

# Knowledge-Driven Stock Trend Prediction and Explanation via Temporal Convolutional Network

Shumin Deng  
Zhejiang University, AZFT Joint Lab  
for Knowledge Engine  
China  
231sm@zju.edu.cn

Jiaoyan Chen  
University of Oxford  
United Kingdom  
jiaoyan.chen@cs.ox.ac.uk

Ningyu Zhang  
Alibaba Group  
China  
ningyu.zny@alibaba-inc.com

Jeff Z. Pan  
University of Aberdeen  
United Kingdom  
jeff.z.pan@abdn.ac.uk

Wen Zhang  
Zhejiang University, AZFT Joint Lab  
for Knowledge Engine  
China  
wenzhang2015@zju.edu.cn

Huajun Chen  
Zhejiang University, AZFT Joint Lab  
for Knowledge Engine  
China  
huajunsir@zju.edu.cn

## ABSTRACT

Deep neural networks have achieved promising results in stock trend prediction. However, most of these models have two common drawbacks, including (i) current methods are not sensitive enough to abrupt changes of stock trend, and (ii) forecasting results are not interpretable for humans. To address these two problems, we propose a novel Knowledge-Driven Temporal Convolutional Network (KDTCN) for stock trend prediction and explanation. Firstly, we extract structured events from financial news, and utilize external knowledge from knowledge graph to obtain event embeddings. Then, we combine event embeddings and price values together to forecast stock trend. We evaluate the prediction accuracy to show how knowledge-driven events work on abrupt changes. We also visualize the effect of events and linkage among events based on knowledge graph, to explain why knowledge-driven events are common sources of abrupt changes. Experiments demonstrate that KDTCN can (i) react to abrupt changes much faster and outperform state-of-the-art methods on stock datasets, as well as (ii) facilitate the explanation of prediction particularly with abrupt changes.

## CCS CONCEPTS

• Information systems → Data streams; Temporal data; • Mathematics of computing → Time series analysis; • Applied computing → Economics.

## KEYWORDS

Knowledge-driven, stock trend prediction, event extraction, structured, unstructured, predictive analytics, explanation

### ACM Reference Format:

Shumin Deng, Ningyu Zhang, Wen Zhang, Jiaoyan Chen, Jeff Z. Pan, and Huajun Chen. 2019. Knowledge-Driven Stock Trend Prediction and Explanation via Temporal Convolutional Network. In *Companion Proceedings of*

*the 2019 World Wide Web Conference (WWW '19 Companion)*, May 13–17, 2019, San Francisco, CA, USA. ACM, New York, NY, USA, 8 pages. <https://doi.org/10.1145/3308560.3317701>

## 1 INTRODUCTION

Stock trend prediction has been widely studied due to its scientific and economic merits, and recent efforts mostly focus on exploring potential of deep neural network models. Although such methods [22, 26, 44] perform well in many tasks, they show weakness in tackling the problem of capturing unexpected abrupt changes, and falling short of giving explanations for prediction results.

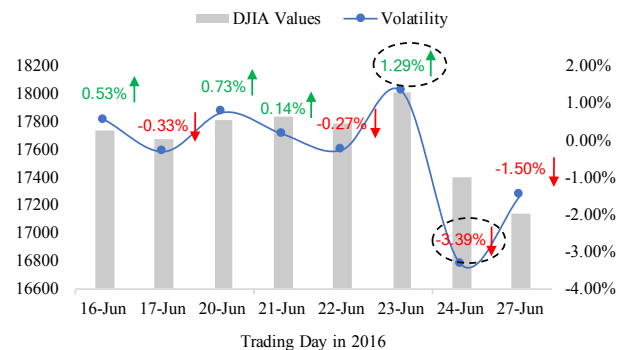


Figure 1: Example DJIA values with abrupt changes

*Stock trend prediction with abrupt changes.* In stock trend prediction, abrupt changes mean that stock prices fluctuate sharply in an extremely short time interval [7, 16, 19, 28, 43]. For example, as shown in Figure 1, the DJIA (Dow Jones Industrial Average) index increased by 1.29% on 23rd June, 2016, while fell by 3.39% sharply on the next day. For the sake of adapting to abrupt trend changes, [26] have proposed a TreNet to learn both local and global numerical features from stock prices. However, it can be insufficient to merely utilize price data. [17] has shown that stock prices can only reflect all known information, and price movements are in response to news or events. Such as in Figure 1, between 23rd June and 24th June, an unexpected event of *British referendum* happened. In order to encode events for stock trend prediction, [11] have demonstrated

This paper is published under the Creative Commons Attribution 4.0 International (CC-BY 4.0) license. Authors reserve their rights to disseminate the work on their personal and corporate Web sites with the appropriate attribution.

WWW '19 Companion, May 13–17, 2019, San Francisco, CA, USA

© 2019 IW3C2 (International World Wide Web Conference Committee), published under Creative Commons CC-BY 4.0 License.

ACM ISBN 978-1-4503-6675-5/19/05.

<https://doi.org/10.1145/3308560.3317701>

the validity of deep learning methods for event-driven stock market prediction, through event-embedding-based news representations. Despite news events help people capture abrupt changes of stock trend swiftly, they are often disordered and sparse. To address this problem, we import exogenous knowledge to represent events. Knowledge, coming from knowledge graphs (KGs), have two major advantages: (i) enriched semantic information in knowledge can help to establish association among discrete events, and (ii) knowledge in KG is structured and facile to be parameterized.

*Deep prediction models lack explanations.* Machine learning (ML) explanation, such as interpreting prediction models or justifying prediction results, can significantly increase decision makers' confidence on prediction and boost its application [4, 33]. Even if deep prediction models successfully detect abrupt changes in the stock market, it is hard for them to make people without ML expertise understand why these changes happen. For example, in event-driven stock trend prediction, people may be concerned with which events have greatly influenced stock fluctuation, and how these events take effects. To tackle this problem, we encode interpretable knowledge in deep prediction models, making prediction explainable.

Above all, in this paper, we propose a novel Knowledge-Driven Temporal Convolutional Network (KDTCN), incorporating background knowledge, news events and price data into deep prediction models, to tackle the problem of stock trend prediction and explanation with abrupt changes. We choose Temporal Convolution Network (TCN) [3] because it outperforms canonical RNNs such as LSTMs across a diverse range of tasks and datasets, while demonstrating longer effective memory. The intuition is that *many events are responsible for abrupt changes in the stock market, and the correlation analysis between events and changes offer explanations.*

Specifically, to address the problem of prediction with abrupt changes, we extract events from financial news and structurize them into event tuples, e.g., "*Britain exiting from EU*" is represented as (*Britain, exiting from, EU*). Then entities and relations in event tuples are linked to KGs, such as Freebase [5] and Wikidata [39]. Secondly, we vectorize structured knowledge, textual news, as well as price values respectively, and then concatenate them together. Finally we feed these embeddings into a TCN-based model. Experiments demonstrate that KDTCN can react to abrupt changes in the stock market more swiftly than state-of-the-art methods. Furthermore, based on prediction results with abrupt changes, we address the problem of making explanations. We visualize the effect of events, and also present the linkage among events with the use of KG. By doing so, we make explanations of (i) how knowledge-driven events influence the stock market fluctuation in different levels, and (ii) how knowledge helps to associate events with abrupt changes in stock trend prediction.

To the best of our knowledge, KDTCN is the first one to utilize Temporal Convolution Network to make stock trend prediction, integrating structured knowledge graph, textual news as well as time-series price values. Additionally, KDTCN is capable of explaining prediction results particularly with abrupt changes. In this paper, the next section reviews related work on stock price prediction and ML explanation. Section 3 presents the architecture of KDTCN and specifically introduces the model. Section 4 presents the experiments and evaluation. Section 5 makes a conclusion of the paper and discusses the future work.

## 2 RELATED WORK

### 2.1 Deep Models for Stock Prediction

Traditional models of stock prediction are mostly based on sequence modeling with sequence data input. Traditional Recurrent Neural Networks (RNNs) [14, 35, 42] are powerful in discovering the dependency of sequence data, however suffer from vanishing gradients and thus have difficulty in capturing long-term dependencies. Long Short-Term Memory (LSTM) [21] overcomes this limitation. There have already existed some RNN-based stock prediction. [26] have proposed TreNet, a novel end-to-end hybrid neural network, to learn local and global contextual features for predicting stock trend. A dual-stage attention based RNN (DA-RNN) [31], is able to capture long-term temporal dependencies appropriately with attention mechanism. These ML methods with time-series values input, are adapted to the relatively stable stock market, but they have difficulty in reacting to abrupt changes of the stock market swiftly.

Except for value-based models [13, 46], there are also some methods using texts. [22] have proposed hybrid attention networks to predict stock trend based on the sequence of recent news. [10, 11] have extracted events from news and demonstrated that deep learning is useful for event-driven stock movement prediction. Although these models convert unstructured text to structural events, they only utilize texts, making performance limited. [1, 44] use both numerical and textual data. [1] have modeled temporal effects of past events on opening prices with LSTM. [44] have presented a deep generative model to predict stock movement from tweets and historical stock prices. However, these two models are unable to represent chaotic social text effectively, as they only utilize bag-of-words or word embeddings without capturing structured relations.

### 2.2 Knowledge-driven Models and Explanation

Despite substantial efforts have been made for stock prediction, most of them only learn features from numerical and textual data, while ignore background knowledge. [9] have demonstrated that incorporating knowledge can help capture inconsistent evolution of stream data, thereby make more accurate prediction. [12] have proposed to incorporate KG into the learning process of event embeddings, which can encode valuable background knowledge. Besides, knowledge-driven ML models also show their strengths in other broad domains, such as recommender systems. [23] have proposed a novel knowledge enhanced sequential recommender, integrating RNN-based networks with Key-Value Memory Network (KV-MN). [41] have proposed a deep knowledge-aware network (DKN), incorporating KG to news recommendation.

In these knowledge-driven models, eXplainable AI (XAI) is very important. In ML literature, work on explanation often focuses on visualizations of prediction. Beyond that, research focuses on two broad approaches to explanation [4]. The first is prediction justification, where a (usually non-interpretable) model [8, 18, 38] and prediction are given, and a justification for the prediction must be produced. The second is interpretable models, aiming to devise models that are intrinsically interpretable and can be explained by reasoning. In this paper, we focus on the first one. There are many producing justifications, and they focus on interpreting predictions of specific complex models, often by proposing to isolate contributions of individual features. [33] have proposed to explain models

by presenting representative individual predictions and their explanations in a non-redundant way. They have explained predictions of each classifier by learning an interpretable model locally.

### 3 METHODOLOGY

In this section, we first present the overview of the proposed knowledge-driven temporal convolutional network (KDTCN) for stock trend forecasting and explanation. Then we introduce each model component in detail.

#### 3.1 Model Overview

The overview of KDTCN architecture is shown in Figure 2. Original model inputs are price values  $\mathcal{X}$ , news corpus  $\mathcal{N}$ , and knowledge graph  $\mathcal{G}$ . The price values are normalized and mapped into the price vector, denoted by

$$\mathcal{P} = \{p_0, p_1, \dots, p_{T-1}\}, \quad (1)$$

where each vector  $p_t$  represents a real-time price vector on a stock trading day  $t$ , and  $T$  is the time span.

In news corpus, pieces of news are represented as event sets  $\mathcal{E}$ , and are structured by open domain information extraction (Open IE [15]) through leveraging linguistic structure. Each event is structured into an event tuple  $e = (s, p, o)$ , where  $p$  is the action or predicate,  $s$  is the actor or subject and  $o$  is the object on which the action is performed. Then, each item in event tuples is linked to KG. Note that event items in this paper refer to the  $s$ ,  $p$  and  $o$  in the event tuple  $(s, p, o)$ , and they also correspond to entities and relations in KG. We obtain event embeddings  $\mathcal{V}$  by training both event tuples and KG triples. Finally, event embeddings, combined with price vectors are input into a TCN-based model [3] for stock trend prediction and explanation particularly with abrupt changes.

Our purpose is to forecast the movement of target stock index trend  $y$  with abrupt changes. We predict the binary movement, in which 1 denotes rise and 0 denotes drop, defined by

$$y = \begin{cases} 1 & \text{if } x_t > x_{t-1} \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

where  $x_t$  denotes the stock price value on the stock trading day  $t$ .

#### 3.2 Event Embedding

The goal of event embedding is to learn low-dimension dense vector representations for event tuples  $e = (s, p, o)$ . We first extract structured event tuples from financial news, and then link them to KG. An event tuple embedding is calculated by multi-channel concatenation of KG embeddings and word vectors for each item.

**3.2.1 Event Extraction and Structuralization.** We convert unstructured news texts into structured event tuples by Open IE [15], the goal of which is to read a sentence and extract tuples with a relation phrase and arguments that are related by that relation phrase. Originally, Open IE extracts binary tuples [29], i.e., two arguments connected by one relation phrase. E.g., “*Britain exiting from the EU*”, with the subject-predicate-object structure, is structured into an event tuple ( $s = \text{Britain}, p = \text{exiting from}, o = \text{EU}$ ).

After event extraction based on Open IE, there can be lots of redundancy in generated event tuples. Thus we have also removed

useless words (e.g., adjectives and adverbs) in fundamental sentence structures, to ensure that event tuples  $\mathcal{E}$  are concise enough.

**3.2.2 Entity Linking and Extension.** After getting concise event tuples, we construct a sub-graph from KG by utilizing the technique of entity linking [36], in order to disambiguate named entities in texts by associating them with predefined entities in KG. Note that the subject  $s$ , predicate  $p$  and object  $o$  in an event tuple may not always have linkage in KG. Besides, information in a single event tuple may be sparse and lack diversity. Thus, we enrich the sub-graph by importing immediate neighbors of linked entities within one hop in KG. To formulate these concepts, we propose  $\text{linking}(e)$  and  $\text{linking}(r)$  to define the entity and relation in an event tuple linked to KG, as well as  $\text{context}(e)$  to define the immediate neighbors of linked entities in KG, denoted by

$$\text{linking}(e) = \{e_i \mid e_i = s \vee o, (s, p, o) \in \mathcal{E}_t \wedge e_i \in \mathcal{G}\} \quad (3)$$

$$\text{linking}(r) = \{r_i \mid r_i = p \wedge (s, p, o) \in \mathcal{E}_t \wedge r_i \in \mathcal{G}\} \quad (4)$$

$$\text{context}(e) = \{e_i \mid (e_i, r, e) \in \mathcal{G} \vee (e, r, e_i) \in \mathcal{G}, e \in \text{linking}(e)\} \quad (5)$$

**3.2.3 Knowledge-driven Multi-channel Concatenation.** We choose TransE [6] as the KG embedding method in this paper, as it could generally preserve structural information in KG with great robustness. As mentioned before, not all entities and relations in event tuples can be linked to KGs. In these situations, zero padding [40] is chosen. If  $s, o \notin \text{linking}(e)$ ,  $\mathcal{V}_{e_i^s} = \vec{0}$  and  $\mathcal{V}_{e_i^o} = \vec{0}$ . If  $p \notin \text{linking}(r)$ ,  $\mathcal{V}_{r_i^p} = \vec{0}$ . Besides, the context of each linked entity tend to be more than one entity and relation, so the context embedding is calculated by averaging. We then parameterize event representations in different channels, denoted by  $\mathcal{V}_l$  in the channel of KG linking,  $\mathcal{V}_c$  in the channel of KG context, and  $\mathcal{V}_w$  in the channel of words.

$$\mathcal{V}_l = [\mathcal{V}_{e_i^s}, \mathcal{V}_{r_i^p}, \mathcal{V}_{e_i^o}] \quad (6)$$

$$\mathcal{V}_c = [\mathcal{V}_{e_c^s}, \mathcal{V}_{r_c^p}, \mathcal{V}_{e_c^o}] \quad (7)$$

$$\mathcal{V}_w = [\mathcal{V}_{e_w^s}, \mathcal{V}_{r_w^p}, \mathcal{V}_{e_w^o}] \quad (8)$$

where  $e_i^s, e_i^o \in \text{linking}(e)$ , and  $r_i^p \in \text{linking}(r)$ ;  $e_c^s, e_c^o \in \text{context}(e)$ , and  $r_c^p \in \mathcal{G}$ ;  $\mathcal{V}_{e_w^s}, \mathcal{V}_{r_w^p}$ , and  $\mathcal{V}_{e_w^o}$  are the word vectors of  $s, p$ , and  $o$  respectively;  $\mathcal{V}_*$  represents the embedding of  $*$ .

Then we concatenate  $\mathcal{V}_l, \mathcal{V}_c$  and  $\mathcal{V}_w$  in multiple channels to get the final event embedding, denoted by

$$\mathcal{V} = F_e(\mathcal{E}, \mathcal{G}) = [\mathcal{V}_l \mathcal{V}_c \mathcal{V}_w] \quad (9)$$

#### 3.3 Temporal Convolutional Network

We refer to the presented TCN architecture proposed by [25, 32], and note that the basic TCN model we adopt in this paper is based on a generic architecture described by [3].

TCN [3, 25, 32] uses a 1-D fully-convolutional network (FCN) architecture [27], where each hidden layer has the same length as the input layer, and zero padding [40] is added to keep subsequent layers the same length as previous ones. In this way, the network can produce an output of the same length as the input. Besides, TCN

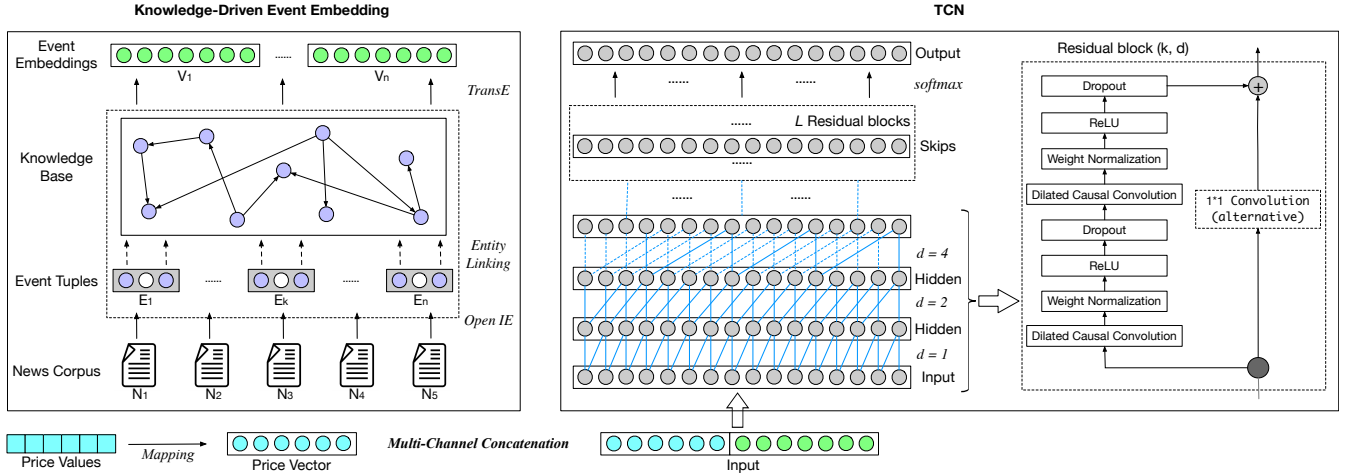


Figure 2: Illustration of the KDTCN framework. (Best viewed in color.)

uses causal convolutions, where an output at time  $t$  is convolved only with elements from time  $t$  and earlier in the previous layer, so that there can be no leakage from the future to the past.

Simply put, TCN = 1-D FCN + causal convolutions.

Then, in the following, we describe how techniques from present convolutional architectures are integrated into a TCN, considering both deep networks and long-range dependence.

**3.3.1 Dilated Convolutions.** Formally, for a 1-D sequence input  $X \in \mathbb{R}^n$  and a filter  $\mathcal{F}$ , the dilated convolution operation on  $j_{th}$  element in the sequence  $X$  is defined as

$$O(X_j) = (X * \mathcal{F}) = \sum_{i=0}^{k-1} \mathcal{F}(i) * X_{j-i \cdot d}, \quad (10)$$

where  $d$  is the dilation factor,  $k$  is the filter size, and the subscript  $X_{j-i \cdot d}$  denotes the direction of the past.

In fact, dilation can be regarded as importing a fixed step between every two adjacent filter taps. Each layer consists of a set of dilated convolutions with rate parameter  $d$ , a non-linear activation  $f(\cdot)$ , and a residual connection that combines the layer's input and the convolution signal.  $d$  increases consecutive layers within a block, calculated by  $d_l = 2^l$ . Convolutions are only applied over two timestamps,  $t$  and  $t-d$ . Specifically, filters can be parameterized by weight matrices  $W = [W_0, W_1]$  and the bias vector  $b$ , where  $W_i \in \mathbb{R}^{F_w \times F_w}$ ,  $b \in \mathbb{R}^{F_w}$ , and  $F_w$  denotes the number of filters.

$\tilde{Z}_t^{(j,l)}$  and  $Z_t^{(j,l)}$  are results after dilated convolution and adding the residual connection at timestamp  $t$  respectively, denoted by

$$\tilde{Z}_t^{(j,l)} = f(W_0 \tilde{Z}_{t-d}^{(j,l-1)} + W_1 \tilde{Z}_t^{(j,l-1)}), \quad (11)$$

and

$$Z_t^{(j,l)} = \tilde{Z}_t^{(j,l-1)} + V \tilde{Z}_t^{(j,l)} + e, \quad (12)$$

where  $V \in \mathbb{R}^{F_w \times F_w}$  denotes the weight matrix and  $e \in \mathbb{R}^{F_w}$  denotes the bias vector for the residual block.

**3.3.2 Residual Connections.** [20] has presented that a residual learning framework can ease network training, indicating residual blocks benefit very deep networks. Referring to [3, 32], we define

a series of residual blocks, each of which contains a series of  $L$  convolutional layers. Activations in the  $l_{th}$  layer and  $j_{th}$  block are  $Z^{(l,j)} \in \mathbb{R}^{F_w \times T}$ , where  $T$  is the time span, and  $F_w$  is same at each layer. The calculation of  $Z^{(l,j)}$  is defined in Equation (12).

Within a residual block, the TCN has two layers of dilated causal convolution and non-linearity, and we use ReLU [30]. For normalization, we apply weight normalization to convolutional filters. In addition, a spatial dropout [37] was added after each dilated convolution for regularization: at each training step, a whole channel is zeroed out.

However, whereas in standard ResNet [20] the input is added directly to the output of the residual function, in TCN (and ConvNets in general) the input and output could have different widths. To account for discrepant input-output widths, we use an additional  $1 \times 1$  convolution to ensure element-wise addition  $\oplus$  receiving tensors of the same shape.

### 3.4 Interpretation of Event Effects

We adopt a well-explored concept, *effect*, in the prediction, meaning its contribution towards or against the predicted class [34, 45]. In this paper, stock trend prediction can be regarded as the problem of binary classification. To make it simple, the discriminant function for data instance (event tuple  $e$ ) in the binary-class classifier is

$$f(y, e) = \sum_i \theta_i e_i \quad (13)$$

where  $e_i$  denotes the instance value of events, and the weight coefficients  $\theta_i$  is learned from training data for each class  $y$ .

Then the classifier predicts the class of the event instance as the one that maximizes the predictor function, through a monotonic non-linear distortion function  $\varphi$ :

$$\hat{y} = \arg \max_y \varphi(f(y, e)) \quad (14)$$

So the effect of the  $i_{th}$  event towards or against predicting class  $y$  for a data instance can be denoted as

$$F_{ef_i} = \theta_i e_i \quad (15)$$

## 4 EXPERIMENTS

The experiments mainly consist of two parts: (i) prediction evaluation and (ii) case-based explanation for prediction.

### 4.1 Datasets & Baselines & Settings

*Datasets.* Datasets in this paper are listed below.

- **Time-series Price Data  $\mathcal{X}$ :** The price dataset contains daily value records of DJIA index, and also has timespan from 08/08/2008 to 01/01/2016. Stock price data are downloaded directly from Yahoo Finance<sup>1</sup>. Note that they are cleaned for bank holidays, and aligned with financial news by time.
- **Textual News Data  $\mathcal{N}$ :** The news dataset is composed of historical news headlines from Reddit WorldNews Channel<sup>2</sup>. For each stock trading day, the top 25 news headlines are selected based on Reddit users' votes. The timespan is from 08/08/2008 to 01/01/2016.
- **Structured Knowledge Data  $\mathcal{G}$ :** The structured data come from two commonly used open knowledge graphs for research, including the Freebase [5] and the Wikidata [39]. We construct a sub-graph based on them, which contains 64958 entities and 716 relations totally.

*Baselines.* We consider commonly-used baseline model variations as shown in Table 1.

| Model     | Input                                     |   |
|-----------|---|---|
|           | Raw Data                                  | Processed Training Data                             |
| ARIMA [2] | $\mathcal{X}$                             | $\mathcal{X}$                                       |
| LSTM      | $\mathcal{X}$                             | $\mathcal{X}$                                       |
| CNN       | $\mathcal{X}$                             | $\mathcal{X}$                                       |
| TCN       | $\mathcal{X}$                             | $\mathcal{X}$                                       |
| WB-TCN    | $\mathcal{N}$                             | <i>word embedding</i>                               |
| EB-TCN    | $\mathcal{N}$                             | <i>event embedding<sup>(a)</sup></i>                |
| PVWB-TCN  | $\mathcal{X} + \mathcal{N}$               | <i>price vector + word embedding</i>                |
| PVEB-TCN  | $\mathcal{X} + \mathcal{N}$               | <i>price vector + event embedding<sup>(a)</sup></i> |
| KDEB-TCN  | $\mathcal{N} + \mathcal{G}$               | <i>event embedding<sup>(b)</sup></i>                |
| KDTCN     | $\mathcal{X} + \mathcal{N} + \mathcal{G}$ | <i>price vector + event embedding<sup>(b)</sup></i> |

**Table 1: Baseline models with different inputs. In the first column, prefix WB means word embeddings, EB means event embeddings, PV means the price vector, and KD means knowledge-driven. Note that *event embedding<sup>(a)</sup>* and *event embedding<sup>(b)</sup>* denote event embedding without and with KG respectively.**

*Settings.* With regard to settings of training process, stochastic gradient descent (SGD) [24] optimizer is used, with  $k$  kernels and  $L$  levels residual blocks. The best performance is gotten when  $k = 2$  and  $L = 10$ . The dimension of hidden units is 100, and the dimension

<sup>1</sup><https://finance.yahoo.com/quote/%5EDJI/history?p=%5EDJI>

<sup>2</sup><https://www.reddit.com/r/worldnews/?hl>

of word embedding, entity embedding as well as relation embedding are all set to 50. In KDTCN, a dropout rate of 0.5 is used to avoid over-fitting, and the learning rate is  $1 \times 10^{-5}$ . We split the raw value dataset into training set and testing set with ratio 0.8 and 0.2. We evaluate prediction performance with two different evaluation metrics, including (i) *Accuracy* and (ii) *F1 score*.

### 4.2 Prediction Evaluation

Performance of KDTCN is shown in three progressive aspects: (i) evaluation of basic TCN architecture, (ii) influence of different model inputs with TCN, and (iii) TCN-based model performance for abrupt changes.

**4.2.1 Basic Evaluation for TCN.** In order to demonstrate that generic TCN architecture can outperform some traditional prediction models, we make comparisons with them, shown in Table 2. Note that all experiments reported in this part are only input with price values.

| Model | Accuracy                             | F1 Score                              |
|-------|--------------------------------------|---------------------------------------|
| ARIMA | 52.40% $\pm$ 0.25%                   | 0.3844 $\pm$ 0.0052                   |
| LSTM  | 56.17% $\pm$ 0.26%                   | <b>0.5348 <math>\pm</math> 0.0039</b> |
| CNN   | 55.41% $\pm$ 0.25%                   | 0.5280 $\pm$ 0.0021                   |
| TCN   | <b>60.96% <math>\pm</math> 0.26%</b> | 0.5317 $\pm$ 0.0036                   |

**Table 2: Stock trend prediction results over the DJIA index dataset with different basic prediction models.**

In Table 2, we observe that TCN greatly outperforms baseline models on the stock trend prediction task. TCN achieves much better performance than either traditional ML models (such as ARIMA), or deep neural networks (such as LSTM and CNN), indicating that TCN has more obvious advantages in sequence modeling and classification problems. Therefore, we choose TCN as our basic prediction model in this paper.

**4.2.2 Different Model Inputs with TCN.** For the sake of validating effectiveness of integrating knowledge graph, financial news corpus, and price values in stock trend prediction, we compare prediction performance of models with different inputs, shown in Table 3.

| Model        | Accuracy                             | F1 Score                              |
|--------------|--------------------------------------|---------------------------------------|
| TCN          | 60.96% $\pm$ 0.26%                   | 0.5317 $\pm$ 0.0036                   |
| WB-TCN       | 62.96% $\pm$ 0.25%                   | 0.5934 $\pm$ 0.0038                   |
| EB-TCN       | 64.74% $\pm$ 0.26%                   | 0.6698 $\pm$ 0.0031                   |
| PVWB-TCN     | 66.49% $\pm$ 0.24%                   | 0.5845 $\pm$ 0.0043                   |
| PVEB-TCN     | 67.51% $\pm$ 0.25%                   | 0.6979 $\pm$ 0.0031                   |
| KDEB-TCN     | 68.51% $\pm$ 0.25%                   | 0.6959 $\pm$ 0.0032                   |
| <b>KDTCN</b> | <b>69.77% <math>\pm</math> 0.25%</b> | <b>0.7073 <math>\pm</math> 0.0030</b> |

**Table 3: Stock trend prediction results over the overall DJIA index dataset with different inputs on TCN-based models.**

As seen, WB-TCN and EB-TCN both get better performance than TCN, indicating textual information helps to improve forecasting. Analogously, compared PVWB-TCN and PVEB-TCN with WB-TCN



and EB-TCN, the comparison results reflect that price values are also useful in stock trend prediction. KDEB-TCN outperforms other baselines, which accounts for structured knowledge greatly working on stock trend prediction. Moreover, KDTCN gets both the highest *accuracy* and *F1 scores*, and such a result demonstrates the validity of model input integration with structured knowledge, financial news, and price values.

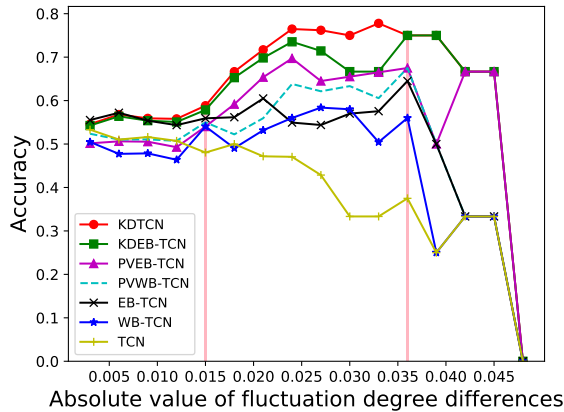
**4.2.3 Model Performance for Abrupt Changes.** We intend to verify if knowledge-driven models can effectively capture abrupt changes of stock trend prediction in this part. We first get time intervals of abrupt changes by figuring out the difference of stock fluctuation degree  $\mathcal{D}_{fluctuation}$  between two adjacent stock trading days [22], calculated by

$$\mathcal{D}_{fluctuation} = \frac{x_t - x_{t-1}}{x_{t-1}}, \quad (16)$$

where  $x_t$  and  $x_{t-1}$  denotes the stock price value on the stock trading day  $t$  and  $t-1$  respectively. Then the difference of fluctuation degree  $C$  is defined by:

$$C_i = \mathcal{D}_{fluctuation_i} - \mathcal{D}_{fluctuation_{i-1}}. \quad (17)$$

Intuitively, the larger  $|C_i|$  is, the more likely for the  $i_{th}$  day locates in the time interval of abrupt changes. If  $|C_i|$  exceeds a certain threshold, it can be considered that the stock price abruptly changes at the  $i_{th}$  day. In order to identify a proper range of thresholds, we show the performance based on data distribution of  $|C|$  in Figure 3.



**Figure 3: Performance on the data distribution of  $|C|$ .**

Seen from Figure 3, overall, higher accuracy is achieved when  $|C|$  are in the interval of 0.015 to 0.036, except for TCN marked in olive. TCN achieves the much worse performance, demonstrating that the numerical-data-based stock prediction model may perform poorly on prices with abrupt changes. Besides, KDTCN achieves more stable and better performance than other baselines, showing advantages of knowledge-driven models with integration of knowledge, texts and values. Then we calculate the average *Accuracy* and *F1 score* of KDTCN and baselines on datasets with abrupt changes, where  $|C|$  are in the interval of 0.015 to 0.036, presented in Table 4.

| Model        | Accuracy                             | F1 Score                              |
|--------------|--------------------------------------|---------------------------------------|
| ARIMA        | 30.18% $\pm$ 0.25%                   | $\infty$                              |
| LSTM         | 36.27% $\pm$ 0.25%                   | $\infty$                              |
| CNN          | 38.33% $\pm$ 0.25%                   | $\infty$                              |
| TCN          | 43.11% $\pm$ 0.25%                   | $\infty$                              |
| WB-TCN       | 54.16% $\pm$ 0.24%                   | 0.5078 $\pm$ 0.0038                   |
| EB-TCN       | 56.64% $\pm$ 0.26%                   | 0.5882 $\pm$ 0.0038                   |
| PVWB-TCN     | 59.56% $\pm$ 0.24%                   | 0.5556 $\pm$ 0.0031                   |
| PVEB-TCN     | 63.54% $\pm$ 0.25%                   | 0.6316 $\pm$ 0.0033                   |
| KDEB-TCN     | 67.32% $\pm$ 0.26%                   | 0.6667 $\pm$ 0.0032                   |
| <b>KDTCN</b> | <b>71.80% <math>\pm</math> 0.25%</b> | <b>0.7388 <math>\pm</math> 0.0030</b> |

**Table 4: Stock trend prediction results over the local DJIA index dataset of abrupt changes, with different model inputs.**

We observe that models with knowledge-driven event embedding input, such as KDEB-TCN and KDTCN, can greatly outperform numerical-data-based and textual-data-based models. These comparison results indicate that knowledge-driven events have significant impacts on the stock trend prediction with abrupt changes, and knowledge-driven models have advantages in reacting to abrupt changes in the stock market swiftly. Furthermore, KDTCN achieves better performance than KDEB-TCN, demonstrating that integrating price data to knowledge-driven models can also benefit stock trend prediction with abrupt changes.

### 4.3 Explanation for Prediction

Explanation in this paper belongs to human-centric justification of ML predictions [4]. We explain why knowledge-driven events are common sources of abrupt changes to human without ML expertise. The explanations are accomplished in two aspects: (i) visualizing effects of knowledge-driven events on prediction results with abrupt changes, and (ii) retrieving background facts of knowledge-driven events by linking the events to external KG.

**4.3.1 Effect Visualization of Events.** We calculate different event effects on stock trend prediction with Equation (14) and (15), then we visualize the results in Figure 4.

The prediction result in Figure 4 is that trend of DJIA index will drop. Note that the bars of the same colour have the same event effect, the height of bars reflects the degree of effects, and the event popularity<sup>3</sup> declines from left to right. Intuitively, events with higher popularity should have greater effects on stock trend prediction with abrupt changes, but not always. As seen, events of *Brexit* and *EU Referendum* both play an important role in forecasting downward trends. Nearly all other events with negative effect are related to these two events, e.g., (*British Pound, drops, nearly 5%*) and (*Northern Ireland, calls for poll on United Ireland*). Although there are also some events have positive effects of predicting stock trend to rise, and have high popularity, i.e., (*Rich, Getting, Richer*), the total effect is negative. Therefore, abrupt changes of the stock index fluctuation can be viewed as the combined result of effects and popularity of events.

<sup>3</sup>The event popularity is obtained through the ranking of top 25 news headlines which are selected based on Reddit users' votes.

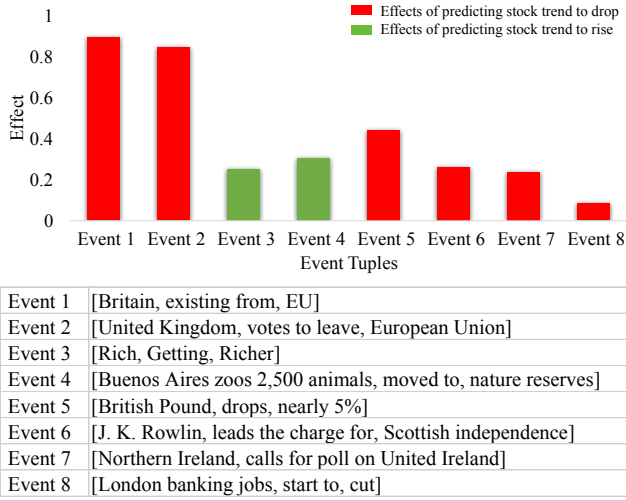


Figure 4: Examples of event effect on stock trend prediction.

4.3.2 *Visualization of Event Tuples Linked to KG.* We present the sample KG triples linked to event tuples in Figure 5.

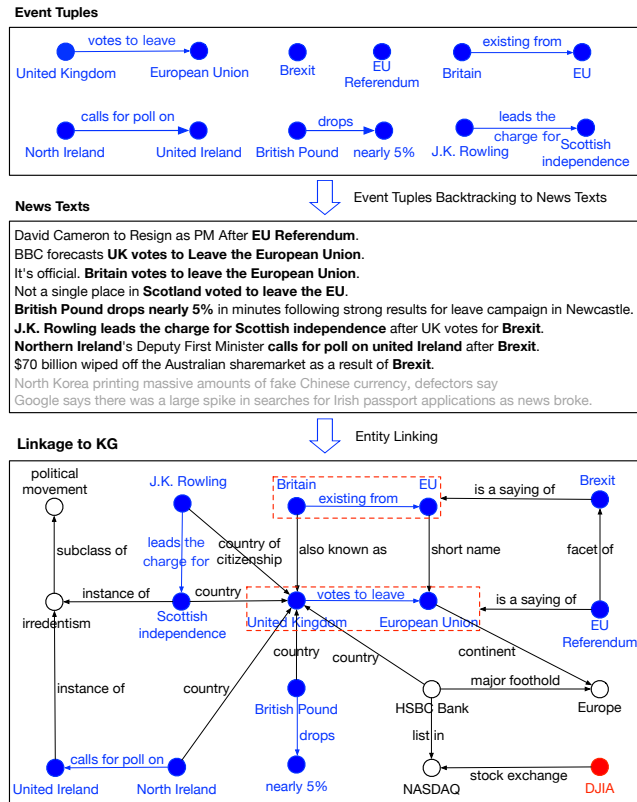


Figure 5: Illustration of triples in KG linked to events.

First, we search the event tuples with great effects or high popularity in stock trend movements. Then, we backtrack to the news texts containing these events. Finally, we retrieve associated KG triples linked to event tuples by entity linking. In Figure 5, each event tuple is marked in blue, and entities in it are linked to KG. We also mark *DJIA* in red. As seen, we illustrate that knowledge can help to associate events with abrupt changes in stock trend prediction together. These listed event tuples, such as (*Britain, existing from, EU*), (*United Kingdom, votes to leave, European Union*), (*British Pound, drops, nearly 5%*), (*J. K. Rowlin, leads the charge for, Scottish independence*), and (*Northern Ireland, calls for poll on United Ireland*), are not strongly relevant literally. However, with the linkage to KG, they can establish association with each other, and strongly related to events of *Brexit* and *EU Referendum*. Besides, these knowledge-enhanced events can also have connections with *DJIA*, thus originally sparse events can be closely linked and have combined effects on *DJIA* index. Thus, examples in Figure 5 explain how knowledge-driven events work on stock movements, and why knowledge-driven models are valid. Moreover, by incorporating explanations of event effects, we justify that knowledge-driven events are common sources of abrupt changes.

## 5 CONCLUSIONS AND FUTURE WORK

In this paper, we propose a novel knowledge-driven temporal convolutional network (KDTCN) to tackle the problem of stock trend prediction and explanation with abrupt changes. We extract structured event tuples from financial news, and utilize background knowledge from KG to associate discrete event tuples with each other. Through training both event tuples and KG triples, we get knowledge-driven event embeddings. Furthermore, we integrate price vectors and event embeddings as prediction model inputs by multi-channel concatenation. We utilize TCN to predict stock trend, and also explain prediction results based on knowledge. The experiments on stock datasets demonstrate that integrating structured knowledge to TCN can (i) greatly outperform present deep models when forecasting stock trend with abrupt changes, and (ii) make explanation on prediction results with abrupt changes. Through the event effect visualization and knowledge-enhanced event tuple visualization, we explain how knowledge influences greatly on stock trend with abrupt changes.

Based on research in this paper, we have identified several potential directions of this work, mainly includes a more general evaluation of different event effects on stock trend, a study on long-range dependency of events, and a more specific experiment on prediction explanations. At a stock trading day, there are various events influencing stock movements. We will figure out different effects of them and categorize these events based on effects, for example, which type of events affects stock trend to a great extent. Beside, the effect of an event may change as time goes by, thus to capture the dynamic effects of events is valuable. Furthermore, in this paper, we only give a case-based experiment on how knowledge help to establish association among sparse events. In the future, we will propose some quantitative indices to evaluate the effectiveness of knowledge, and give more specific explanations.

## 6 ACKNOWLEDGEMENTS

The authors would like to acknowledge that this work is funded by NSFC 61473260/91846204, national key research program YS2018YF-B140004 as well as Natural Science Foundation of Zhejiang Province of China (LQ19F030001), and supported by Alibaba-Zhejiang University Joint Institute of Frontier Technologies.

## REFERENCES

- [1] Ryo Akita, Akira Yoshihara, Takashi Matsubara, and Kuniaki Uehara. 2016. Deep learning for stock prediction using numerical and textual information. In *Computer and Information Science (ICIS)*, 2016. IEEE, 1–6.
- [2] Dimitrios Asteriou and Stephen G. Hall. 2011. *ARIMA Models and the Box-Jenkins Methodology*.
- [3] Shaojie Bai, J. Zico Kolter, and Vladlen Koltun. 2018. An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling. (2018).
- [4] Or Biran and Kathleen R McKeown. 2017. Human-Centric Justification of Machine Learning Predictions.. In *IJCAI*. 1461–1467.
- [5] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*. ACM, 1247–1250.
- [6] Antoine Bordes, Nicolas Usunier, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *International Conference on Neural Information Processing Systems*. 2787–2795.
- [7] Marco Antonio Leonel Caetano and Takashi Yoneyama. 2007. Characterizing abrupt changes in the stock prices using a wavelet decomposition method. *Physica A: Statistical Mechanics and its Applications* 383, 2 (2007), 519–526.
- [8] Emilio Carrizosa, Belen Martin, and Dolores Morales. 2006. *A column generation approach for support vector machines*. Technical Report. Technical report.
- [9] Freddy LeCue Chen, Jiaoyan and, Jeff Z. Pan, and Huajun Chen. 2017. Learning from Ontology Streams with Semantic Concept Drift. In *IJCAI*. 957–963.
- [10] Xiao Ding, Yue Zhang, Ting Liu, and Junwen Duan. 2014. Using structured events to predict stock price movement: An empirical investigation. In *Proceedings of EMNLP 2014*. 1415–1425.
- [11] Xiao Ding, Yue Zhang, Ting Liu, and Junwen Duan. 2015. Deep learning for event-driven stock prediction.. In *Ijcai*. 2327–2333.
- [12] Xiao Ding, Yue Zhang, Ting Liu, and Junwen Duan. 2016. Knowledge-driven event embedding for stock prediction. In *Proceedings of COLING 2016*. 2133–2142.
- [13] Aaron Elliot, Cheng Hua Hsu, and Jennifer Slodoba. 2017. Time Series Prediction: Predicting Stock Price. *arXiv preprint arXiv:1710.05751* (2017).
- [14] Jeffrey L. Elman. 1991. Distributed representations, simple recurrent networks, and grammatical structure. *Machine Learning* 7, 2-3 (1991), 195–225.
- [15] Oren Etzioni, Michael Cafarella, and Michele Banko. 2014. OPEN INFORMATION EXTRACTION.
- [16] Eugene F Fama. 1963. Mandelbrot and the stable Paretian hypothesis. *The journal of business* 36, 4 (1963), 420–429.
- [17] Eugene F Fama. 1965. The behavior of stock-market prices. *The journal of Business* 38, 1 (1965), 34–105.
- [18] Raphael Féraud and Fabrice Clérot. 2002. A methodology to explain neural network classification. *Neural Networks* 15, 2 (2002), 237–246.
- [19] James D Hamilton and Gang Lin. 1996. Stock market volatility and the business cycle. *Journal of Applied Econometrics* 11, 5 (1996), 573–593.
- [20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep Residual Learning for Image Recognition. (2015), 770–778.
- [21] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation* 9, 8 (1997), 1735–1780.
- [22] Ziniu Hu, Weiqing Liu, Jiang Bian, Xuanzhe Liu, and Tie-Yan Liu. 2018. Listening to chaotic whispers: A deep learning framework for news-oriented stock trend prediction. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*. ACM, 261–269.
- [23] Jin Huang, Wayne Xin Zhao, Hongjian Dou, Ji-Rong Wen, and Edward Y Chang. 2018. Improving Sequential Recommendation with Knowledge-Enhanced Memory Networks. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. ACM, 505–514.
- [24] Nikhil Ketkar. 2014. Stochastic Gradient Descent. *Optimization* (2014).
- [25] Colin Lea, Rene Vidal, Austin Reiter, and Gregory D Hager. 2016. Temporal convolutional networks: A unified approach to action segmentation. In *European Conference on Computer Vision*. Springer, 47–54.
- [26] Tao Lin, Tian Guo, and Karl Aberer. 2017. Hybrid Neural Networks for Learning the Trend in Time Series. In *Twenty-Sixth International Joint Conference on Artificial Intelligence*. 2273–2279.
- [27] Jonathan Long, Evan Shelhamer, and Trevor Darrell. 2015. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 3431–3440.
- [28] J. I. Maskawa. 2016. Collective Behavior of Market Participants during Abrupt Stock Price Changes. *Plos One* 11, 8 (2016), e0160152.
- [29] Mausam Mausam. 2016. Open information extraction systems and downstream applications. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*. AAAI Press, 4074–4077.
- [30] Vinod Nair and Geoffrey E. Hinton. 2010. Rectified linear units improve restricted boltzmann machines. In *International Conference on International Conference on Machine Learning*. 807–814.
- [31] Yao Qin, Dongjin Song, Haifeng Chen, Wei Cheng, Guofei Jiang, and Garrison Cottrell. 2017. A Dual-Stage Attention-Based Recurrent Neural Network for Time Series Prediction. (2017), 2627–2633.
- [32] Colin Lea Michael D Flynn René and Vidal Austin Reiter Gregory D Hager. 2017. Temporal convolutional networks for action segmentation and detection. In *IEEE International Conference on Computer Vision (ICCV)*.
- [33] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. 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*. ACM, 1135–1144.
- [34] Marko Robnik-Šikonja and Igor Kononenko. 2008. Explaining classifications for individual instances. *IEEE Transactions on Knowledge and Data Engineering* 20, 5 (2008), 589–600.
- [35] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. 1986. Learning representations by back-propagating errors. *Nature* 323, 6088 (1986), 533–536.
- [36] Avirup Sil and Alexander Yates. 2013. Re-ranking for joint named-entity recognition and linking. 18, 5 (2013), 2369–2374.
- [37] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research* 15, 1 (2014), 1929–1958.
- [38] Henri Jacques Suermondt. 1992. Explanation in Bayesian belief networks. (1992).
- [39] Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Commun. ACM* 57, 10 (2014), 78–85.
- [40] Alexander Waibel, Toshiyuki Hanazawa, Geoffrey Hinton, Kiyohiro Shikano, and Kevin J Lang. 1990. Phoneme recognition using time-delay neural networks. In *Readings in speech recognition*. Elsevier, 393–404.
- [41] Hongwei Wang, Fuzheng Zhang, Xing Xie, and Minyi Guo. 2018. DKN: Deep Knowledge-Aware Network for News Recommendation. *arXiv preprint arXiv:1801.08284* (2018).
- [42] P.J. Werbos. 1990. Backpropagation through time: what it does and how to do it. *Proc. IEEE* 78, 10 (1990), 1550–1560.
- [43] Eugene N White. 1990. The stock market boom and crash of 1929 revisited. *Journal of Economic perspectives* 4, 2 (1990), 67–83.
- [44] Yumo Xu and Shay B Cohen. 2018. Stock Movement Prediction from Tweets and Historical Prices. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Vol. 1. 1970–1979.
- [45] Ghim-Eng Yap, Ah-Hwee Tan, and Hwee-Hwa Pang. 2008. Explaining inferences in Bayesian networks. *Applied Intelligence* 29, 3 (2008), 263–278.
- [46] Liheng Zhang, Charu Aggarwal, and Guo Jun Qi. 2017. Stock Price Prediction via Discovering Multi-Frequency Trading Patterns. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2141–2149.