Machine Learning Algorithms For Fraud Detection

Chentduren Aumkaran z3375310

Table of Contents

1. Introduction

Fraud detection is the problem of detecting a transaction that does not belong in a series of transactions. Fraud is considered as criminal activity. This problem is very relevant in our ever-increasing digital modern word and is getting more complex as fraudsters utilise new ways and opportunities to outsmart existing fraud detection methodologies.

Proactive approaches to solving this problem include statistical methods, machine learning methods and deep learning methods. This report will focus on reviewing the classic machine learning methods, in particular demonstrating their effectiveness in detecting credit card fraud.

Credit card fraud can be either offline fraud or online fraud. Offline fraud is when the physical card itself is stolen and then used by the fraudster to make purchases until the usage of the card is cancelled. Online fraud is when the physical card is not needed as the data associated with the card has been compromised, such as the card number or other information that would be available to a merchant during a legitimate transaction.

The major impact of credit card fraud is financial loss. Other impacts include emotional distress, damage to reputation and lost opportunities. Even other types of fraud such as identity fraud have equally negative impacts. ABS reports that 11% of Australians aged 15 years and over (2.1 million) experienced personal fraud in 2020-21, up from 8.5% in 2014-15. Of this 11%, an estimated 6.9% (1.4 million) experienced credit card fraud in 2020-21, up from 5.9% in 2014-15 [3]. On a related note, a study from the Australian Payments Network showed that credit card fraud cost $495 million in 2021, which is an increase of 5.7% from the previous year [4].

Therefore, using algorithms and mathematical approaches, such as machine learning are extremely important to minimising the impacts of fraud and catching fraudulent transactions as soon as they are made, instead of waiting for humans to report them, which by the time might be too late to do anything. Hence, businesses and corporations incorporating good fraud catching capabilities in their systems is vital.

2. Challenges of Fraud Detection

Machine learning algorithms play an important part in many of the Fraud Detection systems of organisations, but the development of these algorithms have been challenging and difficult due to the massively unbalanced nature of fraud data, lack of confidential transaction data for research and the constantly evolving data as a result of fraudsters finding new ways to commit fraud.

Credit card companies receive and complete a very large number of transactions every day, and the majority of these transactions are simply normal transactions. Fraudulent transactions only make up a very small percentage, normally less than 1% of the total data. This small proportion of fraudulent transactions is the same for other types of fraud as well. Hence in general, fraud data is massively unbalanced. An unbalanced dataset makes it difficult to train a machine learning model that can correctly identify fraud due to unequal misclassification costs. This results in models classifying new transactions as non-fraud since classifying everything as the majority class is an easy way to minimise the error rate. One way to get around this is to weigh the loss so that it is more expensive to misclassify a fraudulent transaction as a normal transaction. Furthermore, it can also be that fraudulent transactions look very similar to normal transactions in the data.

It is also a hindrance to the development of machine learning algorithms when there is either no data or the data available is of low quality. It only makes sense that fraud occurs in transactional data that is confidential. Naturally, organisations and institutions will not give away this data for free due to it having the private information of their customers or clients. If the data is available, it might be that the transactions are not labelled. In this case, transactions must be labelled with the help of subject matter experts or the use of unsupervised machine learning algorithms. The subject matter experts would also have to verify the results of such models. Hence, this could take considerable time and resources to accomplish. Also, it can be possible that the features in the fraud data are just too general or of poor quality to build any good fraud detector models. In this case, better predictive features are needed.

Furthermore, if a machine learning model is successfully trained on a fraud dataset and then deployed into an organisation's fraud detection system, there must also be a model monitoring capability included in the system. A good model monitoring solution will ensure the fraud detection model is maintaining an acceptable level of performance. This is very important in the case of fraud because new transactions are always coming in and fraudsters are constantly coming up with new ways to commit online fraud, so the machine learning algorithms need to be regularly retrained and updated in a feedback loop. The quality of an organisation's model monitoring system is dependent on their engineering capability.

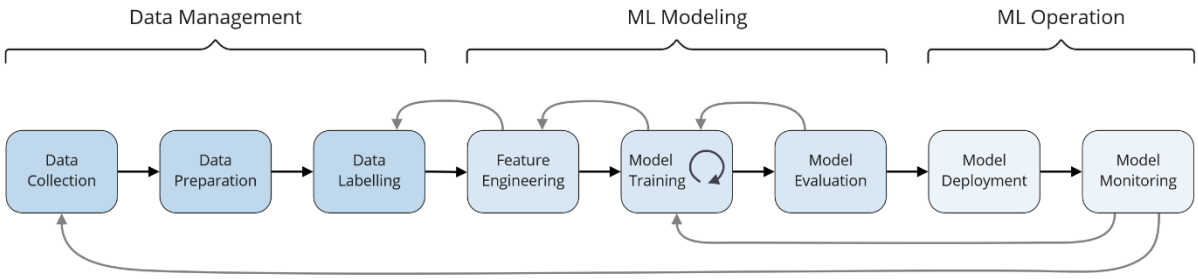

Figure 1: Machine learning development workflow (Fabien Ritz, 2022)

3. Summary of Credit Card Transactions Data

The credit card transactions dataset used in this report is provided by Kaggle [5]. The dataset contains transactions made by credit cards in September 2013 by European cardholders. As expected, the dataset is highly imbalanced, with it having 492 frauds out of 284,807 transactions (0.172%).

Due to customer confidentiality, the majority of features in the dataset are numerical principal components obtained with PCA (V1, V2, … V28). The only 2 features that have not been transformed are 'Time' and 'Amount'. 'Time' containing the seconds elapsed between each transaction and the first transaction in the dataset, and 'Amount' being the transaction amount. The 'Class' variable is 1 in the case of fraud and 0 otherwise.

Machine learning methods will use the input features to determine a decision boundary which will separate the fraud and non-fraud transactions.

```
RangeIndex: 284807 entries, 0 to 284806
Data columns (total 31 columns):
 #   Column  Non-Null Count   Dtype  
---  ------  --------------   -----  
 0   Time    284807 non-null  float64
 1   V1      284807 non-null  float64
 2   V2      284807 non-null  float64
 3   V3      284807 non-null  float64
 4   V4      284807 non-null  float64
 5   V5      284807 non-null  float64
 6   V6      284807 non-null  float64
 7   V7      284807 non-null  float64
 8   V8      284807 non-null  float64
 9   V9      284807 non-null  float64
 10  V10     284807 non-null  float64
 11  V11     284807 non-null  float64
 12  V12     284807 non-null  float64
 13  V13     284807 non-null  float64
 14  V14     284807 non-null  float64
 15  V15     284807 non-null  float64
 16  V16     284807 non-null  float64
 17  V17     284807 non-null  float64
 18  V18     284807 non-null  float64
 19  V19     284807 non-null  float64
 20  V20     284807 non-null  float64
 21  V21     284807 non-null  float64
 22  V22     284807 non-null  float64
 23  V23     284807 non-null  float64
 24  V24     284807 non-null  float64
 25  V25     284807 non-null  float64
 26  V26     284807 non-null  float64
 27  V27     284807 non-null  float64
 28  V28     284807 non-null  float64
 29  Amount  284807 non-null  float64
 30  Class   284807 non-null  int64
```

Figure 2: Table schema for credit card transactions dataset

4. Training and Test Data Split

To evaluate the performance of the machine learning models, we will use stratified sampling to split the data into 2 parts – 70% for training and 30% for testing purposes. Stratified sampling allows the data to be split so that each part has the same proportion of fraud.

Split	Fraud	Non-Fraud	Total
Train	344	199020	199364
Test	148	85295	85443
Total	492	284315	284807

5. Exploring the Data and Feature Selection

Before running machine learning algorithms on the training data, we will do some exploratory analysis to make better sense of the data, as well as remove any features that are redundant.

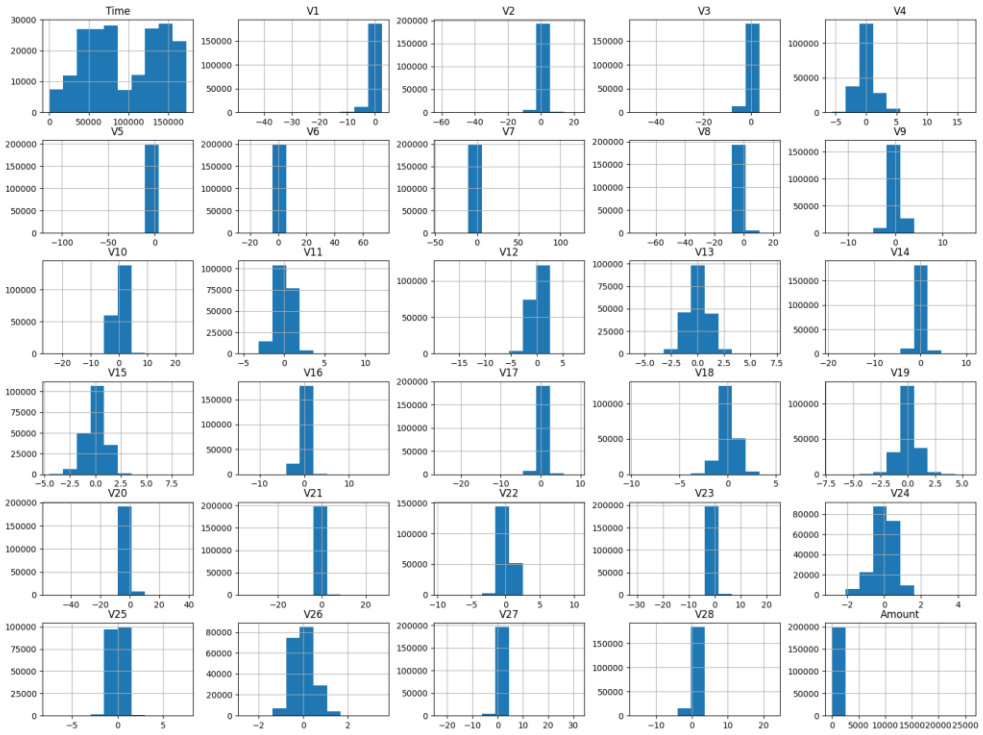

Figure 3: Histograms of the training dataset features

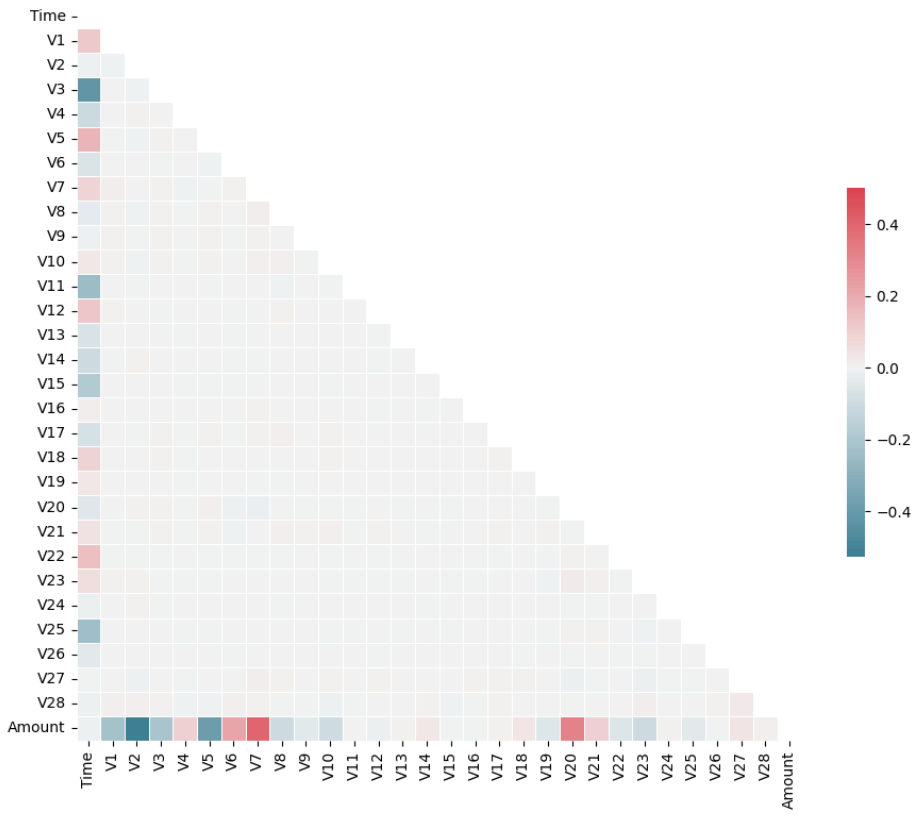

Figure 4: Correlation matrix of training dataset

Since we are not reviewing any time series focused machine learning algorithms in this paper, we can drop the 'Time' column. From the histograms in Figure 2, we need to scale the 'Amount' column so that it is in the same scale as the other features. In addition, we can confirm from the correlation matrix in Figure 3 that there is no significant correlation between the possible pairs of features. Hence, no other features will be dropped. Also, there are no missing values in this dataset, so no need for imputing missing values.

6. Literature Review

Fortunately, in the credit card transactions dataset we have the labels, so we will primarily be reviewing supervised machine learning methods in this report. The objective is to correctly classify the transactions as legitimate or fraudulent. When determining the best machine learning model to use in a fraud detection system, it is important to consider a number of factors such as:

- Performance – Obtains the best score in the relevant evaluation metrics.
- Inference Time – Transactions will come in quickly so the model has to quickly determine which are fraud fast. Depending on the nature of the business or organisation, they might even need real time inference.
- Explainability – Being able to easily Interpret the model and its results will also help choose the final model, as well as help build confidence with stakeholders.

- Training time – The quicker the model can be trained, the faster it can be updated and save costs.

6.1 Random Forest

The Random Forest is an ensembled learning method in which a multitude of decision trees are constructed to obtain a classification. Compared to single decision trees which tend to overfit, the Random Forest is able to get around this problem due to bagging, which reduces the variance, while not increasing the error due to bias. Bagging is when each of the N decision trees are trained on a different sample (with replacement) of the original training set X. Hence, when the Random Forest has completed training and an unseen sample is given, the prediction is made by taking the majority vote of all the trees.

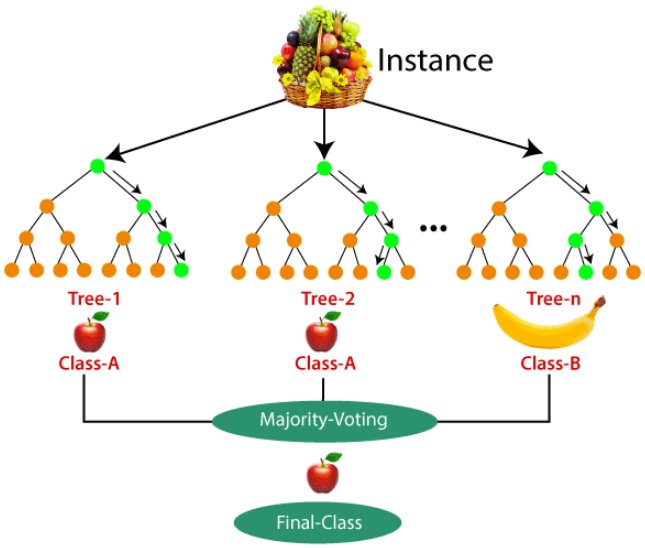

Figure 5: Random Forest majority voting (Sruthi E R, 2021) [6]

Eesha Goel, Abhilasha and Ankit Agarwal (2016) [7] found that for fraud detection, Random Forest had excellent accuracy compared to other algorithms and ran efficiently on large datasets. It has the benefit of being really interpretable as well as it estimates which features are important in the classification.

In addition, Deepashree Devi, Saroj.K Biswas and Biswajit Purkayastha (2019) [14] determined that Random Forests were effective with handling unbalanced datasets if cost-sensitivity was factored into it, meaning that it made it more expensive to misclassify a fraudulent transaction as a normal transaction. In particular, they integrated a misclassification ratio-based cost function into the error-formulation of the generated sub-trees in Random Forest bagging. Hence, the sub-tree with the highest predictive probability has maximum weightage.

6.2 K-nearest neighbours

K-nearest neighbours (KNN) is a classification algorithm where the prediction of a new data point depends on the k data points that are closest in proximity to it. As a result, the prediction is also made by majority voting where the classification of the new data point is dependent on the majority class of the k data points closest to it. This machine learning algorithm is an example of instance-based learning since there is no training, as the training happens when a new prediction is requested. Usually, the distance metric used to calculate the distances between points is the Euclidean Distance. Other distance metrics can also be used, such as the Manhattan distance.

$$Euclidean: d(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

$$Manhattan: d(x,y) = \sum_{i=1}^{n}|x_i - y_i|$$

Where x_i, y_i are the Euclidean vectors starting from the origin of the space and n is the n-space

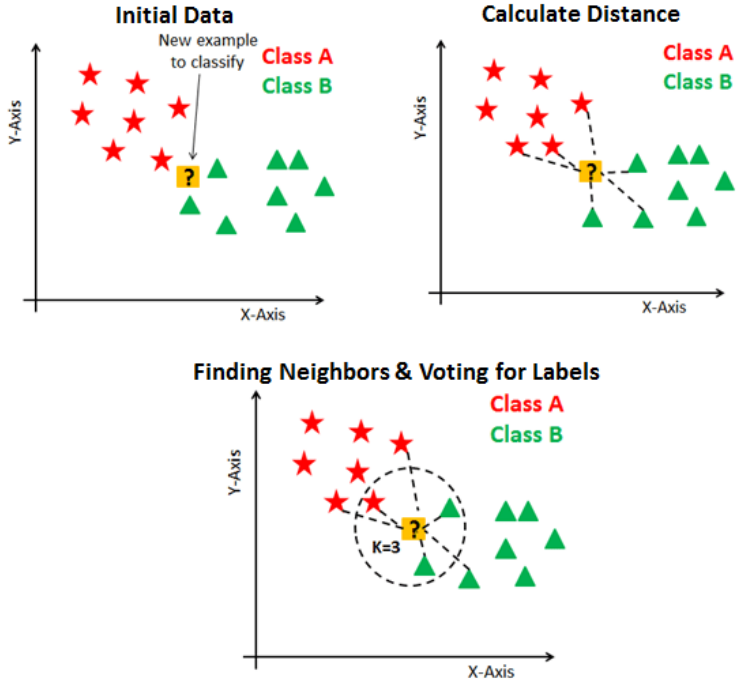

Figure 6: KNN majority voting based on distances and value of K (Avinash Navlani, 2018) [13]

C. Sudha and T.Nirmal Raj (2017) [10] found the KNN to achieve consistently high performance among various fraud detection methods, especially given that it does not take any consideration any priori assumptions about the distributions from which the training examples are drawn. A small and odd numbered K (typically 1,3 or 5) so that it can break any ties, was found to be sufficient to predict fraudulent transactions.

Researchers have also proposed modifications to the traditional KNN algorithm to better help detect fraud. N. Malini and M.Pushpa (2016) [16] proposed implementing Hidden Markov Model with KNN to assist with fraud prediction and prevention. By using KNN to detect the fraudulent transactions and the Hidden Markov Model to determine behaviour pattern and check upcoming transactions, this proposed model could minimise the number of false alarm transactions. Sara Makki, Rafiqul Haque, Yehia Taher, Zainab Assaghir, Mohand-Said Hacid and Hassan Zeineddine [15] proposed a cost-sensitive KNN approach to tackle the imbalance problem by using cosine similarity instead of Euclidean distance to find the neighbours and then calculate a certain score to evaluate the probability of fraud risk.

6.3 Naïve Bayes

Naïve Bayes is a probabilistic machine learning algorithm based on Bayes' theorem which uses the known target classes of the training dataset to determine the probability of a new data point belonging to a class given the effect of attribute values on the classes themselves.

$$Bayes\ Theorem\colon\ P(A \mid B) = \frac{P(B \mid A)\ P(A)}{P(B)}$$

Where A, B are events of interest

This algorithm assumes class-conditional independence, which means the effects of features on a given class is independent to the effect of other features on the same class. Hence, building from Bayes Theorem:

$$P(C_k \mid x) = \frac{P(C_k)\ P(x \mid C_k)}{P(x)}$$

Where $P(C_k \mid x)$ is the posterior probability of a class given predictor (x, attributes), $P(C_k)$ is the prior probability of a class, $P(x \mid C_k)$ is the likelihood which is the probability of predictor given a class and $P(x)$ is the prior probability of predictor. Therefore, assuming the class-conditional independence and ignoring the denominator as it does not impact the final outcome of the classifier, for n features, the probability of a class would be:

$$P(C_k \mid X) = P(x_1 \mid C_k)\ \times P(x_2 \mid C_k)\ \times \ldots \times P(x_n \mid C_k)\ \times P(C_k)$$

Comparing Naïve Bayes with other machine learning and even deep learning methods, Carolyne Milgo (2016)[17] stated that it was easier to interpret due some of the other methods taking a black box approach. It is also superior in its speed of learning. Additional advantages of the Naïve Bayes classifier are that it only requires a small amount of training data to estimate the parameters (means and variances of the features) necessary for classification and is considered robust to both noisy data and missing data. Hence, in the case of fraud detection, the application of the Bayes' theorem is as follows:

$$P(Fraud \mid Evidences) = \frac{P(Evidences \mid Fraud)\ P(Fraud)}{P(Evidences)}$$

I O Eweoya, A A Adebiyi, A A Azeta, F Chidozie, F O Agono and B Guembe (2019) [18] found Naïve Bayes to be an efficient machine learning approach to detect fraud perpetration in banks. It is simple, elegant, robust and flexible. Advantages include training taking a short computation time, model is easily constructed and its iteration parameter estimation is less complicated. If employed in in financial institutions for loan scrutiny, it will save economic loss, reduce human errors and eliminate unnecessary bureaucracies in loan administration.

6.4 Support Vector Machines

Support Vector Machines (SVMs) is classification algorithm which works by finding a hyperplane in N-dimensional space which has the maximum margin between data points of both classes. The reason for maximising the margin is so that unseen data could be classified with more confidence. In the training phase of this algorithm, the data points that are nearest to the maximum margins of the hyperplane are called the support vectors. There is at least one support vector for each class.

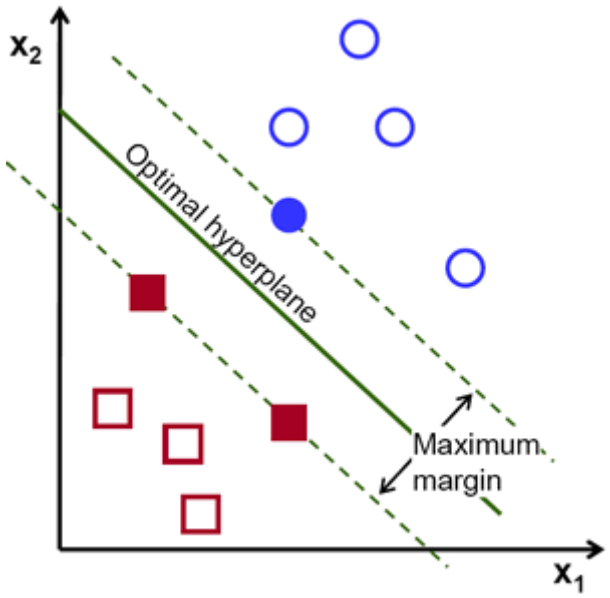

Figure 7: Optimal hyperplane with the maximum margin (Rohith Gandhi, 2018)[19]

Equation of optimal hyperplane: $y = m.x + b = 0$

Equation for margin boundary 1: $y = m.x + b = +1$

Equation for margin boundary 2: $y = m.x + b = -1$

And the margin is $2/||w||$ where $||w||$ is the norm of the vector w

V.Dheepa and R.Dhanapal (2012) [20] found SVM to give effective performance in fraud detection and to be scalable for handling large values of transactions. They also claim that it successfully solves classification problems in noisy and complex domains, has excellent generalisation performance and that the problem of over fitting is less common.

Dongfang Zhang, Basu Bhandari, Dennis Black (2020) [23] found a weighted SVM to outperform other machine learning algorithms in detecting credit card fraud and maintained the most consistency. The number of transactions was used as the weight. They claimed that this model could minimise the financial loss of a bank.

6.5 Isolation Forest

Unlike the other machine learning algorithms mentioned in this literature review, this algorithm is an example of unsupervised learning. It is an implementation of the Random Forest algorithm but is used to detect anomalies instead of a target variable, and thus is used commonly in outlier detection. Specifically, its on the basis that anomalous data points would have shorter paths from the root node of a tree in the forest to the leaf node and thus would be isolated by fewer splits of the data. By averaging the path lengths for each observation, we can find observations that are more distinct within the data set. These average paths can be used to calculate and anomaly score between 0 and 1 for each observation.

$$Anomoly\ score = 2^{-\left(\frac{Average\ path\ length\ for\ observation}{Average\ length\ of\ all\ paths}\right)}$$

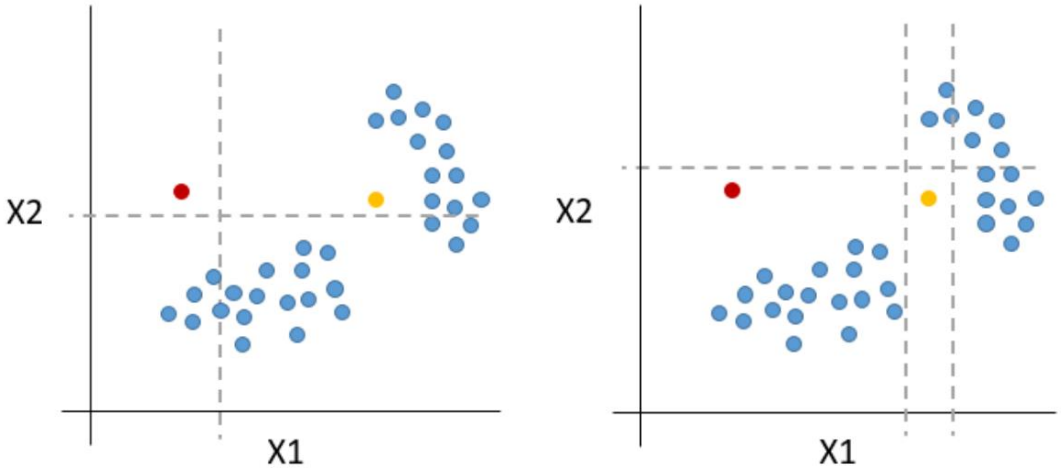

Figure 8: Red observation is more anomalous due to taking 2 divisions to isolate, while orange observations take 3 divisions (Ryan Gillespie, 2019) [8]

Ryan Gillespie (2019) [8] has proposed supplementing supervised methods with unsupervised methods like Isolation Forest. The reason for this being that Fraud patterns can evolve or change due to fraudsters utilising new ways to commit fraud and since unsupervised methods are not limited by the patterns in the historical data, they can find these new patterns as early as possible. Isolation forests were found to have good predictive ability and are easy to interpret.

V. Vijayakumar, Nallam Sri Divya, P. Sarojini and K. Sonika [9] compared Isolation Forest with another unsupervised algorithm called Local Outlier Factor and found Isolation Forest to be more precise in detecting fraudulent transactions. In addition, Isolation Forest had a small memory demand and low complexity in linear time.

6.6 SMOTE

Instead of detecting fraud, Synthetic Minority Oversampling Technique (SMOTE) is machine learning algorithm to augment the aforementioned imbalanced data sets that are inherent to fraud detection so that they become balanced. Compared to random oversampling which just duplicates instances of the minority class, SMOTE synthesises new instances of the minority from the existing data. Thus the use of SMOTE can potentially help the other algorithms mentioned in our literature review perform better. In fact, it uses K-nearest neighbours to synthesise new examples. It works by selecting a random sample from the minority class, then determining the vector between the data point and 1 of its k neighbours and then adding this vector multiplied by a random number between 0 and 1 to the data point. Hence, this effectively moves the data point slight in the directions of its neighbour, making it similar to the data point, but not an exact duplicate of it.

$$x_{i1} = x_i + C_1(x_{i(nn)} - x_i)$$

Where x_{i1} is a new synthetic observation, x_i is the random instance from the minority class, $x_{i(nn)}$ is one of the nearest neighbours and C_1 is the random number

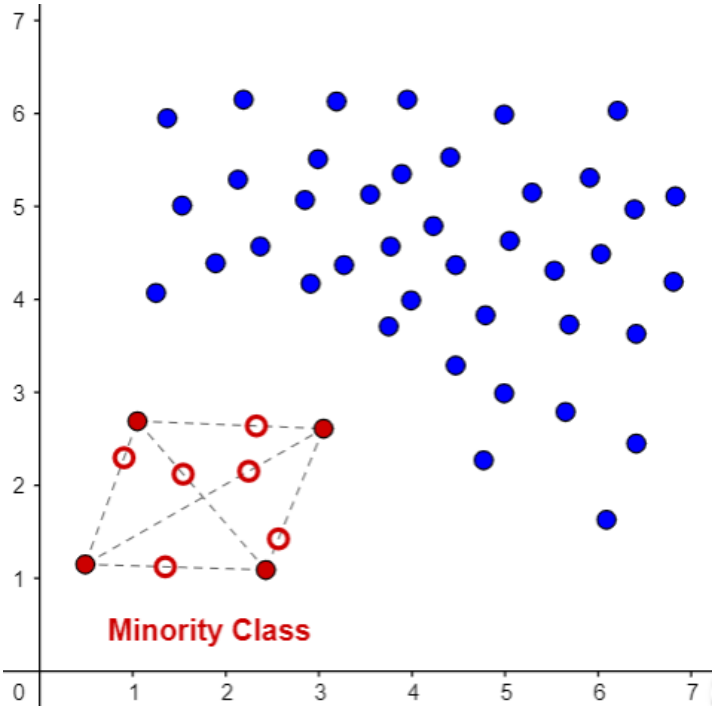

Figure 9: New synthetic instances created from nearest neighbours of the minority class (Aashish Nair, 2022) [21]

Sahayasakila.V, D. AishwaryaSikhakolli and Venkatavisalakshiseshsai Yasaswi (2019) [12] found that by including the SMOTE technique in their Fraud Detection system, it helped to improve the convergence speed and efficiency of detecting fraud compared to their previous system. Similarly, Cuizhu Meng, Li Zhou and Bisong Liu (2020) [22] also found SMOTE to help with training a more stable and generalised fraud detection model.

7. Results And Discussion

The machine learning models explored in the literature review were trained on the training split of the credit card transactions data and then evaluated on the test split with performance metrics like accuracy, recall, precision, f1 score and area under precision-recall curve (AUPRC).

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

$$Recall = \frac{TP}{TP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

Where TP = True Positives, TN = True Negatives, FP = False Positives and FN = False Negatives

Due to imbalanced data, accuracy is normally not a good metric for fraud detection. Instead, it is more important to look at precision, recall and f1 score. In particular, we are interested in the binary precision, recall and f1 score since the fraud detectors being able to correctly classify fraud transactions is most important.

	Accuracy	Precision	Recall	F1
Random Forest	0.999544	0.943089	0.783784	0.856089
KNN	0.999473	0.905512	0.777027	0.836364
SVM	0.999228	0.762821	0.804054	0.782895
Naive Bayes	0.977213	0.060576	0.837838	0.112984
Isolation Forest	0.962993	0.036308	0.797297	0.069453

Figure 10: Evaluations metrics

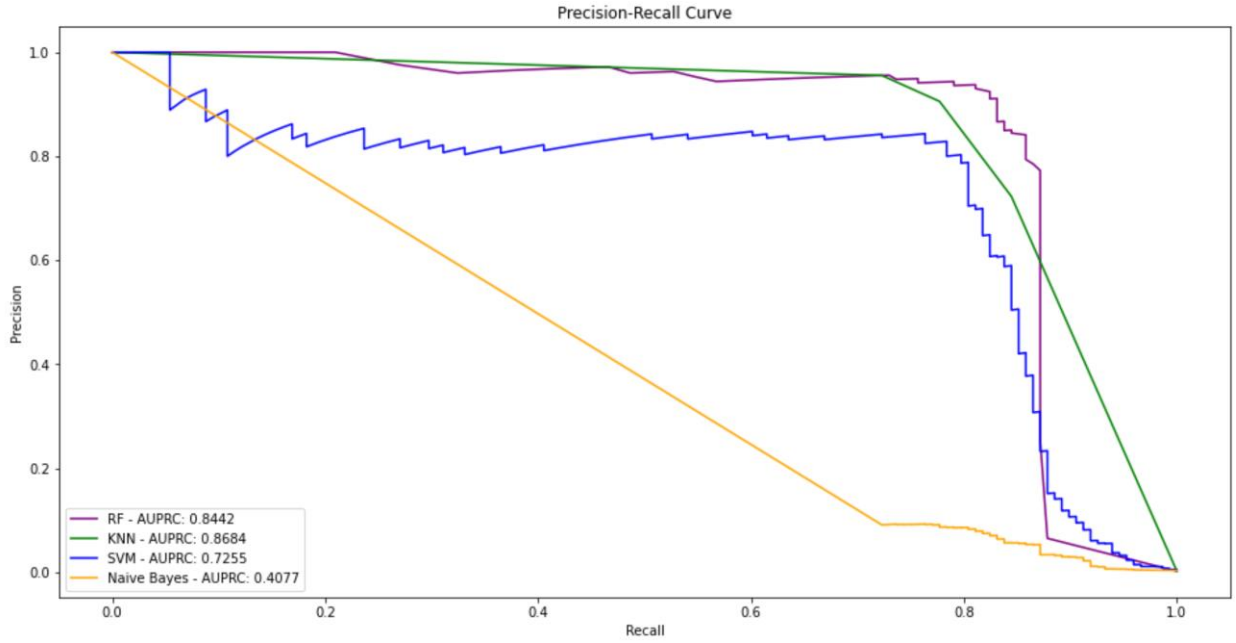

Figure 11: Precision-Recall Curve and AUPRC values

	Accuracy	Precision	Recall	F1
Random Forest	0.999508	0.878571	0.831081	0.854167
KNN	0.998525	0.548673	0.837838	0.663102
SVM	0.976405	0.063959	0.925676	0.119651
Naive Bayes	0.974427	0.055822	0.864865	0.104875

Figure 12: Evaluation metrics for models trained on SMOTE augmented dataset

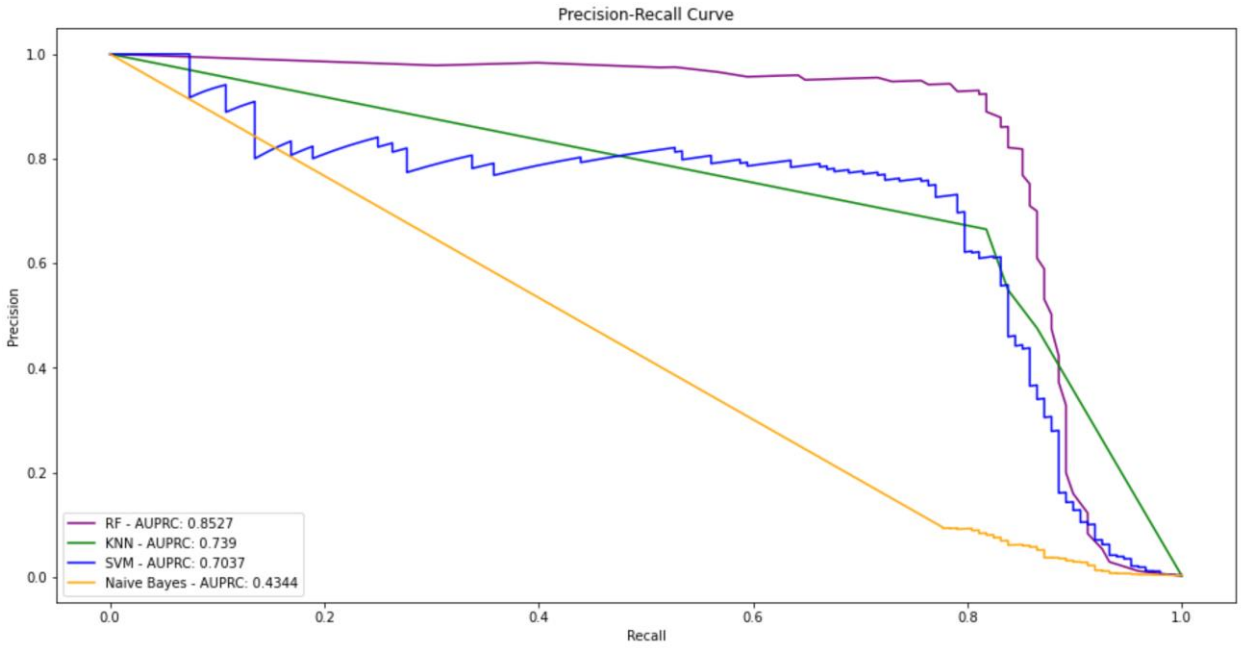

Figure 13: Precision-Recall Curve and AUPRC values for models trained on SMOTE augmented dataset

From Figures 10 and 12, the accuracy scores are all high, but as mentioned before, they do not matter much due to the high imbalance of the test split. All the models trained primarily use their default parameters and thus in this report, these models should be taken more as a benchmark, which could be further improved upon with hypertuning. Interestingly, the models trained on SMOTE augmented data performed overall worse at predicting fraudulent transactions, in particular the SVM whose f1 score decreased significantly from 0.78 to 0.11. It is possible that the synthetic data points that SMOTE created caused the models to overfit in this case.

Based on the f1 scores, Random Forest appears to be the best at correctly predicting fraud with a score of about 0.85. From the algorithms we reviewed, the fact that Random Forest is an ensemble supervising algorithm is probably why it is the best overall predictor. Also, in the training phase, the Random Forest weighs the loss so that it is more expensive to misclassify a fraudulent transaction as a normal transaction. This is then followed by the KNN and SVM models. While the performance of KNN was closest to Random Forest, it took a long time to train. In the case of SVM, the simple linear kernel was used due to its faster training speed, thus it is possible to get better results with another kernel. Naïve Bayes performed the worst out of the supervised machine learning algorithms. This is further shown in Figures 11 and 13, where the AUPRC is around 0.4, which is significantly less than the other 3 supervised models. This Isolation Forest unsupervised machine learning algorithm incorrectly classified a large number of transactions as fraudulent and thus had the worst f1 score. However, in accordance with the literature, this algorithm is reputed to find new patterns of fraud and so is still worth considering.

8. Conclusion

Fraud detection is essential in our modern world where digital transactions through banking and e-commerce are important parts of everyday consumer life. It is a field of study that is ripe with opportunities and where research is being doing to find new and better techniques to catch evolving fraud patterns. The machine learning algorithms that we reviewed and then tested show promise in detecting fraud, particularly credit card fraud. Out of the machine learning algorithms we reviewed,

Random Forest appears to be the best performing. Hence, it is recommended that Random Forest will be a good starting algorithm to build an initial fraud detector which can then be improved upon. The algorithms that we looked at are all very interpretable and this will be very useful in explaining to non-technical stakeholders why certain transactions could be fraudulent.

Due to the imbalanced nature of fraud detection, it is a very challenging field since it involves a trade-off between correctly detecting fraudulent transactions and not misclassifying many non-fraud transactions. This trade-off is a business decision that the company must make. Companies must also be careful that any fraud detectors are ethical and do not marginalise any specific group of people.

Apart from hypertuning the machine learning models that we looked at in this report, there are a number of ways to get more robust fraud detectors. Instead of having just one fraud detector to detect general fraud, we can have user or fraud specific models. This would depend on the capabilities of the fraud investigators and business knowledge. In fact, an ensemble of fraud detectors would most likely produce better results. Furthermore, we only explored machine learning methods in this paper, it would be worthwhile to explore Neural Networks as well, even though they are less interpretable and might make the fraud detection system complex.

Due to data privacy, there is still no standard, comprehensive or benchmark fraud or credit card dataset published publicly for comparative study. Hopefully, there will be one in the near future so that we can benchmark the performance of our fraud detector systems. In the meanwhile, the credit card dataset from Kaggle is quite good and is able to provide a good starting point.

9. References

[1] Alireza Hashemi: *Anomaly & Fraud Detection*, 2020, https://towardsdatascience.com/anomaly-fraud-detection-a-quick-overview-28641ec49ec1

[2] *Fraud Detection: A Complete Overview*, 2022, https://www.inscribe.ai/fraud-detection

[3] *Personal Fraud*, Australian Bureau of Statistics, 2022, https://www.abs.gov.au/statistics/people/crime-and-justice/personal-fraud/latest-release

[4] *Fraud Statistics*, Australian Payments Network, 2022, https://www.auspaynet.com.au/resources/fraud-statistics/2021-Calendar-year

[5] Credit Card Fraud Detection Dataset, https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud

[6] https://www.analyticsvidhya.com/blog/2021/06/understanding-random-forest/

[7] Eesha Goel, Abhilasha and Ankit Agarwal: *Fraud Detection Using Random Forest Algorithm*, 2016, http://www.ijcse.net/docs/IJCSE16-05-05-030.pdf

[8] Ryan Gillespie: *Detecting Fraud and Other Anomalies Using Isolation Forests*, 2019, https://www.sas.com/content/dam/SAS/support/en/sas-global-forum-proceedings/2019/3306-2019.pdf

[9] V. Vijayakumar, Nallam Sri Divya, P. Sarojini and K. Sonika: *Isolation Forest and Local Outlier Factor for Credit Card Fraud Detection System*, https://deliverypdf.ssrn.com/delivery.php?ID=934115017026001123106094096068097087004000290320260500761230270060921021021071171001210560100471060170070640310680690720270640610050330480470660080671131200981140720860150010050981190880270220010090950090721100241010951000860120980160701210840201240000&EXT=pdf&INDEX=TRUE

[10] C.Sudha, T.Nirmal Raj: Credit Card Fraud Detection in Internet Using K-Nearest Neighbour Algorithm, 2017, https://ipasj.org/IIJCS/Volume5Issue11/IIJCS-2017-11-06-6.pdf

[11] Fabian Ritz: *Capturing Dependencies within Machine Learning via a Formal Process Model*, 2022, https://deepai.org/publication/capturing-dependencies-within-machine-learning-via-a-formal-process-model

[12] Sahayasakila.V, D. AishwaryaSikhakolli and Venkatavisalakshiseshsai Yasaswi: *Credit Card Fraud Detection System Using Smote Technique and Whale Optimization Algorithm*, 2019, https://www.ijeat.org/wp-content/uploads/papers/v8i5/D6468048419.pdf

[13] Avinash Navlani: *KNN Classification Tutorial using Scikit-learn,* 2018, https://www.datacamp.com/tutorial/k-nearest-neighbor-classification-scikit-learn

[14] Deepashree Devi, Saroj.K Biswas and Biswajit Purkayastha: *A Cost-sensitive weighted Random Forest Technique for Credit Card Fraud Detection*, 2019, https://www.researchgate.net/publication/353906006_A_Cost-sensitive_weighted_Random_Forest_Technique_for_Credit_Card_Fraud_Detection

[15] Sara Makki, Rafiqul Haque, Yehia Taher, Zainab Assaghir, Mohand-Said Hacid and Hassan Zeineddine: *A Cost-Sensitive Cosine Similarity K-Nearest Neighbor for Credit Card Fraud Detection*, https://ceur-ws.org/Vol-2343/paper10.pdf

[16] N. Malini and M.Pushpa: Investigation of Credit Card Fraud Recognition Techniques based on KNN and HMM, 2016, https://research.ijcaonline.org/icccmit2017/number1/icccmit201707.pdf

[17] Carolyne Milgo: *A Bayesian Classification Model for Fraud Detection over ATM Platforms*, 2016, https://www.iosrjournals.org/iosr-jce/papers/Vol18-issue4/Version-4/F1804042632.pdf

[18] I O Eweoya, A A Adebiyi, A A Azeta, F Chidozie, F O Agono and B Guembe: *A Naive Bayes approach to fraud prediction in loan default*, 2019, https://iopscience.iop.org/article/10.1088/1742-6596/1299/1/012038/pdf

[19] Rohith Gandhi: *Support Vector Machine — Introduction to Machine Learning Algorithms*, 2018, https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47

[20] V.Dheepa and R.Dhanapal: *Behaviour Based Credit Card Fraud Detection Using Support Vector Machines*, 2012, https://ictactjournals.in/paper/IJSCV2_I4_P7_391_397.pdf

[21] Aashish Nair: *Create Artificial Data With SMOTE*, 2022, https://towardsdatascience.com/create-artificial-data-with-smote-2a31ee855904

[22] Cuizhu Meng, Li Zhou and Bisong Liu: *A Case Study in Credit Fraud Detection With SMOTE and XGBoost*, 2020, https://www.researchgate.net/publication/343720451_A_Case_Study_in_Credit_Fraud_Detection_With_SMOTE_and_XGBoost

[23] Dongfang Zhang, Basu Bhandari, Dennis Black: *Credit Card Fraud Detection Using Weighted Support Vector Machine,* 2020, https://www.scirp.org/pdf/am_2020121811281319.pdf