A Counterfactual Modeling Framework for Churn Prediction

Guozhen Zhang, Jinwei Zeng, Zhengyue Zhao, Depeng Jin, Yong Li
Beijing National Research Center for Information Science and Technology,
Department of Electronic Engineering, Tsinghua University, Beijing 100084, China

ABSTRACT

Accurate churn prediction for retaining users is keenly important for online services because it determines their survival and prosperity. Recent research has specified social influence to be one of the most important reasons for user churn, and thereby many works start to model its effects on user churn to improve the prediction performance. However, existing works only use the data's correlational information while neglecting the problem's causal nature. Specifically, the fact that a user's churn is correlated with some social factors does not mean he/she is actually influenced by his/her friends, which results in inaccurate and unexplainable predictions of the existing methods. To bridge this gap, we develop a counterfactual modeling framework for churn prediction, which can effectively capture the causal information of social influence for accurate and explainable churn predictions. Specifically, we first propose a backbone framework that uses two separate embeddings to model users' endogenous churn intentions and the exogenous social influence. Then, we propose a counterfactual data augmentation module to introduce the causal information to the model by providing partially labeled counterfactual data. Finally, we design a three-headed counterfactual prediction framework to guide the model to learn causal information to facilitate churn prediction. Extensive experiments on two large-scale datasets with different types of social relations show our model's superior prediction performance compared with the state-of-the-art baselines. We further conduct an in-depth analysis of the prediction results demonstrating our proposed method's ability to capture causal information of social influence and give explainable churn predictions, which provide insights into designing better user retention strategies.

CCS CONCEPTS

• **Information systems** → *Enterprise applications*; **Social networks**; • **Computing methodologies** → **Neural networks**.

KEYWORDS

Churn prediction, social influence, causal information learning

ACM Reference Format:
Guozhen Zhang, Jinwei Zeng, Zhengyue Zhao, Depeng Jin, Yong Li. 2022. A Counterfactual Modeling Framework for Churn Prediction. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining (WSDM '22), February 21–25, 2022, Tempe, AZ, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3488560.3498468

1 INTRODUCTION

Users are the core of developing online applications. The ability of user acquisition and user retention determine the survival and prosperity of a wide range of online applications, ranging from general online platforms such as social media, columns, and online communities, to online services, such as news, games, and e-commerce [5, 6, 8, 35]. As these online applications spring up, competition becomes fierce, and as a result, acquiring new users is becoming increasingly difficult [32]. According to a recent report, the cost of acquiring new users is five times higher than retaining the existing ones [2]. Therefore, how to retain users is drawing increasing attention. Specifically, as the most critical part of a typical user retention procedure, predicting user churn has become a key concern for both academia and industry [14, 33].

With the maturity of social media, lots of applications have integrated social features such as comments and friend sharing into their services to enhance their user experience. Many even directly build themselves on social media. In these fast-growing scenarios, social influence has become one of the most important reasons for customer churn [20]. Following this lead, many researchers have been trying to model the effects of social influence on user churn to improve churn prediction accuracy. Their attempts can be mainly divided into two types. The first is modeling social network structures as a proxy of social influence. For example, Ahmad et al. [1] use social network analysis to derive network-based features for machine learning model. The second is viewing user churn as a diffusion process. That is, if user A churned after his friend B, we assume user A is "infected" by user B. This type of work typically uses a diffusion model to simulate the effects of social influence [36].

However, existing approaches only utilize the data's correlational information while neglecting the problem's causal nature. Specifically, the fact that a user's churn is correlated with some social factors does not mean he/she is actually influenced by his/her friends. As such, existing deep learning methods mainly suffer from two limitations: (1) their performances can still be improved because using correlational methods to solve causal problems are bound to introduce non-causal noises; (2) their predictions are unexplainable. They cannot answer causal questions such as "did a user churn because of the effects of social influence?" or "would a user stay if his/her friends had not churned?", making it hard to tailor campaigns for user retention.

To bridge these gaps, we set out to utilize causal information to improve the churn prediction performance. Specifically, we focus on designing a churn prediction framework that can learn the causal information of the social influence on user churn to give

WSDM '22, February 21–25, 2022, Tempe, AZ, USA

ACM ISBN 978-1-4503-9132-0/22/02…$15.00
https://doi.org/10.1145/3488560.3498468

accurate and explainable churn predictions. This is challenging because learning causality from data is intrinsically difficult, and one of the most important reasons is the lack of counterfactual data. For instance, if a user churns after his friends, we cannot observe the case when his/her friends did not churn. As a result, the most powerful supervised learning framework is not working without a supervising signal for causal effects.

Faced with the challenge, we present a CounterFactual modeling framework for Churn prediction (CFChurn), which captures the causal effects of social influence on user churn and provides accurate and explainable predictions. Specifically, we first build our causal model on a backbone framework that uses two separate embeddings to model users' endogenous churn intentions and the exogenous social influence, which contains a novel design - social influence guided graph neural network to capture the social influence between users better. Then, we propose a counterfactual data augmentation procedure that solves the problem of missing counterfactual observations based on prior causal knowledge on social influence, which transforms the counterfactual learning problem into a supervised learning problem with partially labeled data. Finally, to help our model to learn causal effects to facilitate churn prediction better, we design a three-headed multi-task prediction framework based on the sufficiency of the propensity score theory.

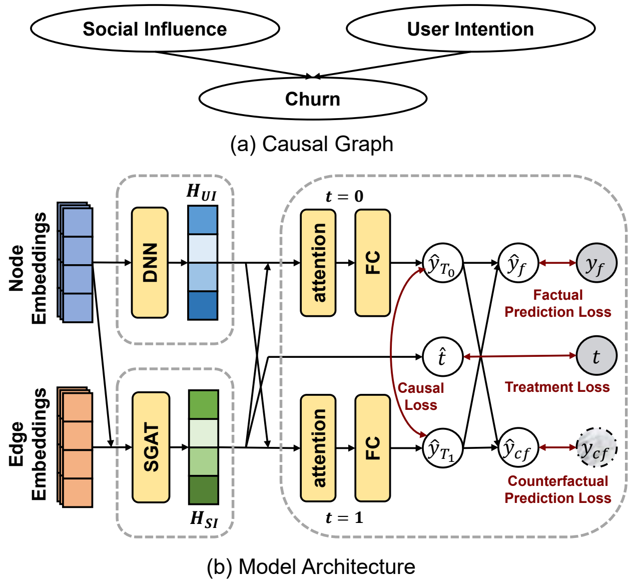
We highlight our contributions as follows:

- To the best of our knowledge, this paper tackles the user churn prediction problem from a causal standpoint for the first time, and we provide a counterfactual prediction framework that can learn the causal information of the social influence on churn.
- We design three novel components, including the social influence guided graph neural network, the counterfactual data augmentation module, and the three-headed multi-task learning module, to facilitate our model to learn the causal effects of social influence from data.
- We conduct extensive experiments on two large-scale real-world datasets with different types of social relationships. On both datasets, our model outperforms the state-of-the-art baselines. Further ablation study shows the effectiveness of each design, and further in-depth analyses on the model predictions show our model's ability to give explanations for the prediction results, which provides insights for online applications to make tailored campaigns.

2 PROBLEM FORMULATION

The goal of churn prediction is to forecast whether a user will stop using a service or a platform in a future period. Specifically, it has three inputs, including a user feature matrix $\boldsymbol{X_v}$ recording users' demographics and historical behaviors, a user interaction feature matrix $\boldsymbol{X_e}$ recording the interactions between users, and a social network G. It outputs a binary prediction for each user, which indicates whether a user will churn in a future period. Formally, it can be formulated as follows:

$$\boldsymbol{y} = F(G, \boldsymbol{X_v}, \boldsymbol{X_e}), \tag{1}$$



(a) Causal Graph

(b) Model Architecture

Figure 1: The causal graph model and our proposed CFChurn framework. (a) According to prior works, exogenous social influence and endogenous user intention are two major types of causes that account for user churn. (b) Our proposed framework models the two different causes by two embeddings separately, and we designed a three-headed counterfactual prediction framework to enable our model to answer counterfactual questions while giving accurate churn predictions.

where $\boldsymbol{X_v} \in \mathbb{R}^{n_{v0} \times N}$, $\boldsymbol{X_e} \in \mathbb{R}^{n_{e0} \times K}$, $\boldsymbol{y} \in \mathbb{R}^N$, with n_{v0} and n_{e0} as the dimension of user features and interaction features, respectively. Here, N, K is the number of users and user relations, respectively.

In this paper, we model the social network G as a graph, with users modeled as nodes and user relations modeled as edges. Thus, we have $G = (\mathcal{V}, \mathcal{E})$ with $|\mathcal{V}| = N$, $|\mathcal{E}| = K$, and its adjacency matrix $\boldsymbol{A} \in \mathbb{R}^{N \times N}$. Further, we model the user features as node features and the user interaction features as edge features.

3 PROPOSED METHOD: CFCHURN

Generally speaking, there are two major types of reasons for user churn [20]. One originates from ones' endogenous intentions, e.g., one lost interest in a service. The other comes from the social influence from one's social connections, which is exogenous. For example, one may stop using a service because he/she feels pressure to conform when most of his/her friends stop using it [3]. We use a causal graph to illustrate the causal relationships between these two causes and churn in Figure 1(a).

We build our model CFChurn based on this causal graph, and its architecture is shown in Figure 1(b). Specifically, CFChurn is based on a backbone that utilizes two separate embeddings to model users' endogenous churn intentions and the exogenous social influence. With the two embeddings as inputs, the proposed counterfactual modeling framework contains two modules: the counterfactual data augmentation module and the three-headed counterfactual prediction module. The augmentation module solves the challenge of lacking counterfactual observations and transforms the counterfactual learning problem into a supervised learning problem with partially labeled data. The prediction module is designed to tackle the challenge of learning the causal effects. Based on the sufficiency of the propensity score theory, it combines the augmented counterfactual data and the original data to train the two embeddings with three tasks and a novel causal regularizer. In the following sections, we first introduce the backbone framework and then elaborate on the counterfactual modeling framework.

3.1 Backbone Model

The exogenous social influence and endogenous user intentions are two major reasons for user churn. In this paper, we model them by two separate embeddings to capture different sources of information. A novel design - SGAT - is proposed to model the relationships between users better so that the embedding it learns can potentially capture social influence information. This section introduces how we get the node embeddings and edge embeddings from the raw features, and then we elaborate on how the model learns the social influence embeddings and user intention embeddings.

3.1.1 *Inputs and Feature Embeddings.*

Our model first takes the user node features $\boldsymbol{X_v}$ and edge features $\boldsymbol{X_e}$ as inputs and transforms them into embeddings. Since we model users as nodes, we suppose the node embedding $\boldsymbol{H_v}$ to contain all the user information. Thus, in addition to two fully connected layers, we also use two GCN layers to model their social relationships automatically so that we do not have to do feature engineering to extract network features, such as the nodes' degree manually. This process can be formulated as follows,

$$\begin{aligned} \boldsymbol{H}_{v_0} &= \sigma\left(\boldsymbol{W}_v^1\sigma\left(\boldsymbol{W}_v^0\boldsymbol{X}_v+\boldsymbol{b}_v^0\right)+\boldsymbol{b}_v^1\right),\\ \boldsymbol{H_g} &= \sigma(\hat{\boldsymbol{A}}\boldsymbol{H}_{v_0}\boldsymbol{W_g}),\\ \boldsymbol{H_v} &= \boldsymbol{H_g}||\boldsymbol{H}_{v_0}, \end{aligned} \tag{2}$$

where $\hat{\boldsymbol{A}} = \boldsymbol{D}^{-1/2}\boldsymbol{A}\boldsymbol{D}^{-1/2} + \boldsymbol{I}$, $\boldsymbol{D} \in \mathbb{R}^{N\times N}$ is the degree matrix of the graph, $\boldsymbol{I} \in \mathbb{R}^{N\times N}$ is an identity matrix. $\boldsymbol{H}_{v_0} \in \mathbb{R}^{n_v\times N}$, with n_v as the dimension of the node embeddings. $\boldsymbol{W}_v^0$, $\boldsymbol{W}_v^1$, $\boldsymbol{W_g}$, $\boldsymbol{b}_v^0$, and $\boldsymbol{b}_v^1$ are model parameters. $(\cdot||\cdot)$ denotes concatenation, and $\sigma(\cdot)$ denotes a non-linear activation function.

We suppose the edge embeddings $\boldsymbol{H_e}$ to records all the user interaction information. Thus, we apply two fully connected layers on the edge features, which can be formulated as follows,

$$\boldsymbol{H_e} = \sigma\left(\boldsymbol{W}_e^1\sigma\left(\boldsymbol{W}_e^0\boldsymbol{X_e}+\boldsymbol{b}_e^0\right)+\boldsymbol{b}_e^1\right), \tag{3}$$

where $\boldsymbol{H_e} \in \mathbb{R}^{n_e\times N}$ with n_e as the dimension of the edge embeddings. $\boldsymbol{W}_e^0$, $\boldsymbol{W}_e^1$, $\boldsymbol{b}_e^0$, and $\boldsymbol{b}_e^1$ are model parameters.

3.1.2 *Learning User Intention Embeddings and User Social Influence Embeddings.*

Users' own intentions can be inferred from three types of information, including who they are, what they have done, and what friends they have. All the above information is included in the node embeddings, and thus we simply use two fully connected layers to learn the user intention embeddings $\boldsymbol{H_{UI}}$.

To model peer influence, Graph Neural Network (GCN) is a common choice [21, 24]. However, the original GCN model cannot effectively utilize the interaction information between users [11], which is one of the most important indicators of social influence. To deal with this problem, we propose a social interaction guided graph attentional network (SGAT), which modifies the propagation step of GAT [31] to capture the information of user interactions. We show the details of the proposed propagation mechanism in Figure 2. Specifically, for each pair of connected nodes, SGAT takes the node embeddings $\boldsymbol{h}_{v_i}, \boldsymbol{h}_{v_j}$ and the embedding $\boldsymbol{h}_{e_{ij}}$ of the edge that connects the two nodes as inputs and learns a vector $\boldsymbol{\epsilon_{ij}}$ to capture peer influence, which can be formulated as follows,

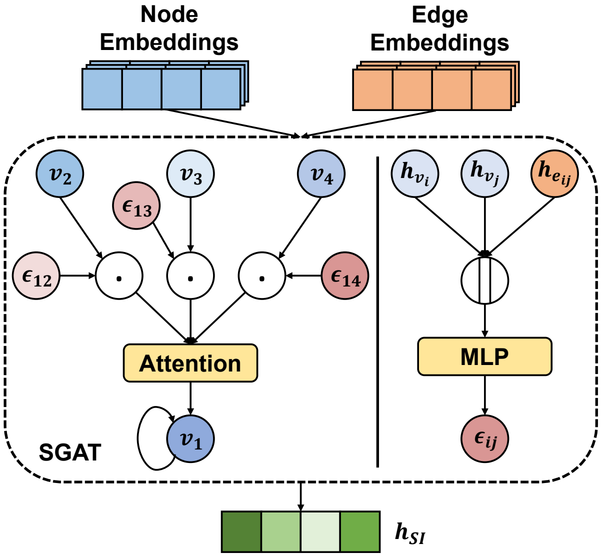


Figure 2: A detailed illustration of the architecture of the social interaction guided graph attentional network (SGAT).

$$\boldsymbol{\epsilon_{ij}} = \sigma\left(\boldsymbol{W_\epsilon}\left(\boldsymbol{h}_{v_i}||\boldsymbol{h}_{v_j}||\boldsymbol{h}_{e_{ij}}\right)+\boldsymbol{b_\epsilon}\right), \tag{4}$$

where $\boldsymbol{W_\epsilon} \in \mathbb{R}^{n_\epsilon\times(n_e+2\times n_v)}$ and $\boldsymbol{b_\epsilon} \in \mathbb{R}^{n_\epsilon\times 1}$ are model parameters with n_ϵ as the dimension of the learned vector. Note here $\boldsymbol{\epsilon_{ij}} \neq \boldsymbol{\epsilon_{ji}}$, which portrays the unbalanced mutual influences.

Then, the model takes the learned vector and the original node embedding as the input of a standard graph attentional network to updates the hidden state $\boldsymbol{h}_{v_i}^{l}$ of node v_i in layer l. Intuitively, this step captures the potential different effects of influence from different friends by an attention mechanism. Formally, the propagation step can be formulated as follows,

$$\begin{aligned} \boldsymbol{h}_{v_i}^{l+1} &= \alpha_{ii}^{l}\boldsymbol{W}_s^{l}\boldsymbol{h}_{v_i}^{l} + \sum_{j\in\mathcal{N}(i)} \alpha_{ij}^{l}\boldsymbol{W}_s^{l}(\boldsymbol{\epsilon}_{ij}^{l}\odot\boldsymbol{h}_{v_j}^{l}),\\ \alpha_{ij}^{l} &= \mathrm{softmax}(\sigma(\boldsymbol{a}^T[\boldsymbol{W}_s^{l}\boldsymbol{h}_{v_i}^{l}||\boldsymbol{W}_s^{l}(\boldsymbol{\epsilon}_{ij}^{l}\odot\boldsymbol{h}_{v_j}^{l})])), \end{aligned} \tag{5}$$

where $\odot$ refers to an element-wise multiplication. $\mathcal{N}(i)$ is the set that contains all the neighbors of the user node i. $\boldsymbol{W}_s^{l}$ and $\boldsymbol{a}$ are model parameters. In our framework, we stack two SGAT layers to enhance the representation power of the model, and the output of the SGAT is the social influence embedding $\boldsymbol{H_{SI}}$.

3.2 Counterfactual Modeling Framework

Before going into the details of our counterfactual modeling framework, we first introduce preliminaries on the causal notations we use for clarity. In this work, we adopt the notations from the most commonly used causal inference framework - the potential outcomes framework [23, 27]. Basically, it links the causal effect on an individual to the difference between the outcomes that would be observed with versus without intervention or treatment.

In our case, we define the treatment t as whether a user has churned friends and the potential outcome y as whether a user will churn in a future period. If a user has churned friends, and he/she churned in the next period, his churn could be potentially attributed to the social influence from his/her friends, and we want our model to distinguish whether he/she is influenced or not. Note that here, we follow prior work to simplify the problem by presuming that social influence on churn originates from the churn of ones' friends [20]. We denote the potential outcome under $t = 0$ as y_{t_0} and the potential outcome under $t = 1$ as y_{t_1}. Since an individual can either have churned friends ($t = 1$) or not ($t = 0$), we can only observe the outcomes under the treatment that has taken place. We denote the outcomes we observe in our data as the factual outcomes y_f and the unobserved outcomes as the counterfactual outcomes y_{cf}. Formally, the relationships between the variables mentioned above are as follows,

$$\begin{aligned} y_f &= t \times y_{t_1} + (1 - t) \times y_{t_0}, \\ y_{cf} &= t \times y_{t_0} + (1 - t) \times y_{t_1}. \end{aligned} \tag{6}$$

3.2.1 *Counterfactual Data Augmentation*.

One of the major challenges for deep learning models to learn causality is the absence of counterfactual data. Existing methods that use causal information to promote the performance of deep learning models typically learn causal information by estimating the counterfactual data distribution from the factual data, and then they use the estimated causal effects as an additional input of the predictor [16]. However, due to the general existence of selection bias that the distribution of the factual outcomes and counterfactual outcomes differs significantly, the observed factual outcomes contain limited information about the counterfactual distribution. Further, this method is also not an end-to-end learning framework, and thus the performance of deep learning models could be constrained.

Inspired by an in-depth analysis of the customer churn problem, we propose to solve the challenge from the root - by counterfactual data augmentation based on causal knowledge. Specifically, we make an assumption that generally applies to user churn scenarios, which can be formally stated as follows,

Assumption: Users' churn probability when they have churned friends is not less than the churn probability when they do not controlling other conditions to be the same, i.e.,

$$P(y_{t_1}) >= P(y_{t_0}). \tag{7}$$

This assumption holds conditioned on the causal graph because for an individual, removing a cause will only reduce his/her churn probability controlling other variables. Based on this assumption, we further derive two corollaries that supplement us with counterfactual data, which we formally stated as follows:

Corollary 1: If a user churns ($y_f = 1$) when none of his/her friends churn ($t = 0$), then the user would also churn ($y_{cf} = 1$) if he/she had churned neighbors ($t = 1$) controlling other conditions.

Corollary 2: If a user does not churn ($y_f = 0$) when he/she has churned friends ($t = 1$), then the user would not churn ($y_{cf} = 0$) if he/she had no churned neighbors ($t = 0$) controlling other conditions.

These two corollaries provide us with counterfactual labels based on prior causal knowledge on customer churn, which can be formulated as the following equation,

$$\begin{aligned} O_{cf1} &= \left\{ y^i_{cf} = y^i_{t_1} = 1 | y^i_f = y^i_{t_0} = 1, i \in O_f \right\}, \\ O_{cf2} &= \left\{ y^i_{cf} = y^i_{t_0} = 0 | y^i_f = y^i_{t_1} = 0, i \in O_f \right\}, \\ O_{cf} &= O_{cf1} \cup O_{cf2}, \end{aligned} \tag{8}$$

where O_{cf1} and O_{cf2} represent the counterfactual datasets we construct according to the two corollaries, and O_f represents the dataset we observe. In this way, we can transfer the counterfactual prediction problem into a supervised learning problem with partially labeled data.

3.2.2 *Three-headed Counterfactual Prediction*.

With the counterfactual augmented dataset, we design a framework that predicts the factual outcomes and the counterfactual outcomes simultaneously to facilitate our model to learn the causal information and provide explainalble predictions. For example, we expect the model can answer counterfactual questions such as would a user stay if he/she was not influenced by his/her friends by providing the counterfactual predictions. Since it is a binary prediction problem, we use the binary cross entropy loss as the prediction loss, which can be formulated as follows:

$$\mathcal{L}_f(y_f, \hat{y}_f) = \frac{1}{N} \sum_{i \in O_f} y^i_f \log(\hat{y}^i_f) + (1 - y^i_f) \log(1 - \hat{y}^i_f), \tag{9}$$

$$\mathcal{L}_{cf}(y_{cf}, \hat{y}_{cf}) = \frac{1}{M} \sum_{i \in O_{cf}} y^i_{cf} \log(\hat{y}^i_{cf}) + (1 - y^i_{cf}) \log(1 - \hat{y}^i_{cf}), \tag{10}$$

where N, M are the number of instances in dataset O_f and O_{cf}, respectively.

In order to facilitate the causal information learning process, we further design two components. First, we introduce a causal regularizer based on our causal assumption. Specifically, we enforce the model prediction in the branch with treatment to be not less than the prediction in the branch without treatment, which can be formulated as a regularization loss as follows,

$$\mathcal{L}_c(\hat{y}_{t_0}, \hat{y}_{t_1}) = \frac{1}{M} \sum_{i \in O_f \cup O_{cf}} \max(0, \hat{y}^i_{t_0} - \hat{y}^i_{t_1}), \tag{11}$$

Second, inspired by previous work [26], we propose to facilitate the model to learn causal information by using the social influence embedding to predict the treatment, as shown in Figure 1(b). The key intuition behind this idea originates from the theory of the sufficiency of the propensity score [22], which illustrates that if the causal effect is identifiable from observational data, it suffices to learn the causal effect from only the information that is relevant to the treatment. In other words, predicting the treatment helps the model to learn useful causal information. Thus, we construct a treatment prediction task to facilitate the social influence embedding to capture the causal information and formulate the corresponding loss as follows,

$$\mathcal{L}_t(t, \hat{t}) = \frac{1}{N+M} \sum_{i \in O_f \cup O_{cf}} t^i \log(\hat{t}^i) + (1 - t^i)\log(1 - \hat{t}^i). \quad (12)$$

The aforementioned tasks, including the factual outcome prediction task, the counterfactual outcome prediction task, and the treatment prediction task, along with the causal regularizer form the core of our proposed counterfactual prediction framework for churn prediction.

3.3 Outputs and Training

Outputs. To transform the social influence embedding and the user intention embedding into churn predictions, we first concatenate them and use a self-attention layer to fuse them together. Then, we use two fully connected layers to calculate the predictions, which can be formulated as follows,

$$\hat{\boldsymbol{y}} = \text{sigmoid}(\boldsymbol{p}^T \sigma(\boldsymbol{W_{fc}} \text{attention}(\boldsymbol{H_{SI}} || \boldsymbol{H_{UI}}) + \boldsymbol{b_{fc}})), \quad (13)$$

where $\boldsymbol{W_{fc}}, \boldsymbol{b_{fc}}, \boldsymbol{p}$ are model parameters. Note that both the factual outcomes and the counterfactual outcomes use this predictor. To predict the treatment, we feed the social influence embedding to a single fully connected layer.

Training. As illustrated in Section 3.2.2, we train our model to predict the factual outcomes, counterfactual outcomes, and treatments simultaneously. Note that the training data for customer churn are typically biased towards negative samples because churned users are less than stayed users, which hinders the training process. To deal with this problem, we add a weight α_d to reward the model to pay more attention to the minority class following prior work on churn prediction [38]. Thus, the objective function can be defined as follows,

$$\mathcal{L} = (1 + y_f \times \alpha_d) \times (\mathcal{L}_f + \alpha_{cf}\mathcal{L}_{cf}) + \alpha_t \mathcal{L}_t + \alpha_c \mathcal{L}_c, \quad (14)$$

where α_{cf}, α_t, and α_c are hyper-parameters that make a balance between different tasks.

4 EXPERIMENTS

To comprehensively evaluate our proposed method, we conduct extensive experiments on two large-scale real-world datasets to answer the following research questions:

- **Q1**: How is the overall prediction performance of CFChurn compared with state-of-the-art methods?
- **Q2**: How do different parts of the counterfactual prediction framework, including the SGAT, the counterfactual data augmentation, and the three-headed counterfactual prediction framework with causal regularizer, contribute to the performance?
- **Q3**: How well does CFChurn learn the causal information, and how does it facilitate the prediction performance?

4.1 Experiment Setup

4.1.1 ***Datasets.*** We evaluate our proposed model on two large-scale real-world datasets that differ in scale, type of social connections, and social network density. Here, we briefly introduce them as follows:

- Beidian: We collect a dataset from a leading social e-commerce platform in China, Beidian, where users can share items with their friends. This dataset covers users' demographic data, historical activity data, and social interaction data based on sharing, along with the users' social relationships on the platform from 01/2019 to 12/2019. We determine users' churn based on a threshold derived from the maximum consecutive logins a user, following prior work's approach [15]. Since 90% of the maximum time between two consecutive logins of users is within 180 days, we regard users without any login for 180 days as churned users. In this way, we can accurately determine the churn time of users whose last login is before 06/2019. Thus, we predict whether a user will churn between 04/2019 and 06/2019 given the data from 01/2019 to 03/2019.
- Epinions: A public dataset[1] from a product review website with 11 years' data [28]. Users can rate products on this website and choose to trust others' reviews. We use the mutual trust relationship between users to form a 'trust' social network, and we use historical review records to generate six basic user features, including the number of times that one trusts others, the number of times one is trusted, the number of times one rates, the number of categories of items one rates, the mean rating score, and the mean rating helpfulness score. Using the same way for the first dataset, we get a maximum re-login time of three years and regard those without any login for three years as churned users. We use the data in the first three years to predict whether one will churn in the 4-6th year.

Statistics	Beidian	Epinions
The number of users	20611	6585
The number of churned users	8241	3457
The number of users' friends	154384	19414
The number of social relationships	2112274	40916
Average degree	20.60	5.24
Average clustering coefficient	0.2670	0.0752

Table 1: The basic statistics of our datasets.

The statistical summaries of these two datasets are reported in Table 1. As we can see, the social network of the Beidian dataset is much larger and denser than that of the Epinions dataset. Denser social networks typically have stronger social influence. Further, the social connections on Epinions are essentially different from the friend relationships on Beidian since users who trust each other's reviews are not necessarily friends in reality. This suggests that the social connections on Epinions are not as strong as those on Beidian.

4.1.2 ***Baseline Methods.*** We compare the performance of CFChurn with state-of-the-art methods from four research lines with minimum modification to adapt them to our problem.

Diffusion-based Methods:

- SPA [7]: A classical diffusion model based on spreading activation techniques.
- IR/SR [36]: A propagation model based on an infection rule (IR) and a stopping rule (SR). The IR is defined as a probability threshold of churn, and the SR is defined as the number of propagations.

[1] https://www.cse.msu.edu/ tangjili/datasetcode/truststudy.htm

		Beidian		Epinions	
Groups	Models	AUC	Accuracy	AUC	Accuracy
Diffusion-based Models	SPA [7]	0.595 ± 0.000	0.619 ± 0.000	0.583 ± 0.000	0.584 ± 0.000
	IR/SR [36]	0.671 ± 0.000	0.663 ± 0.000	0.664 ± 0.000	0.668 ± 0.000
Social Influence Embedding Models	Inf2vec [10]	0.503 ± 0.012	0.503 ± 0.012	0.520 ± 0.009	0.521 ± 0.010
	Inf-VAE [24]	0.505 ± 0.007	0.569 ± 0.068	0.542 ± 0.007	0.585 ± 0.029
GCN-based Models	GCN [11]	0.705 ± 0.003	0.682 ± 0.002	0.645 ± 0.004	0.645 ± 0.004
	GAT [31]	0.707 ± 0.003	0.687 ± 0.002	0.648 ± 0.003	0.646 ± 0.004
SOTA Churn Models	RF [12]	0.692 ± 0.000	0.690 ± 0.000	0.702 ± 0.000	0.705 ± 0.000
	XGBoost [1]	0.696 ± 0.000	0.698 ± 0.000	0.714 ± 0.000	0.716 ± 0.000
	ClusChurn [33]	0.641 ± 0.013	0.633 ± 0.004	0.685 ± 0.004	0.686 ± 0.003
	FIN [29]	0.665 ± 0.015	0.611 ± 0.011	0.644 ± 0.008	0.649 ± 0.007
	Survival Model [17]	0.670 ± 0.000	0.659 ± 0.000	0.654 ± 0.000	0.652 ± 0.000
Ours	**CFChurn**	**0.729 ± 0.003**	**0.706 ± 0.002**	**0.723 ± 0.002**	**0.724 ± 0.002**

Table 2: The performance evaluation results on Beidian and Epinions datasets.

Social Influence Embedding Methods: We compare our method with two state-of-art social influence embedding methods that originally designed for diffusion prediction, i.e., predicting the set of users in the future diffusion path given the seed set of activated users. To fit them into our scenario, we set the churned users in the period of the training set as the seed set and regards the models' prediction outcome as the predicted churned users.

- Inf-VAE [24]: A variational autoencoder framework that jointly models homophily and influence.
- Inf2vec [10]: A latent representation method for social influence embedding.

SOTA Churn Methods. For a fair comparison, we add network-specific features, including the degree, clustering coefficient, average demographics of friends, and average network features of friends, into these models' inputs.

- Random Forest (RF) [12]: An ensemble learning method based on decision trees.
- XGboost [1]: An advanced tree-based boosting method.
- ClusChurn [33]: A two-step method based that first clusters users into groups and uses different LSTM for different groups for prediction.
- FIN [29]: A feature interaction network based on factorization machine.
- Survival Model [17]: A state-of-the-art statistical model for churn prediction. We use a cox proportional hazards regression model for prediction.

GCN-based Methods. Prior work utilizes Graph Neural Networks to model social influence in social recommendation systems and achieves a good performance [9]. We adapt it to the user churn prediction problem by feeding the node embeddings of CFChurn to two stacked GCN layers followed by two fully connected layers for prediction. We use two variants as our baselines:

- GCN [11]: A current state-of-the-art variant of GCN that can efficiently learn from graph-structured data.
- GAT [31]: A variant of GCN that utilizes an attention mechanism to update the embeddings of each node.

4.1.3 ***Evaluation Protocols and Reproducibility.*** We perform 5-fold cross-validation and use two widely used metrics in prior work, including area under the ROC curve (AUC) and accuracy for evaluation [1, 33], and we perform a grid search on hyper-parameters, including the learning rate, batch size, l_2 regularization coefficient, dropout rate, and all the trade-off weights of the multi-task loss function to find the best parameters for CFChurn and all the baselines. For reproducibility, we make our implementation codes of CFChurn avalible[2].

4.2 Overall Performance Comparison (Q1)

To examine whether modeling causality can make the churn prediction model more accurate, we compare CFChurn with different types of state-of-the-art baselines and show their prediction performance on two large-scale datasets in Table 2. Here, we summarize key observations and insights as follows:

- **CFChurn's superior performance**: Our proposed model CFChurn outperforms all different state-of-the-art methods across all evaluation metrics. Specifically, in terms of AUC and accuracy, it provides a relative performance gain of 3.1% and 1.1% on the Beidian dataset and 1.3% and 1.1% on the Epinions dataset, respectively, comparing to the best baseline. All improvements are statistically significant, which demonstrates the effectiveness of our model. Further, AUC reflects the model's ability to give high predicted churn probability to those who actually churned, while accuracy reflects the model's overall ability to distinguish those who churn and those who do not. Our experiments show that baseline models typically make a trade-off between these two metrics, while our model has a consistent improvement, which further demonstrates our model's effectiveness.
- **CFChurn's robustness across different scenarios**: Comparing the results between the two datasets with different scales, social relations, and social network structures, we find that many baseline models are not robust in different scenarios. For example, GCN-based methods perform significantly better on the Beidian dataset than on the Epinions dataset. A plausible reason is that Beidian's social network is much denser than Epinions'. In contrast, our model achieves the best performance in both datasets, which demonstrated our model's robustness across different scenarios.

[2] https://github.com/tsinghua-fib-lab/CFChurn

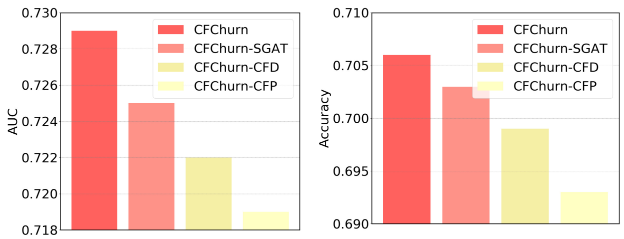


Figure 3: The ablation study on different modules of CFChurn.

	Beidian	Epinions
Right causal relationships	3428	1373
Wrong causal relationships	7	2

Table 3: Examination on whether CFChurn captures the correct causal relationships.

- **Analysis on the poor performance of diffusion-based models**: Diffusion-based models' performances are poor because they treat the sequential information of user churn as the effects of social influence, while the churn order may contain many non-causal noises. This observation indicates the importance of learning causal information for churn prediction models.
- **Analysis on the poor performance of social influence embedding models**: The poor performance of social influence embedding models can be explained by the difference between their original design context and our problem setting. They are originally proposed for diffusion prediction, which requires multiple types of diffusion that happens multiple times as inputs to enable the model to learn the social influence between users in the diffusion process. However, when we treat user churn as a diffusion process, we only have one type of diffusion, and it will only happen once.

4.3 Ablation Study (Q2)

To evaluate how different parts of our proposed counterfactual prediction framework contribute to the performance, we conduct two ablation studies with different granularities, including a coarse-grained module-level experiment and a fine-grained loss-level experiment on the Beidian dataset.

We use the coarse-grained experiment to validate the effectiveness of different modules, including the SGAT module, counterfactual data augmentation module (CFD), and three-headed counterfactual prediction module (CFP). We substitute each module with a simplified one to generate corresponding variants of CFChurn and obtain the performance gain of a module by comparing the performance of CFChurn with the simplified variants. Specifically, we design three variants, including CFChurn-SGAT, CFChurn-CFD, and CFChurn-CFP. Here, CFChurn-SGAT means substituting SGAT with a simple GAT, CFChurn-CFD means training CFChurn without the counterfactual data, and CFChurn-CFP means making CFChurn predict only the factual outcomes. The results are shown in Figure 3, and we have two observations. First, removing any module results in a decrease in the performance, which suggests that all the proposed components are effective. Second, removing the counterfactual prediction module results in the largest performance decrease, which suggests it to be the most effective part.

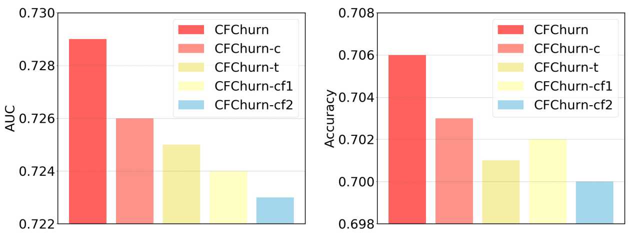


Figure 4: The ablation study on different losses of the counterfactual prediction framework.

We further conduct a fine-grained experiment to examine the effectiveness of the three proposed losses, including $\mathcal{L}_{cf}$, $\mathcal{L}_c$, and $\mathcal{L}_t$. Similarly, we examine the performance gain of a loss by comparing CFChurn with a variant trained without the loss. Specifically, we design four variants - CFChurn-cf1, CFChurn-cf2, CFChurn-c, CFChurn-t. Here, having a specific suffix name means removing the corresponding loss. Note that CFChurn-cf1 stands for removing the counterfactual dataset O_{cf1}, which is equivalent to partially removing $\mathcal{L}_{cf}$. The same applies to CFChurn-cf2. As shown in Figure 4, removing any loss results in a performance decrease, which demonstrates that every task is indispensable in the designed multi-task learning framework. A second observation is that removing any parts of $\mathcal{L}_{cf}$ results in the largest performance decrease, which suggests that the counterfactual prediction task is the most vital one for customer churn prediction with causal effects.

4.4 How Well Does CFChurn Learn the Causal Information and How does It Facilitate the Prediction Performance? (Q3)

After validating that our model provides accurate predictions, we further investigate how well does CFChurn capture the causal effects of social influence to facilitate its prediction performance. This question can be broken down into two parts: (1) Does CFChurn learn correct causal relationships? (2) How does the causal information facilitate the prediction performance? In this section, we first answer these two questions by an in-depth analysis of the predictions of CFChurn, and then we further show how the treatment prediction task helps our model learn the causal effects.

4.4.1 *Does CFChurn learn Correct Causal Relationships?*

To answer this question, we analyze CFChurn's factual predictions and counterfactual predictions on the test set and check if the predictions are in line with the prior causal knowledge, i.e., users' churn probability when they have churned friends is not less than the churn probability when they do not, controlling all other variables. In other words, the factual outcomes for users who have churned friends should be larger than the counterfactual outcomes, and for users who do not have churned friends, vice versa. As shown in Table 3, over 99.9% predictions on both datasets depict the correct causal relationship, which validates our model's ability to learn causality from data. In this way, our model can provide explainable prediction results by comparing the counterfactual outcomes

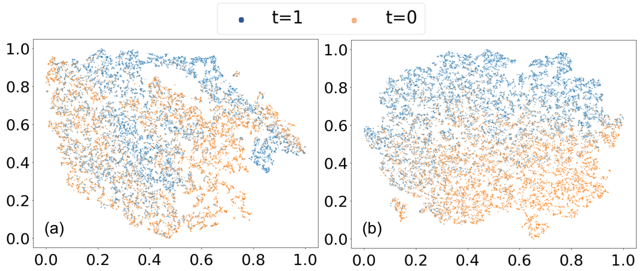


Figure 5: T-SNE plot of embedding h_{SI} of CFChurn. (a) CFChurn without the treatment prediction task. (b) CFChurn with the treatment prediction task.

with the factual outcome. For example, for a user who has churned friends, if the factual outcome predicts him/her to churn, and the counterfactual outcome predicts him/her to stay, our model indicates that the user will churn because of his friends' social influence.

4.4.2 How does the causal information facilitate the prediction performance? To answer this question, we compare the prediction results of CFChurn with its ablation version that does not contain the counterfactual data augmentation and three-headed counterfactual prediction framework. Overall, we found 84 additional correct predictions and 32 additional wrong predictions. Here, the performance gain can be attributed to the CFD and CFP module. Further, among the 84 additional correct predictions, 66 of them have their factual prediction differing from their counterfactual prediction, which means that our model suggests these users' behaviors are influenced by their friends. Put differently, most of the addition correct predictions can be owing to the learning of causal information.

4.4.3 The Effectiveness of the Treatment Prediction Task. We design the treatment prediction task based on the sufficiency of the propensity score theory. Our ablation study validated that it improves the model performance, and here we examine how it works. According to the theory, if the causal effect is identifiable from observational data, it suffices to learn the causal effect from only the information that is relevant to the treatment. In other words, the ability to distinguish the treatment is equivalent to the ability to learn useful information that captures the causal effects. Following this lead, we project h_{SI} into a low-dimensional space and visualize it by t-SNE [30] to see whether the treatment prediction task helps h_{SI} learn useful causal information. As shown in Figure 5, the embeddings without the treatment prediction task are mixed together, while they are separated according to the treatment with the task, which confirms the effectiveness of the treatment prediction task.

5 RELATED WORK

5.1 Modeling Social Influence for Churn Prediction

Modeling the effects of social influence on user churn to improve the prediction performance has been a recent trend [1, 12, 19, 33]. Existing methods mainly approach this problem in two ways. One is to model the network structure as a proxy of social influence. For example, Ahmad et al. [1] use social network analysis to derive network-based features for machine learning model. Yang et al. [33] use network features to cluster users in different groups and predict customer churn with a deep learning model. The other approach is to model the sequential order of churn as a diffusion process and use propagation models such as IR/SR [36] and SPA [7] to simulate the diffusion process and give predictions. However, these two approaches failed to capture the causal nature of social influence. To bridge this gap, this paper tackles the user churn prediction problem from a causal standpoint for the first time, and we provide a counterfactual prediction framework that can effectively capture the causal effects of the social influence on churn.

5.2 Learning Causality in Deep Learning Models

Recently, researchers begin to investigate how to use causal information to build better deep learning models [4, 18, 25, 34, 37]. Applications include eliminating the bias between the observed data and the application scenarios [4, 37] and learning the causal effects to give more accurate predictions [13, 25]. Our work is closely related to the latter one.

One of the major challenges for deep learning models to learn causal information is the lack of counterfactual data. Existing solutions mainly deal with this problem from the perspective of estimating the counterfactual data distribution. For example, Johansson et al. [13] propose to use domain adaption techniques to adapt the model from the factual domain to the counterfactual domain. Yoon et al. [34] propose to generate the counterfactual outcomes by a GAN framework that learns from the factual observations. However, these solutions ignore the characteristics of the problem itself, not to mention that the observed factual outcomes contain limited information about the counterfactual distribution. This work proposes to solve the challenge of missing counterfactual data from the root - by a counterfactual data augmentation procedure based on prior causal knowledge that generally applies to different user churn scenarios.

6 CONCLUSION

In this paper, we investigate the user churn problem from a causal perspective by developing a general framework CFChurn for modeling the causal effects of social influence for user churn prediction. Based on a counterfactual data augmentation procedure and a three-headed multi-task prediction framework, CFChurn consistently outperforms state-of-the-art methods in two large-scale datasets from different application scenarios. The prediction results show that CFChurn effectively learns the correct causal relationships and causal effects, which provides insights into making targeted campaigns for user retention. A meaningful direction for future work is to extend CFChurn to learn beyond binary causal effects. Overall, this work also echos prior work's findings that introducing causal knowledge to help deep learning models to achieve better prediction performance is a promising way [4, 18, 25, 37], and calls for more attention on this research direction.

ACKNOWLEDGMENTS

This work was supported in part by The National Key Research and Development Program of China under grant 2018YFB1800804, the National Nature Science Foundation of China under U1936217, 61971267, 61972223, 61941117, 61861136003, Beijing Natural Science Foundation under L182038, Beijing National Research Center for Information Science and Technology under 20031887521, and research fund of Tsinghua University - Tencent Joint Laboratory for Internet Innovation Technology.

REFERENCES

[1] Abdelrahim Kasem Ahmad, Assef Jafar, and Kadan Aljoumaa. 2019. Customer churn prediction in telecom using machine learning in big data platform. Journal of Big Data 6, 1 (2019), 1–24.

[2] Anol Bhattacherjee. 2001. Understanding information systems continuance: An expectation-confirmation model. MIS quarterly (2001), 351–370.

[3] Sushil Bikhchandani, David Hirshleifer, and Ivo Welch. 1998. Learning from the behavior of others: Conformity, fads, and informational cascades. Journal of economic perspectives 12, 3 (1998), 151–170.

[4] Stephen Bonner and Flavian Vasile. 2018. Causal embeddings for recommendation. In Proceedings of the 12th ACM conference on recommender systems. 104–112.

[5] Pei-Yu Chen and Lorin M Hitt. 2002. Measuring switching costs and the determinants of customer retention in Internet-enabled businesses: A study of the online brokerage industry. Information systems research 13, 3 (2002), 255–274.

[6] Giovanni Luca Ciampaglia and Dario Taraborelli. 2015. MoodBar: Increasing new user retention in Wikipedia through lightweight socialization. In Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing. 734–742.

[7] Koustuv Dasgupta, Rahul Singh, Balaji Viswanathan, Dipanjan Chakraborty, Sougata Mukherjea, Amit A Nanavati, and Anupam Joshi. 2008. Social ties and their relevance to churn in mobile telecom networks. In Proceedings of the 11th international conference on Extending database technology: Advances in database technology. 668–677.

[8] Lisette De Vries, Sonja Gensler, and Peter SH Leeflang. 2017. Effects of traditional advertising and social messages on brand-building metrics and customer acquisition. Journal of Marketing 81, 5 (2017), 1–15.

[9] Wenqi Fan, Yao Ma, Qing Li, Yuan He, Eric Zhao, Jiliang Tang, and Dawei Yin. 2019. Graph neural networks for social recommendation. In The World Wide Web Conference. 417–426.

[10] Shanshan Feng, Gao Cong, Arijit Khan, Xiucheng Li, Yong Liu, and Yeow Meng Chee. 2018. Inf2vec: Latent representation model for social influence embedding. In 2018 IEEE 34th International Conference on Data Engineering (ICDE). IEEE, 941–952.

[11] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In Advances in neural information processing systems. 1024–1034.

[12] Yiqing Huang, Fangzhou Zhu, Mingxuan Yuan, Ke Deng, Yanhua Li, Bing Ni, Wenyuan Dai, Qiang Yang, and Jia Zeng. 2015. Telco churn prediction with big data. In Proceedings of the 2015 ACM SIGMOD international conference on management of data. 607–618.

[13] Fredrik Johansson, Uri Shalit, and David Sontag. 2016. Learning representations for counterfactual inference. In International conference on machine learning. PMLR, 3020–3029.

[14] Ali Khodadadi, Seyedabbas Hosseini, Ehsan Pajouheshgar, Farnam Mansouri, and Hamid R Rabiee. 2020. ChOracle: A Unified Statistical Framework for Churn Prediction. IEEE Transactions on Knowledge and Data Engineering (2020).

[15] Young D Kwon, Dimitris Chatzopoulos, Ehsan ul Haq, Raymond Chi-Wing Wong, and Pan Hui. 2019. GeoLifecycle: User Engagement of Geographical Exploration and Churn Prediction in LBSNs. Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies 3, 3 (2019), 1–29.

[16] Jia Li, Xiaowei Jia, Haoyu Yang, Vipin Kumar, Michael Steinbach, and Gyorgy Simon. 2020. Teaching deep learning causal effects improves predictive performance. arXiv preprint arXiv:2011.05466 (2020).

[17] Junxiang Lu. 2002. Predicting customer churn in the telecommunications industry—-An application of survival analysis modeling using SAS. In SAS User Group International (SUGI27) Online Proceedings, Vol. 114.

[18] James McInerney, Brian Brost, Praveen Chandar, Rishabh Mehrotra, and Benjamin Carterette. 2020. Counterfactual evaluation of slate recommendations with sequential reward interactions. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 1779–1788.

[19] Sandra Mitrović and Jochen De Weerdt. 2020. Churn modeling with probabilistic meta paths-based representation learning. Information Processing & Management 57, 2 (2020), 102052.

[20] Irit Nitzan and Barak Libai. 2011. Social effects on customer retention. Journal of Marketing 75, 6 (2011), 24–38.

[21] Jiezhong Qiu, Jian Tang, Hao Ma, Yuxiao Dong, Kuansan Wang, and Jie Tang. 2018. Deepinf: Social influence prediction with deep learning. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 2110–2119.

[22] Paul R Rosenbaum and Donald B Rubin. 1983. The central role of the propensity score in observational studies for causal effects. Biometrika 70, 1 (1983), 41–55.

[23] Donald B Rubin. 1974. Estimating causal effects of treatments in randomized and nonrandomized studies. Journal of educational Psychology 66, 5 (1974), 688.

[24] Aravind Sankar, Xinyang Zhang, Adit Krishnan, and Jiawei Han. 2020. Inf-vae: A variational autoencoder framework to integrate homophily and influence in diffusion prediction. In Proceedings of the 13th International Conference on Web Search and Data Mining. 510–518.

[25] Rebecca Sharp, Mihai Surdeanu, Peter Jansen, Peter Clark, and Michael Hammond. 2016. Creating causal embeddings for question answering with minimal supervision. arXiv preprint arXiv:1609.08097 (2016).

[26] Claudia Shi, David M Blei, and Victor Veitch. 2019. Adapting neural networks for the estimation of treatment effects. arXiv preprint arXiv:1906.02120 (2019).

[27] Jerzy Splawa-Neyman, Dorota M Dabrowska, and TP Speed. 1990. On the application of probability theory to agricultural experiments. Essay on principles. Section 9. Statist. Sci. (1990), 465–472.

[28] Jiliang Tang, Huiji Gao, Huan Liu, and Atish Das Sarma. 2012. eTrust: Understanding trust evolution in an online world. In Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining. 253–261.

[29] Qi Tang, Guoen Xia, Xianquan Zhang, and Yaxiang Li. 2020. A Feature Interaction Network for Customer Churn Prediction. In Proceedings of the 2020 12th International Conference on Machine Learning and Computing. 242–248.

[30] Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. Journal of machine learning research 9, 11 (2008).

[31] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. arXiv preprint arXiv:1710.10903 (2017).

[32] Fengli Xu, Guozhen Zhang, Yuan Yuan, Hongjia Huang, Diyi Yang, Depeng Jin, and Yong Li. 2020. Understanding the Invitation Acceptance in Agent-initiated Social E-commerce. (2020).

[33] Carl Yang, Xiaolin Shi, Luo Jie, and Jiawei Han. 2018. I Know You'll Be Back: Interpretable New User Clustering and Churn Prediction on a Mobile Social Application. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 914–922.

[34] Jinsung Yoon, James Jordon, and Mihaela Van Der Schaar. 2018. GANITE: Estimation of individualized treatment effects using generative adversarial nets. In International Conference on Learning Representations.

[35] Guozhen Zhang, Yong Li, Yuan Yuan, Fengli Xu, Hancheng Cao, Yujian Xu, and Depeng Jin. 2021. Community Value Prediction in Social E-commerce. In Proceedings of the Web Conference 2021. 2958–2967.

[36] Xiaohang Zhang, Ji Zhu, Shuhua Xu, and Yan Wan. 2012. Predicting customer churn through interpersonal influence. Knowledge-Based Systems 28 (2012), 97–104.

[37] Yu Zheng, Chen Gao, Xiang Li, Xiangnan He, Yong Li, and Depeng Jin. 2020. Disentangling user interest and popularity bias for recommendation with causal embedding. arXiv preprint arXiv:2006.11011 (2020).

[38] Bing Zhu, Bart Baesens, and Seppe KLM vanden Broucke. 2017. An empirical comparison of techniques for the class imbalance problem in churn prediction. Information sciences 408 (2017), 84–99.