Final Project Report

BT4012: Fraud Analytics

Investigating 3 types of Fraud Detection Methods in Detecting Credit Card Fraud

Prepared By: Group 31

Loh Hong Tak Edmund (A0199943H)

Tan Yi Bing (A0204181U)

Yap Hui Yi (A0203707M)

Github Repository:

https://github.com/edologgerbird/repository31

Problem Background

The volume of electronic payments has been growing at unprecedented speeds as more companies and institutions adopt digital payments in their core processes (Surabhi, 2021). While digital payments have streamlined and increased the convenience of large volume transactions (Fiserv, 2021), there exists malicious actors that partake in fraudulent transactions for illicit gains (Tham, 2021). This is especially prevalent in the form of credit card fraud. In 2018, $24.26 Billion was lost due to credit card fraud worldwide, which was an 18.4 percent increase from the previous year (Shift Processing, 2021). To combat this, financial institutions have been adopting data-driven fraud analytics methods to detect and block potential credit card frauds (Reilley,2021). However, fraud analytics methods do possess limitations, which will be addressed in the reviews of our chosen papers.

Chosen Papers and Overviews

Paper 1: ***Sequence classification for credit-card fraud detection (Jurgovsky et. al., 2018)***

Summary: This paper phrased the fraud detection problem as a sequence classification task. Long Short-Term Memory networks are used to incorporate transaction sequences. State-of-the-art feature aggregation strategies were applied to the models and results were reported through traditional metrics such as Precision, Recall and Area Under the Curve. LSTM models are then compared to a baseline random forest classifier.

Paper 2: ***AI^2: Training a big data machine to defend (Veeramachaneni et. al., 2016)***

Summary: The AI^2 system combines four main components - a big data behavioural analytics platform, an ensemble of outlier detection methods, an analyst feedback mechanism and a supervised learning module, to detect fraudulent transactions. As compared to an unsupervised outlier detection method, AI^2 improves detection rate by 3.41 times and reduces false positives by 5 times.

Paper 3: ***FraudMiner: A Novel Credit Card Fraud Detection Model Based on Frequent Itemset Mining (Seeja & Zareapoor, 2014)***

Summary: FraudMiner first constructs a pattern database using Frequent Itemset Mining to form legal and fraud patterns of each customer. Frauds are then detected by a matching algorithm which traverses the pattern databases for a match with the incoming transaction. As compared to other models such as Naives Bayes model, SVM, K Nearest Neighbours, Fraud Miner performed better in terms of higher sensitivity, balanced classification rate, Matthew's coefficient and lower false alarm rates,

Project Objectives and Methodology

Project Objectives

In this project, our group aims to compare and identify the best performing fraud detection method out of the three methods as proposed by the three selected papers.

Methodology

1. We have implemented the respective proposed algorithms in simplified models.
2. We trained and tested the models on the Credit Card Fraud Detection dataset obtained from Kaggle.
    a. Dataset is a simulated credit card transaction dataset containing legitimate and fraud transactions from the duration 1st Jan 2019 - 31st Dec 2020.
    b. It covers credit cards of 1000 customers doing transactions with a pool of 800 merchants.
3. We compared the three different models using the metrics F2-Score, Precision and Recall, to determine which model can most effectively detect fraudulent transactions.
    a. F2-Score is used as it emphasises minimising False Negatives (Lee, 2020), which is costly in fraud detection.

$$F2\ score\ =\ \frac{(1+2^2)*Precision*Recall}{2^2*Precision+Recall}$$

4. We mined for insights through our comparisons and identified assumptions and potential limitations in our simplified implementations.

Paper 1: Sequence classification for credit-card fraud detection

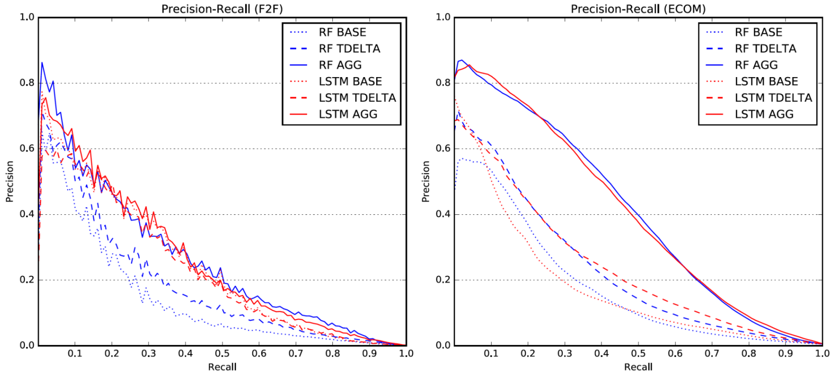
Problem Statement

Fraud detections remain a key concern for modern banking systems due to the direct incurred losses by fraudulent transactions and the difficulty in ensuring that legitimate customers will not be impacted by automated and manual reviews. Since fraudulent transactions represent only a very small fraction of all the daily transactions and are often changing due to new attack strategies, building a model for fraud detection remains a challenge.

Main Results

Firstly, the overall detection accuracy is much higher on e-commerce transactions than on face-to-face transactions. Secondly, longer input sequences does not affect detection accuracy for both face-to-face and e-commerce. Thirdly, there is a noticeable improvement for face-to-face fraud detection after taking into account previous transactions with an LSTM. However, this is not observed for e-commerce fraud detection.

Feature aggregations improve fraud detection and it is more evident for e-commerce transactions. Random forest models have a high peak in precision at low recall levels but decay quickly as the recall increases. LSTM models have a slightly lower precision for low recall levels but retain a higher precision as the recall increases. However, the Precision-Recall curve of the random forest aincreases by a noticeable margin up to a performance that is on par with the LSTM models after adding aggregated features (face-to-face transactions). On e-commerce transactions, the Precision Recall curves of random forest and LSTM are almost identical across all feature sets.



(a) F2F

(b) ECOM

Fig 1. Comparison of Precision-Recall graph between F2F and Ecommerce transactions

Analysis of the Results

We implemented the algorithm through a simplified model to compare its effectiveness in a small scale trial. We implemented RandomForestClassifier and LSTM model. LSTM is also implemented after using RandomForestClassifier to select the most predictive features.

The best performing model is LSTM with RandomForest Feature Selection. By pruning the less important features, we are able to remove noise which allows LSTM to better detect the sequences. Therefore, there is a significant improvement in results from a vanilla LSTM to LSTM with RandomForest.

Since pruning reduces the complexity of the final classifier, a pruned RandomForestClassifier is able to improve its results by the reduction of overfitting. By selecting the top 110 important features, pruned RandomForestClassifier is able to capture important structural information about the sample space. As such, pruned RandomForestClassifier is able to better detect false negatives, which improves both recall and F2 scores.

LSTM performs better in comparison to RandomForestClassifier. Since LSTM is a sequence learner and RandomForestClassifier is a static learning, LSTM is able to learn the past sequences of transactions which then can translate into better prediction of fraud detections.

Model	F2-Score	Precision	Recall
Random Forest	0.206767	1.000000	0.215686
Pruned RandomForest	0.500000	0.950000	0.558824
LSTM	0.626050	0.876471	0.730392
LSTM + RandomForest Feature Selection	0.653061	0.780488	0.784314

Fig 2. Model performance of LSTM, Random Forest, Random Forest after pruning and LSTM with Random Forest

Assumptions, Limitations & Potential Extensions

The paper assumed that consumers' behaviour is controlled by some latent yet consistent qualities which LSTM is able to identify these qualities from the sequence of observations. However, societal conventions impose constraints which can affect consumers' behaviour. Thus, such constraints are hypothesized to be able to reflect more prominently in face-to-face transactions than e-commerce transactions.

Since the results are obtained from aggregating the transaction history through manually engineered features or a sequence learner, one possible extension can find out why aggregating different aspects of history with a sequence learner can be different from the hand-engineered features in some context.

Paper 2: AI^2: Training a big data machine to defend

Problem Statement

Fraud detection methods today suffer from three main issues. Firstly, many institutions lack labeled examples from past frauds, undermining the ability to employ supervised learning models. Secondly, fraud methods are constantly evolving, reducing the effectiveness of supervised models as there are more undetected attacks (false negatives). Lastly, relying on human analysts to investigate individual attacks is costly and time-consuming.

Main Results

Veeramachaneni *et al.* significantly contributed through the paper through the development of Active Model Synthesis, which involves the following steps repeated daily:

1. Analyses the behaviours of different entities within a raw big data set
2. Presents the analyst with an extremely small set of events generated by unsupervised outlier detection models
3. Collects analyst feedbacks (labels) about these events
4. Learns a supervised model using the collected feedback
5. Uses the supervised models in conjunction with the unsupervised models to predict attacks

As a result, detection rate improved by an average of 3.41 times, and false positives were reduced five fold.

Analysis of the Results

We implemented the AI^2 algorithm through a simplified model to verify its effectiveness in a small scale trial. We implemented a RandomForestClassifier for the supervised module and a Replicator Neural Network for the unsupervised module. As for the analyst feedback, we made feedback decisions based on a specified threshold value.

Supervised Model

We predicted the class probabilities of the unseen data at each timestep. If the probability exceeds the threshold, it is labelled as Fraud. Else, non-fraud. For our implementation, we chose an arbitrary probability threshold of 0.30 as we want to be sensitive to positive classes.

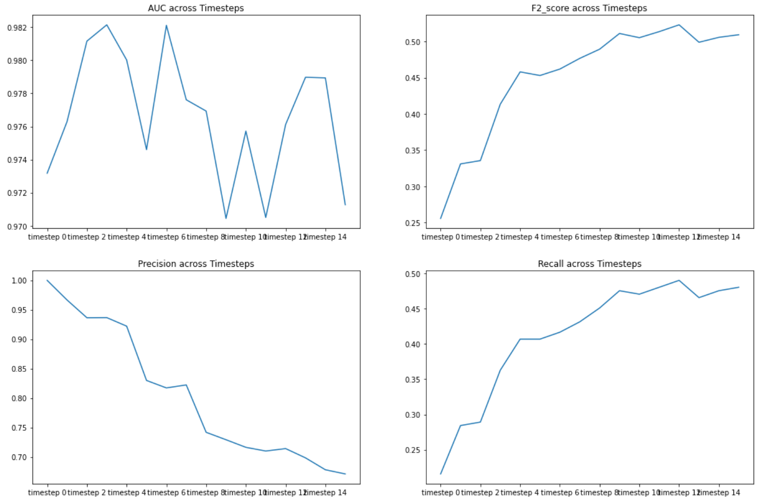


Fig 3. AUC, F2 score, precision and recall across timesteps

Unsupervised Model

We predicted the MSE as the reconstruction error of the unseen data at each timestep. If the MSE exceeds the threshold, it is labelled as Fraud. Else, non-fraud. For our implementation, we chose an arbitrary MSE threshold of 0.03 as we want to be sensitive to positive classes.

When trained the model across 15 timesteps and tested against the same held-out test set for each timestep. We obtain the following results. (Figure 3)

We see an increase in F2-Score and Recall across 15 timesteps from the base RandomForest model at timestep 0. However, precision decreased across timesteps. A decrease in Precision indicates an increase in false positives, while an increase in Recall indicates a reduction in false negatives.

This is ideal in the context of fraud detection, as we would rather the model be more sensitive in the detection of fraud. Even if a false positive is flagged, a secondary analysis by the analysts can proceed to verify the flagged transaction as fraud. We also strive to reduce false negatives so that fewer attacks go unnoticed.

In essence, the cost of false negatives is higher than that of false positives. Hence, the implemented AI^2 model being able to increase recall will reduce cost incurred by stakeholders. Therefore, this small scale trial is a promising indicator that the AI^2 can benefit financial institutions.

Assumptions, Limitations & Potential Extensions

In the paper, it is assumed that the analyst is able to accurately label the fraud transactions. A mistake on the analyst feedback may cause the model to worsen instead, and this mistake will compound over timesteps. In our implementation, we assumed that the outlier threshold score is an accurate predictor of fraudulent transactions. However, as addressed in the paper, an outlier transaction does not always translate to a transaction with malicious intent.

As an extension, AI^2 could consider the usage of synthetic oversampling techniques too to better overcome the issues of data imbalance.

Paper 3: FraudMiner: A Novel Credit Card Fraud Detection Model Based on Frequent Itemset Mining

Problem Statement

Firstly, credit card transaction datasets are highly imbalanced. Generally, in real cases, 99% of the transactions are legitimate while only 1% of them are fraud. Secondly, credit card fraud datasets are anonymous most of the time as banks are unwilling to reveal sensitive customer transaction data due to privacy. Field names are also sometimes changed so that analysts are unaware of the actual fields.

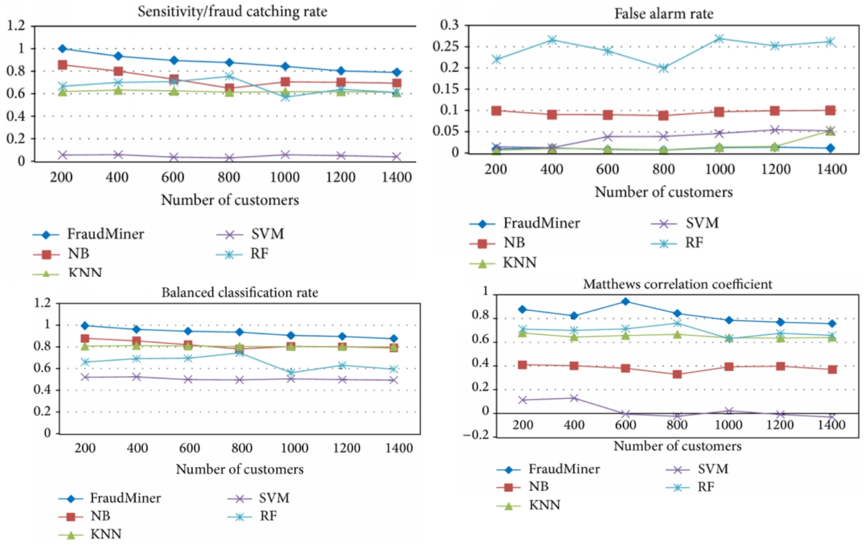


Fig 4. Comparison of sensitivity, false alarm rate, balances classification rate and Matthew correlation coefficient for different models

Main Results

Fraud Miner involves 2 phases. During the **training** phase, Fraud Miner constructs a pattern database using Frequent Itemset Mining to form legal and fraud patterns of each customer. This is done by applying Apriori algorithm to the set of legal and fraud transactions of each customer separately, and storing the largest frequent itemset in the corresponding legal or fraud pattern database. During the **testing** phase, the fraud detection system uses a matching algorithm which traverses the fraud and legal pattern databases for a match with the incoming transaction.

Fraud Miner helps overcome the problem of **anonymous** data as it considers each attribute equally without giving preference to any attribute in the dataset. In addition, creating separate legal transaction patterns (customer buying behaviour pattern) and fraud transaction pattern (fraudster behaviour pattern) for each customer helps to counter the problem of **imbalanced** data. It is also found that fraudsters are intruding into customer accounts after learning their genuine behavior only. Therefore, instead of finding a common pattern for fraudster behavior, it is more valid to identify fraud patterns for each customer.

As compared to other models such as Naives Bayes model, SVM, K Nearest Neighbours, FraudMiner performed better in terms of higher sensitivity, balanced classification rate, Matthew's coefficient and lower false alarm rates.

Analysis of the Results

We implemented a simplified model of FraudMiner to compare its performance on a small scale trial. For the Apriori algorithm, we used a support of 0.9 as stated in the paper, while for the testing algorithm, we used a matching percentage of 0.6 which we found to give the best F1 score after running the model on different matching percentages. The paper did not discretize any data, but we found that discretizing continuous data helped the model to improve its performance.

FraudMiner gave a decent F2 score of 0.712, with a precision of 0.37 and a recall of 0.93. With 922 customers in our training set, our implementation of fraudMiner performed slightly better in terms of a higher recall of 0.93 as compared to approximately 0.7 as stated in the paper.

F2-Score	Precision	Recall
0.712	0.37	0.93

Assumptions, Limitations & Potential Extensions

The paper assumes that customers will have more than one transaction in the database. In the experiments conducted in the research paper, they removed transactions corresponding to those customers who only have one transaction in the training dataset. This is because it is difficult to find patterns from a single transaction. Hence, this also means that if an incoming transaction comes from a customer who is making his/her first or second purchase, the model will not be able to make a prediction.

One limitation of FraudMiner is that, if attributes of both transactions are exactly the same, but the target class is different, FraudMiner will always predict it as non-fraud. This is because both transactions will match either the legal pattern or the fraud pattern of the customer.

A possible extension could be discretizing continuous values such as transaction amount. Transaction amounts usually differ and using an exact transaction amount in the pattern collected to compare to that of the incoming transaction would probably not be indicative. It would be better if the continuous attribute of the incoming transaction is compared to a range of values. This is supported by our implementation of FraudMiner whereby the performance improved significantly after discretizing continuous values in our dataset.

Comparing Implementations

Insights from Model Comparisons

Model	F2-Score	Precision	Recall
FraudMiner	0.712098	0.367589	0.930000
LSTM + RandomForest Feature Selection	0.653061	0.780488	0.784314
AI^2 (after 15 timesteps)	0.509356	0.671233	0.480392
Baseline RandomForest	0.206767	1.000000	0.215686

In terms of F2-Score, the FraudMiner model performed the best out of the 3 implemented models, followed by LSTM + RandomForest Feature Selection, and AI^2. This is possibly due to the FraudMiner and LSTM models being trained on a fully labelled training set. FraudMiner is able to effectively create a pattern database based on the provided labelled datasets, and predict potential frauds as a result too. However, the rigorous pattern matching may have caused FraudMiner to be overly sensitive to positive classes, resulting in it having the lowest Precision score.

The LSTM model is able to harness the labelled dataset to draw sequential relationships between transactions, drawing patterns to classify between fraudulent and non-fraudulent transactions, hence resulting in the second best F2-Score.

AI^2 performed the worst in our trials, with the lowest F2-Score and Recall, indicating the poor performance in identifying true positives. This is most likely due to the simulated analyst feedback process. We assumed that anomalies equate to frauds in our threshold identification process, but this might not be true in reality, as an analyst will have the knowledge to discern and identify actual frauds. Nonetheless, the AI^2 model still outperformed the baseline RandomForest model in terms of F2-Score and Recall.

Although the supervised models were superior in our trial, however, in reality, we may not have fully labelled datasets due to resource constraints. A hybrid between unsupervised and supervised models may be preferred in a real life scenario. AI^2 combines both supervised and unsupervised learning to create a training model that utilises small training budgets. Specifically, it does not expect the analyst to manually provide feedback for every incoming transaction, but only those of higher scores as identified by the model.

In detecting fraud in e-commerce and face-to-face transactions, the LSTM+RandomForest feature selection model may be most suited due to its sensitivity to time and sequence, which were shown to be important variables in the model. Furthermore, the feature selection process is able to identify the most important features of the model, hence utilising only the most important features reduces noise during model building and testing.

Another factor to consider in real life scenarios is the speed requirements for fraud detection. Once a fraud is detected, the bank should identify the fraud in the quickest possible time and cancel the transaction. As the FraudMiner model compares transactions to a pre-trained database, it is much faster as compared to the deep learning models when predicting transaction classes. In this regard, FraudMiner may be more suitable.

Although there exists a model that performs the best in our trial implementations (in terms of F2-score), this does not translate to real life performance, as there exists a multitude of extra factors that will determine if a model is suitable, such as budget and resource constraints, or the speed requirements. Hence, when implementing the models in real life, we need to be aware of the context and choose the model that is best suited.

Limitations

We lack access to real life datasets. The synthetic dataset that we trained our models on may not accurately reflect inherent patterns between real fraudulent transactions. Our trials do not accurately represent real life scenarios, as there are factors that influence the suitability of models, such as speed requirements, budget, time and resource constraints.

Potential Extensions

We could potentially cxplore oversampling methods and how different oversampling methods complement the models we implemented. We could also set up a more realistic training environment that involves factors such as speed requirements and resource constraints. Finally, we could also potentially combine all 3 models into a hybrid model containing multiple supervised and unsupervised training modules.

Conclusion

Our project compared the performances of 3 drastically different fraud detection models. In implementing the 3 different models, we identified the strengths and weaknesses of each model, and the context that each model was most suitable for. In our trial, we tested the 3 models in a singular context, and observed that the LSTM model with RandomForest Feature Selection performed the best in terms of F2-Score, followed by the FraudMiner and AI^2 models. This, however, is not entirely representative of real life environments and limits the conclusiveness of our trails. Hence, further studies should be conducted considering different contexts and their associated constraints.

Bibliography

Brownlee, J. (2020, January 14). A Gentle Introduction to the Fbeta-Measure for Machine Learning. Machine Learning Mastery. https://machinelearningmastery.com/fbeta-measure-for-machine-learning/

Fiserv. (2019). Streamlining Business-to-Consumer Payments With Digital Disbursements. Fiserve. https://www.fiserv.com/en/about-fiserv/resource-center/brochures/streamlining-business-to-consumer-payments-with-digital-disbursements.html

Harris, B. (2020, August 5). Credit Card Transactions Fraud Detection Dataset. Kaggle. https://www.kaggle.com/kartik2112/fraud-detection

Jurgovsky, J., Granitzer, M., Ziegler, K., Calabretto, S., Portier, P. E., He-Guelton, L., & Caelen, O. (2018). Sequence classification for credit-card fraud detection. Expert Systems with Applications, 100, 234–245. https://doi.org/10.1016/j.eswa.2018.01.037

Reilly, J. (2021). Credit Card Fraud Detection With AI: What You Need to Know. Akkio. https://www.akkio.com/post/credit-card-fraud-detection-with-ai

Seeja, K. R., & Zareapoor, M. (2014). FraudMiner: A Novel Credit Card Fraud Detection Model Based on Frequent Itemset Mining. The Scientific World Journal, 2014, 1–10. https://doi.org/10.1155/2014/252797

Shift. (2021, September 29). Credit Card Fraud Statistics [Updated September 2021] Shift Processing. Shift Credit Card Processing. https://shiftprocessing.com/credit-card-fraud-statistics/

Surabhi, S. (2021, October 19). Strong growth in digital payments indicates a lasting shift in consumer payment behaviour. Business Line. https://www.thehindubusinessline.com/data-stories/data-focus/strong-growth-in-digital-payments-indicates-a-lasting-shift-in-consumer-payment-behaviour/article37073025.ece

Tham, D. (2021, September 15). About S$500,000 stolen in fraudulent card payments involving diversion of SMS one-time passwords. CNA. https://www.channelnewsasia.com/singapore/credit-card-fraud-banks-divert-sms-otp-overseas-imda-mas-spf-2179541

Veeramachaneni, K., Arnaldo, I., Korrapati, V., Bassias, C., & Li, K. (2016). AI^2: Training a Big Data Machine to Defend. 2016 IEEE 2nd International Conference on Big Data Security on Cloud (BigDataSecurity), IEEE International Conference on High Performance and Smart Computing (HPSC), and IEEE International Conference on Intelligent Data and Security (IDS). Published. https://doi.org/10.1109/bigdatasecurity-hpsc-ids.2016.79