Modeling the Field Value Variations and Field Interactions Simultaneously for Fraud Detection

Dongbo Xi,[1,2,3] **Bowen Song,**[3,*] **Fuzhen Zhuang,**[1,2,*] **Yongchun Zhu,**[1,2,3]
Shuai Chen,[3] **Tianyi Zhang,**[3] **Yuan Qi,**[3] **Qing He**[1,2]
[1] Key Lab of Intelligent Information Processing of Chinese Academy of Sciences (CAS),
Institute of Computing Technology, CAS, Beijing 100190, China
[2] University of Chinese Academy of Sciences, Beijing 100049, China
[3] Alipay (Hangzhou) Information & Technology Co., Ltd.
{xidongbo17s, zhuangfuzhen, zhuyongchun18s, heqing}@ict.ac.cn,
{bowen.sbw, shuai.cs, zty113091, yuan.qi}@antfin.com

Abstract

With the explosive growth of e-payment industry, online transaction fraud has become one of the biggest challenges for the business. The historical behavior information of users provides rich information for digging into the users' fraud risk. While considerable efforts have been made in this direction, a long-standing challenge is how to effectively exploit user's behavioral information and provide explainable prediction results. In fact, the value variations of same field from different events and the interactions of different fields within one event have proven to be strong indicators of fraudulent behaviors. In this paper, we propose the Dual Importance-aware Factorization Machines (DIFM), which exploits the inter- and intra-event information among users' behavior sequence from dual perspectives, i.e., field value variations and field interactions simultaneously for fraud detection. The proposed model is deployed in Alipay's risk management system, which provides real-time fraud detection service for e-commerce platforms. Experimental results on industrial data under various scenarios in the platform clearly demonstrate that our model achieves significant improvements compared with various state-of-the-art baseline models. Moreover, the DIFM could also give an insight into the explanation of the prediction results from dual perspectives.

Introduction

With the rapid development of e-commerce and e-payment, the problem of online transaction fraud has become increasingly prominent (Cao et al. 2019). As an international Fintech company, Alipay provides e-payment service for many e-commerce platforms, on which millions of transactions occurred each day. A very small portion of fraudulent transactions could easily lead to huge financial loss and introduce great security risk to our business.

Therefore, detecting fraudulent risk in real-time, has become a key factor in determining the security and success of e-payment business. Recently, considerable efforts have shifted from rule-based expert system (Cohen 1995; Brause, Langsdorf, and Hepp 1999; Rosset et al. 1999; Baulier et al. 2000; Stefano and Gisella 2001; Pathak, Vidyarthi, and Summers 2005) to neural network-based models (Fu et al. 2016; Wang et al. 2017b; Zhang, Zheng, and Min 2018; Jurgovsky et al. 2018; Cao et al. 2019; Liang et al. 2019; Xi et al. 2020; Zhu et al. 2020) for fraud detection tasks. The historical behavior events (activities) of users provide rich information for digging into the users' fraud risk as shown in Figure 1. However, due to the limitation of model structures, it is difficult for the above methods to exploit the internal field information thoroughly among events (e.g., the field value variations among the historical events or field interactions inside each event) or to provide explainable prediction results, which is foundational key in fraud detection tasks.

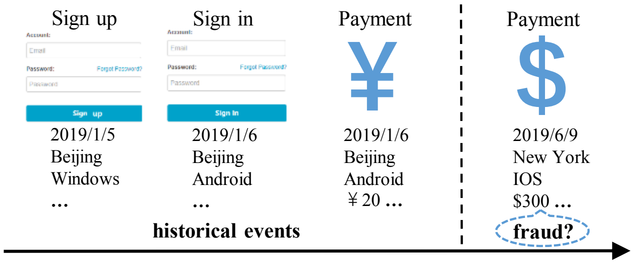


Figure 1: A typical fraud detection task which exploits the user's historical operation events information to determine the fraudulent risk of target payment event.

In this paper, we propose the *Dual Importance-aware Factorization Machines* (DIFM) to more efficiently take advantage of internal field information among events. It could not only improve the performance of fraud detection but also provide explainable prediction results. DIFM captures internal field information from dual perspectives: 1) The **Field Value Variations Perspective** captures the value variations of each field between any two events, and the corresponding Field Importance-aware module perceives the importance of different field value variations. 2) The **Field Interactions Perspective** models the interactions between all fields within each event, and the corresponding Event Importance-aware module perceives the importance of different histor-

[*]Corresponding authors: Fuzhen Zhuang and Bowen Song.

ical events. Besides, the "wide" layer of DIFM can help to assess the fraudulent related risk-level of each field and therefore work as blacklist/whitelist to screen out the fraudulent/good users.

To summarize, the contributions are listed as follows:

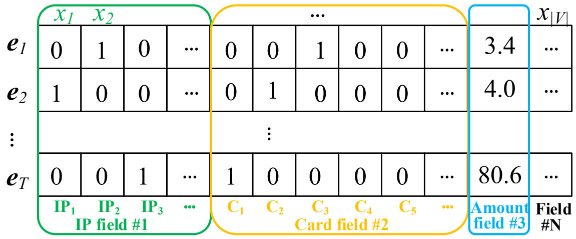
- DIFM effectively and sufficiently exploits both inter- and intra-event information from the field value variations perspective and field interactions perspective simultaneously.
- DIFM perceives the importance of values in field-level and event-level from dual perspectives simultaneously, which could provide explainable prediction results.
- Experimental results on industrial data from different trading scenarios clearly demonstrate that the proposed DIFM model obtains a remarkable improvement comparing to existing state-of-the-art approaches.

Related Work

In this section, we present the related work in two-fold: (1) feature interactions and sequence prediction, (2) fraud detection.

Simply using raw features could rarely yield optimal results in prediction tasks, therefore more information need to be mined. One way is to learn feature interactions from raw data to generate efficient feature representations. Data scientists made a lot of effort to derive efficient feature interactions from raw data (also known as feature engineering) to obtain better prediction performance (Cheng et al. 2016; Lian et al. 2017). Instead of generating feature interactions manually, a solution (Wang et al. 2017a) has been proposed to learn feature interactions automatically from the raw data. Factorization Machine (FM) (Rendle 2010) is a widely used method to model second-order feature interactions automatically via the inner product of raw embedding vectors. Other efforts have also been made to combine the advantages of FM on modeling second-order feature interactions and neural network on modeling higher-order feature interactions (Zhang, Du, and Wang 2016; Qu et al. 2016; He and Chua 2017; Xiao et al. 2017; Guo et al. 2017; Lian et al. 2018). However, in the case of sequence prediction, simply considering the interaction among fields is not sufficient, since such methods only capture the interaction among features regardless of the event they belong to, which results in ignoring the variation of value along the temporal dimension. They do not consider field-in-event-relations and field-between-event-relations, which we believe is of fundamental importance for fraud detection tasks. Other methods have also attempted to take the user's historical event information into consideration (Hidasi et al. 2016; Quadrana et al. 2017; Tang and Wang 2018; Kang and McAuley 2018; Beutel et al. 2018; Ma, Kang, and Liu 2019; Rakkappan and Rajan 2019; Chen et al. 2019; Zhou et al. 2019; Tang et al. 2019). Nevertheless, most of these studies mainly focused on the event sequence but ignored the intra-field field information of historical events.

Fraud detection is one of the most significant applications in e-payment business, early researchers mainly focused on rule-based expert system (Cohen 1995; Brause, Langsdorf, and Hepp 1999; Rosset et al. 1999; Baulier et al. 2000; Stefano and Gisella 2001; Pathak, Vidyarthi, and Summers 2005). There are different types of fraud, for example, credit card fraud (Brause, Langsdorf, and Hepp 1999), telephone fraud (Rosset et al. 1999), insurance fraud (Stefano and Gisella 2001) and so on. With the rapid evolution of fraudster's behavior patterns, only human-summarized rules or expert knowledge are not sufficient to meet the demand of today's real-time fraud detection (Cao et al. 2019). Researchers have attempted to use machine-learning based methods (Tian et al. 2015; Tseng et al. 2015; Fu et al. 2016; Wang et al. 2017b; Zhang, Zheng, and Min 2018; Jurgovsky et al. 2018; Cao et al. 2019; Liang et al. 2019) to detect fraud. Fu et al. (2016) focused on Convolutional Neural Network (CNN) for credit card fraud detection (Fu et al. 2016). Other works utilized Recurrent Neural Networks (RNN) for sequence-based fraud detection (Wang et al. 2017b; Jurgovsky et al. 2018; Zhang, Zheng, and Min 2018). Liang et al. (2019) utilized Graph Neural Network (GNN) to target frauds (Liang et al. 2019). However, most of these methods suffer from the same problem: lack of explainability, which is crucial for fraud detection tasks. In this paper, we propose DIFM model, which could not only exploit the internal field information more thoroughly among events from dual perspective but also give insight into the explanation of the prediction results.



	IP_1	IP_2	IP_3	···	C_1	C_2	C_3	C_4	C_5	···	Amount field #3	Field #N
$\boldsymbol{e}_1$	0	1	0	···	0	0	1	0	0	···	3.4	···
$\boldsymbol{e}_2$	1	0	0	···	0	1	0	0	0	···	4.0	···
⋮							⋮					
$\boldsymbol{e}_T$	0	0	1	···	1	0	0	0	0	···	80.6	···

Figure 2: An example of operation events set $\boldsymbol{E}$.

Methodology

In the following section, we first formulate the problem, then present the details of the proposed DIFM model.

Problem Statement

A simple example of a user's operation events $\boldsymbol{E}$ is shown in Figure 2. Given a user's operation events $\boldsymbol{E} = [\boldsymbol{e}_1, \boldsymbol{e}_2, ..., \boldsymbol{e}_T]$, where T is the number of the events. Each field in a single event despicts certain operation information with the event, e.g. IP city field or payment amount field, and the number of fields for each event is N. Each field could have one or more candidate values (e.g., in Figure 2, the IP field has the values of IP_1, IP_2, IP_3 and so on). $\boldsymbol{e}_t = [x_1^t, x_2^t, ..., x_{|V|}^t]$ $(1 \leq t \leq T)$ is the t-th event of the user in $\boldsymbol{E}$, where $|V|$ is the number of all field values (i.e., the dictionary size of all fields). For categorical fields (e.g., the IP field), x_i^t $(1 \leq i \leq |V|)$ is 1 if $\boldsymbol{e}_t$ has the value in the current categorical field, otherwise is 0. For numerical fields (e.g., the Amount field), x_i^t adopts the real value as its value. Our task is to make a prediction for the current payment event $\boldsymbol{e}_T$ according to the user's historical operation events $[\boldsymbol{e}_1, \boldsymbol{e}_2, ..., \boldsymbol{e}_{T-1}]$ and the available information of the current payment event $\boldsymbol{e}_T$. The task could then be formulated as Classification, Regression or Pairwise Ranking depending on which activation/loss function they use under real application scenarios as described in (Rendle 2010).

Factorization Machines and Importance-aware Module

In this subsection, we first describe two basic modules of the DIFM, which are the Factorization Machines and Importance-aware Module, respectively.

Factorization Machines As shown in Figure 2, there are rich features in each event which describe the event details in real application scenarios. Since most of the features are one-hot encoding categorical features, the dimension is usually high and the vectors are sparse. FM is an effective method to address such high-dimension and sparse problems, and it can be seen as performing the latent relation modeling between any two field values of the same event, e.g., the field value variations or field interactions.

Firstly, we project each non-zero field value x_i to a low dimension dense vector representation $\boldsymbol{v}_i$. Embedding layer is a popular solution in the neural network over various application scenarios. It learns one embedding vector $\boldsymbol{v}_i \in \mathbb{R}^k$ $(1 \leq i \leq |V|)$ for each field value x_i. where k is the dimension of embedding vectors. For both categorical and numerical features, we rescale the look-up table embedding via $x_i\boldsymbol{v}_i$ as done in (He and Chua 2017). Therefore, we only need to include the non-zero features, i.e., $x_i \neq 0$.

Different from traditional FM, which uses inner product to get a scalar, to preserve sufficient information, a vector representation is generated using Hadamard product as done in (He and Chua 2017):

$$FM(\boldsymbol{x}) = \sum_{i \neq j} x_i \boldsymbol{v}_i \odot x_j \boldsymbol{v}_j. \qquad (1)$$

Hadamard product $\odot$ denotes the element-wise product of two vectors: $(\boldsymbol{v}_i \odot \boldsymbol{v}_j)_k = v_{ik}v_{jk}$. The computing complexity of the above Equation (1) is $O(k|V|^2)$, since all pairwise relations need to be computed. Actually, the Hadamard product based FM can be reformulated to linear runtime $O(k|V|)$ (He and Chua 2017) just like the inner product based FM (Rendle 2010):

$$FM(\boldsymbol{x}) = \frac{1}{2}\left[(\sum_i x_i\boldsymbol{v}_i)^2 - \sum_i (x_i\boldsymbol{v}_i)^2\right], \qquad (2)$$

where $\boldsymbol{v}^2$ denotes $\boldsymbol{v} \odot \boldsymbol{v}$. Besides, in sparse settings, the sums only need to be computed over the non-zero pairs x_ix_j. Therefore, the actual computing complexity is $O(k|\hat{V}|)$, where $|\hat{V}|$ is the number of non-zero entries in $\boldsymbol{x}$.

Importance-aware Module In fraud detection tasks, fields and events often play different roles, which indicates different importance for the detection task. For field importance, IP address changing or amount changing over a short period tends to indicate higher risk than value-stable field, therefore we should pay more attention to IP/amount field if they have such pattern; for event importance, if an event has multiple abnormal field values, the event is more important than other normal events.

In order to perceive the relative importance of different fields and events, we design an Importance-aware Module. We expect the model could pay more attention to important fields and events. Attention mechanism has been verified to be effective in machine translation (Bahdanau, Cho, and Bengio 2015), where the attention makes the model focus on useful features for the current task. Inspired by the great success of self-attention in machine translation (Vaswani et al. 2017), we design the self-attention-like Importance-aware Module to learn the importance of different fields and events. The key advantage of the Importance-aware Module is that it can perceive the importance of different fields and events for each user. For a vector set $\boldsymbol{FM} = \{\boldsymbol{FM}_1, ..., \boldsymbol{FM}_m, ...\}$ of different fields or events captured with the FM, the importance weight is defined as scaled dot-product:

$$\hat{a}_m = \frac{< F_1(\boldsymbol{FM}_m), F_2(\boldsymbol{FM}_m) >}{\sqrt{k}}, \qquad (3)$$

$$a_m = \frac{\exp(\hat{a}_m)}{\sum_m \exp(\hat{a}_m)}, \qquad (4)$$

and the output of the Importance-aware Module (IM) is represented as:

$$IM(\boldsymbol{FM}) = \sum_m a_m F_3(\boldsymbol{FM}_m), \qquad (5)$$

where $<, >$ denotes the dot-product and F_1, F_2, and F_3 represent the feed-forward networks to learn for projecting the input vector to one new vector representation space. It is worth noting that we find using multiple feed-forward networks in the designed attention module is more effective than using a single feed-forward network as adopted in (Xiao et al. 2017; Zhou et al. 2018).

Field Value Variations Perspective

In this subsection, we model the field value variations with the above two modules.

For hacked account, it would be super-expensive to mimic the real account owner's complete environment information, so field value variations (e.g., the IP changing) in different events tends to indicate higher risk than stable field.

As mentioned above, FM can help capture the field value variations information, and therefore we apply FM module with our proposed model. For the n-th field $\boldsymbol{f}_n$, we calculate the field value variations among all T events as the brown box shown in Figure 3:

$$\boldsymbol{f}_n = FM(\boldsymbol{x}^n) = \sum_{i=1}^{T-1}\sum_{j=i+1}^{T} x_i^n \boldsymbol{v}_i^n \odot x_j^n \boldsymbol{v}_j^n. \qquad (6)$$

By applying field value variations FM (i.e., Equation (6)) to each field along a user's historical operation events, we get the representations of all fields $\boldsymbol{F} = [\boldsymbol{f}_1, \boldsymbol{f}_2, ..., \boldsymbol{f}_N]$ thereafter for the Field Importance-aware Module.

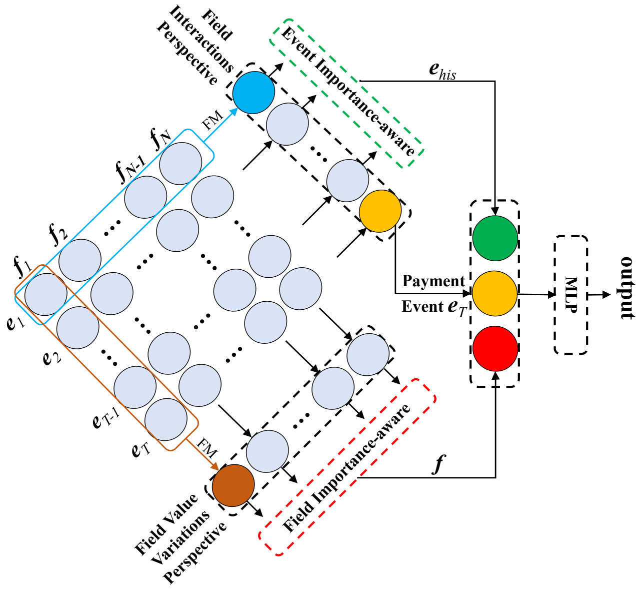


Figure 3: The proposed DIFM model, for simplicity, we do not represent the "wide" part.

In fraud detection, if a user's IP changes over a short period, the field value variation tends to indicate a higher risk. In order to perceive the relative importance of fields, for the vector set $\boldsymbol{F} = [\boldsymbol{f}_1, \boldsymbol{f}_2, ..., \boldsymbol{f}_N]$ of different fields captured with the FM, we apply the Field Importance-aware Module in Equation (5) as follows:

$$\boldsymbol{f} = IM(\boldsymbol{F}) = \sum_{n=1}^{N} a_n F_3(\boldsymbol{f}_n), \tag{7}$$

where a_n is the importance weight learned according to Equation (4). We hope the model could pay more attention to important fields and focus on useful features for the current task. As long as a user's field value variations relate to the fraud label, it will be captured by our model via the field value variations perspective.

Field Interactions Perspective

In this subsection, we capture the field interactions with the above two modules.

In fraud detection, a fraudster's behavior can often be detected by interaction of different features, since combination patterns usually have stronger relevance to fraud than single features. As mentioned above, FM can be seen as performing the field interactions between any two field values. However, simple calculation of interactions among all features is inefficient and will introduce noise to the model, since the interactions of fields between different events provide little info for the prediction. For example, fraudster logged in with normal $IP\#1$ in $event\#1$ and buy high-risk item of category $C\#2$ with high-risk $IP\#2$ in $event\#2$, the interaction between features $IP\#1$ and $C\#2$ provides little information, while the internal interaction in $event\#2$ (e.g., interaction between $C\#2$ and $IP\#2$) can capture the event representation better than any single features, and it will improve the prediction accuracy. Therefore, we capture the field interactions inside each event. For the t-th event $\boldsymbol{e}_t$, we perform the field interactions in all fields as the blue box shown in Figure 3 as follows:

$$\boldsymbol{e}_t = FM(\boldsymbol{x}^t) = \sum_{i=1}^{|V|-1} \sum_{j=i+1}^{|V|} x_i^t \boldsymbol{v}_i^t \odot x_j^t \boldsymbol{v}_j^t. \tag{8}$$

Thus, we can obtain an effective event representation. Some existing methods (Wang et al. 2017b; Zhang, Zheng, and Min 2018; Tang et al. 2019) can also obtain an event representation using a simple dense layer and the embedding concatenation, but they can not effectively extract the internal information of each behavior event. Now, we apply the field interactions FM (i.e., Equation (8)) to each event along a user's historical operations, and we get the representations of all events $\boldsymbol{E} = [\boldsymbol{e}_1, \boldsymbol{e}_2, ..., \boldsymbol{e}_T]$ thereafter for the Event Importance-aware Module. Among them, $\boldsymbol{e}_T$ is the user's current payment event, whose risk we are trying to model.

Besides, the user's final behavior activity is strongly correlated with the user's past several activities and each historical event of users might have different importance. For example, in card-stolen fraud detection scenario, if one event has multiple abnormal field values, the event is prone to have a higher risk than the normal event. Naturally, a good model should pay more attention to such abnormal events. In order to perceive the relative importance of different events, after we get the the vector set $\boldsymbol{E}_{his} = [\boldsymbol{e}_1, \boldsymbol{e}_2, ..., \boldsymbol{e}_{T-1}]$ by utilizing the mentioned FM above, we apply the Event Importance-aware Module in Equation (5) as follows:

$$\boldsymbol{e}_{his} = IM(\boldsymbol{E}_{his}) = \sum_{t=1}^{T-1} a_t F_3(\boldsymbol{e}_t), \tag{9}$$

where a_t is the importance weight learnt according to Equation (4).

Thus, the proposed DIFM can effectively exploit internal features and perceive the importance from dual perspectives simultaneously. Moreover, the Field and Event Importance-aware Module can provide explainable prediction results by indicating which fields or events are crucial for generating the risk score.

The final DIFM architecture is shown in Figure 3, we combine the field value variations feature $\boldsymbol{f}$ in Equation (7), the field interactions feature $\boldsymbol{e}_{his}$ in Equation (9) and the current prediction event $\boldsymbol{e}_T$ and feed them to an MLP. The output of the MLP is combined with a "wide" part (for simplicity, we do not represent this part in Figure 3), and then fed to the $sigmoid$ activation function to form the final DIFM, which seamlessly combines the field value variations and field interactions perspectives:

$$\begin{aligned} \boldsymbol{s} &= [\boldsymbol{f}; \boldsymbol{e}_{his}; \boldsymbol{e}_T], & (10)\\ \hat{y} &= sigmoid(MLP(\boldsymbol{s}) + f(\boldsymbol{x})), & (11) \end{aligned}$$

where $f(\boldsymbol{x})$ is the "wide" part just like the part in Wide&Deep (Cheng et al. 2016) and $\hat{y}$ indicates the probability of fraud. The $f(\boldsymbol{x})$ is defined as follows:

$$f(\boldsymbol{x}) = \sum_{t=1}^{T} \sum_{i=1}^{|V|} w_i x_i^t + w_0, \tag{12}$$

Dataset	#pos	#neg	#fields	#events
C1	15K	1.37M	56	4.28M
C2	10K	1.93M	56	3.57M
C3	5.7K	174K	56	353K

Table 1: Summary statistics for the datasets.

where w_i scores the importance of the field value x_i, which can indicate high-risk or low-risk for directly using in blacklist or whitelist.

For classification tasks, we need to minimize the *cross entropy* loss:

$$L(\theta) = -\frac{1}{S}\sum_{(\boldsymbol{x},y)\in\mathcal{D}}^{S}\left((y\log\hat{y} + (1-y)\log(1-\hat{y})\right), \quad (13)$$

where S is the number of samples, y is the label of sample $\boldsymbol{x}$ and θ is the parameter set, which contains the embedding vector $\boldsymbol{v}_i$, the weight w_i in the "wide" part, and the parameters in $F1, F2, F3$ and MLP.

The model is implemented using Tensorflow and trained through stochastic gradient descent over shuffled mini-batches with the Adam (Kingma and Ba 2014) update rule.

Experiments

In this section, we perform experiments to evaluate the proposed DIFM model against state-of-the-art methods on real industrial datasets. Below, we will introduce the datasets, baseline methods, implementation details and evaluation metrics of our experiments. In the end, we present our experimental results and further analysis.

Datasets

The statistics of all datasets used are shown in Table 1. The datasets contain the card (debit card or credit card) transaction samples from one international e-commerce platform, which utilizes a risk management system to detect the transaction frauds in real-time, i.e., card-stolen cases. We utilize user's historical behavior activity events of the last month in three different trading regions, i.e. countries from Southeast Asia (**C1**, **C2**, **C3**), which consist of 6 event types (i.e., sign up, sign in, digital goods payment, regular goods payment, information modification and card binding). And for all events, the following fields are used in our analysis: IP-information, shipping-information, billing-information, card-information, item-category, operation-result, user-account-information, device-information, etc. The task is to detect whether the current payment event is a card-stolen case. The fraud labels are from the chargeback reports from card-issuing banks (e.g., the card issuer receives claims on unauthorized charges from the cardholders and reports related transaction frauds to the merchants) and label propagation (e.g., the device and card information are also utilized to mark similar transactions). We take the last 10% in chronological order as the validation set in each dataset to verify the convergence of the model, and in order to verify the generalization ability, only the hyper-parameters are tuned on the same C1 validation set.

Baselines

We compare the proposed method with the following competitive and mainstream models which contain feature interactions based models (W&D, DeepFM, NFM, AFM, xDeepFM) and event sequence based models (LSTM4FD, LCRNN, M3). For all feature interactions based models, we use all the features of the user's events as the input:

- **W&D** (Cheng et al. 2016): It consists of "wide" and "Deep" parts, where the "wide" part is a linear model and the "Deep" part is an MLP.
- **DeepFM** (Guo et al. 2017): FM and MLP are combined in this model and fed to the output layer in parallel.
- **NFM** (He and Chua 2017): This is a simple and efficient neural factorization machine model whose FM is fed to MLP for capturing higher-order feature interactions.
- **AFM** (Xiao et al. 2017): It adds the attention mechanism to the FM to consider the importance of different pairs.
- **xDeepFM** (Lian et al. 2018): This is the state-of-the-art feature interactions based model which attempts to learn higher-order feature interactions explicitly.
- **LSTM4FD** (Wang et al. 2017b; Zhang, Zheng, and Min 2018): These works have applied LSTM for fraud detection task, and we call these methods as LSTM4FD.
- **LCRNN** (Beutel et al. 2018): It uses "Latent Cross" to incorporate contextual data in the RNN by embedding the context feature first and then performing an element-wise product of the context embedding with the model's hidden states.
- **M3** (Tang et al. 2019): This is the state-of-the-art event sequence based model which deals with both short-term and long-term dependencies with mixture models, we choose the better one mixture model M3R-TSL.

Implementation Details

For all datasets, we use: embedding dimension k of 64, the maximum number of events T of 20, one hidden layer of MLP and the dimension is 64, mini-batch size of 256 and learning rate of 0.0005. We also use L2 regularization with $\lambda = 1e - 6$, and dropout probability is 0.2. All these values and hyper-parameters of all baselines are chosen via a grid search on the $C1$ validation set. We do not perform any datasets-specific tuning except early stopping on validation sets. The proposed model is trained offline and regularly updated. Meanwhile, the prediction phase is relatively fast, which can meet the requirement of real-time solutions. We conduct experiments of all models with NVIDIA GeForce RTX 2080 GPU.

Evaluation Metrics

To evaluate the performance of our proposed DIFM model and the baselines described above, we follow the existing works (Guo et al. 2017; Lian et al. 2018) to use the standard metric: **AUC** (Area Under ROC). In binary classification tasks, AUC is a widely used metric. In our real card-stolen fraud detection scenario, we should increase the re-

Model	C1 AUC$_{FPR\leq 1\%}$	C2 AUC$_{FPR\leq 1\%}$	C3 AUC$_{FPR\leq 1\%}$
W&D	0.700±0.001	0.777±0.002	0.820±0.009
DeepFM	0.707±0.006	0.773±0.004	0.848±0.007
NFM	0.747±0.003	0.793±0.007	0.831±0.022
AFM	0.709±0.005	0.780±0.007	0.850±0.007
xDeepFM	0.739±0.004	0.783±0.008	0.856±0.014
LSTM4FD	0.712±0.009	0.736±0.008	0.776±0.009
LCRNN	0.719±0.007	0.785±0.011	0.816±0.019
M3	0.729±0.006	0.790±0.005	0.762±0.027
DIFM-same	0.751±0.004	0.828±0.006	0.862±0.015
DIFM-α	0.740±0.007	0.808±0.007	0.854±0.019
DIFM-β	0.764±0.011	0.823±0.009	0.863±0.006
DIFM	**0.768±0.009**	**0.849±0.008**	**0.887±0.011**

Table 2: AUC$_{FPR\leq 1\%}$ performance (mean±95% confidence intervals) on three datasets.

call rate of the fraudulent transactions, at the same time, disturbing as few normal users as possible. In other words, we need to improve the True Positive Rate (TPR) on the basis of low False Positive Rate (FPR). Therefore, we adopt the standardized partial AUC ($\mathbf{AUC}_{\boldsymbol{FPR\leq maxfpr}}$) (McClish 1989) (The standardized area of the head of the ROC curve when the $FPR \leq maxfpr$). In practice, we require FPR to be less than 1%. Hence, we use $\mathbf{AUC}_{\boldsymbol{FPR\leq 1\%}}$ for all experiments. Besides, we also focus on some specific points on the head of ROC curve, i.e., the TPR when the FPR are 0.05%, 0.1%, 0.5% and 1%, respectively. For all experiments, we report the metric with 95% confidence intervals on five runs.

Experimental Results

The results evaluated by AUC$_{FPR\leq 1\%}$ on C1, C2 and C3 are presented in Table 2. We can observe that the performance of baseline models varies on different countries.

For the C1 dataset, the performance of W&D is inferior compared with other baselines with respect to AUC$_{FPR\leq 1\%}$, one possible reason is that W&D could not automatically learn feature interactions, and there is only a simple linear model in its "wide" part. Besides, DeepFM and AFM obtain similar performance compared with W&D. DeepFM and W&D both have parallel structures, and although the FM in DeepFM captures second-order feature interactions, it is directly fed to the output layer and failed to capture higher-order feature interactions. Pairs weighted AFM is also fed directly to the output layer, and therefore it lacks higher-order feature interactions information. On the contrary, the FM in NFM is fed to MLP for capturing higher-order feature interactions, so NFM almost performs the best among all baseline models. Similarly, xDeepFM attempts to learn higher-order feature interactions explicitly, so xDeepFM improves the results compared with W&D and DeepFM. For event sequence based models, LSTM4FD and LCRNN obtain similar performance. Due to dealing with both short-term and long-term sequence dependencies with mixture models, M3 obtains further performance improvement. However, these models do not consider feature interactions, so the improvement is limited.

Similar results can also be observed on dataset C2 for feature interactions based models. For event sequence based models, the performance of LSTM4FD, which simply applies LSTM for fraud detection, is unsatisfactory as well. LCRNN and M3 obtain similar performance, and LCRNN is effective for dataset C2 due to incorporating the contextual data in the RNN.

The label distribution of C3 varies from those of the previous two datasets, and it has fewer positive samples, negative samples and available events than C1 and C2 as shown in Table 1. Therefore, the AUC$_{FPR\leq 1\%}$ performance of these models varies a lot, and the 95% confidence intervals of AUC$_{FPR\leq 1\%}$ of these baseline models is obviously larger than the performance on C1 and C2 datasets. Besides, for event sequence based models, LSTM4FD, LCRNN, and M3 are undesirable due to the fewer number of historical events on the C3 dataset.

Besides, the DIFM which uses multiple feed-forward networks in the Importance-aware Module is more effective than DIFM-same, which only uses a single feed-forward network. Moreover, for better understanding the contribution of different perspectives, we construct ablation experiments over DIFM-α, DIFM-β and DIFM. DIFM-α and DIFM-β utilize the field value variations and field interactions perspectives in their models, respectively. The field interactions perspective brings greater gain than the field value variations perspective. Furthermore, the proposed DIFM model can efficiently take advantage of internal field features among events from dual perspectives and obtain the best performance on all three datasets, which indicates both two perspectives contribute to the performance. Instead of capturing sequence information directly, our proposed DIFM captures more fine-grained value-variations information. Frequent value-variations of certain field in event sequences is a signal which strongly correlates to fraudulent activities. For example, user with IP changing from $IP\#1$ to $IP\#2$ indicates higher risk comparing to user with stable IP. DIFM captures value-variations of each field between any two events in user's operation sequences via the Field Value Variations module. Combined with Field Interactions module, which captures field-in-event-relations, our proposed framework captures information more effectively. And this explains why it outperforms other sequence learning architectures, i.e. RNN-based baselines. These improvements also indicate that the proposed DIFM model can better handle the fraud detection task. Similar conclusions can also be observed in the results evaluated by TPR (when the FPR are 0.05%, 0.1%, 0.5% and 1%, respectively) on datasets of C1, C2 and C3, which are presented in Figure 4.

Case Study

In this subsection, we make some analysis on the explainability of the proposed DIFM on C3 dataset.

Firstly, we extract four highest-risk and lowest-risk field values from four fields according to the learned weights of "wide" part in Equation (12). We present the field values in Table 3. To conform to Data-Protection-Regulation of the

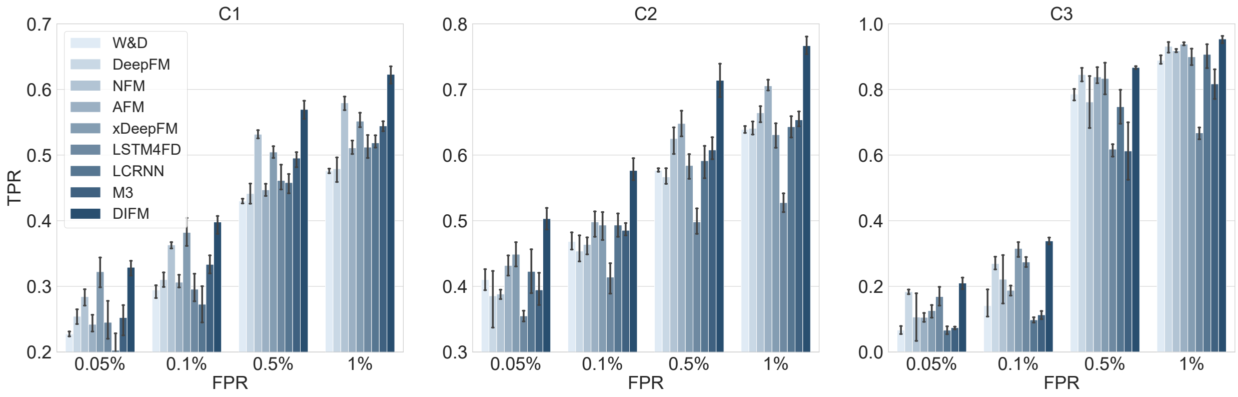


Figure 4: Mean TPR (when the FPR are 0.05%, 0.1%, 0.5% and 1%, respectively) performance over five runs on datasets of C1, C2 and C3, and the short black lines represent 95% confidence intervals.

	Card bin	IP ISP	Email suffix	Issuer
High Risk	#1 (94/105)	#1 (5/8)	#1 (28/28)	#1 (76/79)
	#2 (46/46)	#2 (30/30)	#2 (59/59)	#2 (136/137)
	#3 (12/12)	#3 (22/22)	#3 (56/60)	#3 (105/113)
	#4 (78/82)	#4 (4/5)	#4 (149/183)	#4 (77/85)
Low Risk	#5 (0/58)	#5 (0/81)	#5 (0/1120)	#5 (1/1008)
	#6 (0/181)	#6 (0/384)	#6 (0/101)	#6 (0/34)
	#7 (0/38)	#7 (0/89)	#7 (3/5958)	#7 (0/20)
	#8 (0/239)	#8 (0/38)	#8 (0/79)	#8 (0/54)

Table 3: The extracted high risk (high weight) and low risk (low weight) field values according to the learned weights in Equation (12), the "Card bin" is the last six digits of the card number, the "IP ISP" is Internet Service Provider for the IP, the "Email suffix" is the suffix of the email and the "Issuer" is the name of the issuing bank.

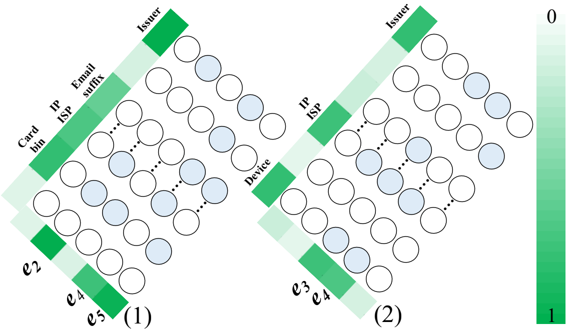


Figure 5: The extracted high-risk fields and events via the Importance-aware Module in two fraud samples.

company, we replace the real field values with #number and present the ratio of each field value with (fraud number/total number). We can clearly observe that these extracted field values have strong correlation to frauds. For example, the Email suffix #2 occurs 59 times and all of them are fraud samples, while the Email suffix #5 occurs 1120 times and all of them are normal usage. Therefore, these field values can be directly added to the blacklist and whitelist. By the way, in our risk management system, model predication is used together with rule-based methods, and this gives double insurance to our system.

Then, we extract some high-risk fields and events for two fraud samples according to the learned importance weights in Equation (4). The field value variations of each field and field interactions of each event can be regarded as modeling of users' fraud patterns. We present the fraud patterns in Figure 5. The solid circle represents that the field value has changed since the last event. The depth of color for each square illustrates the distributions of importance weights. The darker the color is, the higher weight it has.

The fields of Card bin, IP ISP, Email suffix and Issuer should be relatively stable for normal users, but for sample (1), these fields change multiple times in the account's operation history, which indicates a high risk. Therefore, these fields obtain higher weights. Meanwhile, the events 2, 4, and 5 have multiple abnormal field values, which makes them more significant than other normal events. The distribution of the event weights also confirms this. A similar pattern can also be observed in sample (2).

These observations demonstrate that our DIFM model can effectively find the important fields and events from dual perspectives. These above results also demonstrate that the proposed DIFM has the capability to provide explainable prediction results. These explanations are of great help to the human experts in analyzing the fraudulent cases.

Conclusion

In this paper, we proposed DIFM , a model for real-time transaction fraud detection. Specifically, DIFM utilizes the Factorization Machines and the proposed Importance-aware Module to exploit user's behavioral information from dual perspectives, which are field value variations perspective and field interactions perspective. The extensive experimental results on real-world industrial data collected from an e-commerce platform clearly demonstrate the performance improvements of our proposed model compared with various state-of-the-art baseline methods and the case study further approves the explainability of our model.

Acknowledgments

The research work supported by the National Key Research and Development Program of China under Grant No. 2018YFB1004300, the National Natural Science Foundation of China under Grant No. U1836206, U1811461, 61773361, the Project of Youth Innovation Promotion Association CAS under Grant No. 2017146. This work is funded in part by Ant Financial through the Ant Financial Science Funds for Security Research. We also thank Minhui Wang, Zhiyao Chen, Changjiang Zhang and Jiani Li for their valuable suggestions.

Ethical Impact

The assumption that fraudsters like to assume identities of a specific societal group will not affect the normal members of this group. Since our system and algorithm target risky transactions instead of users and cards. It does not blacklist them. For most cases, when trades are marked risky by our system, the trade operator need to provide evidence that they are the authentic owner of the account/card in order to continue the payment process. When applying machine learning algorithm for the detection tasks, there will definitely be false positive cases. On the one hand, we try to improve the precision and recall in the training phase in order to minimize the misclassification rate. On the other hand, when deploying the model to the real-time detection system, there will be different action levels in dealing with the risky cases. For example, transactions with the highest score (highest risk probability) will be rejected directly; transactions with a mid-high score (relatively lower risk) will be transferred to the verification phase. As long as the operator could pass authentication, the payment could continue. Therefore, the authentic owner can pass the verification phase to finish the transaction. Besides, we have customer service waiting for the false positive case's appealing. Such cases will help us improve algorithm performance in the future. Meanwhile, fraudsters usually possess lots of stolen accounts/cards, imitating regular people's operation will largely improve their cost and compress their profitability. Forcing them to such unprofitable situation, it will eventually make them leave. Therefore from a macro perspective we lower the risk level of the platform and protect the users asset in our platform.

References

Bahdanau, D.; Cho, K.; and Bengio, Y. 2015. Neural machine translation by jointly learning to align and translate. In *ICLR*.

Baulier, G. D.; Cahill, M. H.; Ferrara, V. K.; and Lambert, D. 2000. Automated fraud management in transaction-based networks. US Patent 6,163,604.

Beutel, A.; Covington, P.; Jain, S.; Xu, C.; Li, J.; Gatto, V.; and Chi, E. H. 2018. Latent cross: Making use of context in recurrent recommender systems. In *WSDM*, 46–54.

Brause, R.; Langsdorf, T.; and Hepp, M. 1999. Neural data mining for credit card fraud detection. In *ICTAI*, 103–106.

Cao, S.; Yang, X.; Chen, C.; Zhou, J.; Li, X.; and Qi, Y. 2019. TitAnt: Online Real-time Transaction Fraud Detection in Ant Financial. *PVLDB* 12(12): 2082–2093.

Chen, Q.; Zhao, H.; Li, W.; Huang, P.; and Ou, W. 2019. Behavior Sequence Transformer for E-commerce Recommendation in Alibaba. *arXiv preprint arXiv:1905.06874* .

Cheng, H.-T.; Koc, L.; Harmsen, J.; Shaked, T.; Chandra, T.; Aradhye, H.; Anderson, G.; Corrado, G.; Chai, W.; Ispir, M.; et al. 2016. Wide & deep learning for recommender systems. In *DLRS*, 7–10.

Cohen, W. W. 1995. Fast effective rule induction. In *Machine learning proceedings 1995*, 115–123.

Fu, K.; Cheng, D.; Tu, Y.; and Zhang, L. 2016. Credit card fraud detection using convolutional neural networks. In *NeurIPS*, 483–490.

Guo, H.; Tang, R.; Ye, Y.; Li, Z.; and He, X. 2017. DeepFM: a factorization-machine based neural network for CTR prediction. In *IJCAI*.

He, X.; and Chua, T.-S. 2017. Neural factorization machines for sparse predictive analytics. In *SIGIR*, 355–364.

Hidasi, B.; Karatzoglou, A.; Baltrunas, L.; and Tikk, D. 2016. Session-based recommendations with recurrent neural networks. In *ICLR*.

Jurgovsky, J.; Granitzer, M.; Ziegler, K.; Calabretto, S.; Portier, P.-E.; He-Guelton, L.; and Caelen, O. 2018. Sequence classification for credit-card fraud detection. *Expert Systems with Applications* 100: 234–245.

Kang, W.-C.; and McAuley, J. 2018. Self-attentive sequential recommendation. In *ICDM*, 197–206.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* .

Lian, J.; Zhang, F.; Xie, X.; and Sun, G. 2017. Restaurant survival analysis with heterogeneous information. In *TheWebConf*, 993–1002.

Lian, J.; Zhou, X.; Zhang, F.; Chen, Z.; Xie, X.; and Sun, G. 2018. xdeepfm: Combining explicit and implicit feature interactions for recommender systems. In *KDD*, 1754–1763.

Liang, C.; Liu, Z.; Liu, B.; Zhou, J.; Li, X.; Yang, S.; and Qi, Y. 2019. Uncovering Insurance Fraud Conspiracy with Network Learning. In *SIGIR*, 1181–1184.

Ma, C.; Kang, P.; and Liu, X. 2019. Hierarchical Gating Networks for Sequential Recommendation. In *KDD*.

McClish, D. K. 1989. Analyzing a portion of the ROC curve. *Medical Decision Making* 9(3): 190–195.

Pathak, J.; Vidyarthi, N.; and Summers, S. L. 2005. A fuzzy-based algorithm for auditors to detect elements of fraud in settled insurance claims. *Managerial Auditing Journal* 20(6): 632–644.

Qu, Y.; Cai, H.; Ren, K.; Zhang, W.; Yu, Y.; Wen, Y.; and Wang, J. 2016. Product-based neural networks for user response prediction. In *ICDM*, 1149–1154.

Quadrana, M.; Karatzoglou, A.; Hidasi, B.; and Cremonesi, P. 2017. Personalizing session-based recommendations with hierarchical recurrent neural networks. In *RecSys*, 130–137.

Rakkappan, L.; and Rajan, V. 2019. Context-Aware Sequential Recommendations with Stacked Recurrent Neural Networks. In *TheWebConf*, 3172–3178.

Rendle, S. 2010. Factorization machines. In *ICDM*, 995–1000.

Rosset, S.; Murad, U.; Neumann, E.; Idan, Y.; and Pinkas, G. 1999. Discovery of fraud rules for telecommunications—challenges and solutions. In *KDD*, 409–413.

Stefano, B.; and Gisella, F. 2001. Insurance fraud evaluation: a fuzzy expert system. In *FUZZ-IEEE*, volume 3, 1491–1494.

Tang, J.; Belletti, F.; Jain, S.; Chen, M.; Beutel, A.; Xu, C.; and H Chi, E. 2019. Towards neural mixture recommender for long range dependent user sequences. In *TheWebConf*, 1782–1793.

Tang, J.; and Wang, K. 2018. Personalized top-n sequential recommendation via convolutional sequence embedding. In *WSDM*, 565–573.

Tian, T.; Zhu, J.; Xia, F.; Zhuang, X.; and Zhang, T. 2015. Crowd fraud detection in internet advertising. In *TheWebConf*, 1100–1110.

Tseng, V. S.; Ying, J.-C.; Huang, C.-W.; Kao, Y.; and Chen, K.-T. 2015. Fraudetector: A graph-mining-based framework for fraudulent phone call detection. In *KDD*, 2157–2166.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *NeurIPS*, 5998–6008.

Wang, R.; Fu, B.; Fu, G.; and Wang, M. 2017a. Deep & cross network for ad click predictions. In *ADKDD*, 12.

Wang, S.; Liu, C.; Gao, X.; Qu, H.; and Xu, W. 2017b. Session-Based Fraud Detection in Online E-Commerce Transactions Using Recurrent Neural Networks. In *PKDD*, 241–252.

Xi, D.; Zhuang, F.; Song, B.; Zhu, Y.; Chen, S.; Hong, D.; Chen, T.; Gu, X.; and He, Q. 2020. Neural Hierarchical Factorization Machines for User's Event Sequence Analysis. In *SIGIR*, 1893–1896.

Xiao, J.; Ye, H.; He, X.; Zhang, H.; Wu, F.; and Chua, T.-S. 2017. Attentional factorization machines: Learning the weight of feature interactions via attention networks. In *IJCAI*.

Zhang, R.; Zheng, F.; and Min, W. 2018. Sequential Behavioral Data Processing Using Deep Learning and the Markov Transition Field in Online Fraud Detection. In *KDD Workshop*.

Zhang, W.; Du, T.; and Wang, J. 2016. Deep learning over multi-field categorical data. In *ECIR*, 45–57.

Zhou, G.; Mou, N.; Fan, Y.; Pi, Q.; Bian, W.; Zhou, C.; Zhu, X.; and Gai, K. 2019. Deep interest evolution network for click-through rate prediction. In *AAAI*, volume 33, 5941–5948.

Zhou, G.; Zhu, X.; Song, C.; Fan, Y.; Zhu, H.; Ma, X.; Yan, Y.; Jin, J.; Li, H.; and Gai, K. 2018. Deep interest network for click-through rate prediction. In *KDD*, 1059–1068.

Zhu, Y.; Xi, D.; Song, B.; Zhuang, F.; Chen, S.; Gu, X.; and He, Q. 2020. Modeling Users' Behavior Sequences with Hierarchical Explainable Network for Cross-Domain Fraud Detection. In *TheWebConf*, 928–938.