Future-Aware Trend Alignment for Sales Predictions

Yiwei Liu 1,2, Lin Feng 1,2 and Bo Jin 1,2,*

1 School of Computer Science and Technology, Dalian University of Technology, Dalian 116024, China; liuyiwei@dlut.edu.cn (Y.L.); fenglin@dlut.edu.cn (L.F.)
2 School of Innovation and Entrepreneurship, Dalian University of Technology, Dalian 116024, China
* Correspondence: jinbo@dlut.edu.cn; Tel.: +86-130-8410-0305

Received: 21 October 2020; Accepted: 23 November 2020; Published: 28 November 2020

Abstract: Accurately forecasting sales is a significant challenge faced by almost all companies. In particular, most products have short lifecycles without the accumulation of historical sales data. Existing methods either fail to capture the context-specific, irregular trends or to integrate as much information as is available in the face of a data scarcity problem. To address these challenges, we propose a new model, called F-TADA, i.e., future-aware TADA, which is derived from trend alignment with dual-attention multi-task recurrent neural networks (TADA). We utilize two real-world supply chain sales data sets to verify our algorithm’s performance and effectiveness on both long and short lifecycles. The experimental results show that the accuracy of the F-TADA is better than the original model. Our model’s performance could be further improved, however, by appropriately increasing the length of the windows in the decoding stage. Finally, we develop a sales data prediction and analysis decision-making system, which can offer intelligent sales guidance to enterprises.

Keywords: sales forecasting; deep learning; prediction platform

1. Introduction

Accurate sales forecasting is crucial for supply chain management. Overestimation or underestimation can affect inventory, cash flow, business reputation, and profit. Hence, it has attracted attention from both academic and industrial worlds. Essentially, sales prediction can be formulated as a time series forecasting problem, which is usually solved by the autoregressive model (AR) [1] or autoregressive moving average model (ARMA) [2,3]. The AR and ARMA models are suitable for stationary time series, but most time series data are non-stationary, so various linear and non-linear time series models [4], namely autoregressive integrated moving average (ARIMA) [5], seasonal-ARIMA, the seasonally decomposed autoregressive (STL-ARIMA) algorithm [6], the autoregressive conditional heteroscedasticity model (ARCH), and generalized autoregressive conditional heteroskedasticity (GARCH), have come into being. Moreover, autoregression methods that can model cointegration (autoregressive distributed lag (ARDL)) [7] or estimate covariance functions (stochastic autoregressive moving average (ARMA) processes) [8] have been proposed. The above regression algorithms are not domain-specific, and they have been applied in various fields, such as sales forecasting [9] and wind speed prediction [10]. Some works even make the model and code publicly accessible, e.g., Taylor and Letham [11] put forward an additive model and made the model code open-source, which will help analysts to solve a large number of time series prediction problems. Recently, the prediction of time series data (e.g., financial data, electronic health records, and traffic data) has also attracted wide-spread attention in machine learning fields that first need to transform time series into hand-crafted features. For example, Yu et al. [12] used the support vector regression (SVR) algorithm to predict the sales of newspapers and magazines. Based on SVR [13], Kazem et al. proposed a stock prediction method. Gumus et al. predicted crude oil prices by using the XGBOOST algorithm with high accuracy and efficiency [14].
However, both the aforementioned traditional autoregression-based methods and machine learning methods are ineffective for the trend alignment in sales prediction due to the irregular trends or patterns for different products. Hence, deep learning methods have also been utilized for sales forecasting due to the fact that they can better capture context-specific non-linear relationships or patterns among various influential factors (e.g., brand, season, discount, etc.). In general, the recurrent neural network (RNN) is suitable for the prediction of time series data. Khandelwal et al. utilized a combination of ARIMA and the artificial neural network (ANN) for time series prediction [15], while Babai et al., also adopted the autoregressive model to study data as a time series [16]. Tsai et al., leveraged the long short-term neural network (LSTM) algorithm to predict PM2.5 concentrations [17]. Bandara et al., forecasted sales demand with LSTM [18]. Cho et al., proposed a new neural network model called the RNN encoder-decoder model [19]. Due to the defects of the encoder and decoder, an attention mechanism was proposed, for instance, Chen et al. published a trend alignment with the dual-attention multi-task recurrent neural network (TADA) [20]. TADA adopts 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 increase sales prediction performance.

Although the trend adjustments of multiple attention mechanisms help to achieve better prediction results than those of other comparative experimental models, they can be further improved. Even though the definition of the sales forecasting task in the TADA model considers time series sales data, internal (e.g., brand, category, price, etc.) and external (e.g., weather, holiday, promotion, etc.) influential factors in the past, it does not explicitly consider future characteristics or external future prediction features [21–23], such as the weather forecast, promotional schedules, holidays, etc., which are often available for future sales prediction tasks at prediction time. Hence, we propose F-TADA, which adds some known future characteristics of trend alignment to dual-attention multi-task recurrent neural networks, thereby improving on the original model based on our new sales model definition. Note that our model incorporates external future prediction features using deep learning methods, which is different from previous linear regression-based methods [22]. Since it can both model past and future influential factors, it fully utilizes available data, thus solving the data scarcity problem to some degree. Meanwhile, deploying the algorithms into a real prediction system is also important. For instance, the authors in [24] developed a time series prediction system for cross-border e-commerce platforms based on the data from cross-border e-commerce enterprises. However, this system has obvious defects. On the one hand, the data only show the sales value without feature tags. On the other hand, the installation environment is complex, the interface operation is not user-friendly, and the algorithm selection is limited. Though some work [25] has improved the system, the system still has some disadvantages. First, it uses single template content. Second, it cannot be combined freely or provide multi-dimensional data analysis. Third, the system architecture is not separated from the front and rear, its response time is slow, and the interface is simple. Hence, we designed a user-friendly sales prediction system.

In summary, the contributions of this paper are as follows:

- We reproduce and improve the method in the TADA paper and propose a multi-attention machine and trend adjustment algorithms featuring the integration of future known features, which is called F-TADA.
- We analyze two typical real-world time series data. The experimental results show that the performance of the new method is better than that of the original algorithm. To apply the sales forecast algorithm to the intelligent decision-making process, we develop a sales data forecast and analysis decision-making system. This system has a grouping module and a sand table simulation module, which can provide better guidance for the enterprise sales decision-making process.

To derive the proposed modeling procedure and guidelines, the rest of the paper is organized as follows. Section 2 presents a brief introduction about two real-world data sets, including supermarket sales data and pesticide sales data. Section 3 describes the foundational model and shows how we
derived our new model. Section 4 displays the experimental settings and provides a results analysis. Section 5 demonstrates our developed sales prediction system. Section 6 summarizes the findings and draws conclusions on the effectiveness of our model. Further, we outline the shortcomings of the model and list some future possible directions.

2. Data Introduction

Typical sales data can be divided into long period and short period data. We use supermarket chain sales as long-period time-series forecasting data and pesticide sales as short-period time-series forecasting data.

2.1. Supermarket Sales Data

2.1.1. Description of Supermarket Sales Data

The supermarket sales data set comes from a contest by Kaggle, which consists of training data, store data, item data, transaction data, oil data, and holiday event data. The basic information of the data set is as follows:

1. Training data contain the date, store, and items sold.
2. Store data contain the store details, such as store location and store type.
3. Item data contain the characteristics of a commodity, such as perishability, and the type of commodity.
4. Transaction data contain the number of sales per store in the training data.
5. Oil data and holiday event data contain the prices of daily oil and holiday information.

2.1.2. Data Analysis

Figure 1 illustrates the total sales trends. It can be found that the sales increase slowly with the growth of time, with a surge around Christmas at the end of each year, which indicates that the holiday factor is a very important element for sales forecasting.

![Figure 1. Total sales trends over time.](image)

As shown in Figure 2, we drew a heatmap for the sum of 12 different months and seven days a week every year. The sales volume on weekends increased sharply compared with the weekdays sales, which is in line with the consumption habits of human beings, as people are used to buying goods on weekends. Looking at the sales of each month longitudinally, the sales volume in December is greater than that during the remaining 11 months because December sales are generally boosted by Christmas shopping sprees.
2.1.3. Data Processing

For the supermarket sales data, we used Python and Pandas to preprocess the data to obtain the time series prediction data. All the data were first grouped into 54 stores according to the stores, and then each commodity in the 54 stores was also grouped. We used one-hot encoding to encode the characters and numbers. At the same time, we used the embedding dimension reduction method to reduce the dimensions of some features. The final data are composed of multiple files, and each file represents the time series sales data of a certain commodity in a certain store.

2.2. Pesticide Sales Data

2.2.1. Description of Pesticide Sales Data

Data set 2 is the pesticide sales data provided by a pharmaceutical company. After cleaning and preprocessing the original data, the statistical data were obtained. The data were collected from the annual promotion activity table for 2017; the annual county-level consumption data and rice crops were combined to obtain the basic tables. We also used Python to crawl the data of the planting area on the internet. As the original data featured simple regularity, there were many missing values. Moreover, the data with time changes were found to be uncommon, so additional data on the weather and climate were used to help the training.

After data preprocessing and feature extraction using XGBoost [26], we obtained the final information data tables, including the date, downstream distributors, downstream distributor provinces, regions and classifications, regional brands, brands, volumes, artificial grassland, tea gardens, natural grassland, average surface temperature, average pressure, average temperature, average wind speed, sunshine time, and the number of activities in the month.

2.2.2. Data Analysis

The pesticide data are shown in Figure 3. Clearly, the seasonal trends of pesticides sales decreased significantly at the beginning and end of 2017. Seasonal factors affected not only crop growth, but also sales volume.
We sorted the importance of the selected features to see if the information crawled on the site was valid. The advantage of using the XGBoost [26] algorithm to predict the sorted data is that after successfully creating the lifting tree, we obtained the importance score of each attribute, allowing the data to be preprocessed to extract more important features. Then, we discarded less important features. The accuracy of the predictions with the data for Liaoning province was 0.56.

Figure 4 shows the feature importance ranking determined by the XGBoost algorithm, such as the average temperature, average ground temperature, precipitation, and average air pressure. Here, the temperature, precipitation, and number of activities have the greatest impact, while sand features and marsh characteristics are discarded.

2.2.3. Preprocessing

Here, the missing value of the weather class is filled by the average value for the last month and next month, and the missing value of the land area class is filled by the average area of the province. The missing activity information data is set to 0, but the data for missing provinces and cities are abandoned. We process the digital data features and category features in the same way as data set 1.
3. Research Methodology

3.1. Existing Deep Learning Methods

There is no time connection between inputs in traditional neural networks, but the recurrent neural network RNN has a natural advantage for data with a certain time connection. Because of the problem of gradient disappearance, the gated recurrent unit (GRU) is proposed by changing the RNN’s internal propagation structure, giving the model a long-term memory. Another common recurrent neural network that is more powerful and universal is the long short-term memory network (LSTM), which adds a forgetting gate and output gate on the basis of the updating gate. Cho et al. proposed a new neural network model called the RNN encoder–decoder model [19]. This model consists of two recurrent neural networks. Due to the defects of the encoder and decoder, an attention mechanism is proposed. The attention model generates a vector sequence in the encoding stage, and each input in the decoding stage generates a subset in the vector sequence, which focuses more on the part of the vector sequence that is most closely related to the current input.

3.2. Trend Alignment with Dual-Attention Multi-Task Recurrent Neural Networks

Trend alignment with dual-attention multi-task recurrent neural networks (TADA) was published in 2018 by Chen et al. under the ICDM sales forecast method [20]. To a certain extent, this model improves the prediction performance.

The goal of sales forecasting is to predict future sales through a variety of influencing factors and known sales values. The input is defined as the total set of features \( \{x_i\}_{i=1}^T = \{x_1, x_2, \ldots, x_T\} \), and the corresponding sales volume set is \( \{y_i\}_{i=1}^T = \{y_1, y_2, \ldots, y_T\} \). At time \( t \), \( x_i \in \mathbb{R}^n \), where \( n \) is the characteristic dimension, \( T \) is the whole time stage, and the output of the sales forecast is the next \( \Delta \) selling value after the \( T \) time stage, which can be expressed as the following formula: \( \{\hat{y}_i\}_{i=T+1}^{T+\Delta} = \{\hat{y}_{T+1}, \hat{y}_{T+2}, \ldots \hat{y}_{T+\Delta}\} \), where \( \Delta \) is determined by the target of the sales forecast. Assume that \( \Delta \ll T \) and \( \{x_i\}_{i=1}^{T+\Delta} \) are unknown factors in the forecast phase. Compared with the traditional autoregressive model, the performance of the sales forecasting model is different because the scalar value to be predicted has no characteristics in the future. Therefore, we model sales forecasts with the following characteristics:

\[
\{\hat{y}_i\}_{i=T+1}^{T+\Delta} = F(\{x_i\}_{i=1}^T, \{y_i\}_{i=1}^T) \quad (1)
\]

where \( \{x_i\}_{i=1}^T \) is the feature from the first timestamp to time \( T \); \( \{y_i\}_{i=1}^T \) is the historical sales information; \( \{\hat{y}_i\}_{i=T+1}^{T+\Delta} \) is the value to be predicted; and \( F(\cdot) \) is the nonlinear mapping method to be learned.

First, the basic model is coded using the LSTM model, and the activation function \( \sigma \) is the Logistic Sigmoid function; \( \omega, b \) are the weight values to be updated. Internal and external features are the two different semantic features used in time series sales forecasting. In other words, internal and external characteristics have a very different impact on sales forecasting. The formula \( \{x_i^{int}\}_{i=1}^T \) is used to represent the internal characteristics. External features are represented using \( \{x_i^{ext}\}_{i=1}^T \). Internal features include information about internal attributes that are directly related to the product, such as the location of the store and the category of goods. External characteristics refer to the external attribute information of the factors, such as weather conditions at that time, whether it is a special holiday, etc. This is why a single LSTM encoder structure may lose some contextual information by mapping all the original features into a unified space. Therefore, two parallel LSTMs are used to effectively capture the different influencing modes of features by modeling the internal and external features as two subtasks. Accordingly, the problem in Equation (1) is extended to

\[
\{\hat{y}_i\}_{i=T+1}^{T+\Delta} = F(\{x_i^{int}\}_{i=1}^T, \{x_i^{ext}\}_{i=1}^T, \{y_i\}_{i=1}^T) \quad (2)
\]

The structure diagram of the coding phase in the TADA model is shown in Figure 5.
We use \( [h_t^{int}]_{t=1}^T \) and \( [h_t^{ext}]_{t=1}^T \) to represent the final representation learned from the \( \{x_t^{int}\}_{t=1}^T \) internal feature input and \( \{x_t^{ext}\}_{t=1}^T \) external feature input, respectively. When the hidden layer outputs of the internal features and external features are obtained, we put the hidden state into another LSTM layer to obtain unified learning, which we call the context vector, and use \( [h_t^{con}]_{t=1}^T \) to represent the learning coalition in the \( T \) stage. In addition, to strengthen the power of expression, the extra encoder is joined with \( [y_t]_{t=1}^T \), the output of the hidden layers with characteristics of the internal LSTM and the output of hidden layers with characteristics of the external LSTM (the external LSTM characteristics of the output of the hidden layers). This generates joint input as \( \{x_t^{syn}\}_{t=1}^T \), which is shown in Equation (3):

\[
x_t^{syn} = W_{syn}[h_t^{int}; h_t^{ext}, y_t] + b_{syn},
\]

where \( [h_t^{int}, h_t^{ext}, y_t] \) refers to the splicing of \( h_t^{int}, h_t^{ext} \), and \( y_t \), and \( W_{syn} \) and \( b_{syn} \) are weights and offsets that need to be learned. Therefore, the multi-task encoder can be expressed as Equation (4):

\[
h_t^{int} = LSTM^{int}(x_t^{int}, h_{t-1}^{int}), h_t^{ext} = LSTM^{ext}(x_t^{ext}, h_{t-1}^{ext}), h_t^{con} = LSTM^{con}(x_t^{syn}, h_{t-1}^{con}).
\]

The internal-feature coded LSTM, the external-feature coded LSTM, and the combined LSTM encoders are different. Instead of sharing weights and offsets, each LSTM learns its own weights and offsets. After encoding all the historical information of the sales time series, we finally obtain the context vector \( [h_t^{con}]_{t=1}^T \), which carries all the contextual information at time \( t \) for the sales time series. Then, we obtain the internal characteristics of the encoder output of the hidden layers \( \{h_t^{int}\}_{t=1}^T \), as well as the external one, \( \{h_t^{ext}\}_{t=1}^T \). To predict the next sales time sequence \( \{y_t\}_{t=T+1}^{T+\Delta} \), we use the decoder to simulate the next \( \Delta \) time context vector, in which \( T < t \leq T + \Delta \):

\[
a_t^{con} = LSTM^{dec}(x_t^{dec}, d_{t-1}^{con})
\]

where \( d_t^{con} \in \{d_t^{con}\}_{t=T+1}^{T+\Delta} \) is the input of the decoder, and \( LSTM^{dec}(\cdot) \) is the decoder LSTM. \( x_t^{dec} \) is the attention weight, that is, the input of the decoder, and \( a^{con}_{t-1} \) represents the output of the previous decoder’s hidden layers.

As can be seen from Equation (1) of the time series prediction, \( \{x_t^{int}\}_{t=T+1}^{T+\Delta} \) and \( \{x_t^{ext}\}_{t=T+1}^{T+\Delta} \) are both internal features and external features including unknowable features after time \( T \). Therefore, to obtain the input of the decoder, we use the attention mechanism. As shown in Equation (6), the relevant vectors are dynamically selected and merged from \( \{h_t^{int}\}_{t=1}^T \), \( \{h_t^{ext}\}_{t=1}^T \):

\[
x_t^{dec} = W_{dec} \left[ \sum_{t=1}^T \alpha_t^{int} h_t^{int}, \sum_{t=1}^T \alpha_t^{ext} h_t^{ext} \right] + b_{dec}
\]

Figure 5. Structure of multi-task LSTM encoder [16].
where $\alpha_{tt'}^{int}$ and $\alpha_{tt'}^{ext}$ represent the hidden layer attention weight of the internal feature encoder and the external feature encoder at time $t' - \text{th}$. $\sum_{t' = 1}^{T} \alpha_{tt'}^{int} = \sum_{t' = 1}^{T} \alpha_{tt'}^{ext} = 1$. $x_t^{dec}$ refers to the input value at an unknown time, obtained according to the proportion of the attention in the output of the inner hidden layer and the outer hidden layer.

This effect was quantified by the correlation between $d_{tt-1}^{con}$ and both $h_t^{int}$ and $h_t^{ext}$.

$$e_{tt'}^{int} = v_{ext}^T \tanh(M_{int} d_{tt-1}^{con} + H_{int} h_{tt'}^{int}),$$
$$e_{tt'}^{ext} = v_{ext}^T \tanh(M_{int} d_{tt-1}^{con} + H_{ext} h_{tt'}^{ext}).$$  \hspace{1cm} (7)

To obtain the input of the decoder at time $t$, $e_{tt'}^{int}$ and $e_{tt'}^{ext}$ correspond to the correlation score between the output of the hidden layer of the internal characteristic LSTM and the output of the hidden layer of the external characteristic LSTM and $d_{tt-1}^{con}$, respectively, at time $t'$. Here, $v_{int}$, $v_{ext}$, $M_{int}$, $H_{int}$, and $H_{ext}$ are the parameters for the model to learn. Intuitively, the degree of direct correlation between two vectors is shown by projecting those vectors into a common space. The SoftMax function is used to apply the weights of the two attention mechanisms as

$$a_{tt'}^{int} = \frac{\exp(e_{tt'}^{int})}{\sum_{s=1}^{T} \exp(e_{ts}^{int})}, a_{tt'}^{ext} = \frac{\exp(e_{tt'}^{ext})}{\sum_{s=1}^{T} \exp(e_{ts}^{ext})}. \hspace{1cm} (8)$$

After applying the SoftMax function, Equation (6) guarantees that $\sum_{t' = 1}^{T} a_{tt'}^{int} = \sum_{t' = 1}^{T} a_{tt'}^{ext} = 1$. In an ideal state, at time $t$, $\{h_{i_1}^{con}\}_{i=1}^{T}$ and $\{d_{i_1}^{con}\}_{i=T+1}^{T+\Delta}$ will both carry the information of time $t$ and the previous context. With an increase in the length of the prediction stage, the performance of the encoder and decoder network will decrease significantly. To alleviate this problem, a traditional attention mechanism is designed to make the current output consistent with the target input by comparing the current hidden state with the hidden state generated by the previous time step. However, these methods are not applicable to predict the approximate trend of the next $\Delta$ time, so a new trend-adjusted attention mechanism is proposed. It is believed that the subsequent trend changes will be similar to the prior trend, so it is feasible to splice the most similar trends. As shown in Equation (9), $p_i$ represents the joint vector hiding layer:

$$p_i = [h_{i-1}^{con}, h_{i+1}^{con}, \cdots; h_{i+\Delta-1}^{con}], 1 \leq i \leq T - \Delta + 1. \hspace{1cm} (9)$$

It can be seen intuitively that $\Delta$ is the time window, and the original $T$ time is divided into segments of $\Delta$ length. At the same time, the trend $\bar{p}$ to be predicted can be expressed as

$$\bar{p} = [d_{T+1}^{con}, d_{T+2}^{con}, \cdots; d_{T+\Delta}^{con}]$$

where $p_i$ is like a sliding window: It will find the most similar window to $\bar{p}$ from each $p_i e\{p_i|_{i=1}^{T-\Delta+1}\}$ sliding window. The following formula can be used to measure the similarity between them:

$$e_{i}^{\text{trd}} = p_i^T \bar{p}. \hspace{1cm} (10)$$

Then, the formula will find the closest sliding window:

$$i' = \arg\max\left\{e_{i}^{\text{trd}}, e_{i+1}^{\text{trd}}, \cdots, e_{T+\Delta-1}^{\text{trd}}\right\}. \hspace{1cm} (12)$$

We assume that the most similar window is $p_{i'} = [h_{i'}^{con}, h_{i'+1}^{con}, \cdots; h_{i'+\Delta-1}^{con}]$ and combine the windows according to Equation (13):

$$\bar{d}_{i}^{con} = W_{di}[d_j^{con}, h_k^{con}] + b_{di}, T + 1 \leq j \leq T, i' \leq k \leq i' + \Delta - 1. \hspace{1cm} (13)$$
At this point, we have \( \{\tilde{d}_t^{con}\}_{t=T+1} \) and use it to obtain the final predicted value, as shown in Equation (14):

\[
\hat{y}_t = \tilde{v}_y \theta_1^y T_t + b_y
\]  

where \( \hat{y}_t \in [\hat{y}_1]_{t=T+1} \) represents the prediction at time \( t \), while \( \tilde{v}_y \) and \( b_y \) are the parameters for us to learn.

In terms of model prediction, we use the mean square error as the loss function and \( L_2 \) regularization to prevent the model from overfitting during training:

\[
L_F = \frac{1}{N} \left( \sum_{n=1}^{N} \sum_{t=T+1}^{T+\Delta} (\hat{y}_{nt} - y_{nt})^2 \right) + \lambda \sum_{t} \theta_1^2.
\]

### 3.3. A Deep Learning Model that Incorporates Future Known Features

This section focuses on improving the trend adjustment algorithm of the multi-attention mechanism. For the TADA model, the definition of the sales forecasting task shows that the predicted scalar values are featureless in the future.

The algorithm assumes that \( [x_t]_{t=T+1} \) is an unknown factor in the prediction stage, which can be improved in essence. For example, for pesticide data, the probability of temperature, precipitation, and other factors can be obtained through weather forecasts. These characteristics are likely accurate, and if we discard this part of the information, the accuracy of the prediction may be directly reduced.

For data set 1, the city of the store is an internal characteristic, which will not change and has no effect on the prediction. However, the dates, holidays, and other factors are known in the prediction stage with a probability of 1. Thus, it is important for us to predict sales, but the original model does not consider this factor. For data set 2, if the predicted average temperature and other information are known, the prediction accuracy at that time will be improved. At the same time, we add information from the company’s internal management system to obtain the next month’s activity planning and other information. Therefore, it is very helpful to integrate this portion of the information into the task of sales forecasting.

Based on the above concepts, the original TADA model is improved. First, we redefine the sales forecast problem. The input of the improved sales forecast model is defined as all feature sets \( [x_t]_{t=1}^T = [x_1, x_2, \cdots, x_T] \) and the corresponding sales sets \( [y_t]_{t=1}^T = [y_1, y_2, \cdots, y_T] \). In addition, there is some known information between \( T + 1 \) and \( T + \Delta \), which is represented by \( [z_t]_{t=T+1}^{T+\Delta} \). At time \( t \), \( x_t \in \mathbb{R}^n, z_t \in \mathbb{R}^m \), where \( n \) and \( m \) are the characteristic dimensions. The output of the sales forecast is the \( \Delta \) selling value after \( T \), so the definition of the sales forecast can be improved with Equation (16):

\[
[y_t]_{t=T+1}^{T+\Delta} = F([x_t]_{t=1}^T, [y_t]_{t=1}^T, [z_t]_{t=T+1}^{T+\Delta}).
\]

where \( [x_t]_{t=1}^T \) is a feature from 1 to \( T \), and the feature dimension is \( n \). \( [y_t]_{t=1}^T \) is the historical sales information. \( [z_t]_{t=T+1}^{T+\Delta} \) is part of the known probability features between \( T + 1 \) and \( T + \Delta \), and the feature dimension is \( m \). We will next predict the value \( \hat{y}_t \).

Due to the addition of information, there will be some changes to the TADA model to adapt the model to the definition of the new sales forecasting model. In the previous section, the basic models were an encoder and decoder model. Next, some adjustments are made to the encoder and decoder structure. First of all, the input features of the encoder stage are divided into internal feature input and external feature input. After two rounds of the LSTM algorithm, the output of the hidden layer is obtained, combined with the historical sales volume as a context vector that constitutes the coding phase of the model. Since \( [z_t]_{t=T+1}^{T+\Delta} \) cannot be used in the encoding stage of the model, we retain the encoder stage of the model. In the decoder stage, the original model uses the multi-attention
mechanism to obtain the input, so the information for this part can be added to the decoder part for training. The formula of the decoder part is as follows:

$$d_i^{con} = LSTM^{dec}(d_t^{dec}, d_{i-1}^{con})$$

(17)

where $x_t^{dec}$ is the input of the LSTM model in the decoder stage. Since part of the information is unknown, and part of the information is known, during the prediction period, a new method combining the features of the attention mechanism with some known features is used. Here, the characteristic output of the attention mechanism is as follows:

$$a_t^{dec} = W_{dec} \left[ \sum_{t'=1}^{T} \alpha_{t,t'}^{int} h_{t'}^{int}, \sum_{t'=1}^{T} \alpha_{t,t'}^{ext} h_{t'}^{ext} \right] + b_{dec}$$

(18)

Please refer to the input in the decoding stage of the TADA model mentioned above for a detailed explanation of the formula. The improved TADA model adds $[z_t]_{i=T+1}$ to the decoding stage, which is spliced with the result of the attention mechanism to obtain the input $x_t^{dec}$ in the decoder stage:

$$x_t^{dec} = [a_t^{dec}, z_t]$$

(19)

A schematic diagram is shown in Figure 6. The currently known features have been added in the decoder stage, where the category features are the embedded dimension reduction, and the weight vector of the embedded dimension reduction is consistent with (the weight of the embedded dimension reduction) the input in the coding stage. There are two innovations in the improved algorithm: One is the redefined sales forecast and the other is the improvement of the decoder.

Figure 6. Improved decoder stage.

4. Experiments and Results

1. Data set partitioning: In data set 1, we use a total of 365 data from 2016 to 2017 and divide the data into 15:2:2. In data set 2, we use the annual data of each province and city in 2017 and divide them at a ratio of 8:1:1.
2. The evaluation index: We use the mean absolute error, MAE, and the symmetrical mean absolute percentage error, MAPE (or SMAPE).
3. Gradient descent optimization method: We use mini batch gradient descent and Adam descent.

The formulas of MAE and MAPE are defined as follows (20–21):

$$MAE = \frac{1}{N \times \Delta} \sum_{n=1}^{N} \sum_{t=T+1}^{T+\Delta} |y_t - \hat{y}_t|$$

(20)
\[ MAPE = \frac{100\%}{N \times \Delta} \sum_{n=1}^{N} \sum_{t=T+1}^{T+\Delta} \left\{ \begin{array}{ll} 0, & \text{if } y_t = \hat{y}_t = 0 \\ \frac{|y_t - \hat{y}_t|}{|y_t| + |\hat{y}_t|}, & \text{otherwise} \end{array} \right. \]  

(21)

### 4.1. Experimental Results and Analysis

For data set 1, \( \Delta \) is set to 2, 4, and 8 for the experiment. The algorithm needs to divide the data features into internal features and external features to obtain more accurate experimental results. The internal characteristics of data set 1 include the store city, store state, store type, store group, commodity family, and commodity category. The external features include time, total sales volume, whether it is a local holiday, whether it is a national holiday, whether it is a weekend, whether the commodity is perishable, and the price of crude oil. In the decoder stage, the features of the F-TADA algorithm add the date, holiday information, and weekend information. The one-hot coding vector with 365-dimensional date information is embedded into 5 dimensions. The weight matrix of the dimensional reduction is trained by embedding the matrix in the coding stage. This method can solve the problem of insufficient training for date encoding in the decoder stage using the date embedding matrix. Holiday information is a Boolean variable. Therefore, the \( z_t \) input dimension of the F-TADA algorithm is 8 dimensions in the decoding phase, and we also adjust the super parameters during the experiment. The super parameters include the number of hidden layer features and the coefficients of regular terms. Finally, we select 128 as the feature number of the hidden layer and 0.001 as the coefficient of the regular term, which are the best parameters for the TADA algorithm and F-TADA algorithm. To reproduce the experimental results of the original TADA paper, the data set partition, data preprocessing, and best parameters obtained are consistent with those in the earlier paper. However, due to the contingency of deep learning training, there is a slight deviation between the results of the original experiment and the results in this paper. In addition to the original TADA model, we add the encoder decoder model and attention mechanism model for the comparative experiment. The experimental results are shown in Table 1.

| Result | \( \Delta = 2 \) | \( \Delta = 4 \) | \( \Delta = 8 \) |
|--------|-----------------|-----------------|-----------------|
| Encoder and decoder model + attention mechanism | 6.499 40.345 | 6.993 41.317 | 7.160 43.814 |
| TADA | 6.251 39.569 | 6.838 40.762 | 6.853 42.619 |
| F-TADA | 6.273 39.468 | 6.700 40.517 | 6.685 42.155 |

As can be seen from the results in Table 2, when \( \Delta = 2 \), the accuracy of the F-TADA algorithm is similar to that of the original algorithm. The MAE value of the TADA algorithm is slightly better and the MAPE value of F-TADA, so the effect of the two algorithms is almost the same. However, with an increase of \( \Delta \), the advantages of the F-TADA algorithm gradually emerge, and the results are superior.

| Result | \( \Delta = 2 \) | \( \Delta = 4 \) | \( \Delta = 8 \) |
|--------|-----------------|-----------------|-----------------|
| Effect enhancement | -0.021 0.101 | 0.138 0.245 | 0.168 0.464 |

Therefore, based on the results of data set 1, the improved algorithm is effective in sales series forecasting. With an increase in \( \Delta \), the algorithm effectively improves the prediction accuracy of data set 1.
For the prediction results of data set 1, we select $\Delta = 4$ data to visualize and draw a graph for the observations, as shown in Figure 7. After selecting the sales values over 65 days for visualization, the trend learned by the model was observed to be stable compared to the real value.

![1047690 Commodity prediction](image)

**Figure 7.** Results of 1047690 commodity data.

Data set 2 uses the encoder–decoder model with the attention mechanism algorithm, TADA algorithm, and TADA algorithm fusing future known features for training (F-TADA). The TADA algorithm and F-TADA algorithm need to divide the features into internal features and external features in the encoder stage. The internal characteristics of data set 2 are as follows: downstream dealer province, downstream dealer city, region, brand, brand classification, region, and various land area data crawled. The external characteristics of data set 2 include time, date, average surface temperature, average air pressure, average temperature, average wind speed, precipitation, sunshine hours, activity duration in the month, number of activities in the month, and number of activities in the month.

For the sales data of pesticides in data set 2, seasonal factors are particularly important. The sales volume of pesticides is zero in most months but becomes prominent during other months. For example, pesticides and herbicides are generally sold in the summer in northern cities, while sales in winter are 0. Because most of the sales values for the sales data are 0, the total annual sales of all kinds of drugs are 0 in some cities. Therefore, we screened the sales volume of 0 over 12 months in the experimental stage and used a flag to express this situation. It can be seen from Figure 8 that there were 4849 drug sales combinations in cities. Among them, only 223 data were sold throughout the year, while the data without sales over three months or within three months account for about one fifth of the total data. Moreover, the data with a sales volume of 0 over 7, 8, 9, and 10 months account for a large proportion. The predicted results for the stationary time series are better than those for the data with more than 0 values. Therefore, we selected three data sets with flag $\leq 3$, flag $\leq 5$, and flag $\leq 10$ as three data sets for the experiment and 20 urban drug combinations as test sets in each data set.

![Figure 8. Information of the flag.](image)
Data set 2 is characterized by short-period time series predictions. Here, only one year’s worth of sales records is counted by month. Since most of the sales values in the sales records are 0, the problem of sales forecasting for data set 2 is a major challenge. When a certain drug in a city is used for an independent prediction, there are only 12 data without high quality, so it is impossible to carry out an autoregressive prediction independently. For this kind of short period data, the time is not sufficient, but there are many cities and drug combinations. In the data set 2 prediction stage, the activity plan for the next month was uploaded in the company management system, and the weather forecast information can be obtained, as well as whether this month is a rice-growth period. Therefore, the best solution is to use the F-TADA algorithm for this type of sales data. By embedding words in the one-hot vectors of cities, medicines, and types, we can better learn the internal connections between data and improve the accuracy of the predictions.

In data set 2, the training process and decoder stage \( \{z_t\}_{t=T+1}^{T+\Delta t} \) include the following information: date, the number of activities in the month, medication rules, precipitation, average surface temperature, average temperature, average air pressure, sunshine hours, evaporation, and average wind speed. The 12-dimensional one-hot date vector is embedded and encoded into a 2-dimensional vector. The medication pattern is a Boolean type, while the rest are a digital type. Therefore, the \( z_j \) input dimension of the F-TADA algorithm in decoding stage has 11 dimensions. The experiment was conducted by controlling \( \Delta = 3 \), as well as adjusting the super parameters during the experiment, including the number of hidden layer features and the coefficients of regular terms. Like with data set 1, we selected 128 as the feature number of the hidden layer and 0.001 as the coefficient of the regular term, which is the best parameter for the TADA algorithm and F-TADA algorithm. The final experimental results are shown in Table 3.

| Result                          | \( \Delta = 3, \text{Flag} \leq 3 \) | \( \Delta = 3, \text{Flag} \leq 5 \) | \( \Delta = 3, \text{Flag} \leq 10 \) |
|--------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|
|                                | MAE       | MAPE      | MAE       | MAPE      | MAE       | MAPE      |
| Encoder and decoder model + attention mechanism | 2.158  | 143.785  | 1.861  | 166.385  | 0.804  | 190.336  |
| TADA                           | 1.683  | 130.001  | 1.244  | 159.180  | 0.298  | 182.983  |
| F-TADA                         | 1.676  | 128.575  | 1.217  | 157.082  | 0.176  | 181.639  |

As can be seen from the experimental results of data set 2, due to the lower quality of this data set, such as its short period, multiple zero values, etc., the prediction results are worse than those of data set one. We can also see that the F-TADA algorithm is improved compared to TADA. However, with an increase of the 0 value, the prediction accuracy decreases. Based on the effects of the F-TADA algorithm, the single attention mechanism, and the encoder and decoder model, the multi-attention mechanism and trend adjustment algorithm combined with future known features is superior.

The MAE value in the experimental results decreases with an increase of the 0 value. The reason for this phenomenon is that with an increase of the 0 value, the predicted value is close to 0, leading to a decrease in the MAE value, but the MAPE value can be measured across data sets. Therefore, with an increase in the 0 value, the MAPE value becomes increasingly worse. In this case, the MAE is not measured correctly.

Meanwhile, a change was observed in the three indicators with respect to the epochs considered during the training and testing processes after we obtained the optimal hyperparameters according to the validation data set (see Figure 9).
We combine the known features of the prediction stage, improve the decoder stage, and verify both decision-making systems are developed. We combine some behaviors in the business field with artificial intelligence, a sales forecasting and analysis system is developed. Without the assistance of advanced intelligent algorithms, it is inevitable that unnecessary losses will occur. To apply time series algorithms to these real-world scenarios and combine some behaviors in the business field with artificial intelligence, a sales forecasting and analysis decision-making system is developed.

Figure 9. The change of the three indicators with respect to epochs during the training and testing process.

For the prediction results of data set 2, we select a data of flag ≤ 3 to visualize and draw a graph for observation, as shown in Figure 10. It is found that the model can learn the sales volume over the next three months and that the trend is correct.

Figure 10. Results of 5116 data.

4.2. Summary

This section introduces the existing deep learning methods and F-TADA, which adds some future known characteristics of trend alignment with dual-attention multi-task recurrent neural networks. We combine the known features of the prediction stage, improve the decoder stage, and verify both data set 1 and data set 2. The experimental results show that the MAE and MAPE values of the multi-attention mechanism and trend adjustment algorithm are lower than those of the original model based on the encoder-decoder model. According to the results of data set 1, with an increase in the decoding stage length, the improved algorithm offers a great improvement over the original algorithm.

5. Application Technology

5.1. Demand Analysis

In the production and sales process of many small and medium-sized retail enterprises, especially those lacking experience, without the assistance of advanced intelligent algorithms, it is inevitable that unnecessary losses will occur. To apply time series algorithms to these real-world scenarios and combine some behaviors in the business field with artificial intelligence, a sales forecasting and analysis decision-making system is developed.
For most small and medium-sized enterprises, the cost of hiring a professional data analyst or algorithm engineer team is too high, so the research and development of sales forecasting and intelligent decision-making platforms is very necessary. This sales forecasting platform can play a guiding role in a company’s business planning, help develop reasonable strategies for inventory planning, reduce a company’s unsalable risks, and maximize the sales benefits of popular products. Companies can obtain the prediction results and decision-making plans by providing data to the platform. This decision-making scheme can be used as the final solution for inventory management.

Python is used for common machine learning and deep learning algorithms to provide them with data interfaces. Finally, a website-based system was presented to the enterprise. Because this system is an online application, users do not need to install the software locally. Instead, they can simply use a computer or mobile phone to access the website.

The system’s functional requirements include seven modules: the user login and logout module, the data import module, the data grouping module, the data visualization module, the data prediction module, the correlation analysis module, and the sand table simulation module.

5.2. Technology Module

1. System architecture design: We adopt system architecture that separates the frontend and backend.
2. Database design: MongoDB is used as the basic database.
3. Frontend design: The frontend uses a page structure and is written in AngularJS.
4. Backend design: We use the Flash framework with a Celery distributed task queue.

5.3. Function Demonstration

The whole system has a home page that provides a general introduction to the overall functions, as shown in Figure A1 (See Appendix A).

After logging in, users of the system enter the home page, as shown in Figure A2 (See Appendix A). In the middle part of the page, the historical forecast results and data forecast progress are presented.

Users can select and edit templates, and the Excel data template generation page is shown in Figure A3 (See Appendix A). The data import, data edit, and data group modules are shown in Figure A4 (See Appendix A). The data visualization module is shown in Figure A5 (See Appendix A). The data prediction module is shown in Figure A6 (See Appendix A).

6. Conclusions

In the sales prediction task, we often meet three tough challenges: (1) trend or pattern irregularity in sales time series data, (2) complex context-specific relationship among influential factors, including real sales number data, internal factors (e.g., brand, category, price, etc.), and external factors (e.g., weather, holiday, promotion, etc.), and (3) data scarcity. Thus, the state-of-the-art method, i.e., TADA, tries to solve the previous two challenges. To solve all the three challenges, our work differentiates itself by integrating all the available information (both the past and future) based on TADA. More specifically, we proposed a future-aware model called F-TADA. In the decoder stage of the model, the new features are spliced with the obtained decoder input, and the attention mechanism is obtained as the final input of the decoding stage. After trend matching of the TADA model, we obtained the final prediction results.

The experimental results show that the deep learning algorithm integrating known features in the prediction stage improves the accuracy of the original model and offers a better prediction of sales data by using more known information. Moreover, based on the experimental results for supermarket chains, as the length of the decoding stage increases, the F-TADA algorithm increasingly improves compared to the original algorithm. Based on the results of pesticide sales, the F-TADA algorithm can solve the possibility of similar short-term sales data predictions.
Meanwhile, existing works mainly pay attention to the algorithms without the illustration of a real sales prediction management system. In our work, to help the enterprise forecast sales data, we developed a sales data prediction and analysis decision-making system. This system features asynchronous architecture with a separation of the frontend and backend, along with a fast response speed and concurrent prediction of data using multiple machines and processes. This system is divided into four modules: the visualization module, the feature analysis module, the prediction module, and the sand table exercise module. The sand table simulation module is designed to save the model during model training by processing the original data, such as promoting the product, increasing the number of product sales activities, etc. This is an intelligent decision-making element of the system that can obtain the impact of the final prediction results by calling the model. Development of the sand table exercise module is another innovation of this paper and allows the sales forecast to act as a guide for sales in a real sense. In sum, we develop a flexible and user-friendly sales prediction management system. We demonstrate the system in the paper such that it can offer practical insights for both academic and industrial world.

Finally, for the algorithms, some concerns remain unaddressed. For instance, we did not consider seasonal effects in the model. Moreover, we only tested sales data prediction tasks with clear future information. More relevant data sets should be tested in the future. Furthermore, to handle data with missing labels, predictive contrastive coding could be utilized in the sales prediction task.

**Author Contributions:** Conceptualization, B.J.; methodology, Y.L.; software, Y.L.; validation, L.F.; data curation, Y.L.; writing—original draft preparation, Y.L.; writing—review and editing, B.J. All authors have read and agreed to the published version of the manuscript.

**Funding:** Supported by the National Key R&D Program of China (2018YFC0116800), National Natural Science Foundation of China (No. 61772110), CERNET Innovation Project (NGII20170711).

**Conflicts of Interest:** The authors declare no conflict of interest.

**Appendix A**

![Figure A1. Home page of the sales data prediction and analysis decision system.](image-url)
**Figure A2.** Main page of the sales data prediction and analysis decision system. If we click the “bell“ sign, it will show the status (success or failure) of the sales prediction for any sales prediction task.

**Figure A3.** Data template generation page in Excel. To adapt to different preprocessing methods (e.g., one-hot encoding and normalization), we first determined the features template.

**Figure A4.** Loading the real sales prediction data set (csv or Excel format). View of the data import, data edit, and data group modules.
Figure A4. Loading the real sales prediction data set (csv or Excel format). View of the data import, data edit, and data group modules.

Figure A5. View of the Data Visualization Module. Here, we can show the data distribution and characteristics of the data by specifying the available time periods.

Figure A6. View of the Data Prediction Module. On the left, the “Predict Runner” shows allows us to choose some hyperparameters to set up the prediction model, such as the periods of the forecast, the prediction model, and the validation method. On the right, the software can visualize the prediction results.

References
1. Society, R. On a Method of Investigating Periodicities in Disturbed Series, with Special Reference to Wolfer’s Sunspot Numbers Author. *Stat. Pap. George Udny Yule* 1922, 226, 267–298. [CrossRef]
2. Friedlander, B. A recursive maximum likelihood algorithm for ARMA spectral estimation. *IEEE Trans. Inf. Theory* 1982, 28, 639–646. [CrossRef]
3. Cadzow, J. High performance spectral estimation-A new ARMA method. *IEEE Trans. Acoust. Speech, Signal Process.* 1980, 28, 524–529. [CrossRef]
4. Shah, I.; Iftikhar, H.; Ali, S. Modeling and forecasting medium-term electricity consumption using component estimation technique. *Forecasting* 2020, 2, 163–179. [CrossRef]
5. Box, G.E.P.; Pierce, D.A. Distribution of residual autocorrelations in autoregressive-integrated moving average time series models. *J. Am. Stat. Assoc.* 1970, 65, 1509–1526. [CrossRef]
6. Buhl, J.; Liedtke, C.; Schuster, S.; Bienen, K. Predicting the material footprint in Germany between 2015 and 2020 via seasonally decomposed autoregressive and exponential smoothing algorithms. *Resources* 2020, 9, 125. [CrossRef]
7. Busu, M. Analyzing the impact of the renewable energy sources on economic growth at the EU level using an ARDL model. *Mathematics* 2020, 8, 1367. [CrossRef]
8. Schubert, T.; Korte, J.; Brockmann, J.M.; Schuh, W.-D. A generic approach to covariance function estimation using ARMA-models. *Mathematics* 2020, 8, 591. [CrossRef]
9. Kechyn, G.; Yu, L.; Zang, Y.; Kechyn, S. Sales Forecasting Using WaveNet within the Framework of the Kaggle Competition. *arXiv* 2018, arXiv:1803.04037. Available online: https://arxiv.org/abs/1803.04037 (accessed on 24 November 2020).
10. Erdem, E.; Shi, J. ARMA based approaches for forecasting the tuple of wind speed and direction. *Appl. Energy* 2011, 88, 1405–1414. [CrossRef]
11. Taylor, S.J.; Letham, B. Business time series forecasting at scale. *PeerJ Prepr.* 2017, 35, 48–90.
12. Yu, X.; Qi, Z.; Zhao, Y. Support vector regression for newspaper/magazine sales forecasting. *Procedia Comput. Sci.* 2013, 17, 1055–1062. [CrossRef]
13. Kazem, A.; Sharifi, E.; Hussain, F.K.; Saberi, M.; Hussain, O.K. Support vector regression with chaos-based firefly algorithm for stock market price forecasting. *Appl. Soft Comput.* 2013, 13, 947–958. [CrossRef]
14. Gumus, M.; Kiran, M.S. Crude oil price forecasting using XGBoost. In Proceedings of the 2017 International Conference on Computer Science and Engineering (UBMK), Antalya, Turkey, 5–8 October 2017; pp. 1100–1103. [CrossRef]
15. Khandelwal, I.; Adhikari, R.; Verma, G. Time series forecasting using hybrid ARIMA and ANN models based on DWT decomposition. *Procedia Comput. Sci.* 2015, 48, 173–179. [CrossRef]
16. Babai, M.Z.; Ali, M.M.; Boylan, J.; Syntetos, A.A. Forecasting and inventory performance in a two-stage supply chain with ARIMA (0,1,1) demand: Theory and empirical analysis. *Int. J. Prod. Econ.* 2013, 143, 463–471. [CrossRef]
17. Tsai, Y.-T.; Zeng, Y.-R.; Chang, Y.-S. Air pollution forecasting using RNN with LSTM. In Proceedings of the 2018 IEEE 16th International Conference on Dependable, Autonomic and Secure Computing, 16th International Conference on Pervasive Intelligence and Computing, 4th International Conference on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech), Athens, Greece, 12–15 August 2018; pp. 1074–1079. [CrossRef]
18. Bandara, K.; Shi, P.; Bergmeir, C.; Hewamalage, H.; Tran, Q.; Seaman, B. Sales demand forecast in E-commerce using a long short-term memory neural network methodology. In Proceedings of the Public-Key Cryptography-PKC 2018, Rio de Janeiro, Brazil, 25–29 March 2019.
19. Cho, K.; Van Merrienboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 14–21 October 2014; pp. 1724–1734. [CrossRef]
20. Chen, T.; Yin, H.; Chen, H.; Wu, L.; Wang, H.; Zhou, X.; Li, X. TADA: Trend alignment with dual-attention multi-task recurrent neural networks for sales prediction. In Proceedings of the 2018 IEEE International Conference on Data Mining (ICDM), Singapore, 17–20 November 2018; Institute of Electrical and Electronics Engineers (IEEE), 2018; pp. 49–58.
21. Žliobaitė, I.; Bakker, J.J.; Pechenizkiy, M. Beating the baseline prediction in food sales: How intelligent an intelligent predictor is? *Expert Syst. Appl.* 2012, 39, 806–815. [CrossRef]
22. Arunraj, N.S.; Ahrens, D. A hybrid seasonal autoregressive integrated moving average and quantile regression for daily food sales forecasting. *Int. J. Prod. Econ.* 2015, 170, 321–335. [CrossRef]
23. Zhang, M.; Huang, X.-N.; Yang, C.-B. A sales forecasting model for the consumer goods with holiday effects. *J. Risk Anal. Crisis Response* 2020, 10, 69–76. [CrossRef]
24. Liu, J.; Liu, C.; Zhang, L.; Xu, Y. Research on sales information prediction system of e-commerce enterprises based on time series model. *Inf. Syst. e-Business Manag.* 2019, 1–14. [CrossRef]
25. Najera, J.A. *Sales Forecast. Using a Machine Learning Time Series Model*. 2019. Available online: https://openreview.net/forum?id=rYIYcmQ_1H (accessed on 24 November 2020).
26. Chen, T.; Guestrin, C. Xgboost: A scalable tree boosting system. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13 August 2016; pp. 785–794.

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).