**CSCI 5923 01 Capstone in Interprofessional Informatics**

**Analysis of Supervised Machine Learning Algorithms for Credit Card Fraud Detection**

**Maereg Dubale**

**Texas Woman's University**

**Denton, Texas**

**Author Note**

Maereg Dubale, Texas Woman's University, Mathematics & Computer Science

Submitted 2021

**ABSTRACT**

In recent years, credit card fraud has continued to grow, making fraud detection systems increasingly necessary for banks and financial institutions to minimize losses. Machine Learning plays a vital role in data-driven fraud detection systems. Despite the significant research done on fraud detection, no set of standardized machine learning methods or solutions has been developed. This study aimed to review the most popular supervised machine learning algorithms and their effectiveness in identifying fraudulent and non-fraudulent transactions. In this analysis, various scholarly articles covering credit card fraud are studied and discussed. Also, an experiment was conducted to evaluate the performance of six supervised learning algorithms including, decision tree, random forest, XGBoost, kernelized support vector machine, logistic regression, and neural networks on credit card transaction data. For evaluating the models' predictive ability, confusion matrix, accuracy, and F1-score were used. Based on the experiment results, random forest proved to be the most efficient algorithm for detecting fraudulent transactions with a reasonable computation time.

***Keywords****: Machine Learning, Supervised Machine Learning, Credit Card Fraud, Fraud Detection, Decision Tree, Random Forest, XG-Boost, Kernelized Support Vector Machine, Logistic Regression, Neural Networks*

**TABLE OF CONTENTS**

1. Introduction
   1. Background
   2. Supervised Machine Learning
      1. Supervised Learning Algorithms
2. Literature Review
3. Methodology
4. Experiment on Dataset
   1. Data Description
   2. Data Preprocessing and Exploratory Data Analysis
   3. Model Evaluation Metrics
   4. Experimental Setup and Model Performance Evaluation
      1. Experimental Setup
      2. Model Performance Result
5. Summary and Discussion
6. Conclusion and Recommendation
7. References
8. **INTRODUCTION**

**1.1 Background**

Credit cards are prevalent in modern-day society, and credit card fraud cases have continued to grow in recent years. In 2018 alone, $24.26 billion was lost due to payment card fraud worldwide. With 38.6% of reported card fraud losses in 2018, the United States is the most credit fraud-prone country [1]. Credit card fraud is a type of identity theft that occurs when someone uses another person's credit card without their consent to obtain money, goods, or services. Frauds associated with credit cards typically include application fraud, counterfeiting, identity theft, phishing, and skimming [2]. Credit card fraud has a significant impact on the global economy, affecting consumers, merchants, and card issuers alike.

The increase in credit card fraud has made fraud detection systems vital for banks and financial institutions in order to minimize their losses. Frauds have to be explored and identified as quickly as possible in order to stop fraudulent activities [2]. To detect and prevent fraud, various fraud prevention and detection mechanisms are employed. The actions taken against fraud can be divided into two: fraud prevention, which aims to block fraudulent transactions at the source, and fraud detection, which identifies successful fraudulent transactions after they have occurred [3]. As a result, fraud detection has become a vital activity to reduce the impact of a fraudulent transaction on service delivery, costs, and a company's reputation [4].

Machine learning plays a vital role in data-driven fraud detection systems. Financial institutions can automatically analyze customer behavior patterns for signs of abnormal activity, enabling them to catch fraudulent activity in real time. There are many machine learning techniques that can be used to overcome fraudulent activities, which can be categorized into four basic approaches: [supervised](https://searchenterpriseai.techtarget.com/definition/supervised-learning) learning, [unsupervised](https://whatis.techtarget.com/definition/unsupervised-learning) learning, semi-supervised learning, and reinforcement learning. Several algorithm types are included in supervised machine learning; however, this paper focuses on algorithms such as decision tree (DT), random forest (RF), XGBoost, kernelized support vector machine (SVM), logistic regression (LR), and neural networks (NN). In general, this paper aims to review the most popular supervised machine learning algorithms and their effectiveness in identifying fraudulent and non-fraudulent transactions, identify limitations and indicate new research directions in the application of supervised learning in credit card fraud detection systems.

**1.2 Supervised Machine Learning**

Machine learning can be broadly classified into two categories: supervised machine learning and unsupervised machine learning. According to [5], unsupervised machine learning subsumes all kinds of machine learning where there is no known output data is given to the algorithm. In unsupervised learning, the learning algorithm is just shown the input data and asked to extract knowledge from this data. Supervised learning algorithms use a set of examples from previous records that are labeled to make predictions about the future [6]. According to AkinsolaJ. et al. (2017), the process of applying supervised machine learning to a real-world problem is described in Figure 1.1.



*Figure 1.1: The processes of supervised machine learning*

* + 1. **Supervised Learning Algorithms**

Based on [5],[8],[9],[10],[6],[11],[12],[7], the characteristics of the six supervised learning algorithms used in this paper are summarized below in Table 1.1.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Algorithm** | **How it works** | **Tasks used for** | **Strength** | **Weakness** |
| Decision Tree | * Learn a hierarchy of if/else questions, leading to a decision | * Classification and regression | * Very fast * Visually interpretable * Self-explanatory and easy to follow * Don't need scaling of data * High predictive performance * Flexible in handling a variety of input data | * Tendency to overfit and provide poor generalization performance |
| Random Forest | * Build numerous trees in bootstrapped samples and generate an aggregate tree by averaging across trees (reducing overfit) | * Classification and regression | * Nearly always perform better than a single decision tree * Often work well without heavy tuning of the parameters * Very robust and powerful * Don't need scaling of data | * Not suitable for very high-dimensional, sparse data * Are slower to train and predict than linear models * Hard to interpret * Time-consuming on a large dataset |
| Logistic Regression | * Input values (x) are combined linearly using weights or coefficient values (Beta) to predict an output value (y). The output value being modeled is a binary value (0 or 1) rather than numeric. | * Classification | * Suitable for huge datasets * Good for high dimensional data * Work well with sparse data * Very fast to train and predict * Ease of implementation | * Inability to solve a non-linear problem as its decision surface is linear * Prone to overfitting, * It will not work out well unless all independent variables are identified. |
| SVM (Support Vector Machine) | * Using non-linear mapping, it transforms the training data into a higher dimension. Within the new dimension, it finds the optimal separating hyperplane by using support vectors and margins (defined by the support vectors). | * Classification and regression | * Potent for medium-sized datasets of features with similar meaning * Allow for complex decision boundaries * Work well on low-dimensional and high-dimensional data | * Require scaling of data * Sensitive to parameters * Understanding the final SVM model is difficult |
| XG-Boost | * Built based on entropy and information gain. This tree enhanced after multiple iterations of each data point using the residual loss to get a Gradient Boosted tree at the extreme | * Classification and regression | * Efficient for both small and large datasets * High execution speed * Performs well on a wide range of complex machine learning tasks | * It has lots of hyperparameters for fine-tuning and, it may lead to overfitting in some cases. |
| Neural Networks (MLPs) | * The multi-layer perceptron (MLP) is a feed-forward artificial neural network that generates outputs from inputs. * It comprises several layers of input nodes connected by a directed graph between the input and output layers. * Perform multiple stages of processing to come to a decision. | * Classification and regression | * Ability to build incredibly complex models * Capacity to capture the information contained within a large volume of data | * It can be a very complex model, particularly for large datasets * Sensitive to scaling of the data * Susceptible to the choice of parameters * Large models need a long time to train |

*Table 1.1: Characteristics of supervised learning algorithms*

1. **LITERATURE REVIEW**

Over the years, numerous studies have been conducted to determine which machine learning algorithms are most fitting for detecting credit card fraud. In their paper, Varmedja et al. [13] used the Credit Card Fraud Detection dataset to experiment with which algorithm fits well for credit card fraud detection. Their experiment used LR, random forest, Naive Bayes (NB), and multilayerperceptron (MLP) algorithms.Because their dataset was highly imbalanced, they used the SMOTE technique for oversampling. Their result shows that the random forest algorithm gives the best result with an accuracy of 99.96%, followed by MLP with an accuracy of 99.93%, NB with an accuracy of 99.23%, and LR with an accuracy of 97.46%.

Vaishnavi et al. [14] also used the credit card fraud dataset. They experimented on original and SMOTE datasets using Isolation Forest, SVM, LR, DT, RF, and local outlier factor models. Additional evaluation metrics, such as the Matthews Correlation Coefficient, were used in addition to accuracy and precision. In the research, LR was found to be the algorithm that gave a better result. Decision tree and random forest classification algorithms were found to be the next most accurate after LR, accordingly. Paper [15] compares different methods such as XGBoost, DT, k-nearest neighbors (KNN), LR, RF, and SVM. Based on an experiment performed on a credit card dataset, they were able to achieve almost the same high accuracy levels in all six of their models. They concluded that all six could be used for detecting fraud.

A study conducted in paper [16] found that the bagging classifier based on the decision tree algorithms was better suited to detect credit card fraud than other algorithms, such as NB, SVM, and KNN. The study evaluated the performance based on four evaluation metrics: Fraud Catching Rate, False Alarm Rate, Balanced Classification Rate, and Matthews Correlation Coefficient. In [17], MLP machine learning algorithm were applied to a credit card dataset in order to detect fraud. Several MLP parameters were used, and performance was evaluated using sensitivity. In the study, MLP with logistic activation function provided the best results, followed by Tanh activation function. As a result of the relatively small number of hidden layers and the small number of nodes used in the study, the highest sensitivity was only 83%.

Study [18] used sensitivity, precision, and time parameters to compare the various supervised learning models such as KNN, NB, DT, LR, and RF. The results of their analysis show that KNN has a greater sensitivity than DT. However, since KNN takes much longer to test the data, they concluded that the decision tree was the best model for fraud detection. Paper [19] analyzed classical supervised machine learning algorithms using a European credit card transaction dataset based on performance metrics such as accuracy, sensitivity, and specificity. By showing the highest sensitivity and specificity, KNN outperformed other machine learning algorithms. Comparatively, LR gave the lowest sensitivity and specificity.

The review shows that no standardized method or solution has been developed, despite the significant amount of research done on fraud detection. Researchers suggest several ways to classify a transaction as fraud or non-fraud, depending on the data type, choice of models, parametrization, and methods used to pre-process the data.

1. **METHODOLOGY**

The research paper is primarily composed of a systematic literature review and an experiment conducted on real-world data. The first part of the paper explains the background of credit card fraud, describes the characteristics of supervised machine learning algorithms, and reviews related scholarly papers. The second part of the paper discusses the dataset used in the experiment, the performance evaluation metrics employed, model building and evaluation, as well as a discussion and conclusion. Figure 3.1 illustrates the research process of this paper.

Diagram

Description automatically generated

*Figure 3.1: Research methodology*

1. **EXPERIMENT ON DATASET**
   1. **Dataset Description**

The experiment section of the study was carried out using a Credit Card Fraud Detection dataset obtained from Kaggle [21]. The dataset has been collected and analyzed during a research collaboration of worldline and the machine learning group of ULB (Université Libre de Bruxelles) on big data mining and fraud detection. The dataset contains transactions made by credit cards in September 2013 by European cardholders. It presents transactions that occurred in two days, where it has 492 frauds out of 284,807 transactions. The dataset is highly unbalanced; the positive class (frauds) account for 0.172% of all transactions.

Due to confidentiality issues, the dataset did not contain the original features. Also, it did not provide more background information about the data. Therefore, it contains only numerical variables resulting from a PCA transformation. In total, the dataset has thirty-one features from which V1, V2, V3…. V28 are the principal components obtained with principal component analysis (PCA), 'Time' and 'Amount' are the only features that have not been altered. The 'Time' feature contains the number of seconds elapsed between each transaction and the first transaction in the dataset. The feature 'Amount' represents the transaction amount, and the feature 'Class' is the response variable, and it takes the value 1 when fraud is detected; otherwise, it takes the value 0 for non-fraud transactions. Table 4.1 summarizes the description of the features.

Table

Description automatically generated

*Table 4.1: Credit card fraud detection dataset feature description*

* 1. **Data Preprocessing and Exploratory data analysis (EDA)**
  2. ***Class Distribution***

The data used in this study has two predictive class labels for the given observations. First, a bar chart was used to examine the distribution of classifications. The result indicated that the data is highly imbalanced. There are only 492 (0.17 %) frauds out of 284,807 transactions, with the non-fraud transaction being 284,315 (99.83%). As studies indicate, Imbalanced classifications pose a challenge for predictive modeling as most of the machine learning algorithms used for classification were designed around the assumption of an equal number of examples for each class [22]. The imbalance classification may result in models with poor predictive performance, more importantly, the minority class.

Chart, bar chart

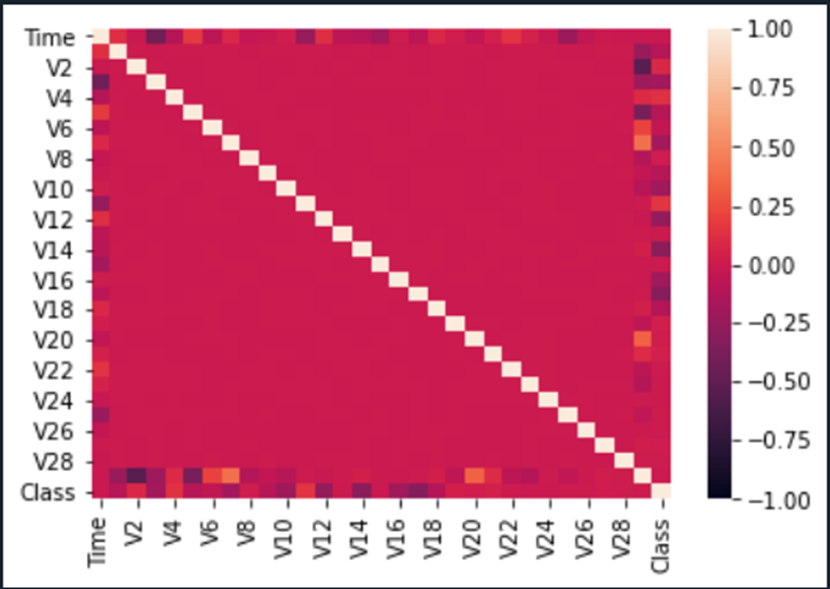
Description automatically generated

*Figure 4.1: An illustration of the distribution of classes in the credit*

*card fraud detection dataset*

* 1. ***Feature Correlation***

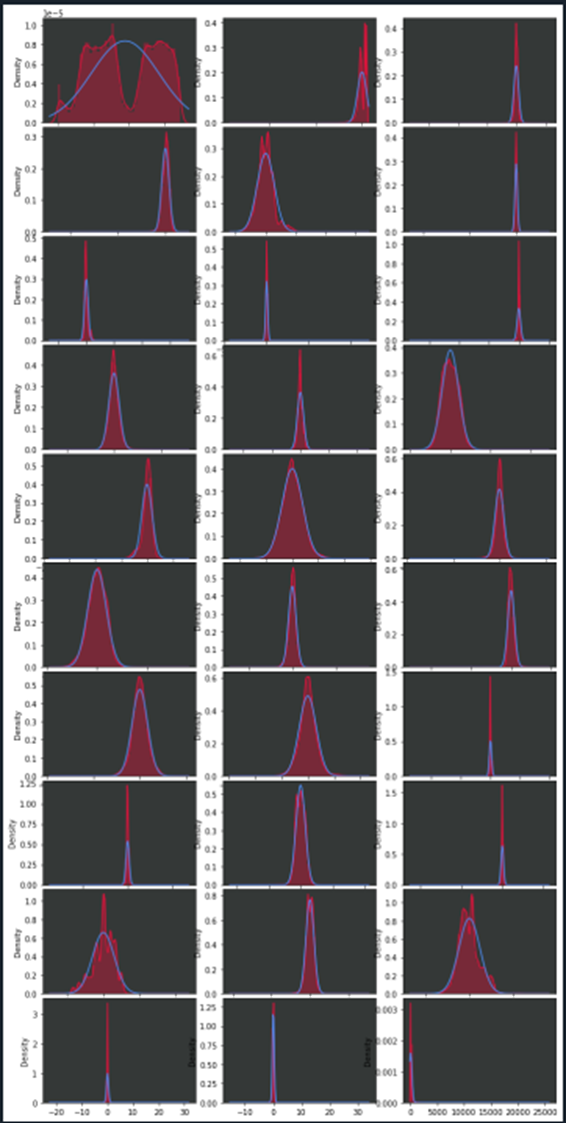
The data has thirty-one features in total. In Figure 4.2, a correlation heat map was used to determine the correlation (a measure of dependence) between all variables. In a heatmap, the closer a correlation is to 1, the stronger the correlation is. As one increases, so do the other, and the closer to 1, the stronger the correlation. A correlation closer to -1 is similar, but one variable will decrease as the other increases rather than both increasing [23]. Therefore, the plot in figure 4.2 illustrates that there are no highly correlated features; however, some features such as 'Amount' 'Time' and 'Class' are slightly correlated with other features. *Note: The 'Amount' feature does not have a label on the heatmap but is located immediately before the 'Class' feature.*



*Figure 4.2: Heat Map illustrating the correlation between features*

* 1. ***Data distribution***

Figure 4.3 gives us an idea of how each feature is distributed. Based on the histograms 'Time' variable comes from the bimodal distribution, while all the other features come from the normal (or gaussian) distribution.



*Figure 4.3: Histogram illustrating the distribution of each feature's values.*

* 1. **Model Evaluation Metrics**

Measuring accuracy alone in binary classification may be misleading. Therefore, this study uses various evaluation metrics such as confusion metrics, accuracy, precision, recall, and F1-score to summarize how well the supervised models performed on the dataset. Based on [5], the evaluation metrics are discussed below.

* 1. ***Confusion metrics***

Confusion metrics are one of the compressive ways to represent the result of evaluating binary classification. The rows correspond to the actual classes, and the columns correspond to the predicted classes. Entries on the main diagonal of the confusion matrix correspond to correct classifications, while other entries tell us how many samples of one class got mistakenly classified as another class.

|  |  |
| --- | --- |
| Predicted negative | Predicted positive |
| Negative class | **TN** | **FP** |
| Positive class | **FN** | **TP** |

*Table 4.2: confusion matrix for binary classification*

* 1. ***Accuracy***

Accuracy is the number of correct predictions (TP and TN) divided by the number of all samples or all entries of the confusion matrix summed up**.**

**Accuracy =**

* 1. ***Precision***

Precision measures how many of the samples that are predicted to be positive are actually turn out to be positive. Precision is used as a performance metric when the goal is to limit the number of false positives.

**Precision =**

* 1. ***Recall***

Recall measures how many of the positive samples are captured by the positive prediction, and it is used as a performance metric when it is necessary to identify all positive samples, in other words, when it is essential to avoid false negatives.

**Recall =**

* 1. ***F1-score***

F1-score, also called F-score or F-measure, is the harmonic mean of precision and recall. Although precision and recall are essential measures, only examining one of them will not give us the complete picture. Since F1-score considers both precision and recall, it can be a better indicator than accuracy for imbalanced binary classification datasets.

**F =**

* 1. **Experimental Setup and Model Performance Evaluation**

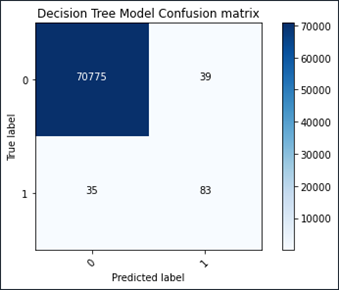
***4.4.1 Experimental Setup***

On the Windows 10 operating system, this experiment was run using Spyder, a scientific Python development environment. Several Python libraries have been used, including Scikit-learn, SciPy, Matplotlib, NumPy, and others. For training and testing purposes, the dataset was split into 75% and 25% for training and testing, respectively. Features such as 'Time' and 'Amount' were pre-processed with the Robust Scaler technique. Moreover, for the confusion matrix analysis, the minority 'Fraud' class was assumed to be a positive class.

***4.4.2 Model Performance Score***

* 1. ***Decision Tree***

Decision trees learn a hierarchy of if/else questions, leading to a decision. In this study, the decision tree model was built upon default parameters with Random state 0. Looking at the confusion matrix, the model has more true positives and true negatives while having fewer false positives and false negatives. Also, looking at the f1-score from the classification report, the non-fraud class has an f1-score of 1, and the fraud class has an f1-score of 0.69. The results suggest that the classifier predicts the classes very well; however, it is more accurate in identifying the majority class (non-fraud) than the minority (fraud) class.

******

|  |
| --- |
| Table  Description automatically generated with medium confidence |

*Figure 4.4: confusion matrix for Decision Tree Table 4.3: Classification report for Decision Tree*

* 1. ***Random Forest***

A random forest is essentially a collection of decision trees, where each tree is slightly different from the others. It often works well without heavy tuning of the parameters. In our case, the random forest consisting of 100 trees was built without tuning any parameters. Analyzing the confusion matrix, the model provides a reasonable result, yielding a small number of false negatives (24) and false positives (2) while yielding a high number of true positives (94). As we can see from the report metrics, both classes have relatively high precision and recall, with an F1-score of 1 for the non-fraud class and 0.88 for the fraud class, respectively.

Chart

Description automatically generated

|  |
| --- |
|  |

*Figure 4.5: confusion matrix for Random Forest Table 4.4: Classification report for Random Forest*

* 1. ***XG-Boost***

XG-Boost(eXtreme Gradient Boosting) is an implementation of Gradient boosted decision trees optimized for speed. In the experiment the XG\_Boost model was built using learning\_rate=0.01, gamma=1, and random\_state=0. The model produces a reasonable f-score for both classes, and based on the evaluation metrics, it can be concluded that the model is effective in identifying the classes.

Chart

Description automatically generated

|  |
| --- |
|  |

*Figure 4.6: confusion matrix for XG-Boost Table 4.5: Classification report for XG-Boost*

* 1. ***Kernelized Support Vector Machine***

The kernelized support vector machine model was built using kernel='rbf' and random\_state=0.

Looking at the confusion matrix, we have more true positives than false negatives and false positives. The report metrics indicate that both classes have comparatively high precision and recall, with an F1-score of 1 for the non-fraud class and 0.83 for the fraud class.

Chart

Description automatically generated

|  |
| --- |
|  |

*Figure 4.7: confusion matrix for SVM Table 4.6: Classification report for SVM*

* 1. ***Logistic Regression***

In order to generate the logistic regression model, the default parameters were chosen, and random\_state was set to 0. Based on the analysis results, both classes have comparatively fair precision and recall, with an F1-score of 1 for the non-fraud class and 0.72 for the fraud class, and in general, the model tends to perform well.

Chart

Description automatically generated

|  |
| --- |
| ***Table  Description automatically generated*** |

*Figure 4.8: confusion matrix for Logistic Regression Table 4.7: Classification report for Logistic Regression*

* 1. ***Neural Networks***

The neural network model has been constructed using the MLP Classifier with two hidden layers each containing 100 hidden units(hidden\_layer\_sizes= [100,100]), max\_iter= 1000, activation='tanh', alpha=1, random\_state=0. Based on the confusion matrix, the Neural Networks model performed poorly at identifying the fraud class and did well in identifying the non-fraud class. It generates more false negatives than true positives. Based on the classification report, the fraud class has a very low recall score but a high precision score. As a result, we obtained an f-score of 0.51 for the fraud class and an f-score of 1 for the non-fraud class.

|  |
| --- |
| **Table  Description automatically generated** |

Chart

Description automatically generated

*Figure 4.9. confusion matrix for Neural Network Table 4.8: Classification report for Neural Network*

1. **SUMMARY AND DISCUSSION**

|  |  |  |  |
| --- | --- | --- | --- |
| **Model** | **Accuracy**  **(On test set)** | **F1-Score**  **(On fraud class)** | **Running Time (In seconds)** |
| **Decision Tree** | 99.90% | 0.69 | 17.23 |
| **Random Forest** | 99.96% | 0.88 | 186.79 |
| **XG-Boost** | 99.96% | 0.87 | 73.25 |
| **Kernelized Support**  **Vector Machine** | 99.95% | 0.83 | 59.26 |
| **Logistic Regression** | 99.92% | 0.72 | 2.12 |
| **Neural Networks** | 99.88% | 0.51 | 52.43 |

*Table 5.1: Model performance summary*

Table 5.1 summarizes the performance of the models. According to the accuracy measure, all the supervised models performed well, yielding an average accuracy score of over 99%. However, since accuracy is an inadequate measure for quantifying predictive performance in an imbalanced setting, it is difficult to determine which of these algorithms is actually most efficient in determining fraud by simply relying on accuracy measures alone. Therefore, it was necessary to consider other evaluation metrics in addition to accuracy in order to assess the effectiveness of the classification models.

As the study's objective is to detect fraud, a high recall and precision score is essential; therefore, the F-measure was used to gauge the model's performance, since it gives equal weight to both precision and recall. Based on Table 5.1, the random forest model obtained the highest F-score of 0.88, followed by the XGBoost and SVM, which had F-scores of 0.87 and 0.83, respectively. The next most effective model was the logistic regression model, with a score of 0.72, followed by the decision tree model, with a score of 0.69. The least effective model was the neural network, with an F-score of 0.51.

Other than accuracy and F-score, the supervised models were analyzed by their computation time, the amount of time it took to run (learn and predict). This was done by measuring the amount of time in seconds. Logistic regression was the fastest model in terms of computation time, followed by decision tree and neural network as the second and third fastest models, and SVM, XGBoost, and random forest as the slowest run models. Overall, each model took approximately 65 seconds or one minute per model based on the average run time.

On the basis of the experimental findings, random forest, XGBoost, and SVM are the best models that can be used to detect fraudulent transactions, yet three of these models take the longest to run. In contrast, the neural network is the fastest in computation time but the worst in classification performance.

1. **CONCLUSION AND RECOMMENDATION**

In general, this study aimed to review the most popular supervised machine learning algorithms and their effectiveness in identifying fraudulent and non-fraudulent transactions. In this paper, various scholarly articles covering credit card fraud were studied and discussed. An experiment was also conducted to evaluate the performance of six supervised learning algorithms on a real-world dataset consisting of credit card transaction data. Based on the experiment results, random forest proved to be the most efficient algorithm for detecting fraudulent transactions with a reasonable computation time.

The author recommends that future studies use resampling techniques to ensure that classes within the data are distributed evenly and use other performance evaluation metrics, such as Area Under the Precision-Recall Curve (AUPRC) to further assess the performance of the models in imbalanced credit card transactions data. Moreover, the experiment can be extended to other supervised learning algorithms not discussed in this article.

1. **REFERENCES**
2. *Credit card Fraud Statistics*. Shift Credit Card Processing. (2021, January 4). https://shiftprocessing.com/credit-card-fraud-statistics.
3. An Empirical Study of AML Approach for Credit Card Fraud Detection–Financial Transactions. By A. Singh, A. Jain
4. Andrea, D. P. (2015). Adaptive Machine Learning for Credit Card Fraud Detection. *Machine Learning Group, Ed.: Université Libre de Bruxelles*.
5. Zareapoor, M., & Shamsolmoali, P. (2015). Application of credit card fraud detection: Based on bagging ensemble classifier. *Procedia computer science*, *48*(2015), 679-685.
6. Müller Andreas Christoph, & Guido, S. (2016). *Introduction to machine learning with python: A guide for data scientists*.
7. Shah, C. (2020). *A hands-on introduction to Data Science*. Cambridge University Press.
8. Rokach, L., & Maimon, O. (2015). *Data mining with decision trees: Theory and applications* (2nd ed.). World Scientific.
9. S. Ray, "A Quick Review of Machine Learning Algorithms," 2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon), 2019, pp. 35-39, doi: 10.1109/COMITCon.2019.8862451.
10. Jiang, T., Gradus, J. L., & Rosellini, A. J. (2020). Supervised machine learning: A brief primer. *Behavior Therapy*, *51*(5), 675–687. https://doi.org/10.1016/j.beth.2020.05.002
11. Brownlee, J. (2019, August 12). *Logistic regression tutorial for machine learning*. Machine Learning Mastery. Retrieved September 28, 2021, from https://machinelearningmastery.com/

logistic-regression-tutorial-for-machine-learning/.

1. Dhingra, P. (2021). Glass Identification Using Extreme Gradient Boosting Algorithm. *context*, *1*(2), 12.
2. Brownlee, J. (2021, February 16). *A gentle introduction to XGBoost for applied machine learning*. Machine Learning Mastery. Retrieved September 28, 2021, from https://machinelearningmastery.com/gentle-introduction-xgboost-applied-machine-learning/.
3. Varmedja, D., Karanovic, M., Sladojevic, S., Arsenovic, M., & Anderla, A. (2019). Credit Card Fraud Detection - Machine Learning methods. 2019 18th International Symposium INFOTEH-JAHORINA (INFOTEH), 1-5.
4. Dornadula, Vaishnavi & Geetha, S. (2019). Credit Card Fraud Detection using Machine Learning Algorithms. Procedia Computer Science. 165. 631-641. 10.1016/j.procs.2020.01.057.
5. A.Chouksey, R.Nimje, J. Saraf (2021) Fraud Detection in Credit Cards with Machine Learning. International Journal for Research in Applied Science & Engineering Technology (IJRASET)ISSN: 2321-9653.
6. Zareapoor, M., & Shamsolmoali, P. (2015). Application of credit card fraud detection: Based on bagging ensemble classifier. *Procedia Computer Science*, *48*, 679–685. https://doi.org/10.1016/j.procs.2015.04.201
7. Pillai, T. R., Hashem, I. A. T., Brohi, S. N., Kaur, S., & Marjani, M. (2018, October). Credit card fraud detection using deep learning technique. In 2018 Fourth International Conference on Advances in Computing, Communication & Automation (ICACCA) (pp. 1-6). IEEE.
8. Khatri, S., Arora, A., & Agrawal, A. P. (2020, January). Supervised machine learning algorithms for credit card fraud detection: a comparison. In 2020 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence) (pp. 680-683). IEEE.
9. D. Dighe, S. Patil and S. Kokate, "Detection of Credit Card Fraud Transactions Using Machine Learning Algorithms and Neural Networks: A Comparative Study," 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA), 2018, pp. 1-6, doi: 10.1109/ICCUBEA.2018.8697799.
10. 8. F.Y, O., AkinsolaJ.E., T., Awodele, O., HinmikaiyeJ., O., Olakanmi, O., & Akinjobi, J. (2017). Supervised Machine Learning Algorithms: Classification and Comparison. International Journal of Computer Trends and Technology, 48, 128-138.
11. ULB, M. L. G.-. (2018, March 23). Credit Card Fraud Detection. Kaggle. Retrieved 2021, from https://www.kaggle.com/mlg-ulb/creditcardfraud.
12. Scientist, D. D. J. D., &amp; Engineer, K. B. S. (2021, September 27). Machine learning mastery. Machine Learning Mastery. Retrieved 2021, from https://machinelearningmastery.com/.
13. Shabuki. (2019, February 14). Retrieved 2021, from (version: 2019-02-14): https://stats.stackexchange.com/q/392519s.