2D-ATT: Causal Inference for Mobile Game Organic Installs with 2-Dimensional Attentional Neural Network

Boxiang Dong
Montclair State University
Montclair, NJ, USA
dongb@montclair.edu

Hui (Bill) Li
Jam City
Los Angeles, CA, USA
bli@jamcity.com

Yang (Ryan) Wang
Jam City
Los Angeles, CA, USA
ryan.wang@jamcity.com

Rami Safadi
Jam City
Los Angeles, CA, USA
rami@jamcity.com

***Abstract*—In the mobile gaming industry, organic installs refer to downloads that cannot be attributed to any advertising channel and thus do not introduce upfront user acquisition (UA) cost. Understanding the causal factors on organic installs is of vital importance for a game's ecosystem, as such knowledge can help bring in more organic users, who tend to be more loyal and active. A major challenge in discovering the causal effects is the potential temporal lag between an UA operation and the growth in organic installs. In this paper, we solve the problem by using a deep attentional neural network to analyze multivariate time series data. The core of our design is a novel attention mechanism, namely 2D-ATT, that can learn the contribution of each feature to the target at different levels of temporal delay. Our experiments on a series of synthetic datasets show that 2D-ATT outperforms existing approaches for discovering complex causal effects. We also use 2D-ATT to analyze a real-world mobile game dataset collected by *Jam City*, a video game company based in California. Our discoveries provide valuable insights to UA operations.**

***Index Terms*—causal inference, attention mechanism, deep learning**

I. Introduction

Since the dawn of smartphones, the mobile gaming industry has witnessed a sharp growth in value. Recent market studies report that mobile games take 51% of the total global gaming industry revenue, leaving console and PC games far behind[1]. By 2021, the mobile gaming market is projected to reach $180 billion. Approximately 3,000 new mobile games are launched in iOS App Store and Google Play Store everyday. The intense competition in the industry exhorts the importance of user acquisition (UA) in mobile gaming operations. Depending on the source of user acquisition, a new app download must fall into one of two categories: *paid install*, i.e., a user obtained by advertising, or *organic install*, i.e., a download that cannot be attributed to any advertisement source.

Organic installs are of crucial value for a mobile game's ecosystem, as organic users tend to be more loyal and active. Besides, there is no upfront UA cost associated with organic installs. In mobile gaming in general, the in-game lifetime value of most paid users cannot even compensate for their UA expense. Therefore, in order for a mobile game to survive in such a competitive market, it is essential to bring in more organic users. This calls for an analysis to discover the driving forces of organic installs, so that UA budgets can be allocated intelligently.

However, it is challenging to distill the causal factors of organic installs. Firstly, there are quite a few aspects that can possibly play a role in capturing organic users, e.g., game quality, app visibility in the app stores, and in-game social referrals. It is difficult to quantitatively evaluate the influences of these factors. More importantly, there exist potential temporal lags between an UA operation and organic installs. For example, when the developer improves the game quality by upgrading the match-making algorithm, we do not observe an immediate growth in the organic installs. This is because it takes time for existing users to feel the improved gaming experience and recommend it to friends, so as to trigger more organic installs. It is necessary to discover the temporal lags of different causal factors, as these insights are crucial for working out a UA strategy that can facilitate the achievement of both short- and long-term objectives.

There are quite a few existing works that focus on uncovering the causal relationships from observational data [13], [17], [32]. However, none of them can be applied to our work. Most statistic inference approaches [13], [17] pose rigid assumptions on the data, e.g., no confounder or additive generative model, which significantly hinders their applicability in real-world datasets. Besides, they cannot identify the temporal lags between the causal factors and the target variable. Recently, attention mechanisms supplemented the "black-box" neural networks with the ability to explain the decisions [2], [16]. However, when dealing with multivariate time series data, most works simply blend the information of all variables into a single hidden state by using recurrent units. It is intractable to distinguish the contribution of individual variables to the prediction through the sequence of hidden states [32].

In this paper, we aim at discovering the causal factors with temporal lags on mobile game organic installs from multivariate time series observational data. To resolve the problem, we firstly capture the temporal dynamics of each

[1]https://www.blog.udonis.co/mobile-marketing/mobile-games/mobile-gaming-statistics

feature/variable by using a deep recurrent neural network. Then we propose a novel attention mechanism named *2D-ATT* that works on the two-dimensional hidden states. By resembling the information flow in the generative process of the target variable, i.e., organic installs, it quantifies the contribution of each feature at every time step. Therefore, *2D-ATT* discovers the causal impact of each factor with different levels of temporal delay.

To the best of our knowledge, we are the first to quantitatively evaluate the causal impact of different factors with temporal lags on organic installs in the mobile game industry. In this paper, we make the following contributions.

- We design a deep recurrent neural network to learn the dynamics of each feature and produce feature-wise subsequence summaries.
- We propose an innovative attention mechanism to learn the importance of each feature with different levels of temporal lag.
- We generate a series of synthetic datasets with various causal structures. The experiment results demonstrate that 2D-ATT accurately discovers causal effects and outperforms 4 baseline approaches.
- We execute 2D-ATT on a real-world mobile game dataset to shed light on organic installs. The causal relationships discovered by 2D-ATT can provide valuable insights to UA operations.

The rest of the paper is organized as follows. Section II illustrates the preliminaries. Section III discusses the detailed method. Section IV presents the experimental findings. Section V introduces related work and Section VI concludes the paper.

II. Preliminaries

In this section, we first elaborate on organic installs and the strategies to boost them. After that, we briefly go over the literature on causal inference.

A. Organic Installs

There are only two ways for mobile game studios to drive downloads of their apps – either organically or through payment. A paid install is directly incentivized by advertising. For example, a user who sees an advert on a social media platform, interacts with it, and later installs the app is considered as a paid install. On the other hand, organic installs refer to the scenario where a user installs an app without directly responding to a mobile advertising campaign. In practice, organic installs are vital for a mobile game's ecosystem, as they bring in loyal and active users. Recent study suggests that the organic users outshine their paid counterparts with 14.8% higher one-day retention rate and 10% more daily sessions[2]. Another important business reason for driving organic installs is the *user acquisition (UA)* budget. Though no organic install is likely to be truly free, organic installs can save on the *cost-per-install (CPI)* considerably, as they are not paid for at the point of install.

[2]https://www.adjust.com/blog/benchmarks-deep-dive-2-paid-vs-unpaid/

Next we introduce three ways to boost organic app installs that are widely recognized in the mobile game industry.

- App store optimization. It is the process of improving app visibility within the app stores, including keywords update, app icon update, and new screenshot publication. Nearly 70% of iOS App Store visitors use search to find new apps. A successful optimization facilitates the app to pop up in the App Store's search results, so as to fascinate more install traffic.
- Encouraging social exposure. The social nature of human beings imply that people trust their friends' and family's recommendations. It is beneficial to prompt current users to share the app across social platforms. Some apps even reward referrals with in-game credits.
- Game quality improvement. The ultimate driving force of organic installs is the game quality. When app developers enhance in-game user experience, an increase in user retention rate and more active daily users are expected. Besides, there are going to be more positive app reviews and higher media visibility, which produces a larger number of organics in the long term.

Note that, in practice, these efforts may take some time to cause an impact on the organic installs instead of an immediate response. For example, when we improve the game quality by launching a new game matching algorithm, the existing users may take a few days to explore the game and recommend it to his/her friends or family members. Therefore, a time lag between the UA operation and the growth in organics is possible.

B. Causal Inference

Causal inference is the process of drawing a conclusion about a causal connection based on the conditions of the occurrence of an effect. By definition, correlation, which describes the observation where two or more variables show the same increasing or decreasing trend, seems close to causality. However, it is well known that "correlation does not imply causation" [1]. For instance, people in the United States spend more on shopping when it is cold does not imply cold weather causes expenditure. To address this problem, the counterfactual framework was proposed to invoke retrospection. It measures causal effects by comparing outcomes in two almost-identical worlds, where two parallel universes are identical in every aspect until a certain point when difference is introduced in one controlled variable. The subsequent disparity in the outcome describes the causal impact of the controlled variable. The gold standard to discover counterfactual causal relations is to perform experiments [18]. However, in many cases, experiments are expensive, unethical, and even impossible to realize. In this paper, we focus on observational causal inference, i.e., infer the casual structure of the data generating system purely from the observational data. In particular, we quantify the contribution of each feature at each time step to the prediction of target variable via a novel attentional neural network.

III. METHODOLOGY

In this section, we first formally define the problem, and then present our solution in details.

A. Problem Statement

Given a set of heterogeneous multivariate time series $D = \{(\boldsymbol{X}_i, y_i)\}_{i=1}^{n}$ collected from n different sources, where each $\boldsymbol{X}_i$ consists of m features of length T, i.e., $\boldsymbol{X}_i = (\boldsymbol{x}_{i,1}, \ldots, \boldsymbol{x}_{i,m})^\intercal \in \mathbb{R}^{m \times T}$, $\boldsymbol{x}_{i,j} = (x_{i,j,1}, \ldots, x_{i,j,T})^\intercal \in \mathbb{R}^\intercal$ represents the measurement of the j-th feature across T time steps, and y_i is the target value at the T-th time step. The objective is to construct a generative model to discover the causal impact of $x_{i,j,t}$ on the target y_i for $i = 1, \ldots, n$, $j = 1, \ldots, m$, and $t = 1, \ldots, T$.

In our setting of mobile game apps, these time series can be obtained from different games, with the features representing the daily UA cost, the number of app store featuring events, the number of paid users, the number of active users, and hashtag counts (i.e., the number of tweets that has a hashtag related to the game) across a fixed time span, e.g., one year or a couple of months. Naturally, y_i represents the number of organic installs on the last day. With the assistance of the model, we will be able to quantify the causal effect of the different features on the organic installs with a temporal lag [9].

B. Solution in a Nutshell

To solve the problem, we design a deep attentional neural network to simultaneously learn the mapping from the multivariate input sequences to the target, and uncover the causal impact across features and time steps. Figure 1 displays the architecture of the network. In particular, the network consists of a deep recurrent neural network (RNN) that computes the hidden state from the input time series, a two-dimensional attention layer (2D-ATT) that quantifies the contribution of the input features to the target, and a fully connected network (FCN) that conducts the nonlinear transformation and calculates the predicated target value. Next, we briefly introduce the functionality of each component.

- **Deep RNN** Given a sequence of features, i.e., $\boldsymbol{x}_{i,j} = (x_{i,j,1}, \ldots, x_{i,j,T})^\intercal$, for any $t = 1, \ldots, T$, the deep RNN computes a compact hidden state $h_{i,j,t}$ to summarize the subsequence until t. By stacking multiple layers of gated LSTM units, the deep RNN can captures the temporal patterns at various granularity, and produce a comprehensive summary of the observed features.
- **2D-ATT** We propose a novel attention mechanism that works on the two-dimensional hidden states across different features and time steps. The attention mechanism evaluates the contribution of each value to the prediction of the target, and reveals the information flow in the generative process.
- **FCN** The FCN takes the synopsis from the previous layer, and predicts the target value with multiple fully-connected layers of hidden units.

In the rest of the section, we discuss the deep RNN and 2D-ATT in details.

C. Deep RNN

Recurrent neural networks (RNNs) are specialized for processing sequences of variable length and incorporating temporal dynamics. Given an input sequence $(x_1, ..., x_T)$, for time step $t = 1, \ldots, T$, a generic RNN computes the hidden state by

$$h_t = \mathcal{H}(x_t, h_{t-1}), \tag{1}$$

where $\mathcal{H}$ is the state transition function. The same transition function, i.e., $\mathcal{H}$, operates on all time steps and all sequences. RNN encloses any useful information about the past into the hidden state h_t . Compared with fully-connected network, RNN is more powerful in that when predicting the output at time step t, it makes use of all the available input information up to the current time frame, i.e., h_t. The RNN in Equation (1) is universal in the sense that any function computable by a Turing machine can be simulated by such a recurrent network with a finite size [27].

Recurrent Unit. In Equation (1), $\mathcal{H}$ is the operation of a recurrent neuron that computes the current hidden state h_t based on the current input value x_t and the previous hidden state h_{t-1}. Due to the well-known difficulty of training an RNN to capture long-term dependencies [19], various gated mechanisms have been proposed to combat the vanishing and exploding gradients problem. In our work, we employ long short-term memory (LSTM) units [14] as the recurrent unit.

Each LSTM unit consists of a memory cell c, an input gate i, a forget gate f, and an output gate o. The graphical illustration is presented in Figure 2. The memory cell maintains the memory content of an LSTM unit. At time t, c_t is updated by partially forgetting the existing memory c_{t-1} and adding a new memory content $\tilde{c}_t$:

$$c_t = f_t c_{t-1} + i_t \tilde{c}_t, \tag{2}$$

where the input gate i_t controls how much new content should be stored in the memory cell, the forget gate f_t indicates to which degree the old content should be obliterated, and the new content $\tilde{c}_t$ at t is computed based on the newly observed input x_t and the previous hidden state h_{t-1}. Then the hidden state h_t of the LSTM unit is computed as:

$$h_t = o_t tanh(c_t), \tag{3}$$

where the output gate o_t modulates the amount of memory content exposure.

An LSTM unit learns to operate these gates to adaptively forget, memorize and expose the memory content. At any time, if the observed input is important, the forget gate will be closed in order to carry the memory content across many time steps. On the other hand, the unit may decide to reset the memory content by opening the forget gate. Since these two modes of operations can happen simultaneously across different LSTM

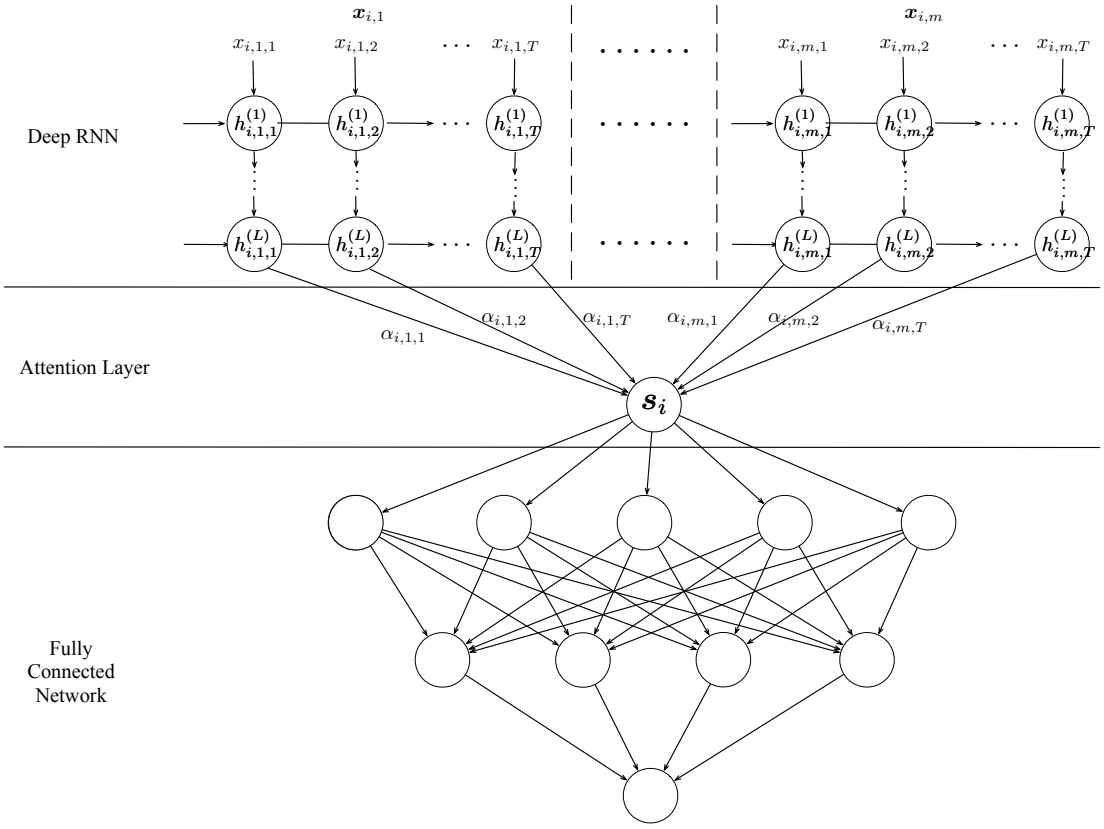

Fig. 1. The architecture of the deep attentional neural network

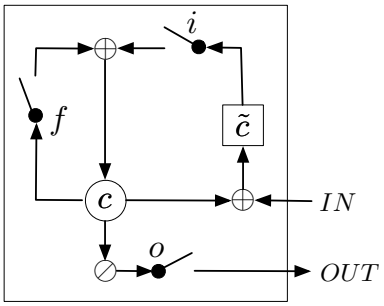

Fig. 2. Illustration of LSTM unit as presented in [5]

units, an RNN with multiple LSTM units may capture both fast-moving and slow-moving temporal patterns [6].

Deep RNN. Since a deep hierarchical model is exponentially more efficient at representing functions than a shallow one [3], we build a deep RNN by stacking multiple recurrent layers, where each layer consists of a list of recurrent units. Most importantly, unlike most existing work [4], [33] that blindly squeeze the information from all features into a single hidden state and make it intractable to quantify the contribution from individual features, at each layer, we allocate a separate set of recurrent units for each feature. In particular, the state transition in a deep RNN with L recurrent layers is defined as:

$$h_{i,j,t}^{(\ell)} = \mathcal{H}_j\big(h_{i,j,t}^{(\ell-1)}, h_{i,j,t-1}^{(\ell)}\big), \tag{4}$$

where $\ell = 1, \ldots, L$, $h_{i,j,t}^{(0)} = x_{i,j,t}$, $\mathcal{H}_j$ is the operation in LSTM dedicated for the j-th feature, and $h_{i,j,t}^{(\ell)}$ is the hidden state for the j-th feature on the ℓ-th layer at time t for the i-th sample. The hidden state in the last recurrent layer is emitted as the output of the deep RNN, i.e., $\boldsymbol{H}_i = (\boldsymbol{h}_{i,1}, \ldots, \boldsymbol{h}_{i,T})^\intercal$, and $\boldsymbol{h}_{i,t} = \big(h_{i,1,t}^{(L)}, \ldots, h_{i,m,t}^{(L)}\big)^\intercal$ for $t = 1, \ldots, T$.

By stacking multiple layers of recurrent units, we explicitly encourage each recurrent level to operate at a different timescale. This allows the deep RNN to capture the temporal patterns of various granularity, and to generate a comprehensive synopsis of the input sequence. On the other hand, the variable-designated recurrent units can learn the specific dynamics of each feature, and produce feature-wise summaries that allow subsequent causal impact analysis.

D. 2D-ATT

Attention mechanisms [2], [16], [31] offer a short-cut access to inputs at different positions based on their importance during the processing of a sequence, and provide long-term context for downstream recognition or prediction. They have become an essential component in compelling sequence modeling. However, most existing attention mechanisms [2], [16] only take single-dimensional hidden states, i.e., a sequence of hidden states with each summarizing all features until a certain time step. Whereas in our work, the hidden states emitted by the deep RNN are two-dimensional, i.e., across

features and time steps. The most relevant existing work to ours was proposed by Guo et al. [12], which describes a 2-layer attention mechanism, with one intra-feature layer and one inter-featuer layer. Unfortunately, the importance of different features at different time steps are not comparable.

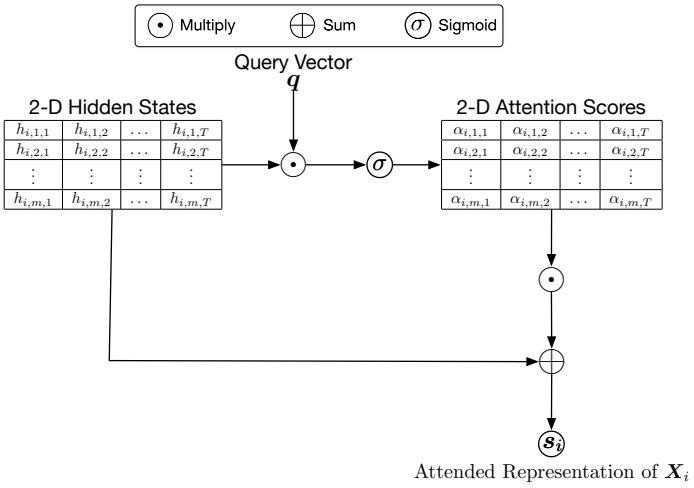

Fig. 3. The graphical illustration of the proposed 2D-ATT attention mechanism

In this paper, we propose a novel query-based self-attention mechanism named 2D-ATT. The graphical illustration of 2D-ATT is displayed in Figure 3. In particular, 2D-ATT directly processes two-dimensional hidden states, and adaptively learns the importance of each feature and each time step with regard to the prediction of the target variable. Given a multivariate time series $\boldsymbol{X}_i$, for each hidden state $h_{i,j,t}$ ($j = 1, \ldots, m$, and $t = 1, \ldots, T$), we first construct its latent representation $u_{i,j,t}$ through one fully-connected layer as

$$\boldsymbol{u}_{i,j,t} = tanh(\boldsymbol{W} h_{i,j,t} + \boldsymbol{b}), \tag{5}$$

where $\boldsymbol{W}$ and $\boldsymbol{b}$ are parameters to be learned. After that, the attention on $h_{i,j,t}$ is calculated as

$$\alpha_{i,j,t} = \frac{\boldsymbol{u}_{i,j,t}\boldsymbol{q}^\intercal}{\sum_{1\le j\le m, 1\le t\le T} exp(\boldsymbol{u}_{i,j,t}\boldsymbol{q}^\intercal)}, \tag{6}$$

where $\boldsymbol{q}$ is the query vector that represents the fixed question "*what are the important hidden states with regard to the prediction of the target variable*" like the one used in the memory networks [29]. The query vector $\boldsymbol{q}$ is randomly initialized and is also learned in the training process. The softmax activation in Equation (6) ensures that $\sum_{1\le j\le m, 1\le t\le T} \alpha_{i,j,t} = 1$. Therefore, $\alpha_{i,j,t}$ represents the relative importance of the input $x_{i,j,t}$ in determining the target value of the sequence, i.e., y_i. Finally, the attended representation of the input sequence $\boldsymbol{X}_i$ is calculated by taking the weighted sum of the hidden states based on the attention scores $\alpha_{i,j,t}$,

$$\boldsymbol{s}_i = \sum_{1\le j\le m, 1\le t\le T} \alpha_{i,j,t} h_{i,j,t}. \tag{7}$$

By resembling the information flow in the generative process of the target variable, the attention scores $\{\alpha_{i,j,t} | i = 1, \ldots, n, j = 1, \ldots, m, t = 1, \ldots, T\}$ represents the contribution of each feature at every time step to the target. The aggregate/average attention scores across all instances reveal the causal impact of each feature with different levels of temporal delay. Our experimental results in Section IV also validates this claim. On the other hand, for each individual instance $\boldsymbol{X}_i$, the attention scores provided by 2D-ATT serve as the explanation of the model's prediction, and significantly boost user confidence in the model. Moreover, as the attention mechanism allows the prediction model to automatically concentrate on the most important input fragments, it can improve the prediction accuracy in return [24]. Our empirical results (Section 5) provide evidence of the effectiveness of the attention mechanism.

IV. Experiments

A. Datasets

In the experiments, we generate synthetic datasets with embedded causal relationships to evaluate the effectiveness of 2D-ATT at causal inference. We also apply 2D-ATT on real-world datasets from mobile game industry to uncover the causal impact of various factors on organic installs. Next, we introduce these datasets in details.

Synthetic 1: Linear, Uniform with instantaneous effects. We follow [7] to generate 2,000 datasets of length 30 from $X_t \sim \mathcal{U}([0, 1])$ and

$$Y_t = \begin{cases} X_{t-3} + N_{X,t} & \text{if } X_t \le 0.5 \\ f(X_{t-5}, X_{t-2}) + N_{X,t} & \text{otherwise,} \end{cases}$$

where $f(\cdot)$ is the average function, and $N_{X,t} \sim \mathcal{U}(0, 0.2)$. The causal relationship is $X_t, X_{t-2}, X_{t-3}, X_{t-5} \to Y_t$.

Synthetic 2: Linear, Gaussian with instantaneous effects. We follow [20] to sample 2,000 datasets of length 30 with

$$\begin{aligned} X_t &= A_1 * X_{t-1} + N_{X,t} \\ W_t &= A_2 * W_{t-1} + A_3 * X_t + N_{W,t} \\ Y_t &= A_4 * Y_{t-1} + A_5 * W_{t-1} + N_{Y,t} \\ Z_t &= A_6 * W_t + + A_7 * Y_{t-1} + N_{Z,t}, \end{aligned}$$

where $N_{\cdot,t} \sim 0.4 \cdot \mathcal{N}(0, 1)$, and $A_i \sim \mathcal{U}\big([-0.8, -0.2] \cup [0.2, 0.8]\big)$. We take Z as the target variable, and regard $W_t, Y_{t-1} \to Z_t$ as correct.

Synthetic 3: Nonlinear, non-Gaussian without instantaneous effects. We simulate 2,000 datasets of length 30 by following [20] with

$$\begin{aligned} X_t &= 0.8 * X_{t-1} + 0.3 * N_{X,t} \\ Y_t &= 0.4 * Y_{t-1} + (X_{t-1} - 1)^2 + 0.3 * N_{Y,t} \\ Z_t &= 0.4 * Z_{t-1} + 0.5 * cos(Y_{t-1}) + sin(Y_{t-1}) + 0.3 * N_{Z,t}, \end{aligned}$$

where $N_{\cdot,t} \sim \mathcal{U}\big([-0.5, 0.5]\big)$. Z is regarded as the target variable. Therefore, $Y_{t-1} \to Z_t$ is the ground truth causal effect.

Synthetic 4: Non-additive interaction. We follow [20] to simulate 2,000 datasets with different lengths from

$$\begin{aligned} X_t &= 0.2 * X_{t-1} + 0.9 * N_{X,t} \\ Y_t &= -0.5 + exp\big(-(X_{t-1} + X_{t-2})^2\big) + 0.1 * N_{Y,t}, \end{aligned}$$

where $N_{\cdot,t} \sim \mathcal{N}(0,1)$. Obviously, X_{t-1} and X_{t-2} are the causal factors of Y_t.

Real-world Dataset. We obtain real-world data from 9 mobile games published on two platforms (i.e., iOS and Android) by *Jam City*[3], a video game company based in California. The multivariate time series data are collected on a daily basis from December 2013 to June 2020. For each day, every game is described by the following variables.

Numerical features

- paid_install: the number of paid users obtained;
- ua_cost: the user acquisition cost;
- DAU: daily active users;
- DAP: daily active payers, i.e., users who make within-app purchases;

Binary features

- featuring: indicates whether there is an app store featuring event, i.e., if the app store displays the game in the main page;
- store_publishing: indicates whether the game's information is updated in the app store;
- test_publishing: indicates whether there is any testing for new game content in the app store;

Target variable

- organic_install: the number of organic installs.

The detailed statistics of the real-world dataset can be found in Table II and III. For the sake of commercial protection, for each game, we multiply the statistics with a fixed random number. We split the multivariate time series with a fixed time window (e.g., 7, 15, 30, 60, 180 and 360). We observe that the prediction based on 180-day historical data is reasonably accurate. Therefore, in the experiments, we only report the results from a 180-day window size.

B. Baselines

In the experiments, we compare 2D-ATT with the following approaches on causal inference.

- T-Causality [30]: It discovers causal effects by following the definition of *transfer causality*.
- G-Causality [20]: It discovers causal effects by following the definition of *Granger causality*. We use the implementation provided by Peters et al.[4], which includes various combinations of regressor (i.e., VAR, GAM and GP) and independence testor (i.e., HSIC with Gaussian kernel and cross-correlation).
- SHAP [15]: It discovers important features on a model's prediction result by approximating the local behavior with simple but interpretable explanation models. We firstly train a gradient boosting tree from the dataset to learn the mapping from features to target by using the *LightGBM* library[5], and identify important features with SHAP[6]. A feature is considered as the causal factor if its SHAP value (impact on model output) is larger than or equal to 1.
- AME [23]: This approach trains an attentional neural network to predict the target. The attention is enforced to be aligned with the G-causality values. We use the open-source implementation provided by the authors[7].

C. Setup

We implement 2D-ATT with in Python by using the *tensorflow 2.0* framework. The neural network is trained with the Adam optimizer to reduce the *mean squared error* between the predicted value and the ground truth. All the hyber-parameters, e.g., the depth and width of the RNN and FCN, the number of attention units, learning rate, the number of epochs, and batch size are fine tuned through random search. We will publish the source code upon the acceptance of the paper. We run the experiments on a workstation with 2 NVIDIA RTX 2080 Ti GPUs and 1 Intel i7-7700K @ 4.2GHz CPU. On average, the training process halts within 2 hours.

D. Causal Validation on Synthetic Datasets

In this section, we compare 2D-ATT and the baseline approaches on discovering causal effects from the synthetic datasets. Table I shows the result. First, we observe that T-Causality [30] fails to detect any causal effect. This is because T-Causality does not take instantaneous effects into consideration, which are involved in the first three synthetic datasets. Moreover, the conditional entropy calculation in T-Causality is vulnerable to noises in the observations. Likewise, G-Causality [20] does not identify any causal effect either. We must note that in the experiments, we try all combinations of regressors and independence testor provided by [20]. As pointed out in Section I, G-Causality is only effective if the data follows a rigid assumption, i.e., no confounder. Similar to T-Causality, it is also sensitive to the observational noise. Among all the baseline approaches, SHAP [15] is the most plausible one. It successfully identifies the causal effects in Synthetic 1 and 4, while only emits a few false effects and misses one effect on Synthetic 2 and 3. In our experiments, we find that the performance of SHAP heavily depends on the accuracy of the prediction which it explains. If the underlying prediction model is not appropriate (e.g., using ARIMA) or is not fine-tuned, the causal effects generated by SHAP have a low fidelity. Even though AME [23] discovers the causal effects on Synthetic 2 and 3, a major limitation is that it only produces variable-level causal effects, without any information about temporal lags. This significantly reduces the value of the uncovered effects.

In Figure 4, we visualize the attention scores generated by 2D-ATT. On Synthetic 1, we observe that 2D-ATT successfully identifies the causal effect of X_t, X_{t-2}, X_{t-5} on Y_t, even though it misses X_{t-3}. This demonstrates the extraordinary capacity of 2D-ATT on discovering causal effects with different levels of temporal lags. On Synthetic 2, 2D-ATT attributes that

[3]https://www.jamcity.com/

[4]http://people.tuebingen.mpg.de/jpeters/onlineCodeTimino.zip

[5]https://lightgbm.readthedocs.io/en/latest/

[6]https://github.com/slundberg/shap

[7]https://github.com/d909b/ame

Approach	Synthetic 1	Synthetic 2	Synthetic 3	Synthetic 4
T-Causality [30]	N/A	N/A	N/A	N/A
G-Causality [20]	N/A	N/A	N/A	N/A
SHAP [15]	$X_{t-3}, X_t, X_{t-5}, X_{t-2} \rightarrow Y_t$ ✓	$Y_{t-1} \rightarrow Z_t$ ✓ $Y_t, W_{t-2}, W_{t-1}, Y_{t-2} \rightarrow Z_t$ ✗	$Y_{t-1} \rightarrow Z_t$ ✓ $Y_t \rightarrow Z_t$ ✗	$X_{t-1}, X_{t-2} \rightarrow Y_t$ ✓
AME [23]	N/A	$W, Y \rightarrow Z$ ✓	$X \rightarrow Z$ ✓ $Y \rightarrow Z$ ✗	N/A
2D-ATT	$X_t, X_{t-2}, X_{t-5} \rightarrow Y_t$ ✓	$W_t \rightarrow Z_t$ ✓ $W_{t-1}, W_{t-2} \rightarrow Z_t$ ✗	$Y_{t-1} \rightarrow Z_t$ ✓	$X_{t-1}, X_{t-2} \rightarrow Y_t$ ✓

TABLE I

CAUSAL EFFECTS DISCOVERED BY ALL APPROACHES ON THE SYNTHETIC DATASETS

(✓denotes a true-positive causal effect, ✗denotes a false-positive causal effect)

Game	# of days	paid_install	ua_cost	DAU	DAP	FD	SPD	TPD	organic_install
Game 1	2309	5765(±4724)	24788(±16586)	1200881(±359630)	140506(±93478)	72	15	40	9774(±6834)
Game 2	1126	1591(±1597)	13966(±9522)	275224(±30665)	45676(±8087)	112	17	0	3525(±5073)
Game 3	1151	2733(±5805)	14210(±21416)	198001(±108431)	42668(±17478)	0	0	0	3694(±7338)
Game 4	1609	3099(±2908)	19233(±14317)	364281(±138480)	48863(±14952)	0	0	0	3423(±3628)
Game 5	896	12837(±35790)	45986(±46174)	770989(±619729)	167773(±86012)	485	7	46	38497(±118031)
Game 6	2394	7977(±7210)	28192(±23455)	773439(±399555)	79803(±58386)	71	17	0	11345(±7882)
Game 7	1071	1375(±2395)	5536(±9849)	81876(±59400)	9074(±3959)	19	1	0	3145(±6425)
Game 8	314	1042(±2426)	9568(±11220)	36502(±14102)	6255(±1146)	40	1	6	1922(±3143)
Game 9	216	2247(±2953)	18348(±8662)	185031(±26034)	19950(±5112)	0	0	0	6448(±7465)

TABLE II

STATISTICS OF THE REAL-WORLD DATASET ON THE IOS PLATFORM

$(x(\pm y)$ STANDS FOR MEAN AND STANDARD DEVIATION, FD: NUMBER OF FEATURING DAYS, SPD: NUMBER OF STORE PUBLISHING DAYS, TPD: NUMBER OF TEST PUBLISHING DAYS$)$

Game	# of days	paid_install	ua_cost	DAU	DAP	FD	SPD	TPD	organic_install
Game 1	2297	14296(±9612)	36909(±17780)	1815921(±694846)	106274(±73129)	129	20	189	27461(±18502)
Game 2	1222	4629(±3229)	20087(±11159)	309025(±71816)	34168(±6575)	74	16	328	5600(±9778)
Game 3	1150	4145(±12442)	10275(±21784)	175343(±139059)	24411(±13172)	0	0	0	5628(±12684)
Game 4	1609	7561(±6882)	19391(±12443)	479868(±212396)	35520(±13126)	0	0	0	8304(±11281)
Game 5	847	19396(±23519)	57114(±40212)	819134(±619381)	132279(±58104)	259	5	165	52774(±135335)
Game 6	2361	11042(±9173)	24657(±15805)	854516(±475386)	44260(±34182)	76	9	78	21330(±13911)
Game 7	1071	1549(±3188)	3717(±7738)	69737(±55846)	4727(±2368)	7	0	31	3271(±5841)
Game 8	315	1610(±2288)	9273(±10457)	53531(±33171)	5604(±1467)	26	1	55	4474(±11625)
Game 9	216	3295(±2470)	16531(±7918)	276994(±42663)	14527(±4089)	0	0	0	27473(±21941)

TABLE III

STATISTICS OF THE REAL-WORLD DATASET ON THE ANDROID PLATFORM

$(x(\pm y)$ STANDS FOR MEAN AND STANDARD DEVIATION, FD: NUMBER OF FEATURING DAYS, SPD: NUMBER OF STORE PUBLISHING DAYS, TPD: NUMBER OF TEST PUBLISHING DAYS$)$

W has a dominant causal effect on Y_t, with closer time steps having a higher importance. This is because in this dataset, we have $W_{t-2} \rightarrow Y_{t-1} \rightarrow Z_t$, which makes 2D-ATT steer most attention on W. On Synthetic 3 and 4, 2D-ATT accurately discovers the causal effects without any false claims. This proves that 2D-ATT is capable of identifying complex causal effects. Overall, 2D-ATT shows the most excellent competency in discovering intricate causal relationships with various levels of temporal delays from multivariate time series.

E. Causal Inference on Real-world Dataset

In Figure 5, we visualize the attention scores on the real-world mobile game dataset. For better visualization quality, we only plot a heatmap from the 30-day historical window. We must note that our attentional neural network predicts the organic installs with a high accuracy. The RMSE is 2,904, which does not account for 10% of the standard deviation. 2D-ATT identifies paid installs and DAU (daily average users) as the most important causal factors of organic installs, with closer statistics playing a more important role. The causal effect between DAU and organic installs are more or less expected. This is because DAU is a proper instantiation of game quality. Games with better quality have higher retention rates, and hence larger DAU. However, the discovery that paid installs are the most significant causal factor of organic installs is innovative. In the mobile game industry, paid users are not one of the widely recognized factors to improve organic installs. In reality, very few game managers recognize the importance of paid installs on bringing in more organic users. This finding suggests that the best way to boost organic installs is to buy more game users. With more paid installs and active users, the operation team can expect a rapid growth in the number of organic users. This demonstrates that the transformation from quantitative change (i.e., more paid users) to qualitative change (i.e., more organic users) also holds in the mobile game industry.

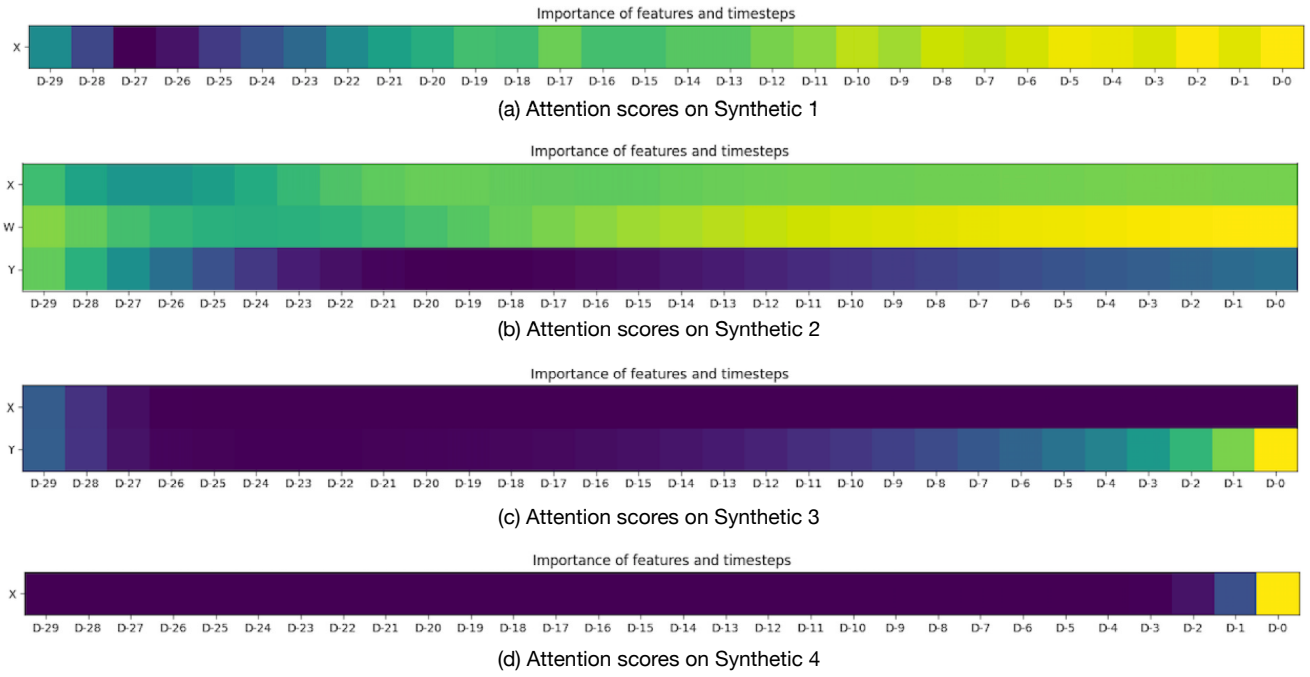

Fig. 4. Causal effects discovered by 2D-ATT on the synthetic datasets
(Yellow color denotes high importance, black color denotes low importance)

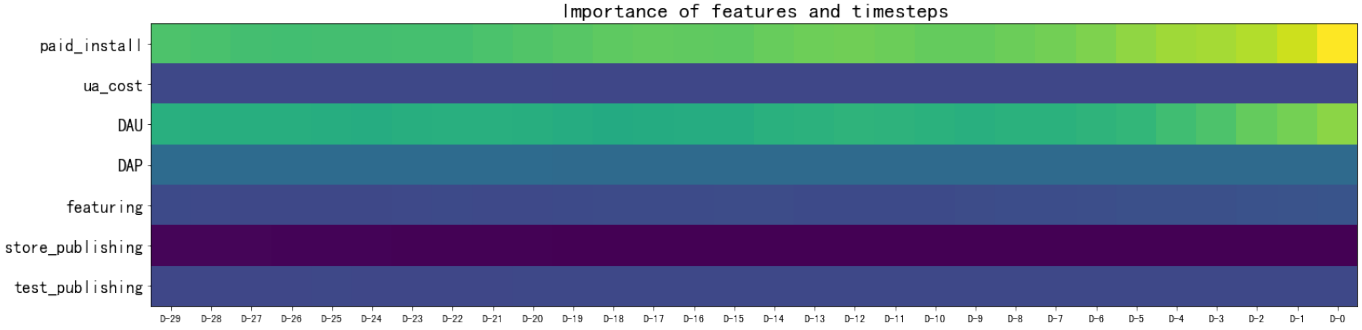

Fig. 5. Causal effects discovered by 2D-ATT on the real-world dataset
(Yellow color denotes high importance, black color denotes low importance)

V. Related Work

Statistical Inference. The classical definition of temporal causality, named *Granger causality*, was proposed five decades ago [11]. According to Granger causality, X_t is causing Y_t if we are better to predict Y_t using all available information than if the information apart from X_t had been used. Since then, it has has become a mainstream choice to determine whether and how two time series exert causal influences on each other. The estimation of linear Granger causality can be done by using vector autoregression model [10], generalized additive model [13], Fourier and wavelet transformation [8]. More complicated nonlinear Granger causality can be generalized from the linear ones, using methods from the theory of Reproducing Kernel Hilbert Spaces (RKHS) [17]. In particular, it embeds the data a Hilbert space, in which linear causality is searched. Various kernels have been adapted for nonlinear Granger causality, such as inhomogeneous polynomial kernel and Gaussian kernel. However, A major limitation of Granger causality is that it does not allow the existence of confounder. If this assumption is violated, the methods may either fail to detect any causal relationship, or draw false causal conclusions. It is very difficult to validate the assumption in real-world datasets.

Aside from Granger causality, Schreiber et al. [22] proposed transfer entropy based on information theory. Transfer entropy, $T_{X\to Y}$, measures the reduction in uncertainty of Y_t when X_t changes from unknown to known. The recent work in [30] defines *transfer causality* with $T_{X,Y} = T_{X\to Y} - T_{Y\to X}$. If $T_{X,Y} > 0$, it means that X is the cause of Y; otherwise, X is the consequence of Y. The inference of transfer causality does not consider any other variable. Therefore, in case of a confounder, it will mis-interpret correlation as causal relation.

Model Explanation. LIME [21] leads the researches on

explaining complex machine learning models. Given a complicated machine learning model or a deep neural network that is trained over largescale datasets, LIME explains the target model's rationale on any testing instance by training an explanation model to approximate the local decision boundary with a simple yet interpretable model, such as linear regression or logistic regression. From the explanation model, users can tell which features play an important role in the target model's decision, so as to infer the causal relationships. After that, a wide range of explanation models have been proposed, e.g., DeepLIFT [25], [26], Shapley [28] and SHAP [15]. The fidelity of the explanation results are unstable, since approximation error is inevitable on some test cases, due to the limited representation power of the simple explanation model.

Attention Neural Networks. Attention mechanisms are the prevailing method to fortify deep neural networks with interpretability when modeling complex sequences. It is embedded into a network, and allows the model to search for input parts that are relevant to make the prediction. The mechanism learns a dynamic weighted average of hidden states from different time steps in a long sequence, which serves as a long-term context for downstream discriminative components. A large weight indicates that the associated input fragment is of high importance with regard to the model's prediction result, and thus allows interpreting the decision. Various attention mechanisms have been proposed for tasks such as neural machine translation [2], [16], and image captioning [31]. However, most attentional neural networks fall short of the aforementioned interpretability for multi-variate time series due to their opaque hidden states. Specifically, the input data is firstly processed by a recurrent neural network (RNN), which blindly blend the information of all variables into the hidden states used for prediction. It is intractable to distinguish the contribution of individual variables into the prediction through the sequence of hidden states [32]. The work that is most relevant to ours is [12], in which one attention layer focuses on multiple steps in each individual variable, and the other one steers attention across the synopsis from all variables. However, we cannot directly apply [12] because we cannot compare the relative importance for inputs at different time steps and different variables.

VI. Conclusion

In this paper, we present a novel attention mechanism to discover causal relationships with temporal lags from multivariate time series data. Experiment results on both synthetic and real-world datasets demonstrate the effectiveness on identifying complex causal effects.

In the future, we plan to extend the work by designing an automated algorithm to intelligently allocate UA budget based on the discovered causal effects. Besides, we also plan to design a user attribution model that evaluates the efficacy of each advertising channel in providing valuable new users.

References

[1] John Aldrich et al. Correlations genuine and spurious in pearson and yule. *Statistical science*, 10(4):364–376, 1995.

[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[3] Yoshua Bengio. *Learning deep architectures for AI*. Now Publishers Inc, 2009.

[4] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[5] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[6] Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. Gated feedback recurrent neural networks. In *International conference on machine learning*, pages 2067–2075, 2015.

[7] Xuan-Hong Dang, Syed Yousaf Shah, and Petros Zerfos. seq2graph: Discovering dynamic dependencies from multivariate time series with multi-level attention. *arXiv preprint arXiv:1812.04448*, 2018.

[8] Mukeshwar Dhamala, Govindan Rangarajan, and Mingzhou Ding. Estimating granger causality from fourier and wavelet transforms of time series data. *Physical review letters*, 100(1):018701, 2008.

[9] Sizhen Du, Guojie Song, Lei Han, and Haikun Hong. Temporal causal inference with time lag. *Neural Computation*, 30(1):271–291, 2018.

[10] Jean-Pierre Florens and Michel Mouchart. A note on noncausality. *Econometrica: Journal of the Econometric Society*, pages 583–591, 1982.

[11] Clive WJ Granger. Investigating causal relations by econometric models and cross-spectral methods. *Econometrica: journal of the Econometric Society*, pages 424–438, 1969.

[12] Tian Guo, Tao Lin, and Nino Antulov-Fantulin. Exploring interpretable lstm neural networks over multi-variable data. *arXiv preprint arXiv:1905.12034*, 2019.

[13] Trevor J Hastie and Robert J Tibshirani. *Generalized additive models*, volume 43. CRC press, 1990.

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

[15] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Advances in neural information processing systems*, pages 4765–4774, 2017.

[16] Minh-Thang Luong, Hieu Pham, and Christopher D Manning. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*, 2015.

[17] Daniele Marinazzo, Mario Pellicoro, and Sebastiano Stramaglia. Kernel-granger causality and the analysis of dynamical networks. *Physical review E*, 77(5):056215, 2008.

[18] Leland Gerson Neuberg. Causality: models, reasoning, and inference, by judea pearl, cambridge university press, 2000. *Econometric Theory*, 19(4):675, 2003.

[19] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In *International conference on machine learning*, pages 1310–1318, 2013.

[20] Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. Causal inference on time series using restricted structural equation models. In *Advances in Neural Information Processing Systems*, pages 154–162, 2013.

[21] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016.

[22] Thomas Schreiber. Measuring information transfer. *Physical review letters*, 85(2):461, 2000.

[23] Patrick Schwab, Djordje Miladinovic, and Walter Karlen. Granger-causal attentive mixtures of experts: Learning important features with neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 4846–4853, 2019.

[24] Attend Show and Kelvin Xu. Tell: Neural image caption generation with visual attention. *Kelvin Xu et. al.. arXiv Pre-Print*, 23, 2015.

[25] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. *arXiv preprint arXiv:1704.02685*, 2017.

[26] Avanti Shrikumar, Peyton Greenside, Anna Shcherbina, and Anshul Kundaje. Not just a black box: Learning important features through propagating activation differences. *arXiv preprint arXiv:1605.01713*, 2016.

[27] Hava T Siegelmann and Eduardo D Sontag. On the computational power of neural nets. *Journal of computer and system sciences*, 50(1):132–150, 1995.

[28] Erik Štrumbelj and Igor Kononenko. Explaining prediction models and individual predictions with feature contributions. *Knowledge and information systems*, 41(3):647–665, 2014.

[29] Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, et al. End-to-end memory networks. In *Advances in neural information processing systems*, pages 2440–2448, 2015.

[30] Haoyan Xu, Yida Huang, Ziheng Duan, Jie Feng, and Pengyu Song. Multivariate time series forecasting based on causal inference with transfer entropy and graph neural network. *arXiv preprint arXiv:2005.01185*, 2020.

[31] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057, 2015.

[32] Liheng Zhang, Charu Aggarwal, and Guo-Jun Qi. Stock price prediction via discovering multi-frequency trading patterns. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 2141–2149, 2017.

[33] Wentao Zhu, Cuiling Lan, Junliang Xing, Wenjun Zeng, Yanghao Li, Li Shen, and Xiaohui Xie. Co-occurrence feature learning for skeleton based action recognition using regularized deep lstm networks. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.