Improving Card Fraud Detection through Suspicious Pattern Discovery

Fabian Braun[1], Olivier Caelen[2], Evgueni N. Smirnov[3],
Steven Kelk[3], and Bertrand Lebichot[4]

[1] R&D, Worldline GmbH, Aachen, Germany
[2] R&D, Worldline SA, Brussels, Belgium
[3] Maastricht University, Department of Data Science & Knowledge Engineering,
Maastricht, The Netherlands
[4] Université catholique de Louvain, Louvain-La-Neuve, Belgium

Abstract. We propose a new approach to detect credit card fraud based on suspicious payment patterns. According to our hypothesis fraudsters use stolen credit card data at specific, recurring sets of shops. We exploit this behavior to identify fraudulent transactions. In a first step we show how suspicious patterns can be identified from known compromised cards. The transactions between cards and shops can be represented as a bipartite graph. We are interested in finding fully connected subgraphs containing mostly compromised cards, because such *bicliques* reveal suspicious payment patterns. Then we define new attributes which capture the suspiciousness of a transaction indicated by known suspicious patterns. Eventually a non-linear classifier is used to assess the predictive power gained through those new features. The new attributes lead to a significant performance improvement compared to state-of-the-art aggregated transaction features. Our results are verified on real transaction data provided by our industrial partner[5].

Keywords: credit card fraud detection, supervised learning, feature engineering, frequent pattern mining, bicliques, graph analysis

1 Introduction

In today's world payments are often effected electronically. Instead of cash, people use credit and debit cards for payments at the point of sale (POS) and can directly issue purchases on shopping websites using their card data (*E-commerce*). However the rise of electronic financial transactions has led to new crime patterns: fraudsters try to misuse the data of legitimate persons to effect payments in their name. Therefore payment processors employ detection techniques to identify fraudulent transactions.

Historically fraud detection is carried out within rule-processing systems where fraudulent transactions are detected if they fulfill certain criteria, e.g. issued at a specific shop at a specific time of a day. The rules of these systems

[5] Worldline `http://www.worldline.com`

are crafted manually by human experts or generated by rule learning algorithms [12]. More sophisticated systems use supervised learning to build classification models which learn to identify fraud from known fraudulent transactions in the past [4].

In the best case a fraudulent transaction is immediately detected and rejected by the system. However, even after acceptance of a fraudulent transaction it is useful to detect it because the fraudster is likely to reuse the same card data for further transactions until the card is blocked. Finding compromised cards becomes easier with each further fraudulent transaction from the card in question, under the condition that detection techniques are not solely analyzing individual transactions. Therefore, feature aggregates built from the transaction history of a credit card are heavily used to improve fraud detection in rule-based systems as well as machine learning approaches [17].

The historical rule based approach has the advantage that it allows human investigators to adapt detection systems according to very specific fraud scenarios. The full expertise of the investigator results in very targeted fraud detection with few false alerts. On the other hand non-linear models such as neural networks are not transparent in regard to how they decide on the label of a transaction, but they are able to find hidden meaning in the data, that investigators are not aware of.

In this work we combine the advantages of historical rule learning and non-linear models to outperform existing fraud detection methods. For this purpose we feed the pattern-indicated "suspiciousness" of a transaction into a non-linear classifier. This additional information boosts the classifier performance by 20% in terms of area under precision-recall-curve (AUCPR).

To provide an example assume that we find in historical data that some fraudsters tend to issue their fraudulent transactions always at the shops $\{E, F, G\}$. After further investigation we find out that actually 50% of the cards which have made a transaction at all the shops $\{E, F, G\}$ are compromised. The pattern $\{E, F, G\}$ is therefore highly suspicious—We have identified an anomaly which can be used to find further fraud cases. Therefore we not only provide classical transaction features such as amount and timestamp to our non-linear model but in addition whether the card is used according to a known suspicious pattern. We show that this combined approach leads to a significant performance increase. A similar approach has been applied to identify companies which might go bankrupt deliberately in order to avoid taxation [16]. We extend and adapt the core ideas from this work to the domain of credit card fraud and show that the relationships between cards and card acceptors similarly carry information indicating which cards might be under the control of a fraudster.

Regarding the structure of this paper we first give a detailed description of our contribution (Section 2). Subsequently we explain the preprocessing and augmentation of our data based on existing scientific work (Section 3). Then we describe the concrete experimental setup (Section 4) for testing our contribution. Finally we report our results (Section 5) before drawing a conclusion (Section 6).

2 Pattern Suspiciousness

Our contribution is based on the hypothesis that compromised credit cards can be identified by inspecting the patterns of card acceptors, e.g. shops, at which they have been used [6]. A pattern is for example “card x has been used in the shops $\{E, F, G\}$” (See definitions 1 and 2).

Definition 1. *Pattern: Unordered set of acceptors containing at least 2, and at most n, elements, where n is the maximally allowed pattern size. We require that a pattern consists at least of* 2 *acceptors.*

Definition 2. *Pattern match: A credit card c matches a pattern at moment t if all acceptors of the pattern appear in the transaction history* $T_{[t-\Delta t,t]}$ *of c.* Δt *denotes a time difference.*

Definition 3. *Pattern support: The absolute number of cards which match a given pattern.*

With techniques originating from the domain of association rule mining we can identify frequent patterns [1] in the transaction data, i.e. patterns that are common among multiple credit cards. We aim to compute the suspiciousness of such patterns by counting how many of the matching cards are compromised. The underlying assumption is that a pattern which is highly exposed to fraud in the past can be used to detect fraud on future transactions. Thus our approach requires a database of patterns and their suspiciousness. However, we do not want to use the suspicious patterns directly to detect fraudulent cards. Instead we incorporate this information in newly defined transaction features and build a model which also takes into account all other given information to detect fraudulent transactions.

When a new transaction arrives we evaluate whether we recognize a pattern from the database in the transaction history of the card. Then we augment the new transaction with information about the matching patterns’ suspiciousness.

2.1 Pattern Enumeration and Scoring

To derive suspiciousness scores for each pattern we need to look at the cards which match the patterns in our historical data. A higher number of compromised cards indicate a higher pattern suspiciousness. For this purpose we introduce the definition of a *biclique* which incorporates a pattern and all its matching cards. We can conclude that for each pattern a corresponding biclique exists and vice versa. The *biclique* definition relies on a graph representation of the data in which the acceptors form a first and the credit cards a second vertex type. A credit card can be linked to an acceptor by a transaction. An example for a *biclique* is depicted in Figure 1. The representation of a bipartite graph to find suspicious bicliques has already been applied successfully in the domain of bankruptcy fraud, i.e. predicting which companies might go bankrupt deliberately in order to avoid taxation by analyzing the business partners of those companies [16].

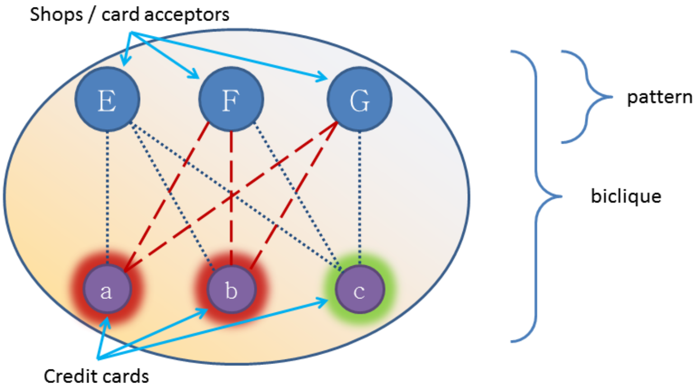

Fig. 1. Example transaction graph to demonstrate the hypothesis of suspicious acceptor patterns. Red long-dashed edges represent fraudulent transactions, blue dotted edges represent genuine transactions. The depicted pattern $\{E, F, G\}$ is suspicious as two out of three (= 66%) of its matching cards are compromised (i.e. have fraudulent transactions).

Definition 4. *Bipartite graph: Let $G = (N, E)$ denote a graph consisting of the vertices N and edges E. G is* bipartite *if N can be divided into two sets U and V such that every edge has one endpoint in U and one endpoint in V.*

Definition 5. *Biclique: Let $G = (V \cup U, E)$ denote a bipartite graph. A subgraph $(V' \cup U', E')$ of G is called a* biclique *if $V' \subseteq V, U' \subseteq U, E' \subseteq E$ and for every $u \in U'$ and $v \in V'$, $\{u, v\} \in E'$.*

The patterns' suspiciousness is directly expressed through the ratio of compromised cards among all cards in its corresponding biclique. The biclique depicted in Figure 1 contains three cards out of which two are compromised because they have fraudulent transactions. Therefore the suspiciousness score of this biclique is 0.66. In addition to the biclique fraud ratio we store the pattern size and the number of matching cards in the pattern database for later use.

Algorithmically the bicliques are identified in two consecutive steps. First, we identify candidates for suspicious patterns only taking into account compromised cards. In a second step we generate full bicliques from those candidates, now taking into account all cards. The first step can be performed by any appropriate frequent pattern mining algorithm such as the apriori-algorithm [2]. In the second step the bicliques are created by identifying the common cards of each acceptor pattern, which is equivalent to a set intersection operation for each pattern [11, chap. 10.3.3].

The time complexity of enumerating all maximal bicliques based on frequent itemset mining can be reduced to $O(mnN)$, where m is the number of edges in the graph (credit card transactions), n the number of vertices (cards and acceptors) and N the number of maximal bicliques [9]. The apriori algorithm does not achieve this complexity but has proven to be adequately efficient for our experiments.

2.2 Feature Aggregation

At this point we have identified suspicious patterns on historical transaction data and quantified this suspiciousness in terms of pattern features. In the next step these pattern features will be used to find compromised cards in a target dataset with unknown labels. Therefore the cards in this distinct dataset must be augmented with the pattern information.

We verify for each card which patterns it matches and derive new features from the matched patterns: the number, the mean and the maximum suspiciousness among them. In the domain of credit card fraud we suspect that patterns having the maximum suspiciousness are most important for detecting future fraud. Therefore we add more information about these most suspicious patterns to the new card features: the number of acceptors (pattern size) and the number of matching cards (pattern *support*). The five new features are summarized in Table 1.

Table 1. Newly proposed pattern features. Each transaction is augmented with these five attributes to achieve a better predictive performance.

attribute	explanation
pattern count	number of patterns matched by the card
mean suspiciousness	of all matching patterns
max suspiciousness	score of most suspicious matching pattern
max suspicious pattern's size	size of m. s. matching pattern
max suspicious pattern's support	supporting cards of m. s. matching pattern

3 Real Data and Preprocessing

To assess our approach we use a real-world transaction dataset from our industrial partner. The dataset contains POS and E-commerce-transactions. Each day of data comprises on average $517,569.7$ transactions with a standard deviation of 59902.9. Out of these transactions on average 0.152% are fraudulent with a standard deviation of 0.040%.

The dataset provides 21 intrinsic transaction features. Those comprise nominal identifiers of transactions, cards and acceptor and more information related to these entities: for example the transaction amount, the timestamp of the transaction and the merchant category. These attributes are common in the domain of fraud detection [13, 12].

A classifier which is only trained on intrinsic attributes is likely to achieve a poor predictive performance. Therefore we augment the given attributes with aggregated new features, which have proven to enhance the prediction performance in other scientific work [3]. The authors derive information on how often the card was used in a similar manner before the current transaction. For example they add the number of transactions issued at the same shop in the past and the average transaction amount of recent payments.

Another approach [5, chap. 5.1.3] computes the risk of discrete attribute values of being associated to fraud transactions, e.g. the risk that a specific acceptor is used for a fraud transaction. In this work transactions are merged with the risk scores associated with their attribute values.

In total a set of 45 transaction features is used to obtain a baseline performance score for a state-of-the-art fraud detection model. These consist of basic features being intrinsic to each transaction (amount etc.), aggregated card features [3] and risk scores [5, chap. 5.1.3] for all categorical attributes in our data. As the data used for the referenced work is not publicly available we have fully reimplemented their work to augment our data with the same attributes.

4 Experimental Setup

To estimate the predictive performance gain we compare the performance of a model trained on a state-of-the-art dataset with the performance reached when adding our newly proposed features (Table 1). We choose a random forest model because it is a standard and well-performing model in the domain of fraud detection [3, 6]. It allows the construction of sophisticated performance metrics because it is capable of returning class likelihoods instead of hard labels. We use random undersampling for training the random forest model such that each training set contains 1500 fraudulent transactions and 13500 genuine transactions to address the imbalance in the data. We choose a sampling ratio of 10% to compare to [3]. This technique performs well in conjunction with random forest models [15]. A random forest model requires two parameters: we fix the number of trees at 501 and leave the number of split candidate variables at $\sqrt{p}$, where p is the number of transaction attributes in the training set—a common default setting [10]. No further tuning of the parameters is required as we want to compare the performance of multiple random forests trained on different attribute subsets rather than producing one highly tuned classifier. For the performance measures requiring hard labels we set a static cutoff threshold of 0.75, i.e. if a transaction is voted to be fraudulent by less than 75% of the trees, it is classified as genuine.

For computing the newly proposed pattern features we restrict the size of the acceptor patterns to two to six. Patterns of size one (i.e. individual acceptors) are already incorporated in the acceptor risk score [5]. Patterns of larger sizes than six are ignored because they would require that a fraudster issues more than six transactions at different acceptors before they can be detected. Additionally we require that each pattern is matched by at least 4 compromised cards to ensure a minimum evidence for a suspiciousness score (*minimum absolute support*). In summary we only assess bicliques consisting of two to six acceptors and at least four cards.

Another parameter is the size of the time-window Δt from which we derive frequent patterns. We choose a window of five days, relying on the fact that fraudsters try to issue their payments within a short time-frame before the card of the customer is blocked. A larger window could be an interesting subject for future research, because the acceptors used by the legitimate cardholder might also carry important information because in many cases one of those acceptors is the source of the data breach.

4.1 Data Splits

When evaluating our approach we have to be careful about setting up training and test data. The risk score [5] and the newly introduced pattern features require the labels of historical transactions as input which might lead to a biased model when the same transactions are used for training. To ensure that this does not occur we split data into three sets: feature learning set, training set and test set. Additionally we want to assess how good the model behaves in a temporal context, i.e. learning on past transactions for predicting future transactions. Therefore we split based on the timestamp of transaction acceptance. We use five days of transaction data for learning suspicious patterns and risk scores. The subsequent five days of data are used for training the random forest model. Finally we test this model on the subsequent day of data (See Figure 2). To obtain statistically sound results we generate 40 different learning, training and test sets from our data.

In fraud detection it is trivial to predict the label of transactions once we know that a card is compromised. To avoid an overoptimistic estimate of the performance we remove transactions from known compromised cards from subsequent splits. For example when a card already has a fraudulent transaction in the training set, its transactions are removed from the test set.

4.2 Performance Measures

The choice of adequate performance indicators is highly influenced by the class imbalance of the fraud detection problem. Standard measures as prediction accuracy and area under ROC-curve (AUC) [14] are not suitable because negative and positive instances contribute equally to them while in unbalanced problems the positive class should be emphasized [17].

We base our conclusions on precision and recall which focus on the fraudulent class. In the fraud domain the precision shows how many transactions might be reported erroneously by the model. The recall captures how many fraudulent transactions are completely missed by the system and are eventually detected by the cardholder in their account statement.

Additionally we use the area under precision-recall-curve (AUCPR) as a cutoff-independent measure, which is better suited to imbalanced problems than the classical AUC [7]. The precision among the top-k-ranked alerts is another

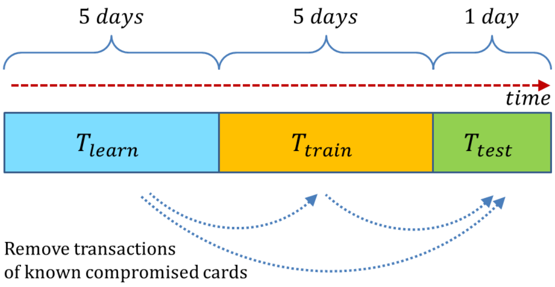

Fig. 2. The transaction dataset is split into three parts: a feature learning dataset, a training set and a test set. The first is used to learn which acceptors patterns are suspicious. The second is used to train a model and the third to test its predictive performance. To avoid reporting already known compromised cards, those are removed from subsequent splits.

Table 2. Performance of classifiers: row 1, 3 our baseline; row 2, 4 baseline + contributed features. We report the average performance and std. deviation obtained from 40 different learning-training-test cycles.

model, attributes	precision	prec. at k	recall	AUCPR	accuracy	AUC
random forest baseline	0.333 ± 0.11	0.381 ± 0.09	0.418 ± 0.10	0.323 ± 0.10	0.999	0.971
random forest + **contribution**	0.371 ± 0.11	0.419 ± 0.09	0.444 ± 0.11	0.387 ± 0.10	0.999	0.971
logistic regression baseline	0.072 ± 0.04	0.129 ± 0.10	0.338 ± 0.10	0.072 ± 0.056	0.995	0.942
logistic regression + **contribution**	0.095 ± 0.05	0.183 ± 0.11	0.406 ± 0.11	0.103 ± 0.071	0.996	0.944

meaningful performance indicator in fraud detection [5, 3]. We fix the parameter k at the number of positive instances in the test set such that a score of 1 indicates perfect prediction.

5 Results

As a first result we observe in Table 2 that our baseline performance differs from what is reported in other scientific work[6], although our dataset contains the same and more features. Our precision is higher, while the recall is lower, i.e. the alerts of our model are more accurate, but it detects less fraudulent transactions. This may be caused by our experimental setup which is oriented towards a real-world scenario in which we use past data to predict future fraud. The differences may also originate from unknown deviations between the used datasets and model parameters.

[6] [3, Table 6c, 7c] reports for another dataset a precision of 0.233, a precision at k of 0.494, a recall of 0.747, an accuracy of 0.987 and an AUC of 0.934.

Looking at our random forest classifier we observe that the new pattern features lead to an average performance improvement of 0.064 in AUCPR. On average the area grows by 20%. In the cycle of largest absolute performance improvement the AUCPR grows from 0.157 to 0.357, while in other cycles there is no significant improvement in performance. This indicates that the importance of the new features changes between the different time windows of data. Generally the mean deviation of the performance measurements is high compared to the performance difference between the models. Therefore we apply the Friedman-Nemenyi test [8] on the performance results. For a significance level of $\alpha = 0.05$ the test confirms that the newly proposed pattern features significantly improve the performance regarding *all* different performance measures reported in Table 2. A logistic regression classifier confirms the positive effect of the new features, although performing generally worse than the random forest model.

The performance fluctuation demonstrates the concept drift in the data, i.e. the constant change in the payment behavior of fraudsters. For some days the task of identifying fraudulent transactions can be simple while it becomes more challenging on other days. Likewise for some days fraudsters might act according to previously identified patterns while on others they change their habits and the patterns lose their explanatory power. During the experiments we observe that the generation of suspicious patterns leads to more than 300 patterns for some time windows of data and sometimes only to around 50. We presume that by storing all previously found patterns in a database and only re-estimating their suspiciousness for new time windows could further improve the performance.

6 Conclusion

We have investigated a pattern-based approach to identify fraud among financial transactions. It is common knowledge that the individual transactions of ongoing fraud can seem entirely unsuspicious. The fraudulent activity only becomes evident once the full sequence of transactions is analyzed. In this work we incorporate pattern information in the form of new attributes into a classical machine learning model and show that the predictive performance improves significantly. The area under precision-recall-curve grows on average by 20% when compared to a state-of-the-art baseline. This result shows that fraudsters tend to use compromised cards at the same set of card acceptors over and over again. This knowledge can be exploited to detect fraud more reliably through generating a database of suspicious acceptor patterns.

Our approach can of course be used to reveal suspicious acceptor patterns, but it can also be used in a wider sense too. It can for example be extended to suspicious patterns of point of sale locations or any other categorical transaction attributes. First experiments into this direction based on the merchant category code show promising results.

While looking at several transactions to detect fraud increases the detection performance it has one drawback: compromised cards are only detected once they have been used for transactions at all acceptors in a suspicious pattern. In practice that means that a fraudster is able to issue multiple transactions before the fraud is detected. However, this drawback comes rather from the nature of fraud—Human experts have found that it is in most of the fraud scenarios impossible to detect them on the first fraudulent transaction.

References

1. Aggarwal, C.C., Han, J.: Frequent Pattern Mining. Springer (2014)
2. Agrawal, R., Imieliński, T., Swami, A.: Mining association rules between sets of items in large databases. In: Int. Conf. on Management of Data. pp. 207–216. SIGMOD '93, ACM, New York (1993)
3. Bhattacharyya, S., Jha, S., Tharakunnel, K., Westland, J.C.: Data mining for credit card fraud: A comparative study. Decision Support Systems 50(3), 602 – 613 (2011)
4. Bolton, R.J., Hand, D.J.: Statistical fraud detection: A review. Statistical Science 17(3), 235–249 (2002)
5. Dal Pozzolo, A.: Adaptive Machine Learning for Credit Card Fraud Detection. Ph.D. thesis, Université libre de Bruxelles (2015)
6. Dal Pozzolo, A., Caelen, O., Borgne, Y.L., Waterschoot, S., Bontempi, G.: Learned lessons in credit card fraud detection from a practitioner perspective. Expert Systems with Applications 41(10), 4915 – 4928 (2014)
7. Davis, J., Goadrich, M.: The relationship between precision-recall and roc curves. In: 23rd Int. Conf. on Machine Learning. pp. 233–240. ICML '06, ACM, New York (2006)
8. Demšar, J.: Statistical comparisons of classifiers over multiple data sets. The Journal of Machine Learning Research 7, 1–30 (2006)
9. Li, J., Liu, G., Li, H., Wong, L.: Maximal biclique subgraphs and closed pattern pairs of the adjacency matrix: A one-to-one correspondence and mining algorithms. IEEE Trans. Knowl. Data Eng. 19(12), 1625–1637 (2007)
10. Liaw, A., Wiener, M.: Classification and regression by randomforest. R News 2(3), 18–22 (2002)
11. Rajaraman, A., Ullman, J.D.: Mining of Massive Datasets. Cambridge University Press, New York, NY, USA (2011)
12. Sánchez, D., Vila, M., Cerda, L., Serrano, J.: Association rules applied to credit card fraud detection. Expert Systems with Applications 36(2), 3630–3640 (2009)
13. Shen, A., Tong, R., Deng, Y.: Application of classification models on credit card fraud detection. In: Int. Conf. Service Systems and Service Management. pp. 1–4. ICSSSM '07, IEEE (2007)
14. Spackman, K.A.: Signal detection theory: Valuable tools for evaluating inductive learning. In: 6th Int. Workshop on Machine Learning. pp. 160–163. Morgan Kaufmann, San Francisco (1989)
15. Van Hulse, J., Khoshgoftaar, T.M., Napolitano, A.: Experimental perspectives on learning from imbalanced data. In: 24th Int. Conf. on Machine Learning. pp. 935–942. ICML '07, ACM, New York (2007)
16. Van Vlasselaer, V., Akoglu, L., Eliassi-Rad, T., Snoeck, M., Baesens, B.: Guilt-by-constellation: Fraud detection by suspicious clique memberships. In: 48th Hawaii Int. Conf. on System Sciences. pp. 918–927. HICSS '15, IEEE (2015)
17. Whitrow, C., Hand, D.J., Juszczak, P., Weston, D., Adams, N.M.: Transaction aggregation as a strategy for credit card fraud detection. Data Min. Knowl. Discov. 18(1), 30–55 (2009)