Multimodal and Contrastive Learning for Click Fraud Detection

Weibin Li*, Qiwei Zhong*, Qingyang Zhao, Hongchun Zhang, Xiaonan Meng
Alibaba Group
Hangzhou, China
{dece.lwb,yunwei.zqw,qingyang.zqy,hongchun.zhc,xiaonan.mengxn}@alibaba-inc.com

ABSTRACT

Advertising click fraud detection plays one of the vital roles in current E-commerce websites as advertising is an essential component of its business model. It aims at, given a set of corresponding features, e.g., demographic information of users and statistical features of clicks, predicting whether a click is fraudulent or not in the community. Recent efforts attempted to incorporate attributed behavior sequence and heterogeneous network for extracting complex features of users and achieved significant effects on click fraud detection. In this paper, we propose a Multimodal and Contrastive learning network for Click Fraud detection (MCCF). Specifically, motivated by the observations on differences of demographic information, behavior sequences and media relationship between fraudsters and genuine users on E-commerce platform, MCCF jointly utilizes wide and deep features, behavior sequence and heterogeneous network to distill click representations. Moreover, these three modules are integrated by contrastive learning and collaboratively contribute to the final predictions. With the real-world dataset containing 3.29 million clicks on Alibaba platform, we investigate the effectiveness of MCCF. The experimental results show that the proposed approach is able to improve AUC by 7.2% and F1-score by 15.6%, compared with the state-of-the-art methods.

KEYWORDS

Click Fraud Detection, Multimodal Learning, Contrastive Learning

1 INTRODUCTION

Online click advertising, widely known as cost-per-click or pay-per-click, is an internet advertising in which an advertiser pays a publisher (typically a search engine, website owner, or a network of websites) when one ad is clicked [1]. Different from traditional advertising, advertisers can track consumers' online behaviors for accurate measurements of advertising profitability [36]. Click fraud detection plays a critical role due to the growing volume of this online advertising. Google implicitly acknowledged the problem when it paid $90 million to settle a click fraud lawsuit [33]. Moreover, the World Federation of Advertisers says ad fraud will cost advertisers $50 billion a year by 2025 [2].

Practically, click advertising is sold on per click basis. Figure 1 shows the four roles in the typical advertising business scenario. Their functions, interest appeals and click fraud motivations are summarized as follows:

- Advertisers: reaching users with advertisements of their products, and further converting users to consumers of their services or products. Advertisers may click rivals' ads with the purpose of driving up their costs or exhausting their ad budget. When a rival's budget is exhausted, it will exit the ad auction.
- Advertising Agency: more professional advertising promotion trader, helping advertisers manage their accounts and providing professional marketing services. They have no incentive for click fraud.
- Advertising Trading Platform: advertising platform that connects internet media and advertisers. It not only provides advertisers with advertising marketing tools and advertising services, but also realizes the commercial value of advertising with the help of internet media traffic. For example, search engine companies, e-commerce companies, and social companies with a large number of users and traffic. They have no incentive for click fraud as well.
- Media Platforms: providers of internet information and services. When users browse their information or use their services, they complete the dissemination of advertising information. The media is generally also called an alliance, such as blogs and address navigation websites. Some of these third parties might click the ads maliciously to inflate advertisers' revenues.
- Users: person who browses information or uses services on the internet is a potential customer of an advertiser. They also have no incentive for click fraud.

Although the existing researches have achieved significant effects in the detection of common frauds such as machine click fraud or click fraud with distinct statistical features [1, 2, 9, 11, 20, 24, 30, 31, 38], the detection of high-level fraud still needs to be resolved. The particular challenges of this issue are summarized as follows:

- Simulate genuine click behavior: fraudsters simulate genuine click, manifesting as more complex abnormality of statistical features.
- Fraudsters frequently switch IP and clear cookies to make their statistical features look like genuine. However, their behavior sequence might be abnormal, such as only visiting search and advertising pages.
- Group fraud involving heterogeneous information: a group of multiple people attack a specific advertiser together.
- Highly imbalanced distribution: the ratio of fraudulent clicks to genuine clicks is less than 1:8 for instance.

Therefore, building a more effective fraud detection system is pivotal for online advertising businesses. Specifically, based on the

*Corresponding author.
[1] https://en.wikipedia.org/wiki/Pay-per-click
[2] https://www.businessinsider.com/wfa-report-ad-fraud-will-cost-advertisers-50-billion-by-2025-2016-6

DeMaL@WWW '21, April 19–23, 2021, Ljubljana, Slovenia

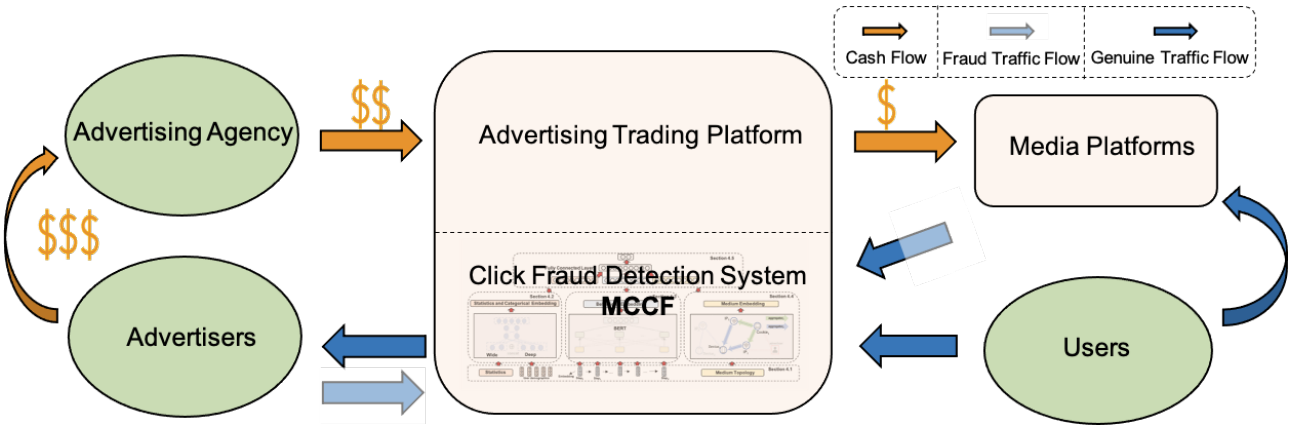

Figure 1: A typical flow of click advertising business.

Table 1: Some typical fields in click fraud detection system.

Field	Description
AbsPos	Absolute position of an ad on website
AdvertiserID	Unique identifier of advertiser
CdTime	Interval between display time and click time
ClickID	Unique identifier of a particular click
ClickTime	Timestamp of a given click
CookieID	Unique identifier of users
CookieTime	Timestamp that cookie was generated
DeviceID	Unique identifier of mobile users
IP	Public IP address of a click
KeywordID	Unique identifier of ad word
PageType	Homepage, Detail, ...

challenges above as well as the analysis and observations below on the real-world dataset, we propose a novel Multimodal and Contrastive learning network for Click Fraud detection (**MCCF**). Firstly, multimodal information including statistic and categorical features, behavior sequences and media relationships modeled by Wide and Deep [5], BERT [7, 34] and GNN [13, 21, 22, 37, 41–43] are involved to perform comprehensive click representations simultaneously. Secondly, we integrate these representations via multiple layer perceptron and output the prediction. Finally, contrastive learning [4] is utilized to solve the imbalance problem in this domain.

Observation 1: The statistical feature of clicks are clearly distinct between genuine users and fraudsters.

Figure 2 (a) and (b) illustrate the cumulative distributions of the average number of clicks per IP per day, and the average time interval between the click time and the time that CookieID was generated for genuine and fraud clicks on Alibaba.com, respectively. We found that the number of clicks per IP of most fraudsters in a single day is much more than that of genuine users. For example, 54.69% of fraud clicks have at least 10 times on their number of clicks per IP, while only 11.53% for genuine clicks. Meanwhile, we observed that time interval between the click time and the time that CookieID was generated for fraudsters are much shorter, e.g., 40.81% v.s. 24.78% of the intervals are $\leq$ 900 seconds for fraudsters and genuine users, respectively. We can easily conjecture the reason is that fraudsters try to fraudulently click as many as possible for a better ROI [3].

Observation 2: The difference of behavior pages between genuine and fraud clickers are significant.

As shown in Figure 2 (c), we demonstrate the ratio of top page types between fraud and genuine clicks. For example, over 99% of fraudsters are concentrated on homepage, detail, and list pages, while the proportion of genuine clicks on each page is relatively even.

Observation 3: Both number of associated media are distinguished between genuine users and fraudsters.

Figure 2 (d) illustrates the cumulative distributions of the average number of media (such as IP, CookieID, DeviceID) of clicks from fraudsters and genuine users. We clearly observe that the number of associated media of most fraudsters in a single day is much more than that of genuine users, which results in flatter trends on the corresponding cumulative distribution curve. For example, 21.86% of fraudsters have at least 3 associated medias, while only 6.31% for genuine users.

The main contributions of this work are summarized as follows:

- To the best of our knowledge, we are the first attempt to incorporate multimodal information and contrastive learning for click fraud detection.
- We propose a novel multimodal and contrastive learning network to solve this problem. Specifically, multimodal information including statistic and categorical features, behavior sequences and media relationships are involved to perform comprehensive click representations, and multiple layer perceptron is utilized to integrate them. Furthermore, to solve the imbalance problem, contrastive learning is elaborately exploited during training.
- Experiments on real-world dataset demonstrate the effectiveness of the proposed approach. It achieves competitive performance and outperforms state-of-the-art methods.

[3] https://en.wikipedia.org/wiki/Return_on_investment

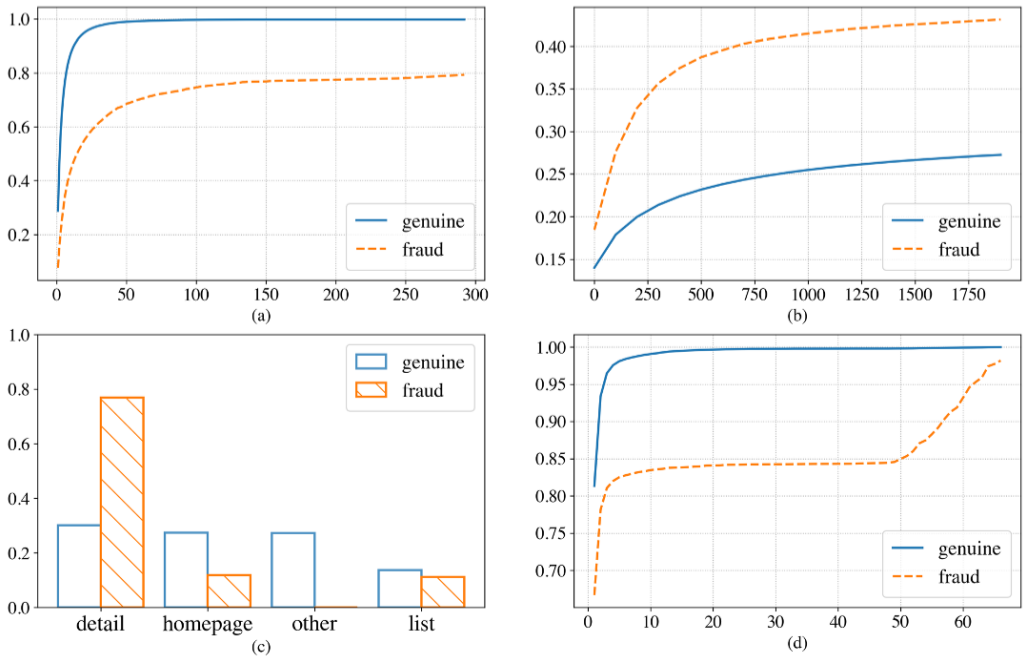

Figure 2: Statistical feature of fraudsters and genuine users: (a) cumulative distributions of the average number of clicks per IP; (b) cumulative distributions of the average time interval between the click time and the time that CookieID was generated; (c) page categorical properties of behaviors; (d) cumulative distributions of the average number of media.

2 RELATED WORK

In this section, we review related studies from three aspects, namely avoid click fraud in advance, anomaly-based and rule-based methods, and classifier-based methods. These related researches are categorized as follows:

2.1 Avoid click fraud in advance

Haddadi [11] presents bluff ads, a strategy to increase the effort of click fraudsters. CAPTCHA is used to ensure that the click is legitimate [6, 30]. Faou [9] follows the traffic to stop click fraud by disrupting the value chain. These methods increase the cost of click fraud, and meanwhile it may hurt the user experience to a certain extent.

2.2 Anomaly-based and Rule-based methods

Kshetri [20] classifies click fraud detection methods into three categories: anomaly-based, rule-based and classifier-based. Antoniou and Zhang [1, 39] analyze the number of visits in a certain time interval to detect duplicate clicks. Badhe [2] uses programmatic scripts to detect machine click fraud. Kitts [19] devises algorithm to detect robot click fraud. Due to the strong interpretability of the rules, Kitts [18] uses rules to filter click fraud early. But as fraud escalates, the rules become difficult to maintain and the detection ability deteriorates.

2.3 Classifier-based methods

Xu [38] constructs a pruned decision tree to classify traffic as valid, casual or fraudulent and introduces additional tests to check whether visiting clients are click-bots. Mouawi and Oentaryo [24, 25] present an important application of machine learning and data mining methods to tackle click fraud detection problems, such as single algorithms (e.g., LR, SVM, kNN, ANN) and ensemble learning algorithms (e.g., Random Forest). Kitts [18] discusses how to design a data mining system to detect large scale click fraud attacks. Berrar, Minastireanu and Oentaryo [3, 23, 25] prove that LightGBM and Random Forest have achieved good results. Thejas [31] combines Cascaded Forest and XGBoost to detect click fraud. Perera [26] utilizes an ensemble method to detect click fraud, which gained higher performance than single classifiers. Thejas [30, 32] proposes a hybrid deep learning model consisting of an Auto Encoder, a Neural Network and a Semi-supervised Generative Adversarial Network (GAN) to predict click fraud in imbalanced dataset. Although the above models can recall some fraud, they cannot effectively detect advanced fraud and group fraud that simulate genuine user behaviors.

3 PROBLEM STATEMENT

In this section, we present the problem formulation for click fraud. A click $\boldsymbol{x}$ in our problem consists of three kinds of information, namely Wide and Deep feature (denoted as $\boldsymbol{x}^{(w)}$, $\boldsymbol{x}^{(d)}$), behavior sequence (denoted as $\boldsymbol{x}^{(b)}$) and graph feature of user (denoted as $\boldsymbol{x}^{(v)}$). Given a set of the corresponding features, the goal of this task aims at predicting whether the click is fraudulent or not. Prior to that, we introduce several definitions which are helpful for problem statement.

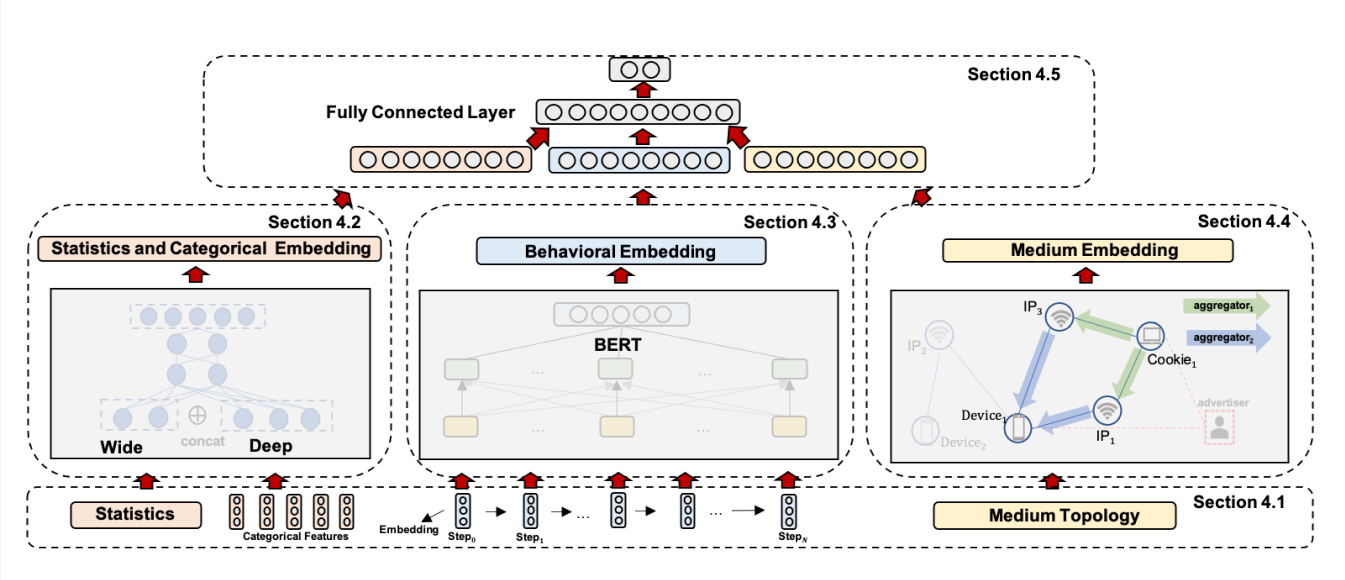

Figure 3: An illustration of the proposed MCCF model.

Definition 1. Wide and Deep feature: wide features are continuous features in each click, including original values (e.g., CdTime), combined features (e.g., AbsPos and CdTime), and demographic features (e.g., the number of cookies in the last day of IP). Deep features are categorical features in each click, such as AdvertiserID, KeywordID. $\boldsymbol{x}^{(w)} = [x_1^{(w)}, x_2^{(w)}, ..., x_l^{(w)}]$ is a vector of l-dimensional wide feature, and $\boldsymbol{x}^{(d)} = [x_1^{(d)}, x_2^{(d)}, ..., x_r^{(d)}]$ is a vector of r-dimensional deep feature.

Definition 2. Behavior sequence: the sequence of pages visited by a user before the ad is clicked, such as "Homepage -> List -> Detail -> $\cdots$", as shown in Figure 3. $\boldsymbol{x}^{(b)} = [x_1^{(b)}, x_2^{(b)}, ..., x_t^{(b)}]$ is a vector of t-dimensional behavior sequence. Specifically, the value of t in our model is 300.

Definition 3. Multi-media heterogeneous network: given a graph $G = (V, E)$, the feature of the node $\boldsymbol{x}^{(v)} = [x_1^{(v)}, x_2^{(v)}, ..., x_s^{(v)}]$ is a vector of s-dimensional feature integrated from itself and neighbors. The node types in our heterogeneous network are IP, CookieID, and DeviceID. For example, If a CookieID uses an IP to visit the website, the two nodes are neighbors, and an edge will be connected between them (as shown in Figure 3). For attributes of heterogeneous network, we collect 542 attributes for each medium (node), such as demographic information and click frequency. For each relation (link), we construct 90 attributes such as link type (e.g., click, login, and pay), first/last related time, and interaction frequency.

4 THE MCCF MODEL

In this section, we present the proposed MCCF model, as shown in Figure 3. We firstly introduce the distilling of feature representations and then illustrate model training via contrastive learning.

4.1 Input layer

Every element in the sequences $\boldsymbol{x}^{(d)}$ and $\boldsymbol{x}^{(b)}$ for each click needs to be transferred into embedding. After looking up from two embedding matrices $\boldsymbol{W}^{(d)}, \boldsymbol{W}^{(b)}$ respectively, $\boldsymbol{x}^{(d)}$ and $\boldsymbol{x}^{(b)}$ are converted to $\boldsymbol{e}^{(d)} = [\boldsymbol{e}_1^{(d)}, \boldsymbol{e}_2^{(d)}, ..., \boldsymbol{e}_r^{(d)}]$, $\boldsymbol{e}^{(b)} = [\boldsymbol{e}_1^{(b)}, \boldsymbol{e}_2^{(b)}, ..., \boldsymbol{e}_t^{(b)}]$, of which each element is an embedding vector, as shown in Figure 3.

$$\boldsymbol{e}^{(d)} = \text{LOOKUP}\left(\boldsymbol{W}^{(d)}; \boldsymbol{x}^{(d)}\right) \tag{1}$$

$$\boldsymbol{e}^{(b)} = \text{LOOKUP}\left(\boldsymbol{W}^{(b)}; \boldsymbol{x}^{(b)}\right) \tag{2}$$

where $\textbf{LOOKUP}\,(\boldsymbol{W}; \boldsymbol{x})$ is an operator to get vectors from $\boldsymbol{W}$ using each element of $\boldsymbol{x}$ as subscript. The embedding vectors are initialized randomly and then the values are trained with the model parameters to minimize the final loss function during training.

4.2 Wide and Deep Network

The wide and deep components are a multilayer neural network, as shown in Figure 3. The original inputs of deep component are categorical features (e.g., AdvertiserID, KeywordID). Each of these sparse, high-dimensional categorical features $\boldsymbol{x}^{(d)}$ are converted into a low-dimensional and dense embedding vector $\boldsymbol{e}^{(d)}$ via equation (1). These low-dimensional dense embedding vectors $\boldsymbol{e}^{(d)}$ and wide feature $\boldsymbol{x}^{(w)}$ are concatenated and then fed into the hidden layers of a neural network in the forward pass. Specifically, the wide and deep components perform the following computation.

$$\boldsymbol{e}^{(wd)} = \text{CONCAT}\left(\boldsymbol{e}^{(d)}, \boldsymbol{x}^{(w)}\right) \tag{3}$$

$$\boldsymbol{v}^{(wd)} = \text{ReLU}\left(\boldsymbol{W}_{wd}^{(L)} \ldots \text{ReLU}\left(\boldsymbol{W}_{wd}^{(1)} \boldsymbol{e}^{(wd)} + \boldsymbol{b}_{wd}^{(1)}\right) + \boldsymbol{b}_{wd}^{(L)}\right) \tag{4}$$

where L is the layer number, $\boldsymbol{W}_{wd}^{(l)}$ and $\boldsymbol{b}_{wd}^{(l)}$ are the model weights and bias at l^{th} layer. $\boldsymbol{v}^{(wd)}$ is the wide and deep component embedding vector.

4.3 Behavior Sequence Network

We utilize BERT [7] which gets SOTA results on many tasks to model behavior sequence, as shown in Figure 3. For the input embedding vector $\boldsymbol{e}^{(b)}$, BERT converts it into representation vector $\boldsymbol{v}^{(b)}$, paying more attention to the page type that can distinguish between fraud and genuine click.

$$\boldsymbol{v}^{(b)} = \mathbf{BERT}\left(\boldsymbol{e}^{(b)}\right) \tag{5}$$

4.4 Multi-media Heterogeneous Network

The core idea of a multi-media heterogeneous network is to aggregate the neighbors' feature information. As shown in Figure 3, the fraudster may exchange multiple cookies and devices of different media for click fraud. After the aggregating of statistical features of neighbor nodes via media heterogeneous network, fraudster's feature distribution might be quite abnormal.

$$\boldsymbol{h}^k_{\mathcal{N}(v)} = \mathbf{AGGREGATE}_k\left(\left\{\boldsymbol{h}^{k-1}_u, \forall u \in \mathcal{N}(v)\right\}\right) \tag{6}$$

$$\boldsymbol{h}^k_v = \sigma\left(\mathbf{W}^k_v \cdot \mathbf{CONCAT}\left(\boldsymbol{h}^{k-1}_v, \boldsymbol{h}^k_{\mathcal{N}(v)}\right)\right) \tag{7}$$

where $\boldsymbol{h}^k_v$ denotes a node's representation at this step. $\mathcal{N}(v)$ are all neighbor nodes of node v. Note that this aggregation step depends on the representations generated at the previous iteration, and representation $\boldsymbol{h}^0_v = \boldsymbol{x}^{(v)}$ is defined as the input node features. We use mean aggregation function here. Our method firstly aggregates the feature vector of the previous step of the neighbor node, then concatenates the node's current representation $\boldsymbol{h}^{k-1}_v$ as shown in equation (7), where σ is nonlinear activation function and k is the depth of the search. For notation convenience, we denote the final output representation at depth k as $\boldsymbol{v}^{(v)} = \boldsymbol{h}^k_v$.

4.5 Integration and Training

Finally, the outputs of the three modules are concatenated, followed by two fully connected layers and an output layer based on softmax function, as shown in Figure 3. The concatenation is represented as $\boldsymbol{v}^{(i)} = \left[\boldsymbol{v}^{(wd)}; \boldsymbol{v}^{(b)}; \boldsymbol{v}^{(v)}\right]$ and the next layers are denoted as

$$\boldsymbol{z}^{(i)}_2 = \boldsymbol{W}^{(i)}_2\left(\mathbf{ReLU}\left(\boldsymbol{W}^{(i)}_1\boldsymbol{v}^{(i)} + \boldsymbol{b}^{(i)}_1\right)\right) + \boldsymbol{b}^{(i)}_2 \tag{8}$$

$$\hat{\boldsymbol{y}} = \mathbf{softmax}\left(\boldsymbol{z}^{(i)}_2\right) \tag{9}$$

where $\boldsymbol{W}^{(i)}_k, \boldsymbol{b}^{(i)}_k (k = 1, 2)$ are weight matrix and bias vector of each layer, and $\mathbf{ReLU}(\cdot)$ is the element-wise rectified linear unit function. Specifically, $\hat{\boldsymbol{y}}$ is defined as the predicted probability vector of a click.

Furthermore, as mentioned previously, there is highly imbalance problem in click fraud detection task generally. To solve it, contrastive learning is elaborately exploited during training. Hadsell, Chopra, and LeCun [12] propose a loss function coined max margin contrastive loss that operates on pairs of samples instead of individual samples. Intuitively, this loss function learns an embedding to place samples with the same labels close to each other, while distancing the samples with different labels. Weinberger and Sohn [29, 35] present a multi-class N-pair loss which is an upgrade of max margin contrastive loss allowing joint comparison among more than one negative samples. Chen and Khosla [4, 16] propose the normalized temperature-scaled cross entropy loss (NT-Xent). It is a modification of multi-class N-pair loss with addition of the temperature parameter (τ).

In this paper, we train our model with SOTA NT-Xent loss with regularization. Specifically, let multiple layer perceptron (**MLP**) be an encoder network mapping $\boldsymbol{z}^{(i)}_2$ to the latent space $\boldsymbol{z}$ firstly.

$$\boldsymbol{z} = \mathbf{MLP}\left(\boldsymbol{z}^{(i)}_2\right) \tag{10}$$

Let $\mathrm{sim}(\boldsymbol{a}, \boldsymbol{b})$ denote the dot product between ℓ_2 normalized $\boldsymbol{a}$ and $\boldsymbol{b}$ (i.e. cosine similarity) in equation (11). When applied on a pair of positive samples $\boldsymbol{z}^{(i)}$ and $\boldsymbol{z}^{(j)}$ and other $2(N-1)$ negative examples, the loss function $\ell(i, j)$ for a positive pair of examples (i, j) is defined in equation (12).

$$\mathrm{sim}(\boldsymbol{a}, \boldsymbol{b}) = \boldsymbol{a}^\top \boldsymbol{b} / \|\boldsymbol{a}\|\|\boldsymbol{b}\| \tag{11}$$

$$\ell(i, j) = -\log \frac{\exp\left(\mathrm{sim}\left(\boldsymbol{z}^{(i)}, \boldsymbol{z}^{(j)}\right)/\tau\right)}{\sum_{k=1}^{2N} \mathbb{1}_{[k\neq i]} \exp\left(\mathrm{sim}\left(\boldsymbol{z}^{(i)}, \boldsymbol{z}^{(k)}\right)/\tau\right)} \tag{12}$$

where $\mathbb{1}_{[k\neq i]} \in \{0, 1\}$ is an indicator function evaluating to 1 if $k \neq i$, and τ denotes a temperature parameter. The final loss is computed across all positive pairs in a mini-batch.

$$\mathcal{L} = \frac{1}{2M}\sum_{k=1}^{M}\left[\ell(2k-1, 2k) + \ell(2k, 2k-1)\right] + \frac{\lambda}{2}\|\boldsymbol{\theta}\|^2_2 \tag{13}$$

where λ is the regularization parameter and $\boldsymbol{\theta}$ is the set of parameters of the proposed model.

4.6 Discussions

It is worth noting that not all kinds of sequences or networks mentioned above are compulsory in our MCCF model. For situations where only part features are available, it works as well. It can accomplish prediction by switching off the corresponding parts in the integration stage. For example, we can use only wide and deep sequences for early detection of click fraud. The corresponding experimental results will be demonstrated in the following sections.

5 EXPERIMENTS

In this section, we investigate the effectiveness of the proposed model. We conduct extensive experiments on a large-scale real-world dataset. Firstly, we verify the performance on detecting frauds from the dataset. Secondly, we perform ablation test and visualization to demonstrate the effectiveness of every component in our model.

5.1 Dataset

We collect a real-world dataset from an online click advertising service on Alibaba.com under the premise of complying with security and privacy policies. It contains 2.54 million clicks for training and 0.75 million clicks for testing, chronologically. User's rich behavioral information such as clicking logs, media relationship logs are collected according to their chronological orders. Based on the dataset, we construct a multimodal attributed information network. As mentioned previously, three modals are adopted, namely wide and deep features, behavior sequence and multi-media heterogeneous network, as shown in Figure 3. It is worth noting that the label (fraud or genuine) of training and testing set are acquired

Table 2: The statistical information of dataset.

Dataset	#Positive	#Negative	#Total	#Positive Rate
Training	276,956	2,265,022	2,541,978	10.89%
Testing	75,999	670,721	746,720	10.17%

via partially forecasting beforehand by the high-precision models deployed online, and manually evaluating and double checking offline afterwards. The data statistical information is exhibited in Table 2.

5.2 Compared Methods

We compare with several state-of-the-art representative methods including tree-based, graph-based and sequence-based to verify the effectiveness of our proposed method. Among them, tree-based baselines use statistical features, graph-based method uses medium topology information and statistical features, and the rests use behavior sequence information.

(a) **Tree-based Methods**

- **Random Forest** [3, 25]: a scalable tree-based model for feature learning and classification task, and widely used in various areas.
- **LightGBM** [15, 23]: an efficient parallel training Gradient Boosting Decision Tree-type method. Random Forest and LightGBM use statistical features, such as the number of cookies in the last day of IP and CdTime.

(b) **Graph-based Method**

- **GraphSAGE** [13]: a general and inductive framework that efficiently generates node embeddings by sampling and aggregating features from a node's local neighbors.

(c) **Sequence-based Methods**

- **BiLSTM** [14]: it mines the contextual information of the behavior sequence, and uses the attention mechanism to extract important information, so as to realize the classification of the sequence.
- **TextCNN** [40]: an algorithm that uses convolutional neural networks to classify text sequence. Different convolutions are used to extract the features of the context at different local locations to obtain semantic information at different levels of abstraction.
- **BERT** [7]: a pretrained model uses the now ubiquitous transformer architecture.

(d) **Our Method and Variants**

- **MCCF**: our proposed method. We also derive four variants of MCCF to comprehensively compare and analyze the performances of its each component. They are:
- **MCCF**$_{\setminus WD}$: removing wide and deep features.
- **MCCF**$_{\setminus B}$: removing behavior sequence.
- **MCCF**$_{\setminus V}$: removing multi-media heterogeneous network.
- **MCCF**$_{CE}$: changing the loss function from NT-Xent to cross entropy [27].

5.3 Implementation Details

For the network structure, the size of wide feature is set to 40, the embedding vector for the input layer of deep feature is set to 128, and the input embedding of behavior sequence is set to 128. For the multi-media heterogeneous network, the aggregating function is mean, the depth of search is set to 2, and the size of node feature vector is set to 500. For training parameters, λ is set to 0.01, the learning rate is set to 0.001, and the batch size is set to 64. We randomly initialize the model parameters with an xavier initializer [10] and choose Adam [17] as the optimizer. Five-run-average values are reported.

Our experiment uses Precision, Recall, micro F1-Score and AUC to compare the effects of all methods. The higher these metrics indicate the higher performance of approaches.

Table 3: Performances of different methods on the dataset.

Method	Precision	Recall	F1-score	AUC
Random Forest	0.867	0.403	0.550	0.685
LightGBM	0.892	0.416	0.567	0.686
GraphSAGE	0.973	0.545	0.699	0.785
BiLSTM	0.966	0.480	0.641	0.755
TextCNN	0.981	0.604	0.747	0.804
BERT	0.984	0.619	0.760	0.861
MCCF	**0.987**	**0.854**	**0.916**	**0.933**

5.4 Main Results

Table 3 demonstrates the main results of all compared methods on the dataset. The major findings from the experimental results can be summarized as follows:

(1) We can clearly observe that our model MCCF outperforms all the baselines by a large margin. Its F1-score, with reported value of 0.916, is at least 21.7% higher than tree-based and graph-based methods, and AUC gets 14.8% higher. Furthermore, MCCF is more advanced than sequence-based methods (i.e., BiLSTM, TextCNN and BERT), with at least 15.6% increased F1-score and 7.2% increased AUC. That is, the usage of sequence information and the further exploring on multimodal features make it more superior to the competitors. Besides, the obvious improvement of F1-score indicates that the model can detect more top-ranked click fraudsters under the same precision. This is critical to the real-world system when leveraging the business effect and interception rate.

(2) For baselines, LightGBM gets better performances than Random Forest among the tree-based methods. It achieves better F1-score due to deeper modeling residuals. BERT gets better performances than TextCNN and BiLSTM among the sequence-based methods. It achieves better F1-score via extracting different semantic information at different levels of abstraction. Moreover, it can be further seen that the graph-based method is more effective than tree-based methods via aggregating the statistical features of multiple media of the same user. F1-score is increased by more than 13.2% and AUC is improved by 9.9%. In addition, we observe that the sequence-based methods, e.g., TextCNN and BERT, are more effective than GraphSAGE due to taking advantage of behavioral information. F1-score is increased by more than 6.1% and AUC is improved by 7.6%.

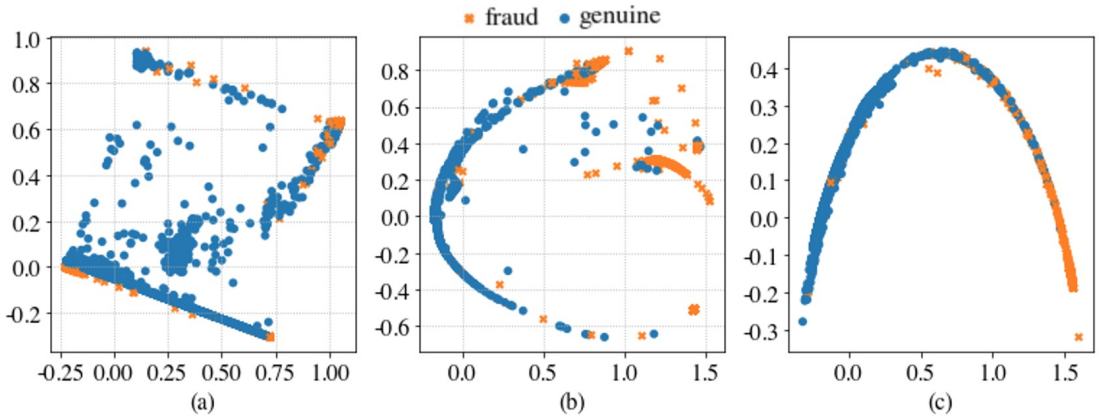

Figure 4: PCA projections: (a) original data; (b) last hidden layer with cross-entropy loss (MCCF$_{CE}$); (c) last hidden layer with NT-Xent loss (MCCF).

Table 4: Performances of ablation test on the proposed MCCF method.

Model	Precision	Recall	F1-score	AUC
MCCF$_{\setminus B}$	0.970	0.735	0.836	0.856
MCCF$_{\setminus V}$	0.975	0.776	0.864	0.882
MCCF$_{\setminus WD}$	0.979	0.807	0.884	0.905
MCCF$_{CE}$	0.985	0.832	0.902	0.918
MCCF	**0.987**	**0.854**	**0.916**	**0.933**

5.5 Ablation Test

Furthermore, we perform the ablation test for our MCCF, and the results are shown in Table 4.

5.5.1 The effects of modals. Firstly, we demonstrate the effectiveness of different modals by removing the corresponding modal information (e.g., removing the behavior sequence) respectively. Compared the second to the fourth rows with the last row in Table 4, we can clearly see that all metrics get worse by removing any modal-specific information. It is the worst by removing behavior sequence, which means behavior sequence has more significant impact on detecting click fraud in our dataset. The results reflect the importance of macroscopically modeling multiple modals as well, since every modal has a positive contribution for our task.

5.5.2 The effects of contrastive learning. Next, to further verify the importance of contrastive learning in model integration and training, we take a comparison with our approach and its variants, as shown in Table 4. The variant MCCF$_{CE}$ is to change the loss function of the model from contrastive learning NT-Xent to cross entropy. We could clearly observe that MCCF$_{CE}$ performs worse than our full model, which illustrates the contrastive learning NT-Xent is more effective in optimizing imbalanced problem. Meanwhile, its decreasing values (v.s. MCCF) reflect that contrastive learning plays a significant role in click fraud detection. In addition, MCCF$_{CE}$ still gets a performance of 0.902 on F1-score and 0.918 on AUC and clearly outperforms BERT, which indicates the effectiveness of multimodal information in click fraud detection.

5.6 Visualization

We next look closer to the data, and visualize the principal components of original data and last hidden layer of model, as shown in Figure 4. The two axes represent two principal components analysis [8, 28] of data respectively. It can be seen that the two principal components of the original data cannot distinguish between fraud and genuine clicks at all. Compared with the principal components of the original data, the system based on the cross entropy loss function, i.e., MCCF$_{CE}$, can clearly distinguish between fraud and genuine clicks. Furthermore, our MCCF model based on the contrastive learning loss function NT-Xent has better discrimination than MCCF$_{CE}$. From the visualizations, we demonstrate again that our MCCF, which incorporating multimodal information and contrastive learning, is effective in click fraud detection.

6 CONCLUSIONS

Advertising click fraud detection plays one of the vital roles in current E-commerce websites. In this paper, we proposed the MCCF model that jointly exploits multimodal information network and contrastive learning for click fraud detection. We carefully analyzed the differences between fraudsters and genuine users in the advertising click scenario on statistical, behavioral and media relation information. The observations motivate the three essential modules in MCCF, extracting features from different perspectives separately. These three modules are integrated and jointly trained via contrastive learning. The experimental results on a real-world click fraud detection task show that our approach achieves promising performance and significantly outperforms the SOTA methods.

ACKNOWLEDGMENTS

We would like to thank all the anonymous reviewers for their thoughtful and constructive comments and suggestions.

REFERENCES

[1] Dimitris Antoniou, Mersini Paschou, Evangelos Sakkopoulos, Efrosini Sourla, Giannis Tzimas, A Tsakalidis, and Emmanouil Viennas. 2011. Exposing click-fraud using a burst detection algorithm. In *ISCC*. 1111–1116.

[2] Anup Badhe. 2017. Click fraud detection in mobile ads served in programmatic inventory. *Neural Networks & Machine Learning* 1, 1 (2017), 1–1.

[3] Daniel Berrar. 2012. Random forests for the detection of click fraud in online mobile advertising. In *Proceedings of the 1st International Workshop on Fraud Detection in Mobile Advertising*. 1–10.

[4] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *ICML*. 1597–1607.

[5] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, et al. 2016. Wide & deep learning for recommender systems. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*. 7–10.

[6] Rodrigo Alves Costa, Ruy JGB de Queiroz, and Elmano Ramalho Cavalcanti. 2012. A proposal to prevent click-fraud using clickable captchas. In *SERE-C*. 62–67.

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[8] George H Dunteman. 1989. *Principal components analysis*. Number 69. Sage.

[9] Matthieu Faou, Antoine Lemay, David Décary-Hétu, Joan Calvet, François Labrèche, Militza Jean, Benoit Dupont, and José M Fernande. 2016. Follow the traffic: stopping click fraud by disrupting the value chain. In *PST*. 464–476.

[10] Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*. 249–256.

[11] Hamed Haddadi. 2010. Fighting online click-fraud using bluff ads. *ACM SIGCOMM Computer Communication Review* 40, 2 (2010), 21–25.

[12] Raia Hadsell, Sumit Chopra, and Yann LeCun. 2006. Dimensionality reduction by learning an invariant mapping. In *CVPR*, Vol. 2. 1735–1742.

[13] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *NeurIPS*. 1024–1034.

[14] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation* 9, 8 (1997), 1735–1780.

[15] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. Lightgbm: a highly efficient gradient boosting decision tree. *NeurIPS* 30 (2017), 3146–3154.

[16] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. *arXiv preprint arXiv:2004.11362* (2020).

[17] Diederik P Kingma and Jimmy Ba. 2014. Adam: a method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[18] Brendan Kitts, Jing Ying Zhang, Gang Wu, Wesley Brandi, Julien Beasley, Kieran Morrill, John Ettedgui, Sid Siddhartha, Hong Yuan, Feng Gao, et al. 2015. Click fraud detection: adversarial pattern recognition over 5 years at Microsoft. In *Real World Data Mining Applications*. 181–201.

[19] Brendan J Kitts, Tarek Najm, and Brian Burdick. 2008. Identifying automated click fraud programs. US Patent App. 11/745,264.

[20] Nir Kshetri. 2010. The economics of click fraud. *IEEE Security & Privacy* 8, 3 (2010), 45–53.

[21] Ting Liang, Guanxiong Zeng, Qiwei Zhong, Jianfeng Chi, Jinghua Feng, Xiang Ao, and Jiayu Tang. 2021. Credit risk and limits forecasting in e-commerce consumer lending service via multi-view-aware mixture-of-experts nets. In *WSDM*. 229–237.

[22] Ziqi Liu, Chaochao Chen, Longfei Li, Jun Zhou, Xiaolong Li, Le Song, and Yuan Qi. 2019. Geniepath: graph neural networks with adaptive receptive paths. In *AAAI*, Vol. 33. 4424–4431.

[23] Elena-Adriana Minastireanu and Gabriela Mesnita. 2019. Lightgbm machine learning algorithm to online click fraud detection. *J. Inform. Assur. Cybersecur* 2019 (2019).

[24] Riwa Mouawi, Mariette Awad, Ali Chehab, Imad H El Hajj, and Ayman Kayssi. 2018. Towards a machine learning approach for detecting click fraud in mobile advertizing. In *IIT*. 88–92.

[25] Richard Oentaryo, Ee-Peng Lim, Michael Finegold, David Lo, Feida Zhu, Clifton Phua, Eng-Yeow Cheu, Ghim-Eng Yap, Kelvin Sim, Minh Nhut Nguyen, et al. 2014. Detecting click fraud in online advertising: a data mining approach. *Journal of Machine Learning Research* 15, 1 (2014), 99–140.

[26] Kasun S Perera, Bijay Neupane, Mustafa Amir Faisal, Zeyar Aung, and Wei Lee Woon. 2013. A novel ensemble learning-based approach for click fraud detection in mobile advertising. In *Mining Intelligence and Knowledge Exploration*. 370–382.

[27] Reuven Rubinstein. 1999. The cross-entropy method for combinatorial and continuous optimization. *Methodology and Computing in Applied Probability* 1, 2 (1999), 127–190.

[28] Lindsay I Smith. 2002. A tutorial on principal components analysis. (2002).

[29] Kihyuk Sohn. 2016. Improved deep metric learning with multi-class n-pair loss objective. In *NeurIPS*. 1857–1865.

[30] GS Thejas, Kianoosh G Boroojeni, Kshitij Chandna, Isha Bhatia, SS Iyengar, and NR Sunitha. 2019. Deep learning-based model to fight against ad click fraud. In *ACM SE*. 176–181.

[31] GS Thejas, Surya Dheeshjith, SS Iyengar, NR Sunitha, and Prajwal Badrinath. 2021. A hybrid and effective learning approach for click fraud detection. *Machine Learning with Applications* 3 (2021), 100016.

[32] GS Thejas, Jayesh Soni, Kianoosh G Boroojeni, SS Iyengar, Kanishk Srivastava, Prajwal Badrinath, NR Sunitha, Nagarajan Prabakar, and Himanshu Upadhyay. 2019. A multi-time-scale time series analysis for click fraud forecasting using binary labeled imbalanced dataset. In *CSITSS*, Vol. 4. 1–8.

[33] Alexander Tuzhilin. 2006. The lane's gifts v. google report. *Official Google Blog: Findings on Invalid Clicks, Posted* (2006), 1–47.

[34] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NeurIPS*. 5998–6008.

[35] Kilian Q Weinberger and Lawrence K Saul. 2009. Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research* 10, 2 (2009).

[36] Kenneth C Wilbur and Yi Zhu. 2009. Click fraud. *Marketing Science* 28, 2 (2009), 293–308.

[37] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. 2020. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems* (2020).

[38] Haitao Xu, Daiping Liu, Aaron Koehl, Haining Wang, and Angelos Stavrou. 2014. Click fraud detection on the advertiser side. In *ESORICS*. 419–438.

[39] Linfeng Zhang and Yong Guan. 2008. Detecting click fraud in pay-per-click streams of online advertising networks. In *ICDCS*. 77–84.

[40] Ye Zhang and Byron Wallace. 2015. A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. *arXiv preprint arXiv:1510.03820* (2015).

[41] Ziwei Zhang, Peng Cui, and Wenwu Zhu. 2020. Deep learning on graphs: a survey. *IEEE Transactions on Knowledge and Data Engineering* (2020).

[42] Qiwei Zhong, Yang Liu, Xiang Ao, Binbin Hu, Jinghua Feng, Jiayu Tang, and Qing He. 2020. Financial defaulter detection on online credit payment via multi-view attributed heterogeneous information network. In *WWW*. 785–795.

[43] Jie Zhou, Ganqu Cui, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. 2018. Graph neural networks: a review of methods and applications. *arXiv preprint arXiv:1812.08434* (2018).