we are able to detect fraud in different ways. According to the professor's suggestion, Jie would like to take advantage of FLAML to work, figure out the most effective algorithm and share the progress and result with the team as a whole.

Jie uses a large amount of data to check whether the logistic regression is effective to detect credit card fraud in advance. The following goals will be progressed in the future work:

1) Classify credit card fraud by logistic regression so that we can get the training and testing data.
2) According to the fraud data unevenly distributed, it is important to figure out which metric is more suitable for this dataset (e.g. accuracy, precision, recall, and f1).
3) Try to visualize the result to show the difference between normal transaction and fraudulent transaction.

*2) **Insurance Claims**:* Insurance claim fraud is the deliberate deception against or by the company or agent in order to gain financially. These can be taken place in any point in the process of a transaction These frauds include: (1) criminals who steal large amounts of money from businesses, (2) people who change service costs, or (3) normal people that cover their deductibles and file claims to make some extra money. Kashish has prior knowledge about Insurance Fraud as she worked on a similar dataset for a previous class. She understands the parameters and factors that play a role in this situation. She will continue to monitor any changes in the claims and work toward the team goal as a whole. The basic idea behind this category will be to work on the dataset and see what factor in insurance acts the most and have the biggest impacts when it comes to claims and frauds. She also wants to make a prediction model that can, for future purposes, predict if the transaction or claims filed might be a fraud. Furthermore, she will use data visualization and prepare charts/graphs for better visual analysis.

3) ***Ethereum Network Transactions:*** Given his interest in blockchains and cryptocurrencies, Ben investigates transactions on the Ethereum blockchain as his area of financial activity. These transactions are structurally similar to transactions using traditional payment systems such as credit cards but there are also some key distinctions. Ben explores whether these key distinctions impact the effectiveness of machine learning algorithms at detecting fraudulent transactions. Formally, he investigates the following Domain Specific questions:

1) Are the machine learning algorithms as a whole more or less effective at detecting fraudulent transactions on Ethereum as compared to on traditional payment systems/?
2) How do specific market conditions or types of transactions (i.e. large values, new users, etc.) impact the effectiveness of the algorithms on Ethereum and in traditional payment systems?

B. Related Works

There has been much prior work exploring the performance of different machine learning algorithms for detecting financial fraud. Most previous research focuses on only one dataset and compares the performance of a few common algorithms using a few common metrics. This research is useful but it leaves a lot of questions on the table regarding why an algorithm may or may not work in a specific context. Our project seeks to answer these questions and is novel in that it compares multiple different machine learning algorithms across different datasets in different areas of financial activity. We hope that this will lead to a better understanding of algorithm performance in varying contexts and set the stage for more in-depth research in this area. What follows is a brief description of the prior work completed on the topic.

All previous work relevant to our problem investigates the performance of many machine learning algorithms on only one dataset (usually credit card transactions). Thus, our problem is unique and there is no prior work that can be used for direct comparison. We can, however, use the results from these prior works to compare the results from each of our areas of financial activities and make interesting comparisons. The common result amongst all of these prior works was that the KNN, Logistic, and DNN models are the most effective at predicting fraud. It should be noted that these results can likely only be extended to Jie's work as she is also looking at credit card transactions. However, we will use this knowledge to inform our model building process such that we can efficiently find the best FLAML models for detecting financial fraud. The main drawback of these related state-of-the-art studies related to our project is that they offer few conclusions that are practically useful. Knowing which kind of machine learning model performs best on a given dataset after rigorous training, tuning, and testing is an interesting research problem but does little to help practitioners in real-world contexts. Our solution is much more helpful in this sense. By investigating which FLAML machine learning models perform best for fraud detection across many different datasets we can provide useful rules of thumb such that FLAML users can efficiently deploy effective fraud detection machine learning models with a high degree of confidence. Overall, we see our project as building off of previous works in a way that makes the results more practically applicable and useful.

III. Methodology

Our Team Goal is to determine whether there is an algorithm that performs best in all three areas or whether algorithm performance is dependent on the area of financial activity so all team members must be consistent in their choices of algorithms and metrics for valid comparisons. We have chosen to use the FLAML python package to facilitate this consistency. The FLAML (A Fast Library for Automated Machine Learning Tuning) package offers our team a standardized way to build and tune our models such that we can be sure that each of us followed the same process and that the comparison of our results is justified. Furthermore, the speed and efficiency of FLAML will allow us to streamline the model building process and focus on our primary question of interest which is a comparison of our results. The primary challenge in the early stages of our project was ensuring that each team member processed the data in the same way before modeling. This was achieved using principal component analysis. In the subsections that follow we describe the algorithms and metrics that we use in our project. Figure 2 shows the methodology framework for achieving our Team goal.

A. Team Methodology

1) Team Metrics: Our metrics are chosen based on simplicity and efficacy. We ensure that all team members have comparable results by simply using the metrics that are built into FLAML. A variety of metrics is used such that we are able to verify the efficacy of our models in different ways. It could be misleading if we only chose to use only a single metric such as accuracy because most observations in fraud detection datasets are negative (imbalanced) and by using accuracy we could get a high accuracy whilst not detecting any of the fraudulent transactions! The following four metrics provide us with a holistic view of the performance of our models: Precision, Recall, F1, and AUC. The combination of recall and precision is our main way by which we evaluate model performance

2) Team Models: Like the metrics, our models were also chosen based on simplicity and efficacy. Based on our research of prior works and preliminary modeling tests, we primarily focus the following models: Xgboost, LGBM, Random Forest, Support Vector Machines, and Ensemble methods.

IV. Experiments and Evaluation

As previously described, we have one dataset for each of the three areas of financial activity. The areas of financial activity and the corresponding datasets were chosen due to both their similarities and differences. They were all similar

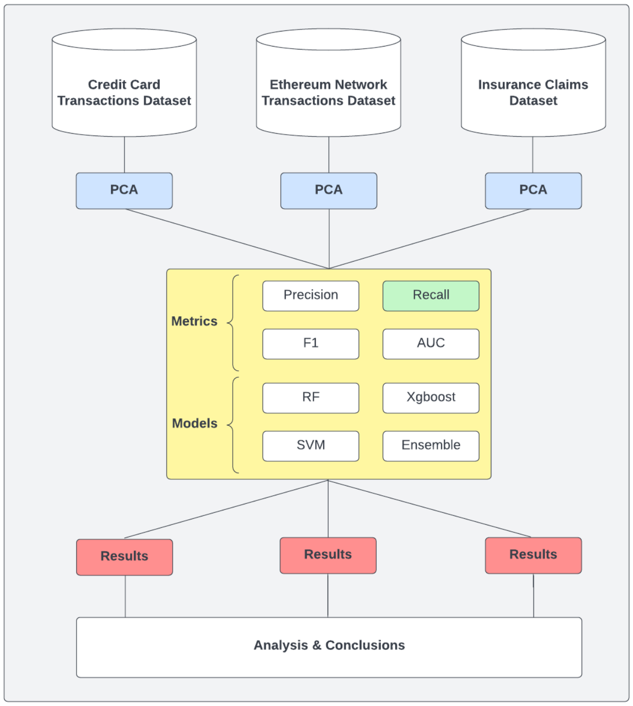
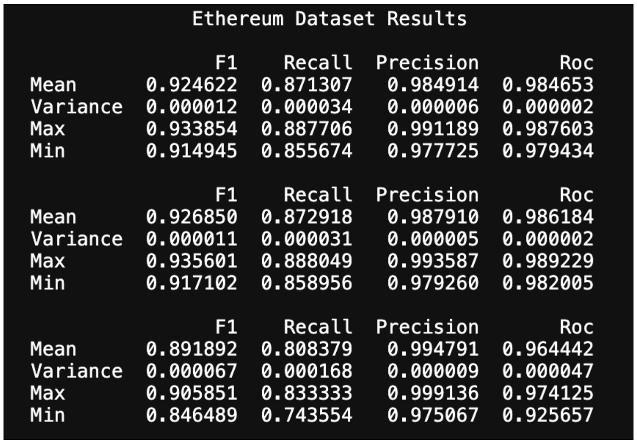


Fig. 2. Team Methodology

to each other in the sense that they are transactional datasets related to finance. If we had chosen to evaluate FLAML machine learning algorithms on datasets containing vastly different kinds of data then the results would not have had any coherent interpretation. In this sense, the similarity of the three datasets allows us to formulate a valid research study. The datsets were different from each other in the sense that the three areas of financial activity have different features and emerge from completely different sources. These differences allowed us to investigate financial fraud under a more diverse set of cirumstances thereby allowing the results our study to potentially apply to all financial fraud and not just one niche situation. In terms of the actual datasets chosen for each area of financial activity, we optimized for simplicity and ease of retrieval. Given that our primary team objective was to investigate the FLAML machine learning algorithms in general and not their use for specific datasets, the specific characteristics of each dataset were not that important. As a result, we obtained the credit card and insurance datasets directly from the top results on kaggle. Similarly, the Ethereum dataset was easily found and obtained on google scholar.

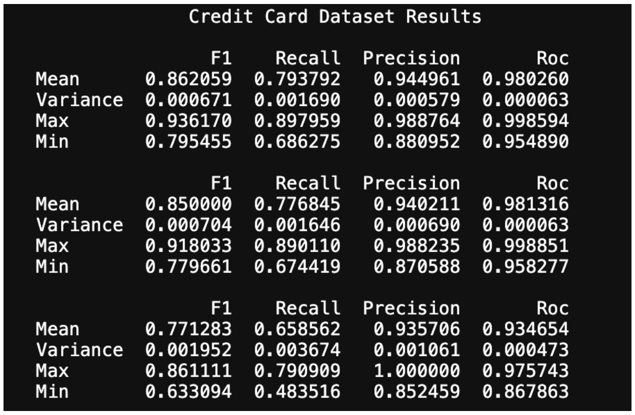
The implementation of the experiments on the three datasets for our team goal can be found on our github in the Team-Goal folder. In accordance with our methodology, we implement Random Forest, Xgboost, LGBM, Ensemble models in conjunction with all other binary classification models offered by FLAML. The best model types found by FLAML for the Ethereum dataset are Xgboost, LGBM, and Catboost models. The best model types found by FLAML for the Credit Card dataset are Xgboost, LGBM, and Extra Tree. The best model types found by FLAML for the insurance dataset are Xgboost, LGBM, and Extra Tree. The implementation, evaluation, and results for each team member's domain specific goals have been omitted in this paper such that the focus remains on the primary question of interest which is the team goal. The code and results corresponding to the domain specific goals can also be found on our github in the corresponding domain specific folders.

After using FLAML to find the best three models and corresponding configurations for each dataset for our team goal, we train and test each using 5-fold cross validation repeated 25 times to ensure that we smooth any noise in our results and achieve an accurate estimation of true model performance. We record the mean, variance, minimum, and maximum values for the F1, Recall, Precision, and Roc-Auc metrics in order to evaluate model performance. Repeating cross validation multiple times reduces the error in our estimate of mean model performance and allows us to record the variance of the evaluation metrics across repeats. This enables us to investigate model performance more comprehensively. A summary of our results across all 25 repeats of 5-fold cross validation for each area of financial activity is shown below:



```
                 Ethereum Dataset Results

                F1    Recall  Precision       Roc
Mean      0.924622  0.871307   0.984914  0.984653
Variance  0.000012  0.000034   0.000006  0.000002
Max       0.933854  0.887706   0.991189  0.987603
Min       0.914945  0.855674   0.977725  0.979434

                F1    Recall  Precision       Roc
Mean      0.926850  0.872918   0.987910  0.986184
Variance  0.000011  0.000031   0.000005  0.000002
Max       0.935601  0.888049   0.993587  0.989229
Min       0.917102  0.858956   0.979260  0.982005

                F1    Recall  Precision       Roc
Mean      0.891892  0.808379   0.994791  0.964442
Variance  0.000067  0.000168   0.000009  0.000047
Max       0.905851  0.833333   0.999136  0.974125
Min       0.846489  0.743554   0.975067  0.925657
```

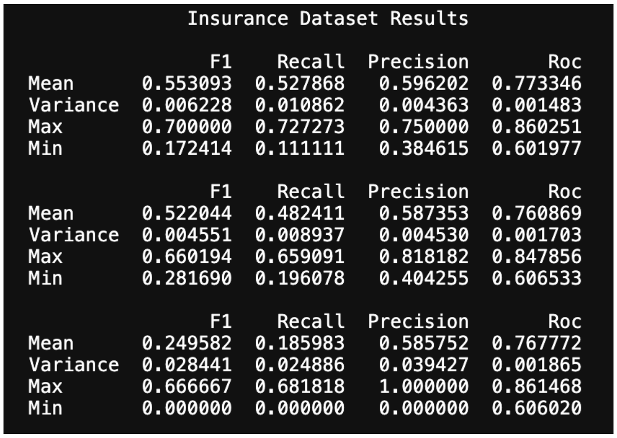
Fig. 3. Ethereum Results



```
                Credit Card Dataset Results

                F1    Recall  Precision       Roc
Mean      0.862059  0.793792   0.944961  0.980260
Variance  0.000671  0.001690   0.000579  0.000063
Max       0.936170  0.897959   0.988764  0.998594
Min       0.795455  0.686275   0.880952  0.954890

                F1    Recall  Precision       Roc
Mean      0.850000  0.776845   0.940211  0.981316
Variance  0.000704  0.001646   0.000690  0.000063
Max       0.918033  0.890110   0.988235  0.998851
Min       0.779661  0.674419   0.870588  0.958277

                F1    Recall  Precision       Roc
Mean      0.771283  0.658562   0.935706  0.934654
Variance  0.001952  0.003674   0.001061  0.000473
Max       0.861111  0.790909   1.000000  0.975743
Min       0.633094  0.483516   0.852459  0.867863
```

Fig. 4. Credit Card Results

As discussed in the methods section of the paper, we primarily use recall as our evaluation metric given that we are attempting to detect fraud. Recall is calculated as the number



```
               Insurance Dataset Results

                F1    Recall  Precision       Roc
Mean      0.553093  0.527868   0.596202  0.773346
Variance  0.006228  0.010862   0.004363  0.001483
Max       0.700000  0.727273   0.750000  0.860251
Min       0.172414  0.111111   0.384615  0.601977

                F1    Recall  Precision       Roc
Mean      0.522044  0.482411   0.587353  0.760869
Variance  0.004551  0.008937   0.004530  0.001703
Max       0.660194  0.659091   0.818182  0.847856
Min       0.281690  0.196078   0.404255  0.606533

                F1    Recall  Precision       Roc
Mean      0.249582  0.185983   0.585752  0.767772
Variance  0.028441  0.024886   0.039427  0.001865
Max       0.666667  0.681818   1.000000  0.861468
Min       0.000000  0.000000   0.000000  0.606020
```

Fig. 5. Insurance Claims Results

of truly fraudulent transactions detected by the model divided by the total number of truly fraudulent transactions in the test set. Thus, a high recall means that the model is detecting a large number of truly fraudulent transactions. A low recall means that the model is failing to detect a large number of truly fraudulent transactions. Failing to detect truly fraudulent transactions is more costly than unnecessarily flagging non-fraudulent transactions which is why we place a heavy emphasis on recall. Note, however, that a perfect recall score of 1 could be achieved by simply flagging every transactions as fraud. Obviously this would not be a useful model. Thus, we use precision in conjunction with recall to measure the extent to which the models are flagging transactions that are actually fraudulent versus just flagging a large amount of transactions as fraudulent. Precision is calculated as the number of truly fraudulent transactions detected by the model divided by the total number of fraudulent flags made regardless of if they are truly fraudulent or not. Thus, a high precision means that most of the transactions that are flagged as fraudulent are actually fraudulent. A low precision means that many of the transactions that are flagged as fraudulent are not actually fraudulent. Thus, if a model flagged every transaction as fraudulent in order to achieve a recall of 1 it would have very low precision because many of the transactions flagged are not actually fraudulent. With these metrics in mind we can effectively investigate the performance of the models for each dataset.

For the Ethereum dataset, the top 3 models found in validation and used for testing were the Xgboost, LGBM, and Catboost models. The results as shown in figure 3 show the mean, variance, minimum values, and maximum values for each of these three models across all cross validation splits and repeats. From top to bottom the results shown correspond to Xgboost, LGBM, and then Catboost. As can be seen, when evaluated on the test set the Xgboost and LGBM models performed comparably and both outperformed the Catboost model significantly. Recall scores for the two top models were around 0.87 while precision scores were significantly higher around 0.98.

For the Credit Card dataset, the top 3 models found in validation and used for testing were the Xgboost, LGBM, and Extra Tree models. The results as shown in figure 4 show the mean, variance, minimum values, and maximum values for each of these three models across all cross validation splits and repeats. From top to bottom the results shown correspond to Xgboost, LGBM, and then Extra Tree. As can be seen, when evaluated on the test set the Xgboost and LGBM models performed comparably and both outperformed the Extra Tree model significantly. Recall scores for the two top models were around 0.78 while precision scores were significantly higher around 0.94. These average scores were surprisingly lower than those for the Ethereum dataset despite there being significantly more data. This indicates that it may actually be easier to detect fraudulent Ethereum transactions than it is to detect fraudulent credit card transactions. Perhaps the more important observation, however, is that the Xgboost and LGBM models again performed best.

For the Insurance dataset, the top 3 models found in validation and used for testing were the Xgboost, LGBM, and Extra Tree models. The results as shown in figure 4 show the mean, variance, minimum values, and maximum values for each of these three models across all cross validation splits and repeats. From top to bottom the results shown correspond to Xgboost, LGBM, and then Extra Tree. As can be seen, when evaluated on the test set the Xgboost and LGBM models performed comparably and both outperformed the Extra Tree model significantly. Recall scores for the two top models were around 0.5 while precision scores were significantly higher around 0.6. These average scores were significantly lower than the other two datasets which is likely due to the drastically smaller size of the dataset. Here again the Xgboost and LGBM models notably performed best.

The results from our tests show that the LGBM and Xgboost FLAML machine learning models perform best in all three areas of financial activity. This provides significant evidence that these FLAML algorithms are the best for fraud detection tasks and that algorithm performance is not dependent on the area of financial activity. The consistency of results across all three areas of financial activity give us high confidence in this conclusion. This has significant implications for potential FLAML users who want to implement fraud detection models. Our study strongly suggests that they should use the LGBM and Xgboost models to achieve an effective model. This being said, the precision metric was consistently greater than the recall metric on the testing set across all three areas of financial activity. As previously discussed, this is not ideal for fraud detection and should be addressed by users who want to implement a FLAML fraud detection model. Luckily, there is an obvious solution. Since our study was focused on which FLAML models perform best at fraud detection and was not necessarily trying to achieve the best possible metrics we did not do any feature engineering on the data besides principal component analysis. By doing more comprehensive feature engineering, users should be able to drastically improve recall scores. This is a potential area of future research. To summarize the findings and implications of our results, we have significant evidence that the Xgboost and LGBM FLAML models perform fraud detection most effectively in all areas of financial activity. Furthermore, the combination of a sufficently large dataset, feature engineering, and a FLAML Xgboost or LGBM model will likely lead to a performant fraud detection model in many contexts.

V. Discussion

In terms of lessons learned from this project, it is clear that the implementation of a project plan often changes throughout the life of the project. We changed courses many times in terms of how to implement each type of model for each dataset and how to process each dataset. Furthermore, we learned that it is very useful to have a well-defined project roadmap and pipeline diagrams from the beginning. Once we had these in place the pace of our project increased tremendously. Furthermore, having two different sets of goals (Team and Individual) was a challenge. It was dificult to stay on track with the team goal while having each team member work on their own dataset independently. We would have changed the structure of our project to be completely team-oriented if we were to do it again. It terms of the outcomes of the project, we are quite happy with our results. We believe that we have effectively extended the results of previous research studies such that they are more practically applicable and helpful for anyone interested in implementing FLAML fraud detection models. Future work building of our project should primarily focus on feature engineering and determine if there are common features across all areas of financial activity that are useful for fraud detection. A study of this nature paired with the results of this project would make the rapid deployment of FLAML fraud detection models very simple for practitioners.

VI. Conclusion

The primary conclusion from our project is that the Xgboost and LGBM FLAML algorithms perform best in all three areas of financial activity and that algorithm performance is largely not dependent on the area of financial activity. Conclusions for each domain specific goal for each of the three datasets can be found of our github. We intend for these results to be useful for anyone interested in implementing FLAML fraud detection algorithms and hope to see our research used to help deploy effective models.

VII. Contribution

Each team member was responsible for the preprocessing and preparation of their own dataset for both their individual goals and the team goal. Ben primarily implemented the algorithms and analyzed the results. Each of the three team members contributed to the project paper. Ben and Jie primarily created the final presentation.

VIII. Appendix and References

A. Github

https://github.com/benjoch/DS440-Project

B. References

D. Varmedja, M. Karanovic, S. Sladojevic, M. Arsenovic and A. Anderla, ”Credit Card Fraud Detection - Machine Learning methods,” 2019 18th International Symposium INFOTEH-JAHORINA (INFOTEH), 2019, pp. 1-5, doi: 10.1109/INFOTEH.2019.8717766.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, Édouard Duchesnay; 12(85):28252830, 2011.

Itoo, F., Meenakshi Singh, S. “Comparison and analysis of logistic regression, Naïve Bayes and KNN machine learning algorithms for credit card fraud detection,” Int. j. inf. tecnol. 13, 1503–1511 (2021). doi: 10.1007/s41870-020-00430-y

J. O. Awoyemi, A. O. Adetunmbi and S. A. Oluwadare, ”Credit card fraud detection using machine learning techniques: A comparative analysis,” 2017 International Conference on Computing Networking and Informatics (ICCNI), 2017, pp. 1-9, doi: 10.1109/ICCNI.2017.8123782.

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Rafal Jozefowicz, Yangqing Jia, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Mike Schuster, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.