Proceedings of Machine Learning Research 71:66–75, 2017 KDD 2017: Workshop on Anomaly Detection in Finance

Collective Fraud Detection Capturing Inter-Transaction Dependency

Bokai Cao[*] CAOBOKAI@UIC.EDU
Department of Computer Science, University of Illinois at Chicago, IL, USA

Mia Mao MMAO@EA.COM
Electronic Arts, Redwood City, CA, USA

Siim Viidu SVIIDU@EA.COM
Electronic Arts, Redwood City, CA, USA

Philip S. Yu PSYU@UIC.EDU
Department of Computer Science, University of Illinois at Chicago, IL, USA
Institute for Data Science, Tsinghua University, Beijing, China

Abstract

In e-commerce, different payment transactions have different levels of risk. Risk is generally higher for digital goods, but it also differs based on product and its popularity, the offer type (packaged game, virtual currency to a game or subscription service), storefront and geography. Existing fraud policies and models make decisions independently for each transaction based on transaction attributes, payment velocities, user characteristics, and other relevant information. However, suspicious transactions may still evade detection and hence we propose a novel approach leveraging a graph based perspective to uncover relationships among suspicious transactions, *i.e.*, inter-transaction dependency. Our focus is to detect suspicious transactions by capturing common fraudulent behaviors that would not be considered suspicious when being considered in isolation. In this paper, we present HitFraud that leverages heterogeneous information networks for collective fraud detection by exploring correlated and fast evolving fraudulent behaviors. First, a heterogeneous information network is designed to link entities of interest in the transaction database via different semantics. Then, graph based features are efficiently discovered from the network exploiting the concept of meta-paths, and decisions on frauds are made collectively on test instances. Experiments on real-world payment transaction data from Electronic Arts demonstrate that the prediction performance is effectively boosted by HitFraud where the computation of meta-path based features is largely optimized. Notably, recall can be improved up to 7.93% and F-score 4.62% compared to baselines.

1. Introduction

Fraud detection has attracted significant research efforts in recent years for various tasks including finance, security and web services. In this work, we investigate the fraud detection problem on electronic game platforms where it is desirable to identify suspicious payment transactions in an early stage in order to avoid chargebacks and to enhance normal users' experience. Current fraud models make independent decisions for each transaction, and

[*] This work was partially done while the author was an intern at Electronic Arts.

detection becomes harder when intelligent adversaries are used, *e.g.*, proxy IP addresses. Graph based methods can detect frauds by leveraging the linkage information between entities of interest Gyöngyi et al. (2004); Hooi et al. (2016); Jiang et al. (2014). Such methods are relatively harder to evade because making a fraud payment transaction unavoidably generates links in the graph which reveals *inter-transaction dependency*.

Existing graph based fraud detection approaches heavily focus on homogeneous information networks and bipartite graphs. Heterogeneous information networks (HINs) Sun et al. (2011) are a special type of information networks that involve multiple types of nodes or multiple types of links. In a HIN, different types of nodes and links have different semantic meanings. Such complex and semantically enriched networks possess great potential for knowledge discovery Ji et al. (2011); Kong et al. (2013); Sun et al. (2009). Its applications to fraud detection, however, are largely unexplored.

Therefore, we are motivated to investigate how to leverage HINs to facilitate the fraud detection task. Most importantly, we seek to capture the inter-transaction dependency. It is critical to explore such relationships among suspicious transactions because fraudulent behaviors are often *correlated* and *fast evolving*. (1) HINs provide us with an effective and compact representation of linked transactions in various semantics, *e.g.*, the same currency, the same IP address, and the same game titles. The statistics of the label information (*i.e.*, fraud or normal) of these linked transactions can be aggregated, and thereby add a new dimension of measurements to distinguish suspicious transactions from normal ones based on the correlated fraudulent behaviors. (2) In order to tackle the problem of fast evolving fraudulent behaviors, we should not only consider the inter-transaction dependency across training transactions and test transactions, but also include the dependency among test transactions. Hence, suspicious transactions are identified in a semi-supervised manner by iteratively obtaining the predicted labels of test transactions and updating the statistics of linked transactions in alternation. Such a collective prediction procedure has the potential to detect a suspicious transaction even if it appears to be normal by itself but its linked transactions (other test transactions in a batch sharing categorical variables) are identified as very suspicious, and thereby improve the recall metric. The main contributions of this work are threefold:

- As fraud payment transactions generally do not occur in isolation, *i.e.*, fraudulent behaviors are often correlated and fast evolving, we formulate the fraud detection task as a collective prediction problem in a HIN to capture relationships among fraud payment transactions.
- To address the daunting challenge on huge feature space, we design a HIN that can effectively capture the various relationships among transactions. Here meta-paths (a sequence of link types) are explored to identify the relevant inter-transaction dependency features. We propose an effective and efficient algorithm to compute meta-path based features in the framework of collective fraud detection.
- We validate that the correlated and fast evolving fraudulent behaviors can indeed be explored to more effectively capture fraud payment transactions. We evaluate the proposed framework on real-world payment transaction data from Electronic Arts payment system, and results show that recall and F-score can significantly be improved by exploring

inter-transaction dependency via the proposed collective fraud detection method based on HINs.

2. Methodology

2.1. HIN Construction

In this work, we use Electronic Arts (EA) payment transaction data as an example to do the study. All transactions on EA digital platform go through a set of policies, rules and models to determine their levels of risk, and a subset of them are sent for additional manual review. An experienced team reviews those transactions and decides if they should be rejected or approved, and this review decision is used as the ground-truth for training and evaluating fraud detection algorithms. We collected manual review data for $n = 130K$ transactions during a recent period. Each transaction is associated with a $d = 2K$ dimensional feature vector, including transaction attributes, payment velocities, user characteristics, and other relevant information.

A HIN is constructed by linking entities of interest from several selected databases. Transactions are the target instances on which fraud decisions are made, so each transaction ID is represented as a node in the network, and the set of transaction IDs compose a node type in the network schema. In addition, other entities that are directly or indirectly related to a transaction are considered here, and they compose other node types in the schema, including billing accounts, user accounts, game titles, IP addresses, *etc.* Links are added based on common semantics. For example, a transaction is linked with a user if the user placed the transaction, and a transaction is linked with an item if the transaction contains the item. As a result, the constructed HIN is composed of over $400K$ nodes and $1.5M$ links. It integrates data involving $m = 12$ types of nodes, such as *transaction, user, item, title, currency, source, country, etc.* which are connected through $r = 15$ types of links, such as "transaction $\xrightarrow{containsItem}$ item", "billing $\xrightarrow{billingIP}$ IP", *etc.* Its network schema is shown in Figure 1 where each rectangle represents a node type, and each line represents a link type.

We can represent a HIN as a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. $\mathcal{V} = \mathcal{V}^1 \cup \cdots \cup \mathcal{V}^m$ denotes the set of nodes involving m node types: $\mathcal{V}^1 = \{v_1^1, \cdots, v_{n_1}^1\}, \cdots, \mathcal{V}^m = \{v_1^m, \cdots, v_{n_m}^m\}$ where v_p^i represents the p-th node of type i. $\mathcal{E} = \mathcal{E}^1 \cup \cdots \cup \mathcal{E}^r \subseteq \mathcal{V} \times \mathcal{V}$ denotes the set of links between nodes in $\mathcal{V}$ involving r link types. Mathematically, a link type k starting from source nodes of type i and ending at target nodes of type j is described by an adjacency matrix $\mathbf{A}^k \in \mathbb{R}^{n_i \times n_j}$ where $\mathbf{A}^k[p, q] = 1$ if there exists a link in $\mathcal{E}^k$ between v_p^i and v_q^j, otherwise $\mathbf{A}^k[p, q] = 0$. We can write this link type as "$\mathcal{V}^i \xrightarrow{\mathcal{E}^k} \mathcal{V}^j$". We may assume without loss of generality that nodes in $\mathcal{V}^1$ are the target entities, *i.e.*, transactions, in the case of fraud detection. The number of target entities is denoted as $n = n_1$. Formally, we are given a data matrix $\mathbf{X} = [\mathbf{x}_1^T; \cdots; \mathbf{x}_n^T]$ where $\mathbf{x}_i \in \mathbb{R}^d$ is the feature vector of the i-th transaction which typically involves extensive feature engineering and domain knowledge. Labels are denoted as $\mathbf{y} = [y_1, \cdots, y_n]$ where $y_i = 1$ if the i-th transaction is a fraud, otherwise $y_i = 0$.

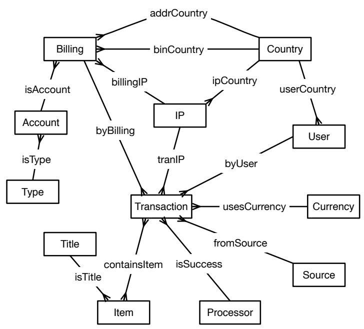
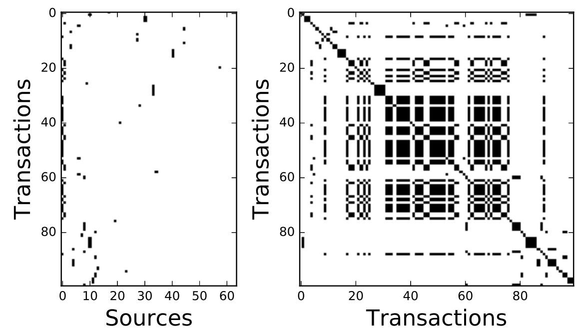


Figure 1: The network schema of EA payment transaction data.



(a) Sparse links $\mathbf{A}$. (b) A meta-path $\mathbf{P}$.

Figure 2: An example of computing a meta-path from sparse links.

2.2. Capturing Inter-Transaction Dependency

First, we briefly review the concept of *meta-path* following previous work Cao et al. (2014); Kong et al. (2013); Sun et al. (2011). In general, a meta-path corresponds to a type of path within the network schema, containing a certain sequence of link types. For example, in Figure 1, a meta-path "transaction $\xrightarrow{containsItem}$ item $\xrightarrow{isTitle}$ title $\xrightarrow{isTitle^{-1}}$ item $\xrightarrow{containsItem^{-1}}$ transaction" denotes a composite relation between transactions where $containsItem^{-1}$ represents the inverted relation of *containsItem*. The semantic meaning of this meta-path is that transactions contain items that belong to the same game title. Different meta-paths usually represent different semantic meanings between linked nodes. In this manner, various relationships among transactions can be described by a set of meta-paths. By capturing such inter-transaction dependency and aggregating the label information of the linked transactions, we could better detect correlated fraudulent behaviors. In other words, we could identify transactions with highly risky values in a categorical variable, *e.g.*, game title. Intuitively, when a recent launch of FIFA attracts many frauds to it, transactions that contain FIFA would be more suspicious than others. Similarly, transactions that are related to user accounts, billing accounts, IP addresses and currencies with high risk could also be identified.

The implementation of meta-paths is essentially a chain of matrix multiplications. Let's denote a meta-path as $\mathcal{P} =< \mathcal{E}^{k_1}, \cdots, \mathcal{E}^{k_l} >$ where the source node of $\mathcal{E}^{k_1}$ is of type s and the target node of $\mathcal{E}^{k_l}$ is of type t. The semantic meaning of this meta-path is mathematically described as $\mathbf{P} = \mathbf{A}^{k_1} \times \cdots \times \mathbf{A}^{k_l} \in \mathbb{R}^{n_s \times n_t}$. It is usually assumed that the strength of connection between v_p^s and v_q^t on such semantics is positively correlated with $\mathbf{P}[p, q]$, because $\mathbf{P}[p, q]$ is the (weighted) count of paths connecting v_p^s and v_q^t that follow the sequence of links in $\mathcal{P}$. Hereinafter, $\mathcal{P}$ and $\mathbf{P}$ will be used interchangeably when the meaning is clear from context.

2.3. Efficient Computation of Meta-Paths

Cardinality, which refers to the maximum number of times a node of the source node type can be linked with nodes of the target node type, is represented by the styling of a line and its endpoint in Figure 1. The notation style is similar to entity-relationship diagrams (ERDs) where a Crow's foot shows many-to-one relationship.

Let's consider a meta-path "transaction $\xrightarrow{fromSource}$ source $\xrightarrow{fromSource^{-1}}$ transaction" and denote it as $\mathcal{P} =< \mathcal{E}^{k_1}, \mathcal{E}^{k_2} >$ where $\mathcal{E}^{k_1}$ is the set of links indicating which source a transaction is from, and $\mathcal{E}^{k_2}$ is its inverted relation. The adjacency matrix of $\mathcal{E}^{k_1}$ is denoted as $\mathbf{A} \in \mathbb{R}^{n \times n_s}$ where $n = 130K$ is the number of transactions and n_s is the number of sources, then that of $\mathcal{E}^{k_2}$ is $\mathbf{A}^T$, and $\mathbf{P} = \mathbf{A} \times \mathbf{A}^T$. Because each transaction is conducted on one of EA game stores, $\mathcal{E}^{k_1}$ here is a many-to-one relation between transactions and sources. That is to say, $\mathbf{A}$ is extremely sparse with one value per row, and its sparsity ratio is $1 - 1/n_s > 98\%$.

The adjacency matrix $\mathbf{A}$ is shown in Figure 2 for 100 randomly sampled transactions, as well as the the meta-path $\mathbf{P} = \mathbf{A} \times \mathbf{A}^T$. As we can see, through the direct application of matrix chain multiplication, the computation of a meta-path almost turns a sparse adjacency matrix $\mathbf{A}$ into a full matrix $\mathbf{P}$. Moreover, there are a lot of redundancy in $\mathbf{P}$. For this particular meta-path, if i-th transaction and the j-th transaction are from the same source k, *i.e.*, $\mathbf{A}[i,k] = \mathbf{A}[j,k] = 1$, they have exactly the same row and column in $\mathbf{P}$, *i.e.*, $\mathbf{P}[i,:] = \mathbf{P}[j,:] = \mathbf{P}[:,i] = \mathbf{P}[:,j]$.

Fortunately, we are not interested in the concrete form of a meta-path itself. For the purpose of obtaining features for fraud detection on transactions, the meta-paths that we need to compute should have the same source node type and target node type which is a transaction. Because each transaction may be linked with different number of transactions through a meta-path, aggregation functions are employed to combine the label information of linked transactions in order to derive a fixed number of meta-path based features. For example, we can use the weighted label fraction of linked transactions as the feature $\mathbf{z} \in \mathbb{R}^n$ for each meta-path Kong et al. (2013, 2012). It is formulated as follows:

$$\mathbf{z} = \mathbf{D} \times \mathbf{P} \times \mathbf{y} \tag{1}$$

where $\mathbf{D} \in \mathbb{R}^{n \times n}$ is a diagonal matrix and $\mathbf{D}[i,i] = 1/\sum_j \mathbf{P}[i,j]$. In this manner, z_i indicates the ratio of being frauds among transactions that are connected with the i-transaction through the meta-path. Note that in the supervised learning setting, the ground-truth label information in $\mathbf{y}$ of test transactions are unknown or withheld, and they can be initialized and updated through collective prediction.

2.3.1. Many-to-One Cases

Next, we will introduce how to obtain the feature $\mathbf{z}$ without explicitly computing the meta-path $\mathbf{P}$. Since the concept of meta-path is defined as a sequence of link types, it can also be considered as a sequence of node types that are endpoints of these links (different link sequences can correspond to the same node sequence though), say, $\mathcal{P} =< \mathcal{V}^{k_0}, \cdots, \mathcal{V}^{k_l} >$. We have $k_0 = k_l = 1$ since we only consider meta-paths whose source node type and target node type are transaction. It is assumed that transaction is always the node set of the largest size, *i.e.*, $\mathrm{argmax}_i\{|\mathcal{V}^i||\mathcal{V}^i \in \mathcal{P}\} = 1$, and t denotes the node set of the smallest size in $\mathcal{P}$, *i.e.*, $\text{argmin}_i\{|\mathcal{V}^i||\mathcal{V}^i \in \mathcal{P}\} = t$. Therefore, we can decompose a meta-path $\mathcal{P}$ into two parts at the node type t, and its matrix form can be written as $\mathbf{P} = \mathbf{P}_1 \times \mathbf{P}_2^T$ where $\mathbf{P}_1, \mathbf{P}_2 \in \mathbb{R}^{n \times n_t}$.

Definition 1 (Simple Meta-path) *A meta-path is a simple meta-path if it is a sequence of many-to-one relations.*

Note that a simple meta-path itself is a (composite) many-to-one relation. For example, "transaction $\xrightarrow{byBilling}$ billing $\xrightarrow{isAccount}$ account $\xrightarrow{isType}$ type" is a simple meta-path. Most meta-paths are a concatenation of a simple meta-path and another meta-path, and their features can efficiently be computed.

Lemma 1 *Given a meta-path $\mathbf{P} = \mathbf{P}_1 \times \mathbf{P}_2^T$ where $\mathbf{P}_1$ is a simple meta-path, the computation of Eq. (1) on $\mathbf{P}$ can be reduced to:*

$$\mathbf{z} = \mathbf{P}_1 \times (\mathbf{D}_2 \times \mathbf{P}_2^T \times \mathbf{y}) \tag{2}$$

where $\mathbf{D}_2 \in \mathbb{R}^{n_t \times n_t}$ is a diagonal matrix and $\mathbf{D}_2[i,i] = 1/\sum_j \mathbf{P}_2[j,i]$.

Proof 1 *The detailed proof is presented in Cao et al. (2017).*

Eq. (2) is important because it enables us to obtain the weighted label fraction of linked nodes without extensive matrix operations. In this manner, we can effectively avoid computing the redundant full matrix of a meta-path as an intermediate result.

2.3.2. Many-to-Many Cases

Let's consider the meta-path "transaction $\xrightarrow{containsItem}$ item $\xrightarrow{isTitle}$ title $\xrightarrow{isTitle^{-1}}$ item $\xrightarrow{containsItem^{-1}}$ transaction". It involves a many-to-many link type, *i.e.*, "transaction $\xrightarrow{containsItem}$ item", in Figure 1.

Definition 2 (Complex Meta-path) *A meta-path is a complex meta-path if it contains at least one many-to-many relation.*

Note that a complex meta-path is "usually" a (composite) many-to-many relation. Now, we dichotomize all meta-paths into simple meta-paths and complex meta-paths. Given any nontrivial meta-path ($l > 1$), we still decompose it into two parts at the node type of the smallest size, $\mathbf{P} = \mathbf{P}_1 \times \mathbf{P}_2^T$, and we propose to compute its features as follows:

$$\mathbf{z} = \mathbf{D}_1 \times \mathbf{P}_1 \times (\mathbf{D}_2 \times \mathbf{P}_2^T \times \mathbf{y}) \tag{3}$$

where $\mathbf{D}_1 \in \mathbb{R}^{n \times n}$ is a diagonal matrix and $\mathbf{D}_1[i,i] = 1/\sum_j \mathbf{P}_1[i,j]$. Obviously, $\mathbf{D}_1 = \mathbf{I}_n$ where $\mathbf{I}_n$ is an identity matrix when $\mathbf{P}_1$ is a simple meta-path. Therefore, Eq. (2) is a special case of Eq. (3) in the many-to-one scenarios.

However, Eq. (3) is not equivalent to Eq. (1) or Eq. (2) when $\mathbf{P}_1$ is a complex meta-path. For example, $\mathcal{P} =$"transaction $\xrightarrow{containsItem}$ item $\xrightarrow{isTitle}$ title $\xrightarrow{isTitle^{-1}}$ item $\xrightarrow{containsItem^{-1}}$ transaction" can be decomposed as $\mathcal{P}_1 = \mathcal{P}_2 =$"transaction $\xrightarrow{containsItem}$ item $\xrightarrow{isTitle}$ title"

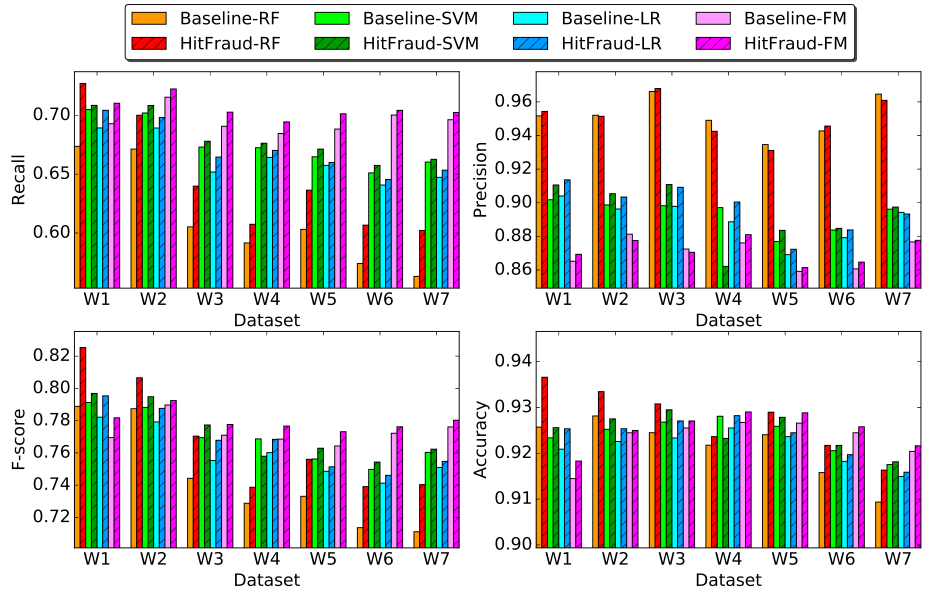


Figure 3: Prediction performance.

where $\mathcal{P}_1$ is a complex meta-path. In Eq. (1), we compute the average fraction of frauds in the linked transactions and each transaction is weighted by the number of common game titles in the current transaction. It does not, however, distinguish which game titles are shared. In other words, each shared game title is counted equally. In contrast, Eq. (3) counts a shared rare title more than a shared popular title because $\mathbf{D}_2$ accounts for a normalization step at the title level. The similarity between two transactions increases proportionally to the number of titles they share, but is offset by the popularity of the title.

2.4. Collective Fraud Detection

So far we have explored the inter-transaction dependency to capture the correlated fraudulent behaviors where the correlations mainly exist between training transactions and test transactions. We further notice that fraudulent behaviors are fast evolving. For example, a batch of new transactions may be made by the same new billing account but with different IP addresses, some of which are rather risky and others might be proxy. It is desirable to mark all these transactions as suspicious.

The inference problem for collective fraud detection in a HIN is to learn a predictive function $f : (\mathcal{V}, \mathcal{E}, \mathbf{X}) \rightarrow \mathbf{y}$. Conventional classification approaches usually make an independent and identically distributed (i.i.d.) assumption, and thus the probability of each transaction being a fraud is inferred independently as $f(\mathbf{x}_i) \propto \Pr(y_i = 1|\mathbf{x}_i)$. In addition to the given features $\mathbf{X}$, we include the meta-path based features $\left\{\mathbf{z}^1, \cdots, \mathbf{z}^c\right\}$ where c is the number of extracted meta-paths. Therefore, the target is to learn

$$f(\mathbf{x}_i) \propto \Pr(y_i = 1|\mathbf{x}_i, z_i^1, \cdots, z_i^c) \tag{4}$$

In this manner, however, the inference of different transactions is essentially not independent, because meta-path based features contain the (predicted) label information of linked transactions in both training and test sets. It can be done in an iterative framework where the label of a transaction is inferred based on the labels of its linked transactions through its meta-path based features, and its predicted label will further be used to infer the labels of its linked transactions by updating their meta-path based features. It is similar to the framework of Heterogeneous Collective Classification (HCC) Kong et al. (2012), and we improve it with a more efficient way of computing meta-path based features.

Definition 3 (Downsized Meta-path) *Given the node sequence of a meta-path* $\mathcal{P} =< \mathcal{V}^{k_0}, \cdots, \mathcal{V}^{k_l} >$*, it is a downsized meta-path if* $n_{k_0} > \cdots > n_{k_l}$.

It is assumed that the meta-paths of interest in this work can be decomposed as two coupled downsized meta-paths which however is not true for all meta-paths. We design the process of meta-path exploration based on discussions in the last section which can be summarized into two facts: (1) Meta-paths that are used for feature computation in our task always start from and end at transaction nodes. (2) Each meta-path can be decomposed into two parts at the node type of the smallest size. Therefore, we can perform a breadth-first-search from transaction nodes to find all downsized meta-paths from $\mathcal{V}^1$ to each other node type. In the search procedure, say, the current meta-path $\mathbf{P}$ is from $\mathcal{V}^1$ to $\mathcal{V}^i$, we enumerate link types "$\mathcal{V}^i \xrightarrow{\mathcal{E}^k} \mathcal{V}^j$" in the network schema. If $|\mathcal{V}^i| > |\mathcal{V}^j|$, a new meta-path $\mathbf{P'} = \mathbf{P} \times \mathbf{A}^k$ is added into $\mathcal{S}_j$. Search will be expanded from the newly added meta-paths until all downsized meta-paths from $\mathcal{V}^1$ to $\mathcal{V}^i$ have been included in $\mathcal{S}_i$.

3. Results

3.1. Experimental Setup

Experiments are conducted on EA payment transaction data that contains $n = 130K$ transactions with manual review labels. For training and evaluation purposes, we segment the data into two consecutive parts so that one-week data is used for testing models and the preceding weeks are used for training. Based on sliding windows, 7 data segmentations are created, each of which is denoted as *W1* to *W7*. Four metrics are reported: recall, precision, F-score and accuracy.

3.2. Effectiveness with Different Classifiers

One claim of this paper is that HitFraud can work well in conjunction with a variety of base classifiers. To evaluate this claim, we conduct experiments using various base classifiers, including random forest (RF), support vector machines (SVM), logistic regression (LR), and factorization machines (FM). The implementations of these base classifiers from `scikit-learn`[1] and `fastFM`[2] are used with default hyperparameter configurations. The baselines are the same base classifiers that explore only the given feature space $\mathbf{X}$. Figure 3 shows the prediction performance comparing HitFraud to the baselines. We can observe that, by conducting collective fraud detection, HitFraud is usually able to outperform the baselines with multiple choices of base classifiers, on multiple datasets, in multiple

1. http://scikit-learn.org
2. https://github.com/ibayer/fastFM

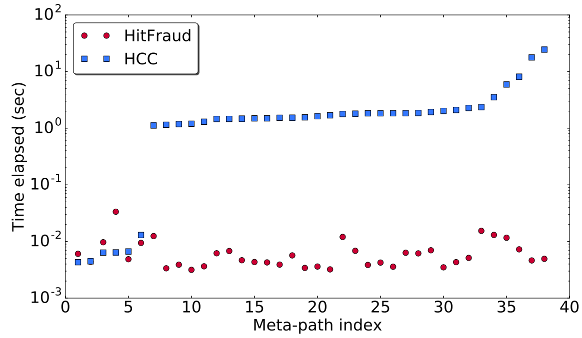


Figure 4: Computation cost of meta-paths.

evaluation metrics. For example, with random forest on the *W1* dataset, recall is boosted 7.93% from 0.6737 to 0.7271 ($p = 0.000$), precision 0.28% from 0.9517 to 0.9543 ($p = 0.218$), F-score 4.62% from 0.7889 to 0.8253 ($p = 0.000$), and accuracy 1.17% from 0.9257 to 0.9366 ($p = 0.003$). According to these raw p-values, recall, F-score and accuracy are all significant, although precision is non-significant. In few cases, precision is sacrificed to boost recall and the overall F-score. However, precision and recall can be improved at the same time by HitFraud in most cases. In general, it demonstrates that HitFraud is flexible and effective in conjunction with a diversity of underlying classification algorithms, and the inter-transaction dependency can indeed be explored to more effectively capture fraud payment transactions.

3.3. Efficiency in Meta-Path Computation

There is much redundancy in the plain-vanilla computation of meta-path based features which aggravates not only the time cost but also the memory consumption. There are in total $c = 38$ meta-paths explored in this work, and some examples of meta-paths are shown in Cao et al. (2017). Figure 4 compares the time cost of computing each meta-path based feature between the approach proposed in Section 2.3 for HitFraud and HCC presented in Kong et al. (2012). The comparison of memory cost is omitted due to space limit whose trend is similar to Figure 4. Note that this experiment is conducted on a network constructed from one-week data, because the time and space cost for HCC is formidable on the whole dataset. We can observe that the discrepancy of time cost is significant even on the log scale. For a few cases where HitFraud and HCC take almost the same time, those are meta-paths that involve nearly one-to-one relation where redundancy is not very severe, *e.g.*, "transaction $\xrightarrow{byUser}$ user $\xrightarrow{byUser^{-1}}$ transaction". Obviously, it is not very likely that many users would place multiple transactions within one week. In general, the time complexity of discovering relevant network features in HitFraud is linear to the number of transactions and insensitive to the type of relationships.

4. Conclusions

In this paper, we propose HitFraud, a collective fraud detection algorithm that captures the inter-transaction dependency. Meta-path based features are efficiently computed through the use of pre-computing downsized meta-paths. Suspicious transactions in the test set are collectively identified when they share common fraudulent behaviors. Experiments on EA payment transaction data demonstrate that the prediction performance is effectively boosted by HitFraud with different choices of base classifiers. It is validated that the correlated and fast evolving fraudulent behaviors can indeed be explored to more effectively capture fraud payment transactions.

5. Acknowledgements

This work is supported in part by NSF through grants IIS-1526499, and CNS-1626432, and NSFC 61672313. We also thank Victor Chen, Shuang Chen, Ying Wang, Frank Yang, Alexandru Doroftei, Marco Sederquist, Thilmin Gee, and Helen Sulic for insightful discussions and valuable feedbacks.

References

Bokai Cao, Xiangnan Kong, and Philip S Yu. Collective prediction of multiple types of links in heterogeneous information networks. In *ICDM*, pages 50–59, 2014.

Bokai Cao, Mia Mao, Siim Viidu, and Philip S Yu. HitFraud: A broad learning approach for collective fraud detection in heterogeneous information networks. *arXiv*, 2017.

Zoltán Gyöngyi, Hector Garcia-Molina, and Jan Pedersen. Combating web spam with trustrank. In *VLDB*, pages 576–587, 2004.

Bryan Hooi, Hyun Ah Song, Alex Beutel, Neil Shah, Kijung Shin, and Christos Faloutsos. Fraudar: Bounding graph fraud in the face of camouflage. In *KDD*, pages 895–904, 2016.

Ming Ji, Jiawei Han, and Marina Danilevsky. Ranking-based classification of heterogeneous information networks. In *KDD*, pages 1298–1306, 2011.

Meng Jiang, Peng Cui, Alex Beutel, Christos Faloutsos, and Shiqiang Yang. Catchsync: catching synchronized behavior in large directed graphs. In *KDD*, pages 941–950, 2014.

Xiangnan Kong, Philip S Yu, Ying Ding, and David J Wild. Meta path-based collective classification in heterogeneous information networks. In *CIKM*, pages 1567–1571, 2012.

Xiangnan Kong, Bokai Cao, and Philip S Yu. Multi-label classification by mining label and instance correlations from heterogeneous information networks. In *KDD*, pages 614–622, 2013.

Yizhou Sun, Yintao Yu, and Jiawei Han. Ranking-based clustering of heterogeneous information networks with star network schema. In *KDD*, pages 797–806, 2009.

Yizhou Sun, Jiawei Han, Xifeng Yan, Philip S Yu, and Tianyi Wu. Pathsim: Meta path-based top-k similarity search in heterogeneous information networks. In *VLDB*, 2011.