HÖGSKOLAN I SKÖVDE

School of Informatics
Master program in Data Science
Scientific Theory in Informatics A1N
IT706A

Comparative Study on Machine learning algorithms in Credit Card Fraud Detection

By Welemhret Welay Baraki

January 2021

University of Skövde, Skövde, Sweden

Table of Contents

1.0. Focus of the Case Study

Cyber and information security is a growing problem for governments, businesses, and individuals. Credit cards are cards issued to bank customers to enable cardholders/customers to pay for purchases in stores and other online payments. Credit card transactions subject to different cyber-attacks. Businesses and individuals should have to ensure and secure their credit card transactions and businesses have to secure their competitive advantage in the market.

This case study mainly focuses on the computation of the informatics discipline, on comparative analysis of some selected machine learning classification algorithms in credit card fraud detection.

During the whole process of the study credit card transaction data set was collected, pre-processing and data transformation, Building of the different types of classification models, evaluating each classifier, and finally compare the classifiers and make conclusions and recommendations to users, and researchers.

The case study is relevant for researchers, data scientists, and businesses (financial institutions) to support their decision on choosing the best, optimal and accurate machine learning algorithm for their application development and research in fraud detection and mitigation.

The focus and main purpose of the study is to provide a recommendation of machine learning algorithms for fraud detection based on comparative analysis of different classification algorithms on imbalanced credit card transaction data sets.

Credit cards are the most common, purchasing method, as a result, it is also exposed to different fraud transaction attacks. An effective way of detecting fraudulent transactions using machine learning algorithms should have to adopt businesses to secure their credit card transactions, unless the traditional rule-based security measures to mitigate new threat/fraud transactions/. The problem of fraud detection and the unavailability of balanced data motivated us to engage and study in this hot area of cybersecurity.

Generally, this study is not designing and developing a new IT artifact but it is a comparative analysis that mainly identifies the weaknesses, and strengths, evaluation and observations, computational complexities of the selected machine learning algorithms on fraud detection.

1.1. Informatics Sub-disciplines

The focus of this case study is on the computation sub-discipline part of the informatics where models and methods are compared to identify the suitable method for the implementation/identification/ of fraudulent and non-fraudulent from real-world imbalanced data sets. The main challenge in fraud detection is the problem of having an imbalanced data set during the development of machine learning models that mainly biases the results of the model (classification model) in which compromises the effective and efficient identification of fraudulent transactions properly. More specifically, this study attempts to compare and find the best, optimal, and effective machine learning algorithm that effectively detect fraudulent transactions, by evaluating the models, and calculating the computational complexities of the selected models.

1.2. Candidate theories and techniques

Fraud detection problem is a typical example of a binary classification problem. Classification is a task of a supervised machine learning task. The main objective of this study is to draw an inference from the comparative analysis of this study of different machine learning

models/algorithms/techniques/ in which fraudulent transactions are effectively identified. There are plenty of machine learning algorithms that are used for fraud detection. However, exploring and comparing all the machine learning algorithms is a vast task that can't be finished in this study. Therefore, this study is limited to Artificial Neural Network, k-nearest neighbors, Decision Tree, and Logistic Regression. The selected algorithms are briefly explained in the following paragraphs.

Logistic regression is widely used in binary dependent variable machine learning problems. K-nearest neighbor is a non-linear classifier that predicts a class by identifying its k-nearest neighbor's class based on the Euclidean distance. Decision Tree is the most powerful classifier where the decision tree is a tree structure, each internal node represents a test on an attribute, the branch represents an outcome of the test (True or False), and leaf nodes hold a class label. Artificial Neural Networks is inspired by the biological neurons, suitable to easily high dimensional data. From the different types of the ANNs, a class of feedforward Multilayer Perceptron that uses backpropagation for training was used for predicting the class.

1.3. Phases of the Development Life Cycle

Since this case study is a data science project the particular development life cycle mainly tracks the following phases of the development life cycle adopted from Domino's Data Science Life Cycle[1]. Figure 1 below illustrates the phases of the development life cycle.

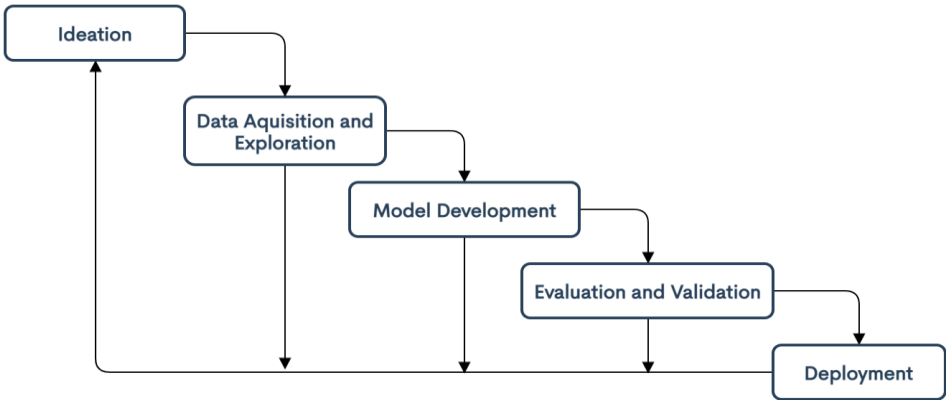

Figure 1: Phases of the development life cycle

The phases of the development life cycle of this study are ideation, data collection and data transformation, building the machine learning models, Evaluation, deployment and recommendation based on the results of the comparative analysis of our models.

The first phase ideation is the process of defining the business problem and conducting business analysis. After defining the business problem the next phase data acquisition, collection, and exploration. At the stage, the underlying data is going to be analyzed, preprocessed, and transformed into suitable formats for the machine learning algorithms. The third phase is Machine Learning model development, setting the required parameters, and train classification models. The fourth phase of the development life cycle is validation and Evaluation with the data sets that are not provided previously to the machine learning algorithms and summarize the results of each model. The final phase is the deployment phase, where the researchers are expected to develop guidelines to the users on how to deploy the application in their business infrastructure and scientific recommendations will be given to consult and consider during deployment and use.

2.0 Specification of the Application Scenario

2.1. Problem Identification

Cyber Security and information security is a broad area that triggers for governments, businesses, institutions, and individuals. Security issues in the digital age still are increasing alarmingly and seriously. Fraud detection and prevention a hot research area in which different stack holders are backing researchers and technologists to develop and implement technologies, methods, and techniques. As a result transition from rule-based security measures to learning systems extensively use machine learning algorithms to model and prevent suspicious business transactions to secure their competitive advantage in the market. The birth of machine learning algorithms envisioned a new hope for cybersecurity. With the rise of hundreds of new threats every day, still, the research and development of fraud detection are not saturated, in which credit card fraud detection systems are one part.

Machine learning algorithms, in particular, faces two major fraud detection problems. The first is the unbalanced class size of transactions that are non-fraudulent and fraudulent to counteract fraudulent ones. For machine learning model development some sampling[1] techniques are used with rational class distributions. The most common sampling techniques are random over and under-sampling techniques. According to [2] random under-sampling is preferred from the oversampling with large data sets. In this study, we change the proportion of fraud to non-fraud cases in the training data using random under-sampling and examine its impact on the four machine learning techniques and considering different performance measures.

The second problem in developing supervised machine learning models for credit card fraud can rise from the potentially undetected fraud transactions, leading to mislabeled cases in the data to be used for building the model.

Generally, the detection of fraudulent credit card transaction occurs twice: there is a real-time fraud detection system at the time of the transaction, and the second one is the transaction records are analyzed to raise warnings about unusual purchase transactions.

The study is based on the data[2] collected in September 2013 from the European credit cardholders. This data set contains credit card transactions that occurred in two days and is highly unbalanced.

2.2. Proposed Solution

Throughout the research process, the domino's[1] data science project life cycle was widely employed starting from ideation to deployment. This life cycle is particularly adopted to create a suitable framework for the development of machine learning models. The selected algorithms for this research are Artificial Neural Network, k-nearest neighbors, Decision Tree, and Logistic Regression.

The motivation to conduct this study is identification of classification algorithms that handle imbalanced data, analyzing computational complexity of the models.

---

[1] Random Under-sampling is used to balance the fraud and legitimate data sets.

[2] The Anonymized credit card transactions labeled as fraudulent or genuine are collected from https://www.kaggle.com/mlg-ulb/creditcardfraud

There is no new system that is going to develop but as a researcher, the study is mainly focused on the deep comparative analysis of machine learning models based on the credit card data sets. The results of models were evaluated using different evaluation metrics, recommendations, future research directions will be forwarded. The comparative study will give a deeper insight on machine learning algorithms in fraud detection, how the performance metrics does affect/bias the decision making, exploring computational complexities and weaknesses and strengths of the models.

Research Questions

As a researcher, there are some research questions that we are going to answer in this study. The research questions are stated as follows:

1. How to identify effective machine learning algorithms for fraud detection?
2. How to compare machine learning algorithms concerning the performance metrics?
3. Do the mathematical foundations affect the performance of these machine learning algorithms?
4. How to compare and contrast machine learning algorithms with their computational complexities?

2.2.1. Objectives

The general objective of this study is to conduct a comparative study on the selected machine learning models and select the best one.
The specific objectives of this study are listed below:

- Collect credit card transactions from different sources.
- Transform the collected data into suitable data formats.
- Build and train Machine learning models
- Test and Evaluate the classification algorithms
- Conduct Comparative analysis and conclude from evaluation results using different evaluation metrics.
- Calculating computational complexities
- Develop deployment guidelines and recommendations

2.2.2. Functional Requirements

Functional requirements are requirements that are compulsory and are defined as a collection of inputs, processes, and outputs.

Table1: Functional requirements of the fraud detection project

Core Function/Task/Module	Functions
Data Transformation Module	• Data Preprocessing and Exploration • Scaling • Handling Missing and null values • Balancing the imbalanced data
Classification Module	• Splitting the balanced data into training and test data • Building and training the Classification models
Evaluation Module	• Prediction/testing the machine learning models • Evaluating the performance of the classification models with different performance metrics.

2.2.3. Non-Functional Requirements

These are supportive and complementary requirements for the core functional requirements of the proposed system.

- **Accuracy**: the system shall produce accurate results and detection of the fraud transactions. Accuracy in this case is very difficult to achieve and ensure.
- **Usability**: The proposed system provides a user-friendly user interface through which users can communicate the system easily and comfortably.
- **Availability**: the system must be available 24/7
- **Performance**: the system should be responsive and it should not frustrate users.

2.3. Problem modeling

Based on the problem specification and functional requirements we have to create a formal model representation to our problem and functional requirements.

The research project consists of the following three main modules:

i. **Data Transformation**: Data can be available in structured, unstructured, and semi-structured formats. To use for machine learning algorithms, we need to preprocess the data. The main functions of the data Preprocessing and Exploration, Scaling, Handling Missing and null values, and balancing the imbalanced data.

ii. **Classification Module**: The data transformation module produces clean and balanced data which used as input to the classification module. The main functions of the classification module are splitting the balanced data into training and test data sets, and Building and training the Classification models. The classification models which are implemented in this module are Artificial Neural Network, K-Nearest Neighbors, Decision Tree, and Logistic Regression.

iii. **Evaluation Module**: The evaluation module takes the predicted values and actual values and measure and assures the performance of each machine learning models to be able to compare and analyzes the results.

2.4. System analysis and specification

Based on the problem statement and required requirements the system specification for the machine learning models was illustrated using the following two-level data flow diagrams.

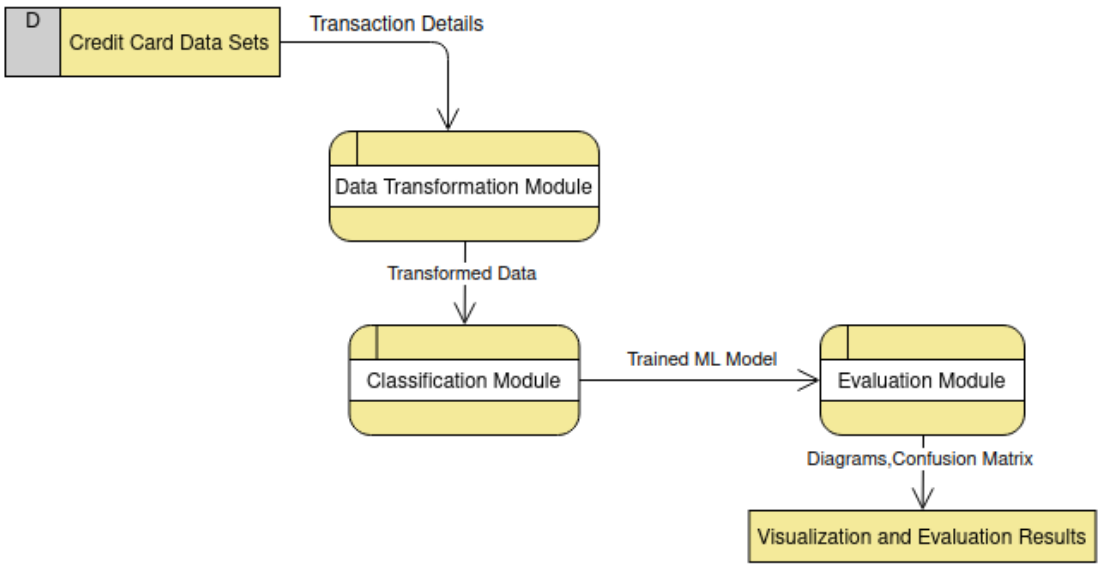

Figure 2: Data flow diagram level 0

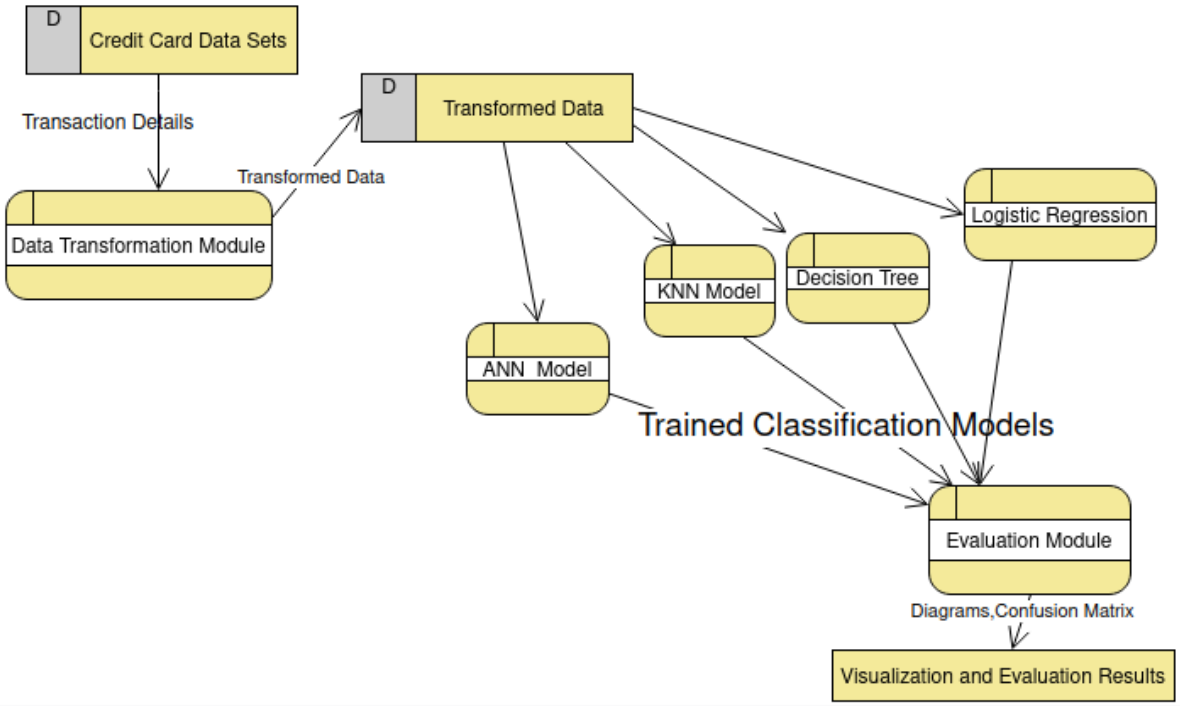

Figure 3 Data flow diagram level 1

The data flow diagram in figure 2 is a high-level abstract representation of the system and the last one figure 3 is a more detailed representation of the application scenario.

2.5. Data Collection[3]

The datasets include purchases made by European cardholders with credit cards in September 2013. This dataset presents transactions that happened within two days and we had 492 frauds out of 284,807 transactions. The dataset is extremely unbalanced, with the positive class (frauds) responsible for 0.172 % of all transactions. It includes only numeric input variables that are the result of a PCA transformation.

3.0 Survey and Selection of theories and techniques

The selection of the best model will be conducted on the comparative analysis section. In this section mainly explores the different types of classification machine learning algorithms from different kinds of literature.

3.1. Logistic Regression

Logistic regression is the statistical fitting of a logistic S-curve or logit function to a dataset to measure the probability of occurrence of a given categorical event based on the values of a series of independent variables. Logistic regression is widely applicable in image segmentation, handwritten recognition, and health care.

The logistic hypothesis can be defined as using the following equations:

$H\theta(x) = g(\theta^T x)$---------- (1)

$g(z) = \frac{1}{1+e^{-z}}$----------(2)

- **Maximum Likelihood Estimation:** Maximum Likelihood Estimation[4] is used to estimate the parameters of the logistic regression model. The parameters of the model can be estimated by maximizing the likelihood function that predicts the mean of a Bernoulli distribution for each data point.
- **Sigmoid Function**[5]: The logistic sigmoid function is used to return a probability value which can then be transformed and mapped to two or more discrete classes.

Logistic regression can be used where the probabilities in binary classes are required whether the credit card transaction is fraudulent (1) or non-fraudulent (0) and is based on the concept of Maximum Likelihood estimation.

3.2. Decision Tree

A Decision Tree classifier is a machine learning algorithm for classification and prediction. The tree is comprised [3] of internal nodes which represent the features of the data set, branches represent the decision rules, and leaf node denotes the outcome.

---

[3] The credit card data are collected from https://www.kaggle.com/mlg-ulb/creditcardfraud (accessed 15 Dec 2020)

[4] Maximum Likelihood Estimation: https://machinelearningmastery.com/what-is-maximum-likelihood-estimation-in-machine-learning/ (accessed 28 December 2020)

[5] Logistic Regression https://ml-cheatsheet.readthedocs.io/en/latest/logistic_regression.html#:~:text=Unlike%20linear%20regression%20which%20outputs,two%20or%20more%20discrete%20classes.

The two impurity[6] methods that measure the homogeneity of the labels for classification are:

- **Entropy**[7] is the amount of information is needed to accurately describe the sample. So if the sample is homogeneous, means all the elements are similar than Entropy is 0, else if the sample is equally divided then entropy is a maximum of 1.

$$Entropy = -\sum_{i=1}^{n} p_i * \log(p_i)$$

- **Gini index/ Gini impurity** is a measure of inequality in the sample. It has a value between 0 and 1. Gini index of value 0 means samples are perfectly homogeneous and all elements are similar, whereas, Gini index of value 1 means maximal inequality among elements. It is the sum of the square of the probabilities of each class.

$$Gini\,index = 1 - \sum_{i=1}^{n} p_i^2$$

After calculating the impurity, the CART[8] the algorithm is used to construct the tree classifier. The decision tree recursively partitions the data set using the depth-first/breadth-first/ greedy approach and finishes the recursion when all the elements have been assigned a class label. The best partitioning will be the one in which the subsets don't overlap.

In the decision tree, pruning[9] is one such important method to optimize its efficiency that typically uses statistical measures to remove the least reliable branches or the branches backed by a few samples. Generally, the Decision tree is computationally fast, easy to implement, can handle any type of data, and is convenient for regression and classification (Fraud /Not fraud) problems.

3.3. K-Nearest Neighbors[10]

K-Nearest Neighbors algorithm is a supervised machine learning algorithm used to solve classification and regression problems. The algorithm assumes that similar data points are close to each other. The similarity of the data points is determined using the distance calculation. The distance calculation can be achieved by Euclidean Distance, Hamming, manhattan, Minkowsky, or Chebychev Distance.

Euclidean Distance formula: $d(x, x') = \sqrt{(x_1 - x_1')^2 + \ldots + (x_n - x_n')^2}$ ----------(1)

Finally, the input x assigned classes with the highest probability(Majority vote) using the following mathematical formula:

$$P(y = j|X = x) = \frac{1}{K} \sum_{i \in \mathcal{A}} I(y^{(i)} = j)$$ ----------(2)

---

[6] Impurity https://spark.apache.org/docs/1.3.0/mllib-decision-tree.html#:~:text=The%20node%20impurity%20is%20a,measure%20for%20regression%20(variance).

[7] Maths behind Decision tree "https://medium.com/@ankitnitjsr13/math-behind-decision-tree-algorithm-2aa398561d6d"

[8] Decision Tree Classification Algorithm "https://www.javatpoint.com/machine-learning-decision-tree-classification-algorithm"

[9] Decision Tree Pruning, time and space complexity: https://heartbeat.fritz.ai/understanding-the-mathematics-behind-decision-trees-22d86d55906

[10] KNN https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761

K-value Selection[11]: This is the responsibility of a data scientist is the selection of the value of k for the KNN algorithm. Generally, a small value for K provides the most adjustable fit, which will have low bias but the high variance and our decision boundary will be more irregular. On the other hand, a higher K averages more voters in each prediction and hence is more flexible to outliers. Larger values of K will have smoother decision boundaries which mean lower variance but increased bias.

According to [3] the performance of KNN depends on three factors: the distance metrics measure to locate the nearest neighbors, the distance rule helps the algorithm to classify new data points into a class, and the value of K which decides the number of neighbors.
Having sufficient computing resources to speedily handle the data you are using to make predictions, KNN can be used in solving problems that have solutions that depend on identifying similar objects.

3.4. Artificial Neural Networks

An ANN is based on a collection of interconnected artificial neurons/nodes/ which receives an input then processes and produces an output to the neurons connected to it. Different layers can perform various transformations on their inputs using different weights and biases.

The training of supervised learning of a neural network from a given example is conducted by determining the difference between the predicted value and a target output and then adjusts its weights and biases according to a learning rule. Successive adjustments will cause the model to produce a similar target output and minimizes the error.

Components of the Artificial Neural Networks:

- **Neurons:** To determine the output of the neuron, we first take the weighted sum of all inputs, measure the weights of the connections from the inputs to the neurons, and apply the bias term. Each artificial neuron has inputs and generates a single output that can be transmitted to several other neurons.
- **Connections and weights:** A given neuron can have multiple inputs and output connections. The network consists of connections, each connection providing the output of one neuron as an input to another neuron. Each connection is assigned a weight that represents its relative importance.
- **Propagation function:** The propagation function computes the input to a neuron from the outputs of its predecessor neurons and their connections as a weighted sum and adds up a bias.

A Multi-Layer Perceptron (MLP) is a composition of an input layer, at least one hidden layer, and an output layer. If an MLP has two or more hidden layers, it is called a deep neural network (DNN).

[11]How KNN Works https://medium.com/@rdhawan201455/knn-k-nearest-neighbour-algorithm-maths-behind-it-and-how-to-find-the-best-value-for-k-6ff5b0955e3d

4.0 Comparative analysis of selected approaches

4.1. Comparison of Selected Machine Learning Models

Table 1 Strengths and weaknesses of Machine Learning models

	Models/Techniques	Strengths	Weaknesses
1	Logistic Regression	1. Efficient and fast that doesn't consume many resources. 2. Logistic regression has low variance and so is less prone to over-fitting 3. Suitable for linearly separable data sets	1. Not suitable for multiclass classification 2. It constructs linear boundaries. 3. It is tough to obtain complex relationships
2	Decision Tree	1. Can be scaled up to very complex. 2. Handle numerical and categorical data well. 3. Requires less effort from users 4. Robust to handle missing values and outliers.	1. Easily face overfitting problem 2. Unstable meaning a small variation in the data point may result in a completely different tree.
3	K-Nearest Neighbors	1. Robust to noisy data 2. No training phase 3. Learns complex data easily 4. Used for classification, and regression	1. Determining the k value will result in a different solution 2. Not suitable for high-dimensional data 3. Not clear which type of distance metric.
4	Artificial Neural Networks	1. ANN have capabilities of parallel processing 2. Failure in one neural network element will not affect the rest of the process. 3. Neural networks are suitable for any type of problem 4. By Implementing an appropriate learning algorithm, ANN can be made to learn without reprogramming.	1. ANN requires a large amount of processing power and time. 2. ANNs requires huge amounts of data.

4.2. Model evaluation results and Observation

Performance metrics or error measures[4] are critical elements of the evaluation frameworks in many machine learning research areas. In machine learning, the performance metrics are used to compare and contrast the predicted values of the models with actual values or classes. Categories of the classification can be measured based on the resemblances (similarities) or differences (dissimilarities) of examples in a specific context. The performance comparison of the classifiers is evaluated based on accuracy, precision, recall, f1-score, Matthews correlation coefficient, and balanced classification rate are explained and analyzed in detail below.

The most common performance metrics accuracy tells us that how much the predicted values are closer to the actual values. Accuracy[5] is often not a good measure, particularly where data is imbalanced and the common challenge is that if the negative class is dominant, high accuracy can only be obtained as long as we predict the dominant classes.

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \text{ ---------- } (i)$$

Where: TP –True Positive, TN –True Negative, FP –False Positive, and FN –False Negative

The observation from the skewed data the 99.9% accuracy in all the machine learning models shows that the negative class is dominating and biasing the accuracy measure which signifies accuracy is not the correct

measure for the imbalanced credit card fraud data sets. The accuracy observation for the balanced data sets is for logistic regression (94%), Decision Tree (92%), KNN (94%), and ANN (95%). The observation on the balanced dataset shows that the accuracy measures are similar with some slight performance difference and the Artificial Neural Network has a high-performance measure in identifying the credit card fraud transaction positive and negative classes. Still, this shows that accuracy is not the correct measure in this context for the evaluation of imbalanced classification in real-world scenarios.

The second metric used was precision[6] which draws our confidence denotes the proportion of predicted positive classes that are correctly real positives.

$$Precision = \frac{TP}{TP+FP} \text{ ---------- } (ii)$$

The precision result from the skewed data was observed that the Artificial Neural Network and K-Nearest Neighbors Models results in approximately 90% precision and achieves promising result in classifying the positive classes as positives than the logistic regression and decision tree. Besides, the precision observation on the balanced data the KNN (97.2%) and Logistic regression (98.5%) models are showing a promising result than that of decision tree and artificial neural networks. The precision of artificial neural network and KNN in both the skewed and balanced data sets is high and have low variability and deviations.

The recall is how many of the true positives were recalled (found) and can be calculated as the ratio of true positives with true positives and false negatives.

$$Recall = \frac{TP}{TP+FN} \text{ ---------- } \mathbf{(}iii\mathbf{)}$$

According to the evaluation results of using the recall metrics among the four machine learning models, the artificial neural network model results in 80% (on skewed data) and 95% (on balanced data sets), which signifies that the model has a high rate of identifying the actual positives in this case fraud transactions correctly. The other model's Logistic regression, decision tree, and KNN showed low recall values on imbalanced data sets and have measures of 55.5%, 73.5%, and 71.3% respectively. This amplifies, logistic regression, decision tree, and KNN models are not suitable and resilient to handle imbalanced data sets. On the other hand, their performance result of recall on the balanced data set showed a remarkable improvement but still, the Artificial Neural Network is leading the score on identifying the fraudulent transactions.

F-1 Score is a measure of a test's accuracy which conveys the balance between the precision and the recall used when there are imbalanced classes as in credit card fraud transactions.

$$F1\ Score = 2 * \frac{Precision*Recall}{Precision+Recall} \text{ ---------- (iv)}$$

The higher the F1-Score values is the better the model. The results of the F1-Score of logistic regression, decision tree, k-nearest neighbors, and artificial neural networks were 66.2%, 74.5%, 80%, and 85% on imbalanced data sets and 93.2%, 92%, 93.5%, and 94.7% for the balanced data set evaluation respectively. The F1-score revealed that the ANN model is a good model in credit card fraud detection prediction both in balanced and imbalanced data sets, whereas the logistic regression is poor in detecting imbalanced credit card transactions.

Matthews correlation coefficient[7] is an evaluation metric for binary classification problems. More importantly, the metric can be used for evaluating imbalanced data sets.

$$Matthews\ correlation\ coefficient = \frac{(TP*TN)-(FP*FN)}{\sqrt{(TP+FP)*(TP+FN)*(TN+FP)*(TN+FN)}} \text{ ---------- (v)}$$

The values of this evaluation metric for linear regression (67.5%) and decision tree (74.5%) is not as good as the results of KNN (80%) and ANN(85%) on the imbalanced data sets. The performance result on the

balanced data sets of the machine learning models range from 84.6%-89.2% and the Artificial Neural Network model shows the highest result among other models in fraud transaction detection.

The balanced classification rate[12] or balanced accuracy is the arithmetic mean of recall (sensitivity) and the true negative rate (specificity) of the classification models.

$$Balanced\ Classification\ Rate = {}^{1}/_{2}\left(\frac{TP}{TP+FN} + \frac{TN}{TN+FP}\right) \text{ ---------- (vi)}$$

Finally, the obtained result of the balanced classification rate metric from the four machine learning models on skewed data in which linear regression (77.6%), decision tree (86.7%), KNN (85.6%), and ANN (89.2%) of the testing examples are correctly classified as positives and negatives. The results of the balanced classification rate on classifying the respected classes correctly on the balanced data for linear regression, decision tree, KNN, and ANN are 93.6%, 92.2% 93.7%, and 94.6% respectively. Except for the linear regression, the other three models shows a promising result on correctly classifying the fraud and legitimate transactions on the imbalanced data. In balanced data, the machine learning algorithms show great improvements and their test result is good.

Table 2 Summary of Performance Evaluation Results of imbalanced data sets of Machine Learning models

Model	Accuracy	Precision	Recall	F-1 Score	MCC	Balanced Accuracy
Logistic Regression	99.91	82.76	55.17	66.21	67.53	77.58
Decision Tree	99.91	75.52	73.46	74.48	74.45	86.71
K-Nearest Neighbors	99.94	89.85	71.26	79.49	80	85.63
Artificial Neural Networks	99.95	90.07	80.27	84.89	85	90.12

Table 3 Summary of Performance Evaluation Results of balanced data sets of Machine Learning models

Model	Accuracy	Precision	Recall	F-1 Score	MCC	Balanced Accuracy
Logistic Regression	93.58	98.48	88.44	93.19	87.60	93.55
Decision Tree	92.23	94.33	89.86	92.04	84.55	92.23
K-Nearest Neighbors	93.58	97.16	90.13	93.52	87.42	93.68
Artificial Neural Networks	94.59	94.08	95.33	94.70	89.19	94.58

The above performance measures shown in tables 2 and 3 discovered that logistic regression and decision tree are not good at classifying the data sets correctly as legitimate and fraudulent transactions and have high variations in the measured results of the balanced and imbalanced data sets. Even though the k-nearest neighbor classifier and Artificial Neural Network are better at detecting fraudulent transactions but in the comparison between the two machines learning models the artificial neural network performance is better than that of KNN on balanced data and imbalanced data. The improvement in performance in different measures shows that balancing data set using sampling techniques greatly improves the performance of different machine learning models on fraud transaction detection.

[12] Balanced Classification Rate: https://statisticaloddsandends.wordpress.com/2020/01/23/what-is-balanced-accuracy/ (accessed on 20 January 2021)

Based on the performance results Artificial Neural Network is going to be selected as the best and recommended machine learning model.

4.3. Computational Complexity the Machine Learning Models

In computing, time complexity[13] is a metric of computational complexity that defines how long computing time it takes to execute an algorithm, whereas space complexity measures how much memory an algorithm needs to run in terms of the input size. Time complexity and space complexity plays an important role in determining the efficiency of machine learning models. Models that require small space and fast running time in identifying and detecting fraud transactions are indispensable for the success of financial businesses and stay in the competitive environment. The timely response for the fraud transactions depends on the early and fast detection of the fraud transaction which leads to risk mitigation and securing legitimate credit card transactions mostly depends on the efficiency of the program.

Even though the training is a one-time process, the training time of machine learning models can be ignored and testing/running time complexity will be computed based on their internal computation. Most importantly, worst-case running and space complexity is considered rather than the average and best case complexities. The developers typically solve the worst-case scenario (Big O Notation) because expecting your algorithm to run on the best or average case scenarios may lead to inefficiency or total program failure.

4.3.1. Computational Complexity of Logistic regression

Logistic Regression[14] is used for binary classification in linearly separable data. The computational complexity of the logistic regression to train the model is focused on finding the w and b values that best separates the classes.

During the training phase, the model tries to calculate the values of w and b using the gradient descent to maximize the sum given below:

$$\mathbf{argmax_w} \sum_{i=0}^{n} \boldsymbol{yi}(w^t xi) + \mathbf{b}$$

In the $\mathbf{W^tX_i}$ operation where W is a vector of size d, the operations/Matrix multiplications/ time complexity is O(d), and then iterating over n data points takes n steps. Finally, the overall training time complexity of logistic regression is **O(n*d)**.

The space complexity of the linear regression, in the training phase of logistic regression we need to store these four variables in memory:

- X - X is a matrix of size n*d, needs O(n*d) steps to store
- Y - Y have size of n*1 is O(n)
- W - W is a vector of size of d, is O(d)
- B -b is a constant , which is an O(1)

Therefore, the overall Space complexity of this model during training is $\boldsymbol{O(nd + n + d + 1) \approx O(nd).}$

Finally, after training the logistic regression model, as W is a vector size d then the operation of $\mathbf{W^tX_i}$ takes O(d) steps. The runtime time complexity of the logistic regression model is O(d). For the runtime space

---

[13] Time and Space Complexity of Linear Regression: https://levelup.gitconnected.com/train-test-complexity-and-space-complexity-of-linear-regression-26b604dcdfa3 (accessed on 21 January 2021)

[14] Train/Test Complexity and Space Complexity of Logistic Regression: https://levelup.gitconnected.com/train-test-complexity-and-space-complexity-of-logistic-regression-2cb3de762054 (accessed on 21 January 2021)

complexity of the logistic regression, the model needs to keep in memory is **W** and **b**, and the space complexity during run-time is in the order of d –O(d +1)≈ $O(d)$.

4.3.2. Computational Complexity of Decision Tree

The Goal[8] of a decision tree induction algorithm is to find an optimal decision tree by minimizing the generalization error. The tree induction algorithms like CART and C4.5 have two main phases: tree growing and tree pruning. For each of the splits in the tree, testing every feature is required for all values to determine the value split that minimizes the information loss function. Generally, In decision tree induction[9], suppose the training data contains n instances and m attributes and the tree is balanced and its depth is $\log(n)$. So the rate of growth of the tree to the n leaves is O(log(n)). The overall time complexity of building the tree during the training phase is $\approx O(n * m * log(n))$. Additionally, the runtime complexity of the decision tree is O(depth) $\approx O(log(n))$.

The space complexity[15] of the decision tree during the training phase are the number of nodes in the tree. If we assume that the tree is balanced and binary the number of nodes are $2^{d+1} - 1$ and d is log(n). So the space complexity of the training phase of the decision tree is $O(2^{logn+1} - 1)$. Besides, the runtime space complexity of the decision tree during the testing phase is the same as the training phase space requirement, that is $O(2^{logn+1} - 1)$.

4.3.3 Computational Complexity of K-nearest Neighbor

The k-nearest neighbor[9] is an instance-based classification each new instance is compared with the existing ones using a distance metric and the closest existing instance is used to assign the class to the new one, based on the majority class of the k closest neighbors. The KNN uses a lazy learning method in which defers the real work as long as possible.

Storing the n data points is mandatory to calculate the distance from the new instance to all data points. Thus, the training phase computational complexity of the K-nearest neighbor[16] are O(n*d) time and space complexity where there are n data points that we are going to calculate the distance and d number of features of each sample(point).

During the testing phase, the number of nearest neighbors k is required and the time complexity of the model is O(n*k*d). According[10] the space complexity during rune time is O(k) which only cares about the k nearest data points.

4.3.4. Computational Complexity of Artificial Neural Networks (MLP)[17]

During the training phase of the artificial neural network, there are three steps to go through the Feed-forward propagation step, back-propagation, and updating weights of the networks. The feed-forward is where the connections don't form a cycle.

In feedforward propagation, to compute the input for a single unit *j* in layer *L1* for a single example x, without using matrices, we would carry out the following operation:

---

[15]Almost Everything You Need To Know About Decision Trees: https://towardsdatascience.com/almost-everything-you-need-to-know-about-decision-trees-with-code-dc026172a284 (accessed on 18 January 2021)

[16]k-nearest-neighbors: https://medium.com/analytics-vidhya/knn-k-nearest-neighbors-1add1b5d6eb2 (accessed on 18 January 2021)

[17]Artificial Neural Networks(MLP): http://www.briandolhansky.com/blog/2014/10/30/artificial-neural-networks-matrix-form-part-5 (accessed on 21 Jan 2021)

$$s_j^{(1)} = \sum_i x_i w_{i \to j}^{(in \to 1)}$$

In the Feed-forward propagation step, the input X propagates to the first layer with $S^{(1)}=XW^{(in\rightarrow 1)}$, and for each of the hidden layers compute $Z^{(i)}=f_i(S^{(i)})$ and $S^{(j)}=Z^{(i)}W^{(i\rightarrow j)}$.

Then store the activation derivatives for the backpropagation step, $F^{(i)}=(f'_i(S^{(i)}))^T$ and $Z^{(i)}$.At the output layer, we have $p(y_i=y)=Z^{(out)}$. Comparable to the matrix multiplication the operations of the activation function and the number of attributes of each example can be ignored and we are assuming the high impact factors in this case.

If we have N hidden layers, this will run N-1 times. Let's assume that N hidden layers of the same number of nodes n, and weight matrix of r rows and c columns, t number of training examples, the sum and multiplication of weights and output of the previous layers is in a single hidden layer is O(r*c*t*n). The back-propagation and forward propagation steps have the same time complexity. Therefore, the overall time complexity of the feedforward propagation for the N hidden layers is O(r*c*t*n*N). Assuming the number of iterations(epoch) e, the overall time complexity of the neural network(feedforward and back propagation) is O(r*c*t*n*N*e).

The training space complexity of the neural network is O(r*c*t*n*N). This is due to the need of storing the weight of each layer, derivatives of the activation functions, and outputs of each node. Therefore the overall training space complexity of the multilayer perceptron (ANN) is O(r*c*t*n*N).

The runtime time complexity is the same as a single feedforward neural networks time complexity, that's O(r*c*t*n*N). The overall runtime space complexity is O(n).

4.3.5. Summary

The summarized computational complexity below in the table 4 below shows that artificial neural network consumes high computational resources due to the high dimensional matrix multiplications. A matrix multiplication creates high computational complexity, but it is easier to parallelize with a GPU.

Table 4 Time and Space Complexity

Models	**Time Complexity**		**Space Complexity**		**Remarks**
	Training	Runtime	Training	Runtime	
Logistic Regression	O(n*d)	O(d)	O(nd+n+d)	O(d)	d –vector size, n –number of samples
Decision Tree	O(n*m*log(n))	$O(log(n))$	$O(2^{logn+1}-1)$	$O(2^{logn+1}-1)$	n-instances and m-attributes
KNN	O(n*d)	O(n*k*d)	O(n*d)	O(k)	k-nearest neighbors, n-number of points d-number of features
ANN	O(r*c*t*n*N*e)	O(r*c*t*n*N)	O(r*c*t*n*N)	O(n)	r-rows, c-columns, t-examples, n-nodes in hidden layer, N- Hidden layers ,e-epoch

4.4. Model Selection and the application Scenario

The main aim of this section is to select the final suitable machine learning model/strategy/ based on the detailed analysis of weaknesses and strengths of each model, evaluation results, and computational complexity. Based on the above detailed discussions and evaluations Artificial Neural Networks Machine learning models are the most effective model in detecting fraud transactions based on the evaluation results and is suitable for high-dimensional data. The KNN, Logistic regression and decision tree classifier have poor performance on handling the imbalanced data than the Artificial Neural Network. Even though the artificial neural network needs huge computational resources, it is effective in classifying and identifying credit card fraud transactions and parallel processing of matric multiplications of each node using GPU capability greatly improves its speed. As a result, Artificial Neural Network is selected and recommended as the best model in this scenario.

5.0 Applications of Selected Approaches

In this section, the selected machine learning model for Artificial Neural Network will be discussed in detail with real-world scenarios on the framework of the data science project life cycle (phases) on the credit card fraud detection.

5.1. Ideation

At the initial stage of the project, the financial institutions (data science team) are expected to model how the credit card transaction is processing on the client's side, payment, and deposit processing methods. Moreover, how the business processing transactions, how the business currently handles fraud transactions? Is it manual or not? For each of the components and internal functions of the business, the whole transaction processing and detailed analysis of the business needs are identified. If fraud detection is implemented in the traditional way, identifying the problem, institution should have to adapt and use the new technology and methods. To adopt the new technique and integrate it into their model, they should identify the current ICT infrastructure and business needs. After building a deep understanding of the existing business process and then identifying and isolating the business problem either to build a new machine learning system or adopt a new technique on detecting credit card fraud transactions.

Then map the existing process and system of the credit card systems of the organization and identify the exact place that this machine learning model is going to be implemented. In the credit card system, the transactions are accumulated in the main office of the ICT Infrastructures and the new system that is going to be adopted in the future should have to integrate with the existing infrastructures, credit card systems, and accumulated transactions storage Medias. The team of analysts on the credit card system is expected to convince the managers and stakeholders how much it is important if the machine learning credit card fraud transaction adopted and used, even consider creating synthetic data to show them how the future system works, provide prototypical deliverables, project costs and profits should have been addressed clearly.

5.2. Data Acquisition and Exploration

The team should have to identify the sources of data, from where to collect credit card transaction data with the help of the credit card issuer businesses, banks, and financial institutions. After credit card transaction data is collected, exploratory data analysis is conducted and is prepared for both the present project modeling and as re-usable constituents for future developments. At this stage detailed preprocessing of the collected data will be conducted and the main responsibility of the

data science project team is to organize the data suitable for modeling. Close cooperation with the IT team of the credit card issuer system is required in every step of the project modeling and implementation.

5.3. Research and Model development

In this section, the fraud detection data science project is going to be developed. Before diving into the development of a full-fledged fraud detection system the team is expected to develop and build simple models instead of using the whole data features. The team of developers should have to have a regular time frame to publish incremental models and parts of the systems and communicate with the stakeholders.

For example, if some change is needed in the data like dimension reduction, scaling and other changes, in their regular meetings with the stakeholders they should have to communicate with their works and small deliverables that why such changes are required. If the stakeholders don't know what is happening, maybe the project is going to fail and will not fulfill the credit card systems requirements and business needs.

The data science team should have to ensure that the relevant key performance indicator (KPI) of credit card issuer (Banks and Financial Institutions) are ensured and aligned. Moreover, users should have allowed experimenting with the new deliverables of the fraud detection system with cloud resources by establishing standard hardware and software configurations and balance the gap and business needs of the institution. Finally, receive feedback from the organization and tune the project accordingly. The suitable ANN machine learning model is going to be developed based on the business requirements and the data collected from the financial institution.

5.4. Evaluation and Validation

Rigorous evaluation of data rules and conventions, codebase, model performance, and machine learning model outputs assures that we can steadily increase business performance with the new credit card fraud detection system. The quality validation team is expected on dividing the fraud detection model and testing assumptions, business rules, and understandings from the initial sampling to the hyper-parameters and front-end implementation. Automated validation checks can also be used to support human review.

For example, the team of developers can develop an automated validation checkup system that will validate the system by generating synthetic fraud and legitimate data or impose real data to the system if the system effectively detects fraud and legitimate credit card transactions and produce validation results and statistical results. If there are some technical problems on some parts of the fraud detection system, the validation team and developers will test and patch.

It may be necessary to have a debate and discussion on the system with multiple stakeholders to receive feedback and increase the confidence of the stakeholders in your system.

During the validation and evaluation of your system, since the fraud and legitimate transactions are imbalanced and the performance measures like accuracy are not good at detecting on the positive classes so selecting proper evaluation methods like F1-Score, recall and balanced classification rate are good for such biased data is critical. The organization should have to set up its infrastructure and process real world credit card transactions to capture how the ANN Model works and behaves and receive comments and discuss with the stakeholders/IT Team/Experts.

5.4. Deployment

After validating the ANN fraud detection model, the product should have to be delivered and integrated into the credit card transaction and processing systems. Finally, to deploy the system in the real environment, the team should have to develop a deployment plan, develop monitoring and maintenance documents to avoid issues during the operational phase of the ANN Fraud detection system. The final ANN fraud detection system documentation is expected to be developed by the team. Review the machine learning model what is happening and what improvements are expected to do in the future, and give training to the experts of the credit card transaction operators and users.

6.0 Guidelines for deployment

As we mentioned earlier in this document, this case study mainly focuses on the comparative analysis of machine learning models on detecting credit card fraud transactions. Based on the comparative analysis, the Artificial Neural Network is the most important and recommended model by achieving high performance on detecting fraud credit card transactions. If financial institutions or individuals want to develop and deploy an Artificial Neural Network based-credit card fraud detection system the following deployment considerations should have to think through:

1. Identify the business needs and conduct detailed analysis from the customer/stakeholder that you are going to develop and deploy.
2. Due to the sensitivity of the business data especially credit card transactions, the organization/banks/financial institution may not afford you the required data and requirements. So, you as a development team should have to prepare documents and agreements not to disclose and give data to third-party.
3. On the data acquisition and exploration communicate with stakeholders regularly about the changes that you are going to do in the data, missing values, manipulations, and scaling, etc.
4. Show deliverables and give privileges to personnel to the stakeholder's side to experiment on the ongoing project.
5. Provide the hardware and software requirements and specification of the system, since such machine learning models consume a huge amount of computational resources, like Powerful Servers, CPU, GPU, RAM, etc.
6. Before deploying and delivering your project conduct thorough validation checks, and make performance measurements using the Mathewos Correlation coefficient, balanced classification rate, F1 score, and recall.
7. Give training to the experts and users of the credit card system/bank employees.
8. Don't use Accuracy measurement on such imbalanced data sets.
9. Use optimization techniques to create an optimal Artificial Neural Network.
10. Test, deploy and monitor the machine learning model.
11. Use the machine learning models as a way of triggering to the humans in some new fraud and unusual fraud transactions instead of mitigating and making decisions. It can be a legitimate transaction that is new to the model.
12. During the development of the model, you should have to consider and plan for future changes in transactions and technology infrastructures, and others.
13. Prepare and give required system documentations, Maintenance and monitoring documents

Therefore, bearing in mind the above and other deployment considerations, the artificial neural network is the power of current and future systems. The neural network model deployment in your system will create a huge impact on your profit and on detecting fraud transactions. Artificial Neural Network consumes a lot of memory but due to the capabilities of GPU parallel processing of matrix multiplication of the perceptron's speeds up the network. Artificial Neural Network is recommended for big financial institutions that can afford high and expensive computational resources.

7.0. Conclusions

This is a comparative analysis on fraud transaction detection on some of the selected machine learning models of logistic regression, Decision tree classifier, K-Nearest Neighbors, and Artificial Neural Networks based on credit card transactions collected in 2013. Based on the detailed analysis and observation results of these models Artificial Neural Networks is performing excellently on both the balanced and imbalanced data sets. The others perform fine on balanced data sets but have poor performance on the imbalanced data sets.

As we have observed from the implementation and comparative analysis, Artificial Neural Network consumes a lot of memory but due to the capabilities of GPU parallel matrix multiplication of the perceptron’s speed up the network. Artificial Neural Network is not suitable for low resourced devices (RAM and CPU) whereas the others don't consume a huge memory and time but their performance is poor. So Artificial Neural Network is recommended for big financial institutions that want to afford high and expensive resources.

Special focus is required for machine-learning-based security systems to create a secure transaction of credit card systems, especially Artificial Neural Network models that are compact and scaled to small devices much research and devotion is required. To further enhance the efficiency of the model, there are many ways to investigate the input variables, execute certain techniques of preprocessing, feature engineering, and optimization techniques.

References

[1] Domino, “The Practical Guide to Managing Data Science at Scale,” *Domino*, pp. 1–25, 2017.

[2] J. C. W. siddhartha Bhattacharyya, Sanjeev Jha, Kurian Tharakunnel, “Data mining for credit card fraud: A comparative study.” pp. 602–613, 2011.

[3] Y. Jain, N. Tiwari, S. Dubey, and S. Jain, “A comparative analysis of various credit card fraud detection techniques,” *Int. J. Recent Technol. Eng.*, vol. 7, no. 5, pp. 402–407, 2019.

[4] A. Botchkarev, “Performance Metrics (Error Measures) in Machine Learning Regression, Forecasting and Prognostics: Properties and Typology.”

[5] B. Juba and H. S. Le, “Precision-Recall versus accuracy and the role of large data sets,” in *33rd AAAI Conference on Artificial Intelligence, AAAI 2019, 31st Innovative Applications of Artificial Intelligence Conference, IAAI 2019 and the 9th AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019*, Jul. 2019, vol. 33, no. 01, pp. 4039–4048, doi: 10.1609/aaai.v33i01.33014039.

[6] D. M. W. Powers and Ailab, “EVALUATION: FROM PRECISION, RECALL AND F-MEASURE TO ROC, INFORMEDNESS, MARKEDNESS & CORRELATION.”

[7] V. N. Dornadula and S. Geetha, “Credit Card Fraud Detection using Machine Learning Algorithms,” *Procedia Comput. Sci.*, vol. 165, pp. 631–641, 2019, doi: 10.1016/j.procs.2020.01.057.

[8] H. M. ; Sani, C. ; Lei, and D. Neagu, “Computational complexity analysis of decision tree algorithms Item Type Conference paper,” 2018, doi: 10.1007/978-3-030-04191-5_17.

[9] I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal, *Data Mining: Practical Machine Learning Tools and Techniques*. 2016.

[10] S. Raschka, “Nearest Neigbor Methods,” 2018. [Online]. Available: https://sebastianraschka.com/pdf/lecture-notes/stat479fs18/02_knn_notes.pdf.