TADA: Trend Alignment with Dual-Attention Multi-Task Recurrent Neural Networks for Sales Prediction

Tong Chen[†], Hongzhi Yin[†*], Hongxu Chen[†], Lin Wu[‡†], Hao Wang[§], Xiaofang Zhou[†], Xue Li[†⊥]
[†]*School of Information Technology and Electrical Engineering, The University of Queensland, Brisbane, Australia*
[‡]*School of Computer and Information, Hefei University of Technology, Hefei, China*
[§]*Alibaba AI Labs, China*
[⊥]*Nanjing University of Aeronautics and Astronautics, Nanjing, China*
{tong.chen, h.yin1, hongxu.chen, lin.wu}@uq.edu.au, cashenry@126.com, {zxf, xueli}@itee.uq.edu.au

***Abstract*—As a common strategy in sales-supply chains, the prediction of sales volume offers precious information for companies to achieve a healthy balance between supply and demand. In practice, the sales prediction task is formulated as a time series prediction problem which aims to predict the future sales volume for different products with the observation of various *influential factors* (e.g., brand, season, discount, etc.) and corresponding historical sales records. However, with the development of contemporary commercial markets, the dynamic interaction between influential factors with different semantic meanings becomes more subtle, causing challenges in fully capturing dependencies among these variables. Besides, though seeking similar trends from the history benefits the accuracy for the prediction of upcoming sales, existing methods hardly suit sales prediction tasks because the trends in sales time series are more irregular and complex. Hence, we gain insights from the encoder-decoder recurrent neural network (RNN) structure, and propose a novel framework named TADA to carry out trend alignment with dual-attention, multi-task RNNs for sales prediction. In TADA, we innovatively divide the influential factors into *internal feature* and *external feature*, which are jointly modelled by a multi-task RNN encoder. In the decoding stage, TADA utilizes two attention mechanisms to compensate for the unknown states of influential factors in the future and adaptively align the upcoming trend with relevant historical trends to ensure precise sales prediction. Experimental results on two real-world datasets comprehensively show the superiority of TADA in sales prediction tasks against other state-of-the-art competitors.**

***Index Terms*—Sales Prediction, Time Series Data, Deep Neural Network**

I. INTRODUCTION

Keeping a balance between supply and demand is crucial to retailers, and accurate prediction of sales volume is becoming indispensable for commercial success [1]. Overestimated sales can result in excessive inventory, unhealthy cash flow and even bankruptcy, while the underestimated sales may lead to unfulfilled orders, decreased business reputation and profit [2]. In practice, sales prediction is formulated as a time series forecasting problem, which aims to predict future sales volume based on the observed multivariate time series data which consists of historical sales volume and ***influential factors*** (e.g., brand, season, discount, etc.). Thus, a reasonable modelling of the influential factors and historical sales information should be performed to successfully predict sales volume.

In recent years, time series prediction algorithms are widely adopted in many areas such as financial market prediction [3], [4], recommender systems [5], [6] and medical data processing [7], [8]. Among these techniques, the discovery of trending events or repeating patterns based on the clues from historical observations has inspired some interesting applications like traffic modelling [9], solar intensity prediction [10] and argument discovery [11]. Undoubtedly, the discovery of recurring trends will greatly benefit the forecast of sales by aligning relative contextual information learned from the influential factors, and this insight is referred to as ***trend alignment*** in this paper. However, both traditional autoregressive based methods [12]–[14] and recent trend mining models [9], [15] are ineffective for the trend alignment in sales prediction. This is because these methods assume the trend in time series data recurs periodically (i.e., distributes with a fixed time period), thus requiring domain knowledge for every application area and carefully chosen parameters based on the data. Hence, existing techniques are unable to align similar trends in sales time series where the sales patterns are much more subtle and irregular due to the effect from complicated real-world situations, and the difficulty increases when there are a large number of different products.

The formation of a trend in sales time series has specific contexts which can be modelled from the interaction among various influential factors. In regards to contextual information learning from raw time series, recurrent neural network (RNN) models have been intensively studied and applied to learn vector representations from sequential inputs [9], [16]. Compared with previous efforts on time series prediction like kernel methods and Gaussian process [17], [18] which are limited by their predefined non-linear form, RNNs show their advantages in flexible yet discriminative non-linear relationship modelling. Moreover, two variants of RNN, namely long short-term memory (LSTM) [19] and gated recurrent unit (GRU) [20] further advance the performance in

*Corresponding author.

tasks related to neural machine translation [21] and image captioning [22]. Among these applications, the encoder-decoder RNN architecture leverages two independent RNNs to encode sequential inputs into latent ***contextual vectors*** and decode these contexts into desired interpretations [21]–[23]. After showing its superiority in recent time series modelling tasks [4], [8], it is natural to consider encoder-decoder based RNN for sales prediction by leveraging its capability to fully capture the non-linear relationship between the influential factors and the sales volume.

However, even with the state-of-the-art encoder-decoder based RNN models, sales prediction is still a challenging research problem because when multiple influential factors interact with each other, they have different influences on different products. For instance, temperature has more impact on the sales of down jackets than shirts because shirts are intrinsically cheaper and can be worn all year round. Furthermore, the influential factors are dynamic and unpredictable in many cases, so it is impractical to assume their future availability. For example, though environmental policy significantly affects electrical car sales, and fashion trend dominates the clothing industry, we have very limited prevision on these influential factors. To make things worse, when performing trend alignment using contexts learned from the past, the decoder cannot generate rich contexts with the unknown states of influential factors. Though contextual features can be learned via multimodal modelling [24]–[27], these approaches are inapplicable as they need to be retrained once the sales information is updated. Hence, the main challenges in sales prediction are summarized as follows. The first is how to ***fully capture the dynamic dependencies among multiple influential factors***. Secondly, without any prior knowledge of mutative variables in the future, how can we possibly ***glean wisdoms from the past to compensate for the unpredictability of influential factors***. Third, as different sales trends recur irregularly due to complex real-world situations, it is necessary to ***align the upcoming trend with historical sales trends***, thus selectively gather relative contextual information for accurate prediction of sales volume.

In light of these challenges, we propose a novel sales prediction framework, namely Trend Alignment with Dual-Attention Multi-Task Recurrent Neural Network for Sales Prediction (TADA). TADA consists of two major components: the multi-task based LSTM encoder and the dual-attention based LSTM decoder, which are illustrated in Figure 1.

In order to solve the first challenge, we make our own observation on the characteristics of sales time series based on previous discussions. The semantics of influential factors in sales prediction are diverse, which however has been ignored by the conventional time series prediction methods. Specifically, for each product, its influential factors come with its intrinsic properties which are directly related to customers' subjective preference, e.g., brand, category, price, etc. Meanwhile, there are also many factors that objectively affects the sales, e.g., weather, holiday, promotion, etc. In this paper, we categorize the intrinsic properties of a product as its ***internal feature*** and the other influential factors as the ***external feature***. While internal features and external features express different semantic meanings, they both contribute to the fluctuations of the product sales volume at the same time. Hence, compared with predictive models that treats all kinds of features in a unified way [4], [9], [28], we propose a multi-task based LSTM encoder to learn contextual vector representations of historical sales time series. As shown in Figure 1, to solve the first challenge, we novelly model the internal feature and external feature in parallel via two individual LSTM layers. Then, we use a synergic LSTM layer to simultaneously join these two learned latent representations at each time step. The insight of a multi-task based encoder structure is to comprehensively leverage all available resources by modelling internal and external features separately first, and then pose a dynamic interaction between different features to generate contextual representations of historical sales time series.

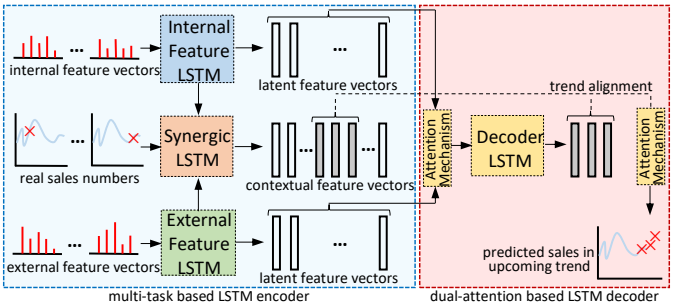

Fig. 1. Illustration of the proposed framework TADA.

To address the challenges of trend alignment and unknown influential factors, we propose an innovative dual-attention based LSTM decoder to tackle the difficulties. Grasping intuitions from existing attention mechanisms [21], [29] which aim to select relevant parts of hidden states learned by the encoder to attend, we develop our simple yet effective attention mechanisms which perfectly blend into the neural network for accurate sales prediction. As illustrated in Figure 1, in the decoding stage, the first attention models the effect of unknown influential factors using relevant contextual vectors from the encoder. After new sales contexts are generated within the look-ahead time interval, the second attention gathers contextual information of this upcoming trend, and then actively aligns the new trend with historical ones. Eventually, we combine the representation from the aligned trends to produce a sequence of estimated sales volume in the future.

We summarize the primary contributions of our research as follows:

- We are the first to categorize the influential factors in sales time series into internal features and external features, and innovatively model these two aspects with multi-task based LSTM encoder. We also adopt a synergic LSTM layer to model the dynamic interaction between different types of influential factors.
- We propose TADA, a dual-attention multi-task recurrent neural network to tackle the aforementioned challenges in sales prediction. The novel approach allows the encoder-decoder structure to comprehensively model variables

with different semantic meanings. Also, the embedded dual-attention increases both the interpretability and accuracy of the model by simulating unknown states of future contexts and aligning the upcoming sales trend with the most relevant one from the past.

- We conduct extensive experiments on two real-life commercial datasets. The results showcase the superiority of our approach in sales prediction by outperforming a group of state-of-the-art predictive models. We validate the vigorous contribution of each component in TADA via ablation tests and visualizations. Additional experiments on training efficiency further show promising scalability of TADA.

The rest of this paper is organized as follows. Section II formulates the sales prediction task and explains our proposed TADA in detail. Section III verifies the asymptotic time complexity of TADA is linearly associated with the scale of the data. In Section IV, we report experimental results of our model in comparison with state-of-the-art baselines. After outlining related research backgrounds in Section V, we conclude our findings with Section VI.

II. TADA: THE MODEL

In this section, we first mathematically formulate the definition of sales prediction and then we present the technical details of our proposed model TADA. Finally, we introduce the loss function and optimization strategy.

A. Problem Formulation

The objective of sales prediction is to predict future sales volume according to multivariate observations (e.g., previous sales, weather, price, promotion, etc.) from the past. The formulation of sales prediction is similar to, but different from multivariate time series forecasting and autoregressive models (AR). Formally, for an arbitrary product, the input is defined as its fully observed feature vector set $\{\mathbf{x}_t\}_{t=1}^{T} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_T\}$ and the corresponding sales volume $\{y_t\}_{t=1}^{T} = \{y_1, y_2, ..., y_T\}$ at time step t. Here, $\mathbf{x}_t \in \mathbb{R}^n$, $y_t \in \mathbb{R}$ and n is variable according to the feature dimension, while T is the amount of total time steps. The output of sales prediction is the estimated sales volume of following Δ time steps after T, denoted as $\{\hat{y}_t\}_{t=T+1}^{T+\Delta} = \{\hat{y}_{T+1}, \hat{y}_{T+2}, ..., \hat{y}_{T+\Delta}\}$, where Δ is adjustable according to the business goal. In this paper, we assume $\Delta \ll T$ to ensure the prediction accuracy because $\{\mathbf{x}_t\}_{t=T+1}^{T+\Delta}$ is non-available in the prediction stage.

Importantly, compared with multivariate time series forecasting and AR, sales prediction models behave differently. This is because our target is to acquire the one-dimensional scalar representing the sales volume without prior knowledge of the features in the future. Meanwhile, in multivariate time series forecasting, the output is specifically $\{\mathbf{x}_t\}_{t=T+1}^{T+\Delta}$, which has the same form and contextual meaning of its input [9]. Also, the AR assumes $\{\mathbf{x}_t\}_{t=T+1}^{T+\Delta}$ is available when predicting $\{\hat{y}_t\}_{t=T+1}^{T+\Delta}$ [4] because it is designed to model a mapping function between conditions and consequences.

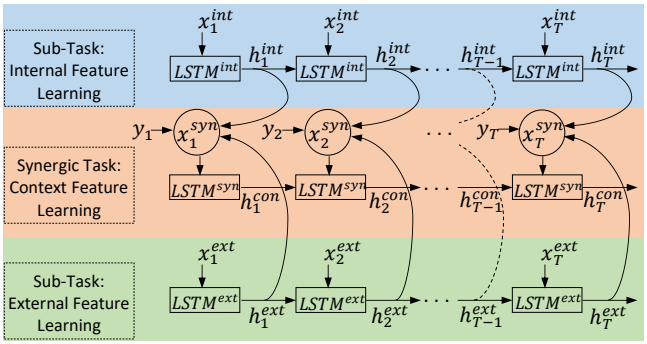

Fig. 2. The unfolded structure of our proposed multi-task LSTM encoder. Two sub-tasks consist of internal feature learning and external feature learning LSTMs, denoted by $LSTM^{int}$ and $LSTM^{ext}$ respectively. After latent representations of both internal and external features are generated, they are combined with the real sales number $\{\mathbf{y}_t\}_{t=1}^{T}$ to compute the contextual vectors $\{\mathbf{h}_t^{con}\}_{t=1}^{T}$ via the synergic task LSTM ($LSTM^{syn}$).

Hence, we formulate sales prediction as a non-linear mapping from time series features $\{\mathbf{x}_t\}_{t=1}^{T}$ and real sales $\{y_t\}_{t=1}^{T}$ in the history to the estimation of sales volume $\{\hat{y}_t\}_{t=T+1}^{T+\Delta}$ with Δ time steps ahead:

$$\{\hat{y}_t\}_{t=T+1}^{T+\Delta} = F\Big(\{\mathbf{x}_t\}_{t=1}^{T}, \{y_t\}_{t=1}^{T}\Big), \tag{1}$$

where $F(\cdot)$ is the non-linear mapping function to learn.

B. Multi-Task based Encoder Structure

Taking a time series $\{\mathbf{x}_t\}_{t=1}^{T}$ as input, recurrent neural network (RNN) encodes $\{\mathbf{x}_t\}_{t=1}^{T}$ into hidden states $\{\mathbf{h}_t\}_{t=1}^{T}$ via $\mathbf{h}_t = f(\mathbf{x}_t, \mathbf{h}_{t-1})$, where $f(\cdot)$ is a non-linear mapping function. To capture the long-range dependency, we leverage RNNs with long short-term memory architecture (LSTM) via the following formulation [19]:

$$\begin{aligned}
\mathbf{i}_t &= \sigma(\mathbf{W}_i\mathbf{x}_t + \mathbf{U}_i\mathbf{h}_{t-1} + \mathbf{b}_i),\\
\mathbf{f}_t &= \sigma(\mathbf{W}_f\mathbf{x}_t + \mathbf{U}_f\mathbf{h}_{t-1} + \mathbf{b}_f),\\
\mathbf{o}_t &= \sigma(\mathbf{W}_o\mathbf{x}_t + \mathbf{U}_o\mathbf{h}_{t-1} + \mathbf{b}_o),\\
\mathbf{c}_t &= \mathbf{f}_t\mathbf{c}_{t-1} + \mathbf{i}_t \odot \tanh(\mathbf{W}_c\mathbf{x}_t + \mathbf{U}_c\mathbf{h}_{t-1} + \mathbf{b}_c),\\
\mathbf{h}_t &= \mathbf{o}_t \odot \tanh(\mathbf{c}_t),
\end{aligned} \tag{2}$$

where $\odot$ denotes element-wise multiplication and the recurrent activation σ is the *Logistic Sigmoid* function. $\mathbf{i}$, $\mathbf{f}$, $\mathbf{o}$ and $\mathbf{c}$ are respectively the input gate, forget gate, output gate, and cell state vectors. When updating each of them, there are corresponding trainable input-to-hidden and hidden-to-hidden weights $\mathbf{W}$ and $\mathbf{U}$ along with the bias vectors $\mathbf{b}$.

For sales prediction, ***internal feature*** and ***external feature*** are two kinds of features with different semantic meanings in sales time series. We use $\{\mathbf{x}_t^{int}\}_{t=1}^{T}$ and $\{\mathbf{x}_t^{ext}\}_{t=1}^{T}$ to denote the feature vectors of internal and external information in sales time series respectively. As we discussed in previous sections, internal features carry information of intrinsic attributes directly linked with the product like store location and item category, while the external features store information of extrinsic attributes viewed as external influential factors like weather condition and holiday. As a result, a single LSTM structure may suffer from loss of contextual information as it maps all

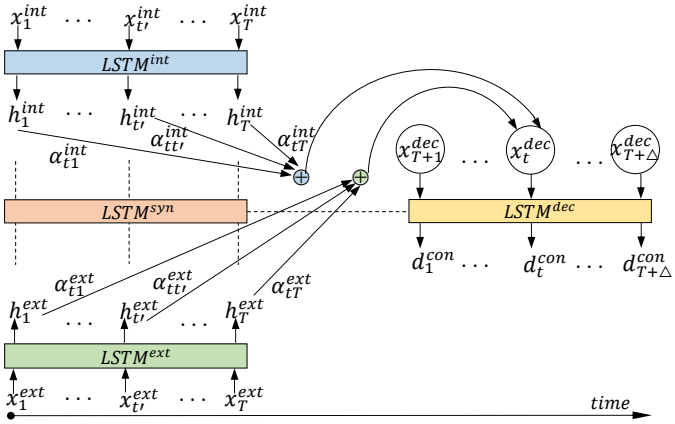

Fig. 3. Demonstration of proposed attention mechanism for weighted input mapping. The details of $LSTM^{syn}$ are omitted for a clearer view. With the calculated attention weights $\alpha_{tt'}^{int}$ and $\alpha_{tt'}^{ext}$, the latent representations generated by $LSTM^{int}$ and $LSTM^{ext}$ are mapped into the input vectors $\{\mathbf{x}_t^{dec}\}_{t=T+1}^{T+\Delta}$ for the decoder $LSTM^{dec}$.

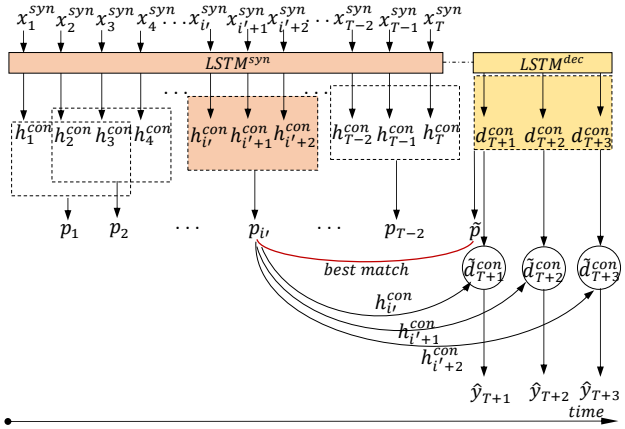

Fig. 4. Demonstration of proposed attention mechanism for trend alignment. The process for generating output label is included as well. We omit $LSTM^{int}$ and $LSTM^{ext}$ to be succinct. Note that we assume $\Delta = 3$ in this figure for better readability. The essence is to find a best match denoted by $\mathbf{p}_{i'}$ for the current trend $\tilde{\mathbf{p}}$. Afterwards, we sequentially join the aligned contextual vector pairs within two trends to produce the final contextual vectors $\{\tilde{\mathbf{d}}_t^{con}\}_{t=T+1}^{T+\Delta}$ and then predict the upcoming sales $\{\hat{y}_t\}_{T+1}^{T+\Delta}$.

raw features into one unified space, as we will reveal in Section IV. Hence, we use two LSTMs in parallel to effectively capture the different semantics by treating internal and external feature modelling as two sub-tasks. Correspondingly, we extend the problem formulation in Eq.(1) as:

$$\{\hat{y}_t\}_{t=T+1}^{T+\Delta} = F\Big(\{\mathbf{x}_t^{int}\}_{t=1}^{T}, \{\mathbf{x}_t^{ext}\}_{t=1}^{T}, \{y_t\}_{t=1}^{T}\Big). \quad (3)$$

Figure 2 demonstrates our proposed encoder architecture. We use $\{\mathbf{h}_t^{int}\}_{t=1}^{T}$ and $\{\mathbf{h}_t^{ext}\}_{t=1}^{T}$ to denote the latent representations learned from $\{\mathbf{x}_t^{int}\}_{t=1}^{T}$ and $\{\mathbf{x}_t^{ext}\}_{t=1}^{T}$. After the hidden states are learned from both sub-tasks, we simultaneously feed those hidden states into a synergic LSTM layer to learn a joint representation, namely ***contextual vectors*** denoted by $\{\mathbf{h}_t^{con}\}_{t=1}^{T}$ at all T time steps in the sales time series. Furthermore, to enhance the expressive ability of the encoder, instead of adopting $\{y_t\}_{t=1}^{T}$ to calculate the prediction loss, we fuse $\{y_t\}_{t=1}^{T}$ with hidden states from both internal and external encoding LSTMs to calculate the input $\{\mathbf{x}_t^{syn}\}_{t=1}^{T}$ for the synergic layer:

$$\mathbf{x}_t^{syn} = \mathbf{W}_{syn}[\mathbf{h}_t^{int}; \mathbf{h}_t^{ext}; y_t] + \mathbf{b}_{syn}, \quad (4)$$

where $[\mathbf{h}_t^{int}; \mathbf{h}_t^{ext}; y_t]$ represents the concatenation of $\mathbf{h}_t^{int}$, $\mathbf{h}_t^{ext}$ and y_t while $\mathbf{W}_{con}$ and $\mathbf{b}_{con}$ are weights and biases to be learned. For notation convenience, we format the multi-task encoder structure into the following equations:

$$\begin{aligned} \mathbf{h}_t^{int} &= LSTM^{int}(\mathbf{x}_t^{int}, \mathbf{h}_{t-1}^{int}), \\ \mathbf{h}_t^{ext} &= LSTM^{ext}(\mathbf{x}_t^{ext}, \mathbf{h}_{t-1}^{ext}), \\ \mathbf{h}_t^{con} &= LSTM^{syn}(\mathbf{x}_t^{syn}, \mathbf{h}_{t-1}^{con}), \end{aligned} \quad (5)$$

where $LSTM^{int}(\cdot)$, $LSTM^{ext}(\cdot)$ and $LSTM^{syn}(\cdot)$ denote internal, external and synergic LSTM encoders respectively. Note that the trainable weights are not shared across different LSTM layers in our multi-task encoder structure.

C. Dual-Attention based Decoder Structure

After encoding the entire historical sales time series with the multi-task encoder, we have the ***contextual vectors*** $\{\mathbf{h}_t^{con}\}_{t=1}^{T}$ where each $\mathbf{h}_t^{con}$ carries contextual information of the sales time series at time step t. The latent representations, $\{\mathbf{h}_t^{int}\}_{t=1}^{T}$ and $\{\mathbf{h}_t^{ext}\}_{t=1}^{T}$ for internal and external features are also learned. To predict the desired sales volume $\{\hat{y}_t\}_{t=T+1}^{T+\Delta}$, we adopt a LSTM decoder to mimic the ***contextual vectors*** in the following Δ time steps. Similar to Eq.(5), when $T < t \leq T + \Delta$, we have:

$$\mathbf{d}_t^{con} = LSTM^{dec}(\mathbf{x}_t^{dec}, \mathbf{d}_{t-1}^{con}), \quad (6)$$

where $\mathbf{d}_t^{con} \in \{\mathbf{d}_t^{con}\}_{t=T+1}^{T+\Delta}$ is the contextual vector to learn in the decoding stage at time step t, $LSTM^{dec}(\cdot)$ is the decoder with the same formulation as Eq.(2), $\mathbf{x}_t^{dec}$ is the ***attention-weighted*** input for the decoder and $\mathbf{d}_{t-1}^{con}$ is the previous decoder hidden state.

*1) **Attention for Weighted Decoder Input Mapping**:* According to the problem formulation, we assume that both $\{\mathbf{x}_t^{int}\}_{t=T+1}^{T+\Delta}$ and $\{\mathbf{x}_t^{ext}\}_{t=T+1}^{T+\Delta}$ are non-available in the decoding stage because both of them contain attributes unknown to the future, such as price as an internal feature and weather as an external feature. Thus, to formulate the decoder input at time $t > T$, we propose an attention mechanism to dynamically select and combine relevant contextual vectors from $\{\mathbf{h}_t^{int}\}_{t=1}^{T}$ and $\{\mathbf{h}_t^{ext}\}_{t=1}^{T}$ with:

$$\mathbf{x}_t^{dec} = \mathbf{W}_{dec}\left[\sum_{t'=1}^{T}\alpha_{tt'}^{int}\mathbf{h}_{t'}^{int}; \sum_{t'=1}^{T}\alpha_{tt'}^{ext}\mathbf{h}_{t'}^{ext}\right] + \mathbf{b}_{dec}, \quad (7)$$

where $\alpha_{tt'}^{int}$ and $\alpha_{tt'}^{ext}$ denote the attention weights mapped to t'-th hidden states of internal and external feature encoder, respectively. We use Fig.3 to illustrate the attention for weighted decoder input mapping process. We enforce $\sum_{t'=1}^{T}\alpha_{tt'}^{int} = \sum_{t'=1}^{T}\alpha_{tt'}^{ext} = 1$, so that $[\cdot]$ in Eq.(7) can be viewed as the concatenation of two probability expectations from $\{\mathbf{h}_t^{int}\}_{t=1}^{T}$ and $\{\mathbf{h}_t^{ext}\}_{t=1}^{T}$. The rationale is that we simulate $\mathbf{x}_t^{dec}$ by summarizing varied influences from all $2T$ historical hidden states of both internal and external feature.

The influences are computed through quantifying the relevance between $\mathbf{d}_{t-1}^{con}$ and each $\mathbf{h}_{t'}^{int}$, $\mathbf{h}_{t'}^{ext}$:

$$\begin{aligned} e_{tt'}^{int} &= \mathbf{v}_{int}^{\top} \tanh(\mathbf{M}_{int}\mathbf{d}_{t-1}^{con} + \mathbf{H}_{int}\mathbf{h}_{t'}^{int}), \\ e_{tt'}^{ext} &= \mathbf{v}_{ext}^{\top} \tanh(\mathbf{M}_{int}\mathbf{d}_{t-1}^{con} + \mathbf{H}_{ext}\mathbf{h}_{t'}^{ext}), \end{aligned} \tag{8}$$

where $e_{tt'}^{int}$ and $e_{tt'}^{ext}$ are the relevance scores mapped to t'-th hidden states in $\{\mathbf{h}_t^{int}\}_{t=1}^{T}$ and $\{\mathbf{h}_t^{ext}\}_{t=1}^{T}$ for the decoder input at time t, while $\mathbf{v}_{int}$, $\mathbf{v}_{ext}$, $\mathbf{M}_{int}$, $\mathbf{v}_{ext}$, $\mathbf{H}_{int}$ and $\mathbf{H}_{ext}$ are parameters to learn. In particular, Eq.(8) compares two hidden states with different semantic meanings. Intuitively, this is a scoring scheme that shows how well two vectors are correlated by projecting them into a common space. Afterwards, we apply $SoftMax$ on both attention weights:

$$\begin{aligned} \alpha_{tt'}^{int} &= \frac{\exp(e_{tt'}^{int})}{\sum_{s=1}^{T}\exp(e_{ts}^{int})}, \\ \alpha_{tt'}^{ext} &= \frac{\exp(e_{tt'}^{ext})}{\sum_{s=1}^{T}\exp(e_{ts}^{ext})}, \end{aligned} \tag{9}$$

which enforces $\sum_{t'=1}^{T}\alpha_{tt'}^{int} = \sum_{t'=1}^{T}\alpha_{tt'}^{ext} = 1$.

*2) **Attention for Trend Alignment**:* Idealy, at time t, each acquired contextual vector in $\{\mathbf{h}_t^{con}\}_{t=1}^{T}$ and $\{\mathbf{d}_t^{con}\}_{t=T+1}^{T+\Delta}$ carries contextual information of both time t and previous time steps. However, as discussed in [4], [20], the performance of the encoder-decoder networks decrease significantly when the length of time series grows. To alleviate the problem, traditional attention mechanisms have been designed to align the current output with the targeted input by comparing the current hidden state with the ones generated at previous time steps. Meanwhile, these methods are not applicable as we aim to match similar trends for the prediction period Δ, and we propose a novel attention mechanism for trend alignment. Mathematically, we represent a Δ-step trend in sales time series as the concatenation of Δ successive contextual vectors in $\{\mathbf{h}_t^{con}\}_{t=1}^{T}$:

$$\mathbf{p}_i = [\mathbf{h}_i^{con}; \mathbf{h}_{i+1}^{con}; ...; \mathbf{h}_{i+\Delta-1}^{con}], \quad 1 \le i \le T-\Delta+1 \tag{10}$$

where $\mathbf{p}_i$ denotes the i-th trend in T with a timespan of Δ. Similarly, we represent the upcoming trend $\widetilde{\mathbf{p}}$ in the $[T+1, T+\Delta]$ time interval via the concatenation of all contextual vectors in $\{\mathbf{d}_t^{con}\}_{t=T+1}^{T+\Delta}$:

$$\widetilde{\mathbf{p}} = [\mathbf{d}_{T+1}^{con}; \mathbf{d}_{T+2}^{con}; ...; \mathbf{d}_{T+\Delta}^{con}]. \tag{11}$$

We explain the workflow of attention for trend alignment in Fig.4. As demonstrated in Fig.4, when the trend index i increases from 1 to $T-\Delta+1$, $\mathbf{p}_i$ can be viewed as a sliding window that dynamically captures temporary contextual information learned from existing sales time series with respective step and window size as 1 and Δ. Hence, we compute the relevance score between $\widetilde{\mathbf{p}}$ and each $\mathbf{p}_i \in \{\mathbf{p}_i\}_{i=1}^{T-\Delta+1}$, with:

$$e_i^{trd} = \mathbf{p}_i^{\top}\widetilde{\mathbf{p}}, \tag{12}$$

and then find out the best match of $\widetilde{\mathbf{p}}$:

$$i' = \text{argmax}(e_i^{trd}, e_{i+1}^{trd}, ..., e_{T+\Delta-1}^{trd}), \tag{13}$$

where e_i^{trd} denotes the relevance between $\widetilde{\mathbf{p}}$ and $\mathbf{p}_i$, while i' indicates the i'-th trend in $\{\mathbf{p}_i\}_{i=1}^{T-\Delta+1}$ is the most relevant to $\widetilde{\mathbf{p}}$. Because $\widetilde{\mathbf{p}}$ and $\mathbf{p}_i$ express similar contextual semantics with the same dimensionality, we don't use the scoring scheme in Eq.(8) but adopt the dot product to be computational efficient. Intuitively, the closer $\widetilde{\mathbf{p}}$ and $\mathbf{p}_i$ are, a larger e_i^{trd} will be generated and vice versa (e_i^{trd} = 0 when orthogonal), so we can align the upcoming trend $\widetilde{\mathbf{p}}$ with its best match $\mathbf{p}_{i'} = [\mathbf{h}_{i'}^{con}; \mathbf{h}_{i'+1}^{con}; ...; \mathbf{h}_{i'+\Delta-1}^{con}]$.

More importantly, now the contextual vectors within both trends, i.e., $\{\mathbf{d}_t^{con}\}_{t=T+1}^{T+\Delta}$ and $\{\mathbf{h}_t^{con}\}_{t=i'}^{i'+\Delta-1}$ are also aligned as trend components instead of individual hidden states. With the upcoming sales trend $\widetilde{\mathbf{p}}$ aligned with the i'-th historical trend, we merge each pair of contextual vector in $\{\mathbf{d}_t^{con}\}_{t=T+1}^{T+\Delta}$ and $\{\mathbf{h}_t^{con}\}_{t=i'}^{i'+\Delta-1}$ into the aligned representation of contextual vectors:

$$\begin{aligned} \widetilde{\mathbf{d}}_t^{con} &= \mathbf{W}_{ali}[\mathbf{d}_j^{con}; \mathbf{h}_k^{con}] + \mathbf{b}_{ali}, \\ T+1 \le j \le T, &\quad i' \le k \le i'+\Delta-1, \end{aligned} \tag{14}$$

where $\widetilde{\mathbf{d}}_t^{con}$ is the aligned contextual vectors at time t, $\mathbf{W}_{ali}$ and $\mathbf{b}_{ali}$ are parameters to learn, $[\mathbf{d}_j^{con}; \mathbf{h}_k^{con}]$ is the concatenation of aligned contextual vector pair. We use the following algorithm to acquire the full set of aligned contextual vectors for sales prediction:

Algorithm 1 Generate Aligned Contextual Vectors

1: **Input:** prediction time steps Δ; aligned trend index i'; encoded time length T; sales contextual vectors $\{\mathbf{d}_t^{con}\}_{t=T+1}^{T+\Delta}$ and $\{\mathbf{h}_t^{con}\}_{t=i'}^{i'+\Delta-1}$
2: **Output:** aligned representations of contextual vectors $\{\widetilde{\mathbf{d}}_t^{con}\}_{t=T+1}^{T+\Delta}$
3: initialize with $j = T+1$, $k = i'$;
4: **while** $j \le T+\Delta$ **and** $k \le i'+\Delta-1$ **do**
5: update $\widetilde{\mathbf{d}}_t^{con}$ via Eq.(14);
6: $j++$;
7: $k++$;
8: **end**

Here, $\{\widetilde{\mathbf{d}}_t^{con}\}_{t=T+1}^{T+\Delta} = \{\widetilde{\mathbf{d}}_{T+1}^{con}, \widetilde{\mathbf{d}}_{T+2}^{con}, ..., \widetilde{\mathbf{d}}_{T+\Delta}^{con}\}$ contains the final latent representation at each upcoming time step in the simulated sales context.

D. Sales Prediction and Model Learning

With the aligned contextual vectors $\{\widetilde{\mathbf{d}}_t^{con}\}_{t=T+1}^{T+\Delta}$ generated, we approximate the future sales with regression:

$$\hat{y}_t = \mathbf{v}_y^{\top}\widetilde{\mathbf{b}}_t^{con} + b_y, \tag{15}$$

where $\hat{y}_t \in \{\hat{y}_t\}_{t=T+1}^{T+\Delta}$ denotes the predicted sales at time t, $\mathbf{v}_y^{\top}$ and b_y are parameters to learn.

For model learning, we apply the simple yet effective mean squared error coupled with *L2* regularization (to prevent overfitting) on model parameters:

$$\mathcal{L}_F = \frac{1}{N}\left(\sum_{n=1}^{N}\sum_{t=T+1}^{T+\Delta}(\hat{y}_{nt} - y_{nt})^2\right) + \lambda\sum_{l}^{L}\theta_l^2, \tag{16}$$

where $n \leq N$ is the number of training samples, $l \leq L$ is the index of model parameters, y_{nt} is the actual label of sales at t-th time step, θ_l is the model parameter, and λ is the weight decay coefficient that needs to be tuned.

In the training procedure, we leverage mini-batch Stochastic Gradient Decent (SGD) based algorithm, namely Adam [30] optimizer. Specifically, we set the batch size as 128 according to device capacity and the start learning rate as 0.001 which is reduced by 10% after each 10,000 iterations. We iterate the whole training process until the loss converges.

III. Time Complexity of TADA

Because the proposed multi-task, dual-attention RNN model is heavily associated with multiple parameters, here we discuss its time complexity in detail. We prove that like a standard LSTM system, with the model parameters fixed, the asymptotic time complexity of TADA is linear to the size of data.

For a basic LSTM cell in Eq.(2), we denote the number of hidden dimensions as q (i.e., $\mathbf{h} \in \mathbb{R}^{q \times 1}$). According to [19], [31], ignoring the biases, a single-task LSTM with T time steps has the complexity of $O(q^2T)$. Similarly, we formulate the time complexity for our encoder-decoder structure. Assuming all LSTMs in TADA have q hidden dimensions, and the multi-task encoder structure with $LSTM^{int}$, $LSTM^{ext}$ and $LSTM^{syn}$ are deployed in parallel, the time complexity is $O\big(q^2(T+\Delta)\big)$, which is identical to a basic encoder-decoder LSTM structure.

Then, we focus on the dual-attention mechanism. Since Eq.(8) can be viewed as two parallel feed-forward networks, the complexity is $O(q^2)$ for each time step. Coupled with Eq.(7), the time complexity of attention mechanism in section II-C1 is $O(q^2T\Delta + q^2\Delta) = O\big(q^2(T+1)\Delta\big) \simeq O(q^2T\Delta)$. According to [29], dot product based attention mechanism Eq.(12) has the complexity of $O\big(q\Delta(T-\Delta+1)\big) \simeq O(qT\Delta - q\Delta^2)$. Combining with Eq.(14), the overall complexity of attention mechanism in section II-C2 is $O(q^2\Delta + qT\Delta - q\Delta^2)$.

With the complexity of encoder-decoder and dual-attention mechanism sorted, we aggregate the complexity for generating the aligned contextual vectors $\{\widetilde{\mathbf{d}}_t^{con}\}_{t=T+1}^{T+\Delta}$. Note that the complexity of Eq.(4) throughout time T is $O(q^2T)$, and the complexity of Eq.(15) throughout time Δ is $q\Delta^2$. Finally, the overall complexity of TADA comes to $O\big(2q^2(T+\Delta) + qT(q+\Delta)\big)$. In practice, we have $\Delta \ll T$ and $\Delta \ll q$, so T and q are dominating in dimensionality. Therefore, we simplify the final time complexity as $O(3q^2T) \rightarrow O(q^2T)$. For a dataset with N samples (time series), it takes $O(Nq^2T)$ to go through the entire dataset once. In summary, when the hidden dimension q and total time step T is fixed, the time complexity of TADA is linearly associated with the scale of the data.

IV. Experiments

In this section, we conduct experiments on real commercial datasets to showcase the advantage of TADA in the task of sales prediction. In particular, we aim to answer the following research questions via the experiments:

TABLE I
Statistics of datasets in use.

Dataset	Time Series	Granularity	Time Range	Variables
Favorita	11,536	1 day	365 days	13
OSW	1,585	1 week	106 weeks	11

- How effectively and accurately TADA can predict continuous sales volume with observed sales time series from the past.
- How TADA benefits from each component of the proposed structure for sales prediction.
- How efficiently TADA can be trained when handling training data with different sizes.

A. Datasets and Features

To validate the performance of TADA, we use two real-life commercial datasets shown in Table I, namely **Favorita** and **OSW**. Here we briefly introduce the properties of these two datasets below:

- **Favorita**: It contains the daily features and sales volume of all products in 56 Ecuadorian-based grocery stores. Note that the original Favorita dataset covers the time range from 1 January 2013 to 15 August 2017, but we only use the portion from 15 August 2016 to 15 August 2017 (365 days) due to two reasons: (1) a magnitude 7.8 earthquake struck Ecuador on 16 April 2016, which exerted abnormal sales patterns in the following few weeks[1]; (2) shorter time series suits the real-life conditions better as it is faster for the model to learn.
- **OSW**: One Stop Warehouse[2] is one of the largest solar energy appliance suppliers in Australia. The dataset covers 12 warehouses' weekly sales volume of various products (e.g., solar panels, batteries, etc.) from 22 February 2016 to 4 March 2017 (106 weeks). Empirically, sales prediction on OSW dataset is more challenging from two perspectives: (1) the sales volume of solar energy appliances is more dependent on external causes (e.g., policy, electricity price, promotion, etc.), which are unavailable in this dataset; (2) the sales volume in OSW dataset fluctuates more significantly than Favorita.

The features we used from the datasets are listed in Table II. Features consist of binary (represented as 1 or 0), categorical (represented via one-hot vectors) and numerical data, which are marked by superscripts of b, c and n respectively. To accelerate the training process, we process all the numerical features by performing $\log_{10}$ transfer (a small bias of 0.001 was added to all numerics to avoid the case of 0). In addition, we leverage embedding to reduce the original dimensionality of categorical data, and combine all these features together as the model input. As suggested by the Tensorflow research team from Google[3], we set the embedding dimension of each categorical feature by taking the 4th root of the total amount of

[1] https://www.kaggle.com/c/favorita-grocery-sales-forecasting/data

[2] https://www.onestopwarehouse.com.au

[3] https://developers.googleblog.com/2017/11/introducing-tensorflow-feature-columns.html

TABLE II
SUMMARIZATION OF FEATURES EXTRACTED FROM DATASETS.

Dataset	Type	Feature	Dimension	
Favorita	internal feature	city of store^c	3*	17
		state of store^c	2*	
		store type^c	2*	
		store group^c	2*	
		item family^c	3*	
		item class^c	5*	
	external feature	promotion state^b	1	11
		date^c	5*	
		store transaction^n	1	
		oil price^n	1	
		local holiday^b	1	
		national holiday^b	1	
		pay day^b	1	
OSW	internal feature	item index^c	5*	16
		city of store^c	2*	
		item category^c	5*	
		battery type^c	3*	
		item price^n	1	
	external feature	week number^c	4*	9
		discontinued state^b	1	
		solar exposure^n	1	
		temperature^n	1	
		week(s) after last holiday^n	1	
		week(s) to next holiday^n	1	

categories. In Table II, numbers with '$*$' mean the dimension of embedding for categorical features.

In both datasets, each time series is actually a log file for a specific product. Hence, we don't split different products up for training and test because it means many products are totally new to the model during test, which is not realistic in real business. So, we first randomly take 3,000 and 400 time series out of Favorita and OSW dataset for validation. Then, given time series with the total time steps of M (365 for Favorita and 106 for OSW) and Δ steps to predict, we apply the 'walk-forward' split strategy on the remaining data. For training, we encode the information with $t \in [1, M-2\Delta]$ and predict sales with $t \in [M-2\Delta+1, M-\Delta]$. For evaluation, we encode the information with $t \in [\Delta+1, M-\Delta]$ and predict sales with $t \in [M-\Delta+1, M]$ to test the accuracy. This test strategy has more practical meaning in the real world, where most businesses tend to predict future sales volume according to previous records.

B. Parameters and Experimental Settings

In TADA, we apply the same size to the hidden states of all LSTM systems to maintain the consistency of contextual feature dimension. That is to say, there are only two hyper parameters in TADA to be determined, namely the size of hidden states and the weight decay penalty λ. We conduct grid search for the number of hidden states and λ over $\{32, 64, 128, 256, 512\}$ and $\{0.001, 0.01, 0.1, 1, 10\}$ respectively. The settings with the best performance on the validation set ($\lambda = 0.01$ on Favorita, $\lambda = 0.1$ on OSW, and 128 hidden states for both datasets) are used in the test.

We conduct experiments against the following state-of-the-art predictive methods:

- **Random Forest (RF)**: We implement a widely-used, predictive decision tree based model, random forest to predict sales from the observed features.
- **XGBoost**: It stands for extreme gradient boosting, proposed by Chen *et al.* [32]. It is a state-of-the-art, gradient boosted regression tree approach based on the gradient boosting machine (GBM) [34].
- **SAE-LSTM**: From the cutting edge of economics research, we adpot the stacked auto encoder with LSTM (SAE-LSTM) [33] which is a neural network based model proposed for financial time series prediction.
- **A-RNN**: Attention RNN (A-RNN) was originally designed by Bahdanau *et al.* for machine translation tasks [21], with the output of a probability distribution over the word dictionary. We modify the output layer by mapping the learned hidden states into scalar values and use the loss function in Eq.(16) for sales prediction task.
- **DA-RNN**: This is a non-linear autoregressor (AR) with attentions in both encoder and decoder RNNs [4]. Compared with A-RNN, the proposed encoder attention in DA-RNN assumes the inputs must be correlated along the time, which is not always true in sales time series.
- **LSTNet**: It is a deep learning framework (long- and short-term time series network) designed for multivariate time series prediction [9]. This method combines convolutional neural network and a recurrent-skip network to capture both short-term and long-term trending patterns of the time series.

Furthermore, to fully study the performance gain from each component of our proposed model, we implement three degraded versions of TADA:

- **TADA-SE**: We replace the multi-task based encoder with a single-task, 1-layer LSTM encoder. The internal and external feature vectors are concatenated as its input.
- **TADA-SA$_1$**: We remove the attention mechanism for decoder input mapping to build a single-attention variant.
- **TADA-SA$_2$**: We remove the attention mechanism for trend alignment to build another single-attention variant.

To measure the effectiveness of all the methods in sales prediction, we adopt two evaluation metrics, namely mean absolute error (MAE) and symmetric mean absolute percentage error (SMAPE). Mathematically, they are defined as follows:

$$\begin{aligned} \textbf{MAE} &= \frac{1}{N \times \Delta} \sum_{n=1}^{N} \sum_{t=T+1}^{T+\Delta} |y_t - \hat{y}_t|, \\ \textbf{SMAPE} &= \frac{100\%}{N \times \Delta} \sum_{n=1}^{N} \sum_{t=T+1}^{T+\Delta} \left(\begin{array}{c} 0, \ \ if \ y_t = \hat{y}_t = 0 \\ \frac{|y_t - \hat{y}_t|}{(|y_t| + |\hat{y}_t|)/2}, otherwise \end{array} \right), \end{aligned} \tag{17}$$

where y_t and $\hat{y}_t$ denote real and predicted sales volume respectively. We choose them because MAE is scale-dependent while SMAPE is not, so MAE is suitable for comparison of different methods on the same dataset and SMAPE suits comparison across different datasets. We test all methods on two datasets with $\Delta \in \{2, 4, 8\}$ to showcase their robustness in multiple sales prediction scenarios.

TABLE III
SALES PREDICTION RESULTS. NUMBERS IN BOLD FACE ARE THE BEST RESULTS WITHIN EACH COLUMN.

Method	Favorita						OSW					
	$\Delta = 2$		$\Delta = 4$		$\Delta = 8$		$\Delta = 2$		$\Delta = 4$		$\Delta = 8$	
	MAE	SMAPE(%)	MAE	SMAPE(%)	MAE	SMAPE(%)	MAE	SMAPE(%)	MAE	SMAPE(%)	MAE	SMAPE(%)
RF	32.483	200(max)	35.507	200(max)	41.329	200(max)	29.147	89.482	35.576	137.892	43.096	200(max)
XGBoost [32]	16.705	87.433	19.833	91.230	22.547	158.461	21.496	49.556	24.916	53.243	30.322	82.633
SAE-LSTM [33]	7.364	39.447	8.033	44.384	8.116	46.932	17.828	44.241	19.805	46.887	20.823	49.873
A-RNN [21]	11.610	60.781	12.226	62.397	13.005	65.812	17.391	44.635	18.823	44.603	22.129	49.180
DA-RNN [4]	7.816	43.859	8.234	44.704	8.566	46.281	17.634	44.215	19.578	47.139	20.693	48.365
LSTNet [9]	7.419	43.523	7.982	45.662	8.729	48.469	16.625	42.317	18.989	45.782	21.246	49.191
TADA-SE	9.995	58.715	11.076	60.332	10.955	60.257	19.635	53.017	20.884	49.370	21.687	51.685
TADA-SA_1	8.152	46.732	8.273	43.951	8.968	49.079	16.585	42.620	18.624	44.331	21.699	51.195
TADA-SA_2	7.635	42.883	8.247	44.942	8.626	48.609	17.087	42.199	18.643	45.219	21.190	49.825
TADA	**6.955**	**38.770**	**7.323**	**40.588**	**7.422**	**43.675**	**15.418**	**41.354**	**17.572**	**43.265**	**19.618**	**47.782**

C. Discussion on Effectiveness Results

We report the results of all tested methods on all Δ settings in Table III, where the best performance is highlighted with bold face. MAE measures the error with the deviation between predicted and real sales volume, and SMAPE quantifies such error with a proportional perspective.

It is as expected that all neural network based predictive models outperform decision tree based models (RF and XGBoost) by a significant margin in both datasets. Hence, we can empirically suggest that deep neural networks better suit the task of sales prediction in the real-world scenario. Apparently, the performance of all methods start to drop when we gradually increase the time range for sales prediction with $\Delta \in \{2, 4, 8\}$. However, among this observation, TADA demonstrates the least negative impact from the increasing Δ and presents the dominating prediction performance against all state-of-the-art baselines. In other words, the trend alignment scheme from TADA can practically meet the requirement of sales prediction when merchants are trying to look ahead at more upcoming time steps. When comparing with other deep neural network based approaches (SAE-LSTM, A-RNN, DA-RNN and LSTNet) the results also support the superiority of TADA. This is because: (1) the multi-task based encoder in TADA is better at capturing the interactive effect from both internal and external features to the real sales than modelling all influential factors in the unified way; (2) the dual attention architecture in TADA successfully captures latent trends from the past which are similar to the upcoming one, especially when comparing with existing attention mechanisms (A-RNN and DA-RNN) and periodic trend modelling method (LSTNet). The effectiveness of each proposed component in TADA is initially revealed in Table III by its degraded versions, which we will further discuss in the following section.

D. Discussion on Model Components

We implement three variants of our proposed model TADA, namely TADA-SE, TADA-SA_1 and TADA-SA_2, by removing one of the key components each time. With the degraded versions of TADA, we carry out the ablation study on the performance gain from every proposed component within TADA. As shown in Table III, the evaluation results on two real datasets indicate that these variants suffer from noticeable drops in the prediction performance. Specifically, TADA-SE shows more obvious infection. This provides evidence for our assertion that by dividing the influential factors in sales time series into semantically different internal and external features, the multi-task based encoder structure can extract more latent contextual information related to the real sales volume. In TADA-SE, the dynamic interaction of internal and external features are no longer modelled, causing insufficient performance accuracy.

According to Table III, when we remove each one of the two proposed attention mechanisms in TADA-SA_1 and TADA-SA_2, the prediction performance both drops. Combining their performance on both datasets, the performance reduction is similar when either part of the dual attention mechanism is blocked. So, we draw the observation that both attentions contribute positively and almost equally, and they are indispensable to each other for precise sales prediction. Thus, after the contextual vectors are learned from the encoder, it is crucial to leverage the dual attention based decoder to mimic the contextual information in the future as well as aligning the upcoming trend with historical ones to enhance the prediction of sales. Furthermore, as the attention mechanism provides TADA (full version) with a better interpretability, we visualize the intermediate results of aligned trends in the predicting (decoding) stage, along with the predicted sales. Fig.(5) visualizes the results of trend alignment from samples selected from both Favorita and OSW datasets by highlighting the sales trend with the highest attention weight. As a result, we find that similar sales contexts lead to similar sales volume, which confirms the rationale of performing trend alignment for sales prediction and the effectiveness of all components in TADA.

E. Discussion on Training Efficiency and Scalability

Due to the importance of practicality in real-life applications, we validate the scalability of TADA. As we proved in Section III, when all the parameters in the network are fixed (in our case, the dimension for all hidden states is 128, and T is determined according to Δ), the training time for TADA is only associated with the number of training samples. Ideally, the training time for TADA should increase linearly as we enlarge the scale of the training data. Note that we set $\Delta = 8$ ($T = 349$ correspondingly) for this validation.

We test the training efficiency and scalability of TADA by using different proportions of the whole training set from Favorita, and then report the corresponding training time

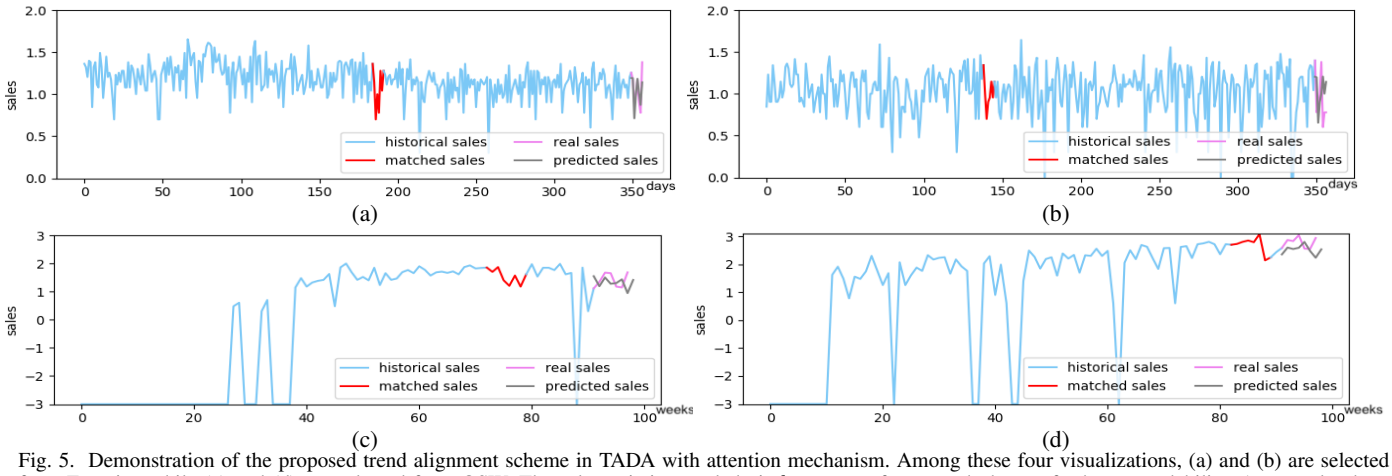

Fig. 5. Demonstration of the proposed trend alignment scheme in TADA with attention mechanism. Among these four visualizations, (a) and (b) are selected from Favorita, while (c) and (d) are selected from OSW. The sales axis is rescaled via $\log_{10}$ transfer on each dataset for better readability. Apparently, there are no obvious recurring trends in all these sales records, but TADA successfully selects the most relevant one to assist the prediction. The figures illustrate that aligned trends in sales time series not only share similar contexts, but also have close sales volume.

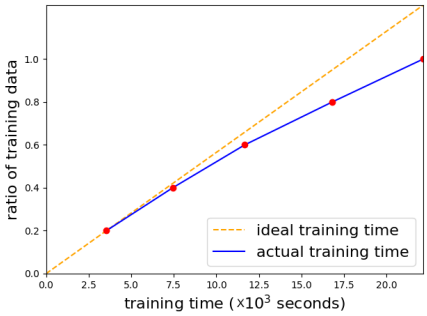

Fig. 6. The training time of TADA with varied proportions of training data.

(excluding I/O). The growth of training time along with the data size is shown in Fig.6. When the ratio of training data gradually extends from 0.2 to 1.0, the training time for TADA increases from 3.54×10^3 seconds to 22.15×10^3 seconds. It shows that the link between training time and the data scale is approximately linear. Hence, we conclude that since its linear time complexity can ensure high scalability, TADA can be efficiently trained with large-scale datasets.

V. Related Work

When performing sales prediction using multivariate time series, the techniques can be divided into linear models and non-linear models. While linear models like autoregressive integrated moving average (ARIMA) [14], support vector machine (SVM) [35] and robust regression [36] mostly aim at finding parameterized functions from statistics, non-linear models like Gaussian process [17], [37] and gradient boosting machines [32], [34] can better model complicated dependencies by leveraging machine learning techniques. However, due to the high computational cost and unsatisfying scalability in real applications [9], [38], these approaches are not ideal for sales time series which usually carries high dimensionality and long time range. In addition, these methods mainly rely on carefully designed mapping functions, so sufficient domain knowledge of the data is a prerequisite. To address this issue, recurrent neural network (RNN) [39], along with its two popular variants, namely long short-term memory (LSTM) [19] and gated recurrent unit (GRU) [20] have been proposed to dynamically capture long-range dependencies among the sequential data via a flexible non-linear mapping from the inputs to the outputs.

Attempts on time series modelling using RNNs have demonstrated the efficacy of RNNs in various time series prediction tasks, such as dynamic location prediction [40], [41] and user satisfaction prediction [42]. In aforementioned applications, a single RNN is leveraged to learn discriminative hidden states from the raw sequential inputs, and the last hidden state in a sequence is used to generate desired output. As real-life tasks get more complex, the one-step prediction result generated from the last hidden state of a single RNN no longer suit the demand. Consequently, the encoder-decoder network is first proposed in neural machine translation scenarios [20], [23], which further inspires relevant researches on multi-step ahead time series prediction [33], [43], [44].

With the repetitive patterns in different time series, the discovery of recurring trends in time series is worth more investigations [10], [15], [45], [46]. Unfortunately, these methods are either too rough to capture the subtle trend in sales time series or can only be applied to periodic trends within the time series. Besides, it is doubtful that whether these approaches can be effectively embedded into the network structure. On the basis of encoder-decoder RNN structure, several attention mechanisms are designed to align the output state with relevant encoded hidden states, thus selectively picking valuable contextual information to enhance the model's performance [4], [21], [22]. However, these attention mechanisms are incompatible with the requirement of trend alignment in sale time series because of the unknown state of influential factors and the timely interaction between semantically different influential factors (i.e., internal and external features). Recently, a RNN framework incorporating a regular recurrent layer and a recurrent layer with skipping schemes is developed in [9] to capture repetitive trends in the time series. However, the skipping step size in [9] needs to be observed from the data or obtained with a manual tuning process, which lacks enough flexibility to tackle the irregular patterns in sales time series.

VI. Conclusion

Sales prediction is a significant yet unsolved problem due to the subtle influential patterns among different factors and the irregular sales trends triggered by complex real-life situations. In this paper, we propose TADA, a novel model that performs trend alignment with dual-attention, multi-task recurrent neural networks to predict sales volume in real-life commercial scenario. With TADA, we first model the internal and external features within the influential factors in the sales time series in a multi-task fashion, thus maintaining their unique semantic meanings when timely modelling their mutual influences to the sales. Besides, we propose a dual-attention based decoder to simulate the sales contextual information in the future, and then align the generated representation of the upcoming trend with the most relevant one from the past. By this mean, TADA conquers existing challenges in the sales prediction task and outperforms the state-of-the-art baselines in two real datasets. In the future work of sales prediction, it will be appealing to further investigate cold-start predictions and the mutual influence between two products in the retail commerce.

Acknowledgment

This work is supported by ARC Discovery Early Career Researcher Award (Grant No. DE160100308, LP150100671 and DP160104075), ARC Discovery Project (Grant No. DP170103954) and New Staff Research Grant of The University of Queensland (Grant No. 613134).

References

[1] R. Carbonneau, K. Laframboise, and R. Vahidov, "Application of machine learning techniques for supply chain demand forecasting," *European Journal of Operational Research*, 2008.

[2] A. Kochak and S. Sharma, "Demand forecasting using neural network for supply chain management," *International journal of mechanical engineering and robotics research*, 2015.

[3] Y. Wu, J. M. Hernández-Lobato, and Z. Ghahramani, "Dynamic covariance models for multivariate financial time series," *ICML*, 2013.

[4] Y. Qin, D. Song, H. Cheng, W. Cheng, G. Jiang, and G. Cottrell, "A dual-stage attention-based recurrent neural network for time series prediction," *IJCAI*, 2017.

[5] Q. Wang, H. Yin, Z. Hu, D. Lian, H. Wang, and Z. Huang, "Neural memory streaming recommender networks with adversarial training," in *SIGKDD*, 2018.

[6] H. Yin, B. Cui, X. Zhou, W. Wang, Z. Huang, and S. Sadiq, "Joint modeling of user check-in behaviors for real-time point-of-interest recommendation," *Transactions on Information Systems*, 2016.

[7] K. L. Caballero Barajas and R. Akella, "Dynamically modeling patient's health state from electronic medical records: A time series approach," in *SIGKDD*, 2015, pp. 69–78.

[8] W. Chen, S. Wang, X. Zhang, L. Yao, L. Yue, B. Qian, and X. Li, "Eeg-based motion intention recognition via multi-task rnns." SDM, 2018.

[9] G. Lai, W.-C. Chang, Y. Yang, and H. Liu, "Modeling long-and short-term temporal patterns with deep neural networks," *SIGIR*, 2018.

[10] S. Papadimitriou, J. Sun, and C. Faloutsos, "Streaming pattern discovery in multiple time-series," in *VLDB*, 2005, pp. 697–708.

[11] N. Q. V. Hung, C. T. Duong, N. T. Tam, M. Weidlich, K. Aberer, H. Yin, and X. Zhou, "Argument discovery via crowdsourcing," *VLDB J.*, 2017.

[12] J. D. Hamilton, *Time series analysis*. Princeton university press Princeton, 1994, vol. 2.

[13] H.-F. Yu, N. Rao, and I. S. Dhillon, "Temporal regularized matrix factorization for high-dimensional time series prediction," in *NIPS*, 2016.

[14] G. E. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

[15] M. Shokouhi, "Detecting seasonal queries by time-series analysis," in *SIGIR*. ACM, 2011, pp. 1171–1172.

[16] A. Graves and N. Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," in *ICML*, 2014, pp. 1764–1772.

[17] A. Wilson and R. Adams, "Gaussian process kernels for pattern discovery and extrapolation," in *ICML*, 2013, pp. 1067–1075.

[18] T. Idé and S. Kato, "Travel-time prediction using gaussian process regression: A trajectory-based approach," in *Proceedings of the 2009 SIAM International Conference on Data Mining*. SDM, 2009.

[19] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[20] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," *SSST-8 ACL*, 2014.

[21] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *ICLR*, 2015.

[22] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," in *ICML*, 2015, pp. 2048–2057.

[23] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *NIPS*, 2014, pp. 3104–3112.

[24] H. Yin, W. Wang, H. Wang, L. Chen, and X. Zhou, "Spatial-aware hierarchical collaborative deep learning for poi recommendation," *TKDE*, 2017.

[25] H. Yin, B. Cui, Y. Sun, Z. Hu, and L. Chen, "Lcars: A spatial item recommender system," *Transactions on Information Systems*, 2014.

[26] H. Yin, X. Zhou, B. Cui, H. Wang, K. Zheng, and Q. V. H. Nguyen, "Adapting to user interest drift for poi recommendation," *TKDE*, 2016.

[27] H. Chen, H. Yin, W. Wang, H. Wang, Q. V. H. Nguyen, and X. Li, "Pme: Projected metric embedding on heterogeneous networks for link prediction," in *SIGKDD*, 2018.

[28] H. Yin, H. Chen, X. Sun, H. Wang, Y. Wang, and Q. V. H. Nguyen, "Sptf: a scalable probabilistic tensor factorization model for semantic-aware behavior prediction," in *ICDM*, 2017.

[29] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *NIPS*, 2017.

[30] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *ICLR*, 2015.

[31] A. P. Parikh, O. Täckström, D. Das, and J. Uszkoreit, "A decomposable attention model for natural language inference," *EMNLP*, 2016.

[32] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *SIGKDD*. ACM, 2016, pp. 785–794.

[33] W. Bao, J. Yue, and Y. Rao, "A deep learning framework for financial time series using stacked autoencoders and long-short term memory," *PlOS ONE*, vol. 12, no. 7, 2017.

[34] J. H. Friedman, "Greedy function approximation: a gradient boosting machine," *Annals of statistics*, pp. 1189–1232, 2001.

[35] G. Ristanoski, W. Liu, and J. Bailey, "Time series forecasting using distribution enhanced linear regression," in *PAKDD*, 2013.

[36] P. J. Rousseeuw and A. M. Leroy, *Robust regression and outlier detection*. John wiley & sons, 2005, vol. 589.

[37] W. Yan, H. Qiu, and Y. Xue, "Gaussian process for long-term time-series forecasting," in *IJCNN*. IEEE, 2009, pp. 3420–3427.

[38] J. Zhou and A. K. Tung, "Smiler: A semi-lazy time series prediction system for sensors," in *SIGMOD*. ACM, 2015, pp. 1871–1886.

[39] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, 2015.

[40] D. Yao, C. Zhang, J. Huang, and J. Bi, "Serm: A recurrent model for next location prediction in semantic trajectories," in *CIKM*, 2017.

[41] X. Zheng, J. Han, and A. Sun, "A survey of location prediction on twitter," *TKDE*, 2018.

[42] R. Mehrotra, A. H. Awadallah, M. Shokouhi, E. Yilmaz, I. Zitouni, A. El Kholy, and M. Khabsa, "Deep sequential models for task satisfaction prediction," in *CIKM*. ACM, 2017, pp. 737–746.

[43] A. Sordoni, Y. Bengio, H. Vahabi, C. Lioma, J. Grue Simonsen, and J.-Y. Nie, "A hierarchical recurrent encoder-decoder for generative context-aware query suggestion," in *CIKM*. ACM, 2015, pp. 553–562.

[44] S. Bengio, O. Vinyals, N. Jaitly, and N. Shazeer, "Scheduled sampling for sequence prediction with recurrent neural networks," in *NIPS*, 2015.

[45] B. Chiu, E. Keogh, and S. Lonardi, "Probabilistic discovery of time series motifs," in *SIGKDD*. ACM, 2003, pp. 493–498.

[46] T. T. Nguyen, C. T. Duong, M. Weidlich, H. Yin, and Q. V. H. Nguyen, "Retaining data from streams of social platforms with minimal regret," in *IJCAI*, 2017.