IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received May 3, 2020, accepted June 3, 2020, date of publication June 10, 2020, date of current version June 23, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3001287

Multivariate Financial Time-Series Prediction With Certified Robustness

HUI LI[1], YUNPENG CUI[2,7], SHUO WANG[3,4], (Member, IEEE), JUAN LIU[2,7], JINYUAN QIN[5], AND YILIN YANG[6]

[1]School of Labor Economics, Capital University of Economies and Business, Beijing 100026, China
[2]Key Laboratory of Agricultural Big Data, Ministry of Agriculture and Rural, Beijing 100081, China
[3]Faculty of Information Technology, Monash University, Melbourne, VIC 3800, Australia
[4]CSIRO, Canberra, ACT 2601, Australia
[5]Faculty of Business and Economics, University of Melbourne, Melbourne, VIC 3010, Australia
[6]Faculty of Science, The University of British Columbia, Vancouver, BC V6T 1Z4, Canada
[7]Agriculture Information Institute, Chinese Academy of Agricultural Sciences, Beijing 100081, China

Corresponding authors: Shuo Wang (shuo.wang@monash.edu) and Juan Liu (liujuan@caas.cn)

This work was supported in part by the Innovation Program of Agricultural Information Institute, Chinese Academy of Agricultural Sciences, under Grant 2020CX043, and in part by the Fundamental Research Funds of Chinese Academy of Agricultural Sciences under Grant Y2020PT14.

ABSTRACT The futures market's forecasts are significant to investors and policymakers, where the application of deep learning approaches to finance has received a great deal of attention. In this study, we propose a multivariate financial time-series forecasting method. Our model addresses the long- and short-term features, multimodal and non-stationarity nature of multivariate time-series by incorporating the improved deep neural networks and certified noise injection. Specifically, multimodal variational autoencoder is used to extract deep high-level features of multivariate time-series, Long- and Short- Term recurrent neural network is applied for multivariate time-series forecasting, and certified noise injection mechanism, inspired by differential privacy, is proposed to improve the robustness and accuracy of prediction. Extensive empirical results on real-world agricultural commodity futures price time series and relevant external data demonstrate that our model achieves better performance over that of several state-of-the-art baseline methods.

INDEX TERMS Futures prices, deep neural networks, prediction, multivariate, Gaussian noise.

I. INTRODUCTION

Considering the significance of the futures market in the financial field, forecasting futures price movements is critical to investors and policymakers. For example, a good prediction of agricultural commodity futures prices is crucial for providing price information for decision support of agricultural commodity, while reducing the uncertainty and risks of agricultural markets and crop insurance programs [1], [2]. Traditional prediction models, e.g., linear model, Auto-Regressive Integrated Moving Average (ARIMA), and Vector Auto-Regression (VAR), are commonly assumed to be independent variables, normal distribution, which is contradicted with the real market. On the other hand, machine learning models, such as Deep Neural Networks (DNNs), have received a great deal of attention for predicting financial time series and gain high predictive accuracy [3]–[8]. Such methodologies have several distinct advantages such as non-parametric, self-learning, non-assumption, and noise-tolerant [9].

However, there are three major concerns faced by the DNN-based prediction on agricultural commodity futures, resulting in poor prediction performance:

(1) It is necessary but challenging to recognize high-level features from the high dimensionality financial time series for effective and accurate prediction. Existing works cannot effectively learn the complex and high dimensionality data distribution of time series data, and hence cannot make an accurate prediction. On the other hand, lower-dimensional representation can reduce the total number of data points but easily lose critical information.

(2) Ignoring the local dependency patterns among multi-dimensional input variables and the connection between linear and non-linear structures usually yields unsatisfactory outcomes as the price of agricultural commodities is susceptible to several factors, such as weather or government policies.

The associate editor coordinating the review of this manuscript and approving it for publication was Alba Amato.

Namely, agricultural commodity futures prices are multivariate time series in nature, consisting of multiple correlated univariate time series or channels. Besides, many studies prefer time series prediction of short-term repeating patterns rather than a mixture of short-term (e.g., the effects of extreme weather) and long-term repeating patterns (e.g., the change between days vs. nights, summer vs. winter).

(3) Futures market prediction is usually considered as one of the most challenging issues among time-series predictions due to its noise and volatile features. Agricultural commodity futures may change dramatically in different periods, and there is insufficient information to observe past behaviors [10].

In this study, we propose a deep neural network-based prediction model with certified robustness. It is a combination of the three components to address the above three challenges, and we refer to this novel model as DP-MAELS hereafter. Our contributions can be summarized as follows:

- A multimodal latent representation model is used to learn compressed representations of multivariate time series. Specifically, the Multimodal-VAE model is used to hierarchically extract invariant and abstract deep high-level features of multivariate time-series in an unsupervised manner. Gaussian Mixture priors are applied in the latent space to characterize multimodal data.
- Long- and Short- Term prediction model (LST-Prediction) is applied for agricultural commodity futures price forecasting, considering the combination of short-term and very long-term repeating patterns.
- Certified noise injection mechanism, inspired by differential privacy (DP), is proposed to improve the robustness and accuracy of prediction and address the third challenge.
- We conduct an extensive empirical evaluation using real-world global agricultural commodity futures price time series and relevant external data, to validate the efficiency and robust performance of DP-MAELS.

The structure of the paper is organized as follows. Section 2 describes the background concepts. Sections 3 explains the system design, optimization strategies, and setting details. Section 4 describes our experimental evaluation of the prediction accuracy and robustness results. Related works are summarized in Section 5, and Section 6 concludes the work.

II. BACKGROUND

A. AUTOENCODERS AND VARIATIONS

Autoencoders (AEs) are common deep models in unsupervised learning [11]. It aims to represent high-dimensional data through the low-dimensional latent layer, a.k.a. Bottleneck vector or code. Basically, an encoder E, parameterized by $q_\phi(z|x)$, is trained to convert high-dimensional data x into the latent representation bottleneck vector z in latent space that follows a specific Gaussian distribution $p(z) \sim N(0, 1)$. The decoder $p_\theta(x|z)$ is trained to reconstruct the latent vector z to x. The training process of autoencoders is to minimize the reconstruction error. Formally, we can define the encoder and the decoder as transitions τ_1 and τ_2:

$$\tau_1(X) \rightarrow Z, \tau_2(Z) \rightarrow \tilde{X}, \tau_1, \tau_2 = \underset{\tau_1, \tau_2}{argmin} \left\| X - \tilde{X} \right\|^2 \tag{1}$$

The VAEs [12] model shares the same structure with the autoencoders, but is based on the assumption that the latent variables follow some kind of distribution, such as Gaussian or uniform distribution. It uses variational inference for the learning of the latent variables. In VAEs, the hypothesis is that the data is generated by a directed graphical model $p(x|z)$ and the encoder is to learn an approximation $q_\phi(z|x)$ to the posterior distribution $p_\theta(z|x)$ estimated by decoder. The encoder and decoder are trained simultaneously based on the negative reconstruction error and the regularization term, i.e. Kullback-Leibler (KL) divergence between $q_\phi(z|x)$ and $p(z)$, by optimizing the variational lower bound:

$$L(\theta, \phi; x) = KL(q_\phi(z|x)||p(z)) - \mathbf{E}_{q_\phi(z|x)}[logp_\theta(x|z)] \tag{2}$$

The left part is the KL divergence regularization term to match the posterior of z conditional on x, i.e., $q_\phi(z|x)$, to a target distribution $p(z)$, e.g., Gaussian distribution whose mean μ and diagonal covariance $\sum$ are the encoder output.

The right part denotes the reconstruction loss for a specific sample x. In a training batch, the loss can be averaged as:

$$\begin{aligned} L_{VAE} = \mathbf{E}_{p_{data}(x)}[L(\theta, \phi; x)] = \mathbf{E}_{p_{data}(x)}[KL(q_\phi(z|x)||p_\theta(z))] \\ -E_{p_{data}(x)}[\mathbf{E}_{q_\phi(z|x)}[logp_\theta(x|z)]] \end{aligned} \tag{3}$$

β-VAE [37] is a modification of the VAE framework that introduces an adjustable hyperparameter β to the original VAE objective:

$$\mathcal{L} = \mathbb{E}_{q_\phi}(logp_\theta(x|z)) - \beta D_{KL}(q_\phi(z|x)||p_\theta(z)) \tag{4}$$

Well-chosen values of β (usually $\beta > 1$) result in more disentangled latent representations z. As shown in Figure 2, the structure of SAEs is stacking n autoencoders into n hidden layers by an unsupervised layer-wise learning algorithm and then finetuned by a supervised method. Generally, the SAEs can be achieved into three stages: (1) Train the first autoencoder by input data and obtain the learned feature vector; (2) The feature vector of the former layer is used as the input for the next layer, and this procedure is repeated until the training completes. (3) After all the hidden layers are trained, the backpropagation algorithm is used to minimize the cost function and update the weights with labeled training set to achieve finetuning.

B. CNN AND LSTM

1) CNN

For supervised classification, Convolutional Neural Networks (CNNs) are among the most successful models The full CNN framework and formula derivation can be referred to [13]. CNNs are hierarchical models whose convolutional layers alternate with subsampling layers, reminiscent of simple and complex cells in the primary visual cortex [14].

The previous layer feature maps are convolved with learnable kernels, which form the output feature map through the activation function. Multiple input maps may be combined as the output with convolutions. For convenience, we just introduce the convolution layer:

$$x_j^l = f(\sum_{i \in M_j} x_i^{l-1} * k_{ij}^l + b_j^l) \tag{5}$$

where M_j represents a selection of input maps.

2) LSTM

Recurrent neural networks have the capability to incorporate experience due to internal recurrence [15] dynamically. RNNs can project the dynamic properties of the system automatically, thus they are computationally more powerful than feed-forward networks, and the valuable approximation results are obtained for chaotic time series prediction [16]. One of the RNN models is long-short-term memory which works when there is a long delay, and the signals with a mixture of low and high-frequency components can be able to handle. However, the learning process of RNN models requires a relatively long time because there is a recurrent network architecture [17]. A schematic of the vanilla LSTM block [18] is illustrated in Figure 1.

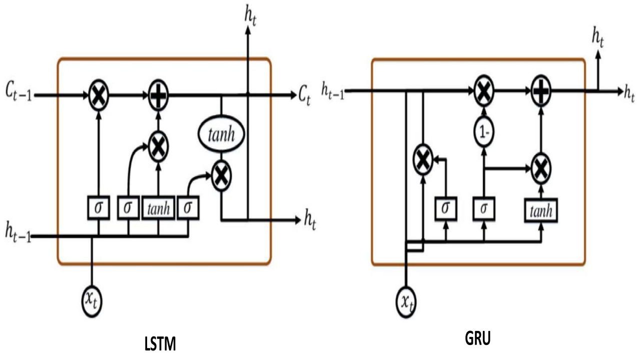

FIGURE 1. Structure of LSTM and GRU.

LSTM consists of three gates (input, forget and output), block input, a single cell (the Constant Error Carousel), an output activation function, and peephole connections. The output of the block is recurrently connected back to the block input and all of the gates. The vector formulas for LSTM layer forward pass can be expressed as follows [18]:

$$\begin{aligned} z^t &= g(W_z x^t + R_z y^{t-1} + b_z) \textit{ block input} \\ i^t &= \sigma(W_i x^t + R_i y(t-1) + p_i \odot c^(t-q) + b_i) \textit{ input gate} \\ f^t &= \sigma(W_f x^t + R_f y(t-1) + p_f \odot c^{t-1} + b_f \textit{ forget gate} \\ c^t &= i^t \odot z^t + f^t \odot c^{t-1} \textit{ cell state} \\ o^t &= \sigma(W_o x^t + R_o y^{t-1} + p_o \odot C^t + b_o) \textit{ output gate} \\ y^t &= o^t \odot h(c^t) \textit{ block output} \end{aligned} \tag{6}$$

where x_t is the input vector at time t, the W are input weight matrices, the R are square recurrent weight matrices, the p are peephole weight vectors and b are bias vectors. Functions, g and h are point-wise non-linear activation functions: logistic sigmoid $\frac{1}{1+e^{-x}}$ is used for as activation function of the gates and hyperbolic tangent is used as the block input and output activation function. The point-wise multiplication of two vectors is denoted as $\odot$.

C. (ϵ, δ)-DIFFERENTIAL PRIVACY

Definition 1: (ϵ, δ)-differential privacy [19]. For scalars $\epsilon > 0$ and $0 \leq \delta < 1$, mechanism $\mathcal{M}$ is said to preserve (approximate) (ϵ, δ)-differential privacy if for all adjacent datasets $D, D' \in \mathcal{D}^n$ and measurable $S \in \text{range}(\mathcal{M})$,

$$\Pr\{\mathcal{M}(D) \in S\} \leq \exp(\epsilon) \cdot \Pr\{\mathcal{M}(D') \in S\} + \delta.$$

Given the deterministic function f, differential privacy can be satisfied by adding random noise into the output of the deterministic function, where the scale of noise is decided by the sensitivity of f. The sensitivity measures the maximum change in the query answers due to the change of a single database entry.

Definition 2: Sensitivity [20]. The sensitivity of query function f is the maximum change in the query results:

$$\Delta(f) = \max_{D_1, D_2} \parallel f(D_1) - f(D_2) \parallel_1 \tag{7}$$

If sensitivity of f is defined using $\mathcal{L}2$ norm, the Gaussian mechanism is selected for randomizing the output of f [19]. The Gaussian noise mechanism used to achieve differential privacy is defined by

$$\mathcal{M}(d) \triangleq f(d) + \mathcal{N}(0, (\Delta f)^2 \sigma^2 \mathcal{I}), \tag{8}$$

Here $\mathcal{N}(0, (\Delta f)^2 \sigma \mathcal{I})$ is a Gaussian distribution with zero mean and covariance matrix $(\Delta f)^2 \sigma^2 \mathcal{I})$ and $\mathcal{I}$ is the identity matrix. A single application of the Gaussian mechanism to function f of sensitivity Δf satisfies (ϵ, δ)-differential privacy if $\delta \geq \frac{4}{5} exp(-(\sigma\epsilon)^2/2)$ and $\epsilon < 1$.

III. DP-MAELS: MULTIVARIATE TIME-SERIES PREDICTION OF AGRICULTURAL COMMODITY FUTURES PRICES

A. ARCHITECTURE

To address the challenges mentioned in the introduction, our prediction model DP-MAELS is composed of three components: (**C1**) multimodal-VAE for addressing the local dependency patterns among multi-dimensional input variables and feature extraction, (**C2**) LST-prediction for the multivariate time-series prediction, including a recurrent component to discover the short-term patterns in the time dimension, a recurrent-skip component to discover long-term patterns for time series trends (**C3**) provable noise injection to improve the robustness of prediction against non-stationary nature.

The scheme of the DP-MAELS is illustrated in Figure 2. The input of the scheme consists of multivariate time-series, including fundamental data and external information. The encoder of the trained multimodal-VAE model is applied to produce a set of features for each time step, as the feature

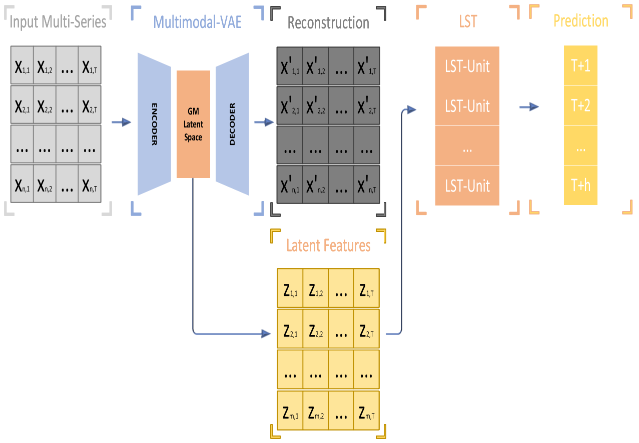

FIGURE 2. Scheme of DP-MAELS.

extractor, resulting in a new time series of features with the same length as the original one but much smaller dimensions. An LST-prediction is then applied on the time series produced by the multimodal-VAE, and in turn, produces a prediction for the desired time step. To enhance the robustness and utility, the provable noise injection mechanism is incorporated into the DP-MAELS.

B. MULTIMODAL-VAE FOR LATENT REORIENTATION OF MULTIVARIATE TIME-SERIES

Since agricultural commodities futures price is correlative with many external factors, e.g. weather, macro data, and other future markets, the dimension of the multivariate time series would be very huge. The autoencoder (AE) or a variational autoencoder (VAE) is commonly adopted to learn the low-dimensional representation of complex data. However, since only a single Gaussian distribution is assumed as the prior in the data generative procedure, such representations cannot well approximate the original data distribution, especially when input data distributions are strongly multimodal [21].

In this section, we propose a multimodal latent representation model to capture the multivariate time-series data's high-level latent representation. The multimodal latent representation.

The agricultural commodity futures prices at time t is denoted by $\boldsymbol{x}^{(t)} = [x_1^{(t)}, x_2^{(t)}, \cdots, x_i^{(t)}, \cdots, x_D^{(t)}]$, where $x_i^{(t)}$ reveals the value of attributes i (D=299 in this case) at time t. The time series data are grouped into time windows of size T. The jth time window represents a $D \times T$ matrix $\boldsymbol{W}^{(j)} = [\boldsymbol{w}^{(j,1)}, \boldsymbol{w}^{(j,2)}, \cdots, \boldsymbol{w}^{(j,D)}]_{D\times T}$, where $\boldsymbol{w}^{(j,t)} = \boldsymbol{x}^{(j-1)\times T+t}$. Formally, given a series of observed time series data $\boldsymbol{X} = \{\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \cdots, \boldsymbol{x}^{(T)}\}$, where$x^{(t)} \in \mathbb{R}^n$ and n is the variable dimension, we aim to estimate a series of agricultural commodity futures prices in a rolling forecasting manner. Namely, to predict the $\boldsymbol{x}^{T+h)}$, where h is the desirable horizon steps ahead of the current timestamp. Thus, the input of the prediction model is $n \times T$ matrix of futures prices at successive T timestamps, and the output is predictions for the $(h+T)^{th}$ timestamp.

a mixture of Gaussians is used as prior in the latent space, so that the Multimodal-VAE could learn complex hidden distributions and better approximate the original data distribution.

The input matrix is mapped to the Gaussian Mixture latent codes, followed by further being transported to the decoder part to reconstruct the original input. The loss function measures the difference between the reconstructed data from the original input. Specifically, a mixture of Gaussian distributions is used as prior in the latent codes. Given K components in the Gaussian Mixture, a categorical prior distribution is denoted by π_c and the prior probability of the kth component is denoted by pr_k^t. For each data sample, one component is chosen at first based on the categorical prior distribution among K components. Given a determined component, the corresponding latent Gaussian distribution is also determined. Next, the Kullback-Leibler divergence of the approximate from the true posterior is calculated as follows:

$$\begin{aligned}
&D_{KL}[q_\phi(z^{(t)}, w_k^{(t)}|\boldsymbol{x}^{(t)})||p_\theta(z^{(t)}; w_k^{(t)}|\boldsymbol{x}^{(t)})] \\
&\quad = -\boldsymbol{E}_{q(z^{(t)}, w_k^{(t)}|\boldsymbol{x}^{(t)})} log \frac{p_\theta(z^{(t)}, w_k^{(t)}, \boldsymbol{x}^{(t)})}{q_\phi(z^{(t)}, w^{(t)}|x^{(t)})} + log\ p_\theta(\boldsymbol{x}^{(t)}) \\
&\quad = -L_{VAE} + log\ p_\theta(\boldsymbol{x}^{(t)}) \qquad (9)
\end{aligned}$$

The training is to minimize the KL divergence, i.e., maximize the variational lower bound L_{VAE} under Gaussian Mixture prior. The $q_\phi(z^{(t)}, w^{(t)}|x^{(t)})$ is assumed to follow a mean-filed distribution as [21], then $q_\phi(z^{(t)}, w^{(t)}|x^{(t)}) = q_\phi(z^{(t)}|x^{(t)})q_\phi(z^{(t)}|x^{(t)})$. Consequently, L_{VAE} can be calculated as follows:

$$\begin{aligned}
L_{VAE} &= \boldsymbol{E}_{q(z^{(t)}, w_k^{(t)}|x^{(t)})}[log\ p_\theta(x^{(t)}, z^{(t)}, w_k^{(t)}) \\
&\quad - log\ p_\theta(z^{(t)}, w^{(t)}|x^{(t)})] \\
&= \boldsymbol{E}_{q(z^{(t)}, w_k^{(t)}|x^{(t)})}[log\ p_\theta(\boldsymbol{x}^{(t)})|z^{(t)}] \\
&\quad - D_{KL}(q_\phi(z^{(t)}, w_k^{(t)}|\boldsymbol{x}^{(t)})||p_\theta(z^{(t)}, w_k^{(t)})) \qquad (10)
\end{aligned}$$

Here, the first term is the reconstruction error, in which both the $z^{(t)}$ and $w_k^{(t)}$ are considered into the reconstruction of input. The second term is the regularization that push the Gaussian prior as close to the variational posterior as possible. The L_{VAE} can also be rewritten as follows:

$$\begin{aligned}
&L_{VAE} \\
&= \int_{\boldsymbol{z}^{(t)}} q_\phi(\boldsymbol{z}^{(t)}|\boldsymbol{x})^{(t)}) log \frac{p_\theta(\boldsymbol{x}^{(t)})|\boldsymbol{z}^{(t)})p_\theta(\boldsymbol{z}^{(t)})}{q_\phi(\boldsymbol{z}^{(t)}|\boldsymbol{x})} d\boldsymbol{z}^{(t)} \\
&\quad - \int_{\boldsymbol{z}^{(t)}} \sum_{w_k^{(t)}} q_\phi(\boldsymbol{z}^{(t)}|\boldsymbol{x})^{(t)}) D_{KL}(q_\phi(w_k^{(t)}|\boldsymbol{x}^{(t)})||p_\theta(w_k^{(t)}|z^{(t)}))d\boldsymbol{z}^{(t)} \\
&\qquad (11)
\end{aligned}$$

If we consider the unknown $q_\phi(w^{(t)_j}|\boldsymbol{x}^{(t)})$ to be equal to $p_\theta(w^{(t)_j}|\boldsymbol{z}^{(t)})$, i.e. the second term is always 0, then the L_{VAE} achieves the maximum w.r.t the first term.

The convolutional network is used to construct encoder and decoder networks to extract short-term patterns and interdependence among the input multi variables. The filters used in the convolutional layer are with width w and height n (namely the input dimension). The output of the k^{th} filter is vector $h_k = RELU(W_k * X + b_k)$, where $*$ is the convolution operation, and the RELU function is $RELU(x) = max(0, x)$. The output of the convolutional layer is a $\#filter \times T$ matrix, and then simultaneously fed into LST-prediction components.

C. LST-PREDICTION COMPONENT

1) SCHEME OF THE LST-PREDICTION

Agricultural commodities futures price is multivariate time series in nature, characterizing correlation, and a combination of short and long-term repeating patterns. In this work, we consider many variables regarding the agricultural commodities futures price. These variables represent a mixture of long-term and short-term patterns. In this section, we apply the LST-prediction to conduct the multivariate time-series prediction. Specifically, local dependencies between variables and important features extracted by the pre-trained encoder of the Multimodal-VAE component are used as the input for the LST-prediction. The recurrent component is used to discover the short-term patterns in the time dimension. Besides, the Recurrent-skip component is used to discover long-term patterns for time series trends. Therefore, we could discover short-term and long-term repeating patterns of multivariate agricultural commodity futures prices and predict prices more accurately.

The LST-prediction model consists of two components: (1) non-linear part, including recurrent component to capture the short-term dependence patterns, a recurrent-skip component and temporal attention layer to capture the very long-term dependence patterns; (2) linear part in parallel to the non-linear neural network part, applying an autoregressive model (AR) to enhance the robustness of the non-linear deep learning model for the time series with violate scale changing.

2) RECURRENT COMPONENT AND RECURRENT-SKIP COMPONENT

Recurrent component and recurrent-skip component are used to capture long-term and very long-term information, respectively. The Recurrent component is a recurrent layer with the Gated Recurrent Unit (GRU) [22] and uses the RELU function as the hidden update activation function. A gated recurrent unit (GRU) [22] has been demonstrated to have a better performance in discovering the correlations among sequence data over the traditional LSTM network due to its fewer parameters [23]. The input of the encoder is firstly fed into the GRU that enables each recurrent unit to adaptively capture dependencies of different time scales. Specifically, the GRU has gating units that modulate the flow of information inside the unit, having no separate memory cell. The activation h_t^j of the GRU at time t is a linear interpolation between the previous activation h_{t-1}^j and the candidate activation $\tilde{h}_t^j$. Namely, $h_t^j = (1 - z_t^j) \odot h_{t-1}^j + z_t^j \odot \tilde{h}_t^j$. Here, an update gate z_t^j is used to decide how much the unit updates its activation, or content, where $z_t^j = sigmoid(W_z x_t + U_z h_{t-1})^j$. The candidate activation $\tilde{h}_t^j = tanh(Wx_t + U(r_t \odot h_{t-1}))^j$, where r_t is a set of reset gates and is an element-wise multiplication. When $r_t \rightarrow 0$, the reset gate effectively makes the unit act as if it is reading the first symbol of an input sequence, allowing it to forget the previously computed state. The reset gate r_t^j is computed as $r_t^j = sigmoid(W_r x_t + U_r h_{t-1})^j$. The $W,\ W_r,\ ,W_z,\ U,\ U_z,\ U_r$ are the learned weight metrics.

The outputs of the recurrent layer and recurrent-skip layer are the hidden states at each timestamp. The hidden state of GRU at time t is computed as follows, to capture relatively long-term dependencies. However, due to gradient vanishing, the GRU may fail to capture a very long-term correlation. A recurrent structure with temporal skip-connections, Recurrent-Skip Component, is applied to extend the temporal span of the information flow then memory history information in a longer period.

This is achieved by adding skip-links between the current hidden cell and the hidden cells in the same phase in adjacent periods. The hidden state of Recurrent-Skip Component at time t is computed as,

$$\begin{aligned} r_t &= sigmod(x_t W_{xr} + h_{t-p} W_{hr} + b_r) \\ u_t &= sigmod(x_t W_{xu} + h_{t-p} W_{hu} + b_u) \\ c_t &= RELU(x_t W_{xc} + r_t \odot (h_{t-p} W_{hc}) + b_c) \\ h_t &= (1 - u_t) \odot h_{t-p} + u_t \odot c_t \end{aligned} \tag{12}$$

where p is the number of hidden cells skipped through. The value of p can be easily determined for datasets with clear periodic patterns, and has to be tuned otherwise.

The output of the GRU-based Recurrent component and Recurrent-skip component are connected by a dense layer. Specifically, the input of this dense layer is the hidden state of GRU-based Recurrent component h_t^R, and p hidden states of Recurrent-skip component from timestamp $t - p + 1$ to t denoted by $h_{t-p+1}^S, h_{t-p+2}^S, \cdots, h_t^S$. The output of the dense layer is calculated as follows:

$$h_t^D = W^R h_t^R + \sum_{i=0}^{p-1} W^S h_{t-1}^s + b \tag{13}$$

3) AUTOREGRESSIVE LAYER

Since the scale of input time-series constantly change in a non-periodic manner, we decompose the final prediction of LSTNet into a linear part aiming at the local scaling issue via Autoregressive (AR), concatenated with the non-linear part containing recurring patterns as the final prediction. Let h_t^A denote the prediction result of the AR, and $W^a \in \mathbb{R}^B$ and b^a denote the coefficients, the AR model is given as follows:

$$h_{t,i}^a = \sum_{k=0}^{B} W_k^a y_{t-k,i} + b^a \tag{14}$$

The final prediction of LSTNet at timestamp t is calculated by integrating the outputs of the neural network part and the AR component:

$$\tilde{Y}_t = h_t^D + h_t^A \tag{15}$$

D. ENHANCE ROBUSTNESS VIA CERTIFIED NOISE

To improve the robustness of the prediction model against the noisy and volatile nature of data, we apply a provable Gaussian noise injection method that inserts certified noise at the neural network training, inspired by differential privacy [24], [25]. The robustness of the neural network reveals that accuracy is guaranteed to be slightly affected by certain volatile information.

By considering the input matrix from the first step as databases in DP definition, randomizing the scoring function $f(x)$ to enforce DP on a small number of elements in a matrix input guarantees robustness of predictions even with that number of greatly biased elements. To achieve the goal, noise $\mathcal{N}(0, \sigma_l^2)$ is injected into each layer l of a deep neural network, similar to [26]. This results in the following (ϵ_l, δ_l)-DP condition, with a budget ϵ_l and a broken probability δ_l of robustness, as follows:

$$\mathbb{E}f_k(x) \leq e^{\epsilon_l}\mathbb{E}f_k(x+a) + \delta_l, a \in \mathcal{N}(0, \sigma_l^2) \tag{16}$$

where $\mathbb{E}f_k(x)$ is the expected value of $f_k(x)$. SGD trains the network. At the prediction time, a certified robustness check is applied for each prediction.

IV. EVALUATION

A. DATASETS

In this section, we conduct an extensive evaluation to validate our DP-MAELS based on the multivariate agricultural commodity futures prices and relevant external data. COBT soybean is used as two typical agricultural commodity futures for the prediction of settle prices and trends.

Many factors affect changes in futures prices. Four The external information are considered in this article: (1) supply factors, including agricultural commodity supply; (2) demand factors, including agricultural commodity consumption, GDP, CPI, LIBOR, inflation level, crude oil market; (3) agricultural commodity futures/spot Market factors, including CBOT agricultural commodity futures price, trading volume and open interest (soybean oil, corn, rice, oats, soybean meal, wheat); (4) unforeseen factors, including climate and environmental factors. Sources of data: USDA United States Department of Agriculture, CBOT Exchange, Quandl[1] platform. Closing prices of COBT bean between 07/01/1956 and 02/01/2020 are collected for evaluation. To match quarterly macro data (e.g., GDP) with other daily data, we assumed the macro features to be constant throughout the same quarter. Each sample data of agricultural commodity futures prices is split into a training set (60%), validation set (20%), test set (20%) in chronological order. The study uses a validation set to tune hyperparameters, while uses a test set to evaluate and compare the forecasting performance of DP-MAELS and other models. Besides, the Null values are immediately dropped due to its little scale. To detect the agricultural commodity futures price pattern, it is necessary to normalize the agricultural commodity futures price data. Since the LSTM neural network requires the agricultural commodity futures patterns during training, we use the "min-max" normalization method to reform dataset, which keeps the pattern of the data, as follow:

$$x_t^n = \frac{x_t - min(X)}{max(X) - min(X)} \tag{17}$$

where x_t^n denotes the data after normalization. Accordingly, denormalization is required at the end of the prediction process to get the original price, which is given by:

$$\tilde{x}_t = \tilde{x}_t^n[max(X) - min(X)] + min(X) \tag{18}$$

where $\tilde{x}_t^n$ denotes the predicted data and $\tilde{x}_t$ denotes the predicted data after denormalization. Note that the text representation values are not normalized. To better understand historical variation, four sub-figures in Figure 3 show agricultural commodity futures prices in a daily, monthly, quarterly, and yearly scale, respectively. The short-term and long-term repeating patterns are not clear due to non-stationary time series or patterns with a flexible period.

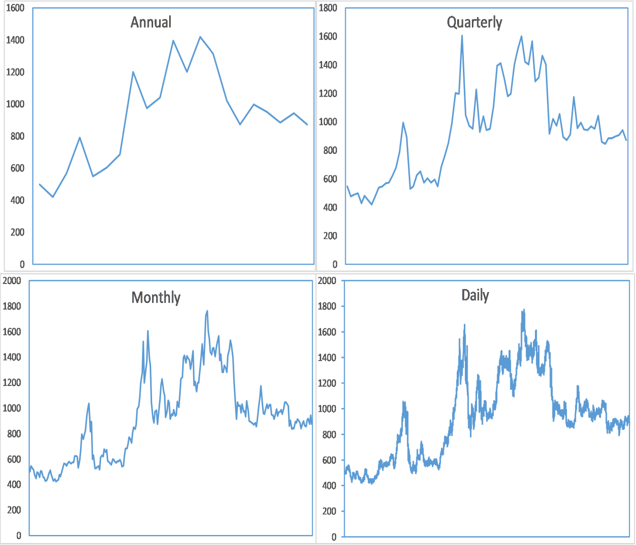

FIGURE 3. Illustration of the settle price value in daily, monthly, quarterly and yearly scale, respectively.

B. EVALUATION METRICS

The prediction performance of our approach MAELS is compared to the state-to-the-arts machine learning based predicions, such as CNN, RNN, LSTM, ARIMA, and SVM-VAR. CNN, RNN and LSTM approaches enable the handling of multivariate input and output, while ARIMA is a single output method in which n models are trained separately.

[1] https://www.quandl.com/

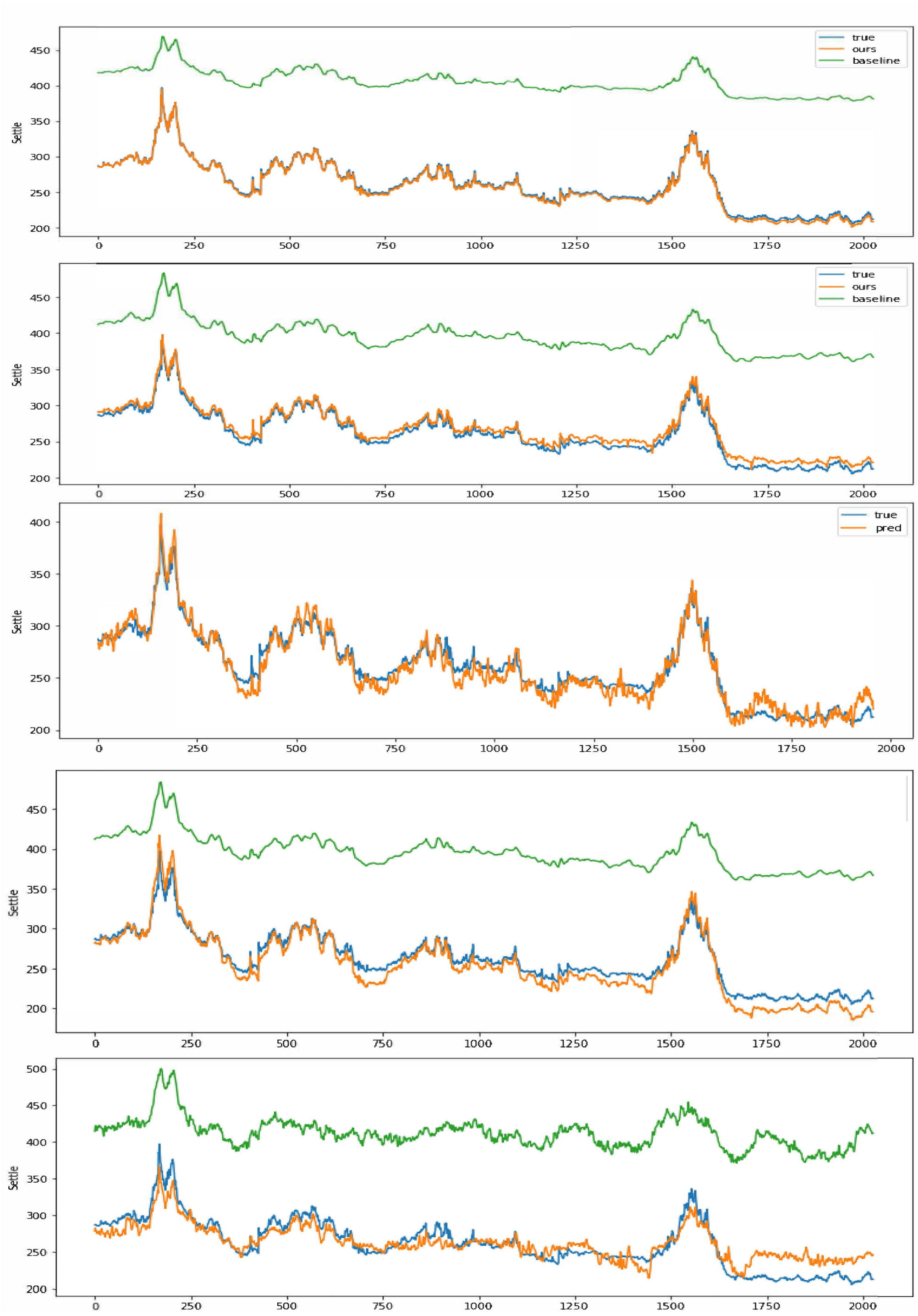

FIGURE 4. Illustration of settle price value prediction using DP-MEALS, MEALS, LSTM-Skip, LSTM, CNN, from top to bottom respectively. Baseline methold is SVM-VAR.

Performance measures include mean absolute percentage error (MAPE), Root Relative Squared Error (RSE), Relative Absolute Error (RAE), Empirical Correlation Coefficient (R), and Theil U. The RSE and RAE are used to compare the efficiency of prediction in a scaled version.

MAPE measures the size of the error in terms of the relative average of the error. Theil U is a relative measure of the difference between two variables. It squares the deviations to give more weight to large errors and to exaggerate errors. If R is bigger, it means that the predicting value is similar to the actual value, while if MAPE and Theil U are smaller, this also indicates that the predicted value is close to the actual value [27], [28].

$$
\begin{aligned}
MAPE &= \frac{1}{T}\sum \left|\frac{\hat{x}_{t,s} - x_{t,s}}{x_{t,s}}\right| \\
RSE &= \frac{\sqrt{\sum_{s,t}(\hat{x}_{t,s} - x_{t,s})^2}}{\sqrt{\sum_{s,t}(mean(\boldsymbol{x}) - x_{t,s})^2}} \\
RAE &= \frac{\sum_{s,t}|\hat{x}_{t,s} - x_{t,s}|}{\sum_{s,t}|mean(\boldsymbol{x}) - x_{t,s}|} \\
R &= \frac{1}{n}\sum_{1}^{T}\frac{\sum(mean(\boldsymbol{x}) - x_{t,s})(mean(\hat{\boldsymbol{x}}) - \hat{x}_{t,s})}{\sqrt{\sum(mean(\boldsymbol{x}) - x_{t,s})^2(mean(\hat{\boldsymbol{x}}) - \hat{x}_{t,s})^2}} \\
TheilU &= \frac{\sqrt{\frac{1}{T}\sum(\hat{x_{t,s}} - x_{t,s})^2}}{\sqrt{\frac{1}{T}\sum \hat{x_{t,s}}^2} + \sqrt{\frac{1}{T}\sum x_{t,s}^2}} \qquad (19)
\end{aligned}
$$

Here, $x_{t,s}$ and $\hat{x}_{t,s}$ are true values and predicted values of the s^{th} agricultural futures at the timestamp t, respectively, n is the number of out of sample forecasts, τ is forecast horizon, T is total sample size. Our experiments are implemented in the Linux system (Ubuntu 18 LTS) with GPU (device 0: NVIDIA V100) and 64.00GB RAM with running Python 3.7 source codes. In the price prediction of agricultural commodity futures, a skip length τ of 24 is found to achieve the best possible results. The default values of the hidden dimension of Recurrent and Convolutional layer, the dropout rate, the horizon h and the optimization algorithm are arbitrarily chosen to be 50, 0.2, 12, respectively.

C. EXPERIMENTAL RESULTS

We demonstrate the accuracy measurements to demonstrate the predictive performance and illustrate the performance of our enhancement strategies.

D. PREDICTIVE EVALUATION

We first directly plot the settle price value prediction results of DP-MAELS on soybean futures, as shown in Figure 4. We compare the predicted price value of the DP-MAELS to benchmark baselines, and DP-MAELS without the robustness enhanced strategies (legend as MAELS), as illustrated in these sub-figures of Figure 4.

The prediction of settle price we can find that all DNN-based approaches outperform the baseline method SVM-VAR, where LSTM and CNN have larger variations and distances to the real settle price data than DP-MAELS and MAELS. Furthermore, comparing DP-MAELS with MAELS, the former outperforms the latter: DP-MAELS has less volatility and is closer to the actual trading data than MAELS.

In addition, we conduct predictive performance measures on soybean future trading using these approaches in terms of our accuracy measure metrics. As shown in Figure 5, We illustrate the performance of the DP-MAELS in terms of MPAs, and RSE, RAE, R and TheilU between the predicted and real price values, compared to the LSTM, RNN, CNN, ARIMA, VAR, and DP-MAELS without robustness enhanced strategies (MAELS). Although the periodic patterns of agricultural commodity futures prices are not clear and the dataset is noisy, DP-MAELS still performs better than other DNN-based approaches and traditional econometric methods on average. It can be seen from the figure that DP-MAELS shows much better performance than the other models in predicting soybean future prices. For example, the average value of MAPE and Theil U of DP-MAELS reach 0.015 and 0.011, respectively, much less than those of the other three models. Besides, indicator R has an average value of 0.945, which is the highest among these models. Specifically, DP-MAELS outperforms the baseline RNN, CNN by 20.45%, 22.05% in RSE metric and 30.52%, 32.15% in RAE metric respectively, and more than 20 times better than ARIMA and VAR, suggesting the much better performance of the proposed method. The RAE and RSE of DP-MAELS are 18.18% and 25.01% smaller than the DP-MAELS without robustness enhanced strategy (MAELS), revealing the efficiency of the noise injection mechanism. Also, we find that the performance of DP-MAELS is much stable compared with other models. This could be because our MAE-based feature representation and noise injection mechanism are efficient in processing noise data than the other three. Besides, the R evaluation reveals that the performance of DP-MAELS is much better than others as well.

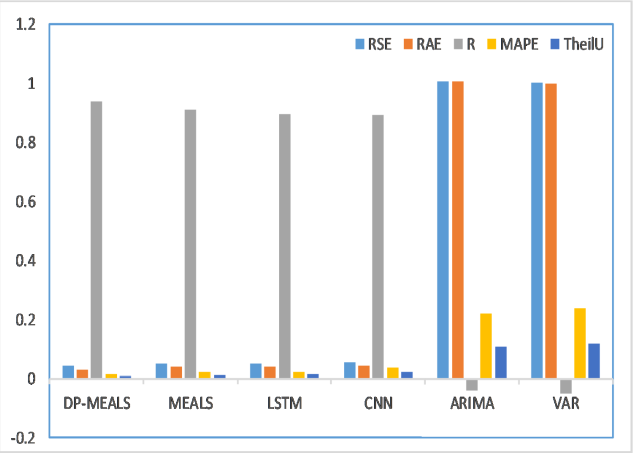

FIGURE 5. Accuracy evaluation on price value prediction.

Further, we provide trend prediction results for soybean futures over a period in Figure 6, compared to the SVM, CNN, LSTM, MAELS and LSTM-Skip (LSTM with the recurrent skip strategy). The performance of our DP-MAELS is consistently better than others. In general, DP-MAELS achieves 11.7% improvement in accuracy for trend prediction compared with MAELS, revealing the effectiveness of our enhancement strategies. Compared with normal CNN and LSTM, the DP-MAELS achieves more than 15% accuracy improvement. All these results confirm that the DP-MAELS framework achieves better accuracy and robustness.

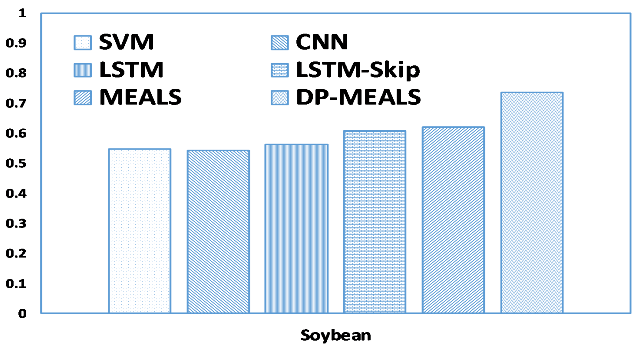

FIGURE 6. Accuracy evaluation on trend prediction.

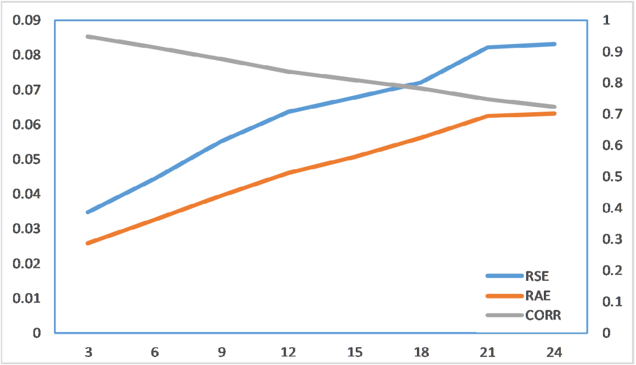

FIGURE 7. Evaluation on horizon values.

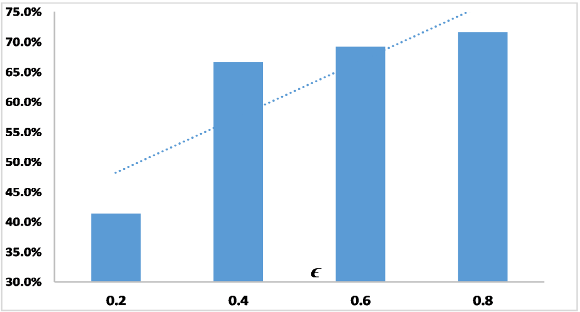

FIGURE 8. Accuracy change of trend prediction when varying ϵ.

We also test the impact of horizon value on the prediction results on the horizon set {3, 6, 9, 12, 15, 18, 21, 24}. As shown in Figure 7, the larger the horizons, the worse the prediction results.

E. PERFORMANCE ON VARYING NOISE MAGNITUDE

To answer "*How does DP noise affect the accuracy of our models?*", we evaluate the performance of the DP-MAELS for varying privacy budgets ϵ. The overall trend prediction results on soybean settle prices are shown in Figure 8. As shown, the accuracy drops as the noise scale increases while it is feasible to select a suitable noise magnitude to balance the trade-offs between robustness and utility, e.g., 0.4.

V. RELATED WORK

Recently, with more computational capabilities and the availability to handle massive databases, it is possible to use more complex machine learning models, such as deep learning models, which presents a remarkable performance in various tasks, including financial domains. Some examples of deep learning models for financial time series forecasting. ANNs were used as a prediction system for forecasting stocks listed on the S&P 500 index and achieved excellent profits using the prediction system in a simulation exercise [29]. A wavelet denoising-based back propagation neural network was applied to predict the Shanghai Composite Index and achieved more accuracy than did a conventional backpropagation neural network [30]. For time-series data, such as text, signals, stock prices, and so on, a long short-term memory (LSTM) is superior for learning temporal patterns in deep neural networks. LSTM model was applied to predict ten company's closing stock prices using textual and numerical information [31] and to predict stock price movement using five historic price measures (open, close, low, high, volume) and 175 technical indicators [32]. The LSTM model was also applied to a large-scale financial market prediction task on S&P 500 data from December 1992 until October 2015. They showed that the LSTM model outperforms the standard deep net and traditional machine learning methods [5]. A three-stage process [8] was proposed to predict six market index futures and confirmed that the performance of the proposed model was better than those of other models, such as RNN, LSTM, and wavelet-LSTM omitted second-stage models. Financial time series data can be used not only as numeric data but also as image data that is transformed in predicting stock prices. Convolutional neural networks (CNNs) can learn or extract features themselves. Not only CNNs but also data visualization methods were used to transform stock price data into image data to eliminate noise [33]. Ba *et al.* [34] proposed a deep multimodal neural network using a CNN and a multilayer perceptron to classify the dataset. Some researchers have taken a further step by examining the joint effects of various types of information, which has proven helpful in capturing stock movements [35], [36].

However, few explorations were conducted to demonstrate the performance of deep neural networks in futures prediction is rarely explored [10], especially in the agricultural domain. Existing studies on agricultural commodity futures prediction using deep neural networks mainly focused on interval forecasting of futures prices while ignoring the point forecasting. The dynamic dependencies among multiple variables were rarely explored in the existing few studies. Very long-term information and a mixture of short-term and long-term repeating patterns were not fully considered in existing works, due to gradient vanishing. Besides, the deep features of financial time series were not sufficiently studied, especially in an unsupervised manner.

To the best of our knowledge, our proposed prediction is the first attempt to incorporate deep feature extraction, the dynamic dependencies among multiple variables, a mixture of short-term and long-term repeating patterns, and certified noise robustness, to conduct accurate and robust prediction on multivariate agricultural commodity futures.

VI. CONCLUSIONS

In this paper, we propose a robust forecasting framework to predict the prices of several global agricultural commodity futures, called DP-MAELS. It integrates three strategies to address significant challenges faced by the application of deep neural networks to multivariate time-series. Specifically, a multimodal latent representation model is used to learn compressed representations of multivariate time series to relief the curse of dimensions; the recurrent-skip component is used to combine short-term and very long-term repeating patterns and certified noise injection mechanism, inspired by differential privacy, is proposed to improve the robustness and accuracy of prediction. With the empirical results, we test our proposed model in terms of predictive accuracy compared to other state-to-the-art models. The experimental results demonstrate that DP-MAELS can outperform comparisons and is a promising alternative for multivariate time series forecasting in multivariate time-series.

REFERENCES

[1] D. Wang, C. Yue, S. Wei, and J. Lv, "Performance analysis of four decomposition-ensemble models for One-Day-Ahead agricultural commodity futures price forecasting," *Algorithms*, vol. 10, no. 3, p. 108, Sep. 2017.

[2] H. Ouyang, X. Wei, and Q. Wu, "Agricultural commodity futures prices prediction via long-and short-term time series network," *J. Appl. Econ.*, vol. 22, no. 1, pp. 468–483, 2019.

[3] E. Chong, C. Han, and F. C. Park, "Deep learning networks for stock market analysis and prediction: Methodology, data representations, and case studies," *Expert Syst. Appl.*, vol. 83, pp. 187–205, Oct. 2017.

[4] Z. Guo, H. Wang, Q. Liu, and J. Yang, "A feature fusion based forecasting model for financial time series," *PLoS ONE*, vol. 9, no. 6, Jun. 2014, Art. no. e101113.

[5] T. Fischer and C. Krauss, "Deep learning with long short-term memory networks for financial market predictions," *Eur. J. Oper. Res.*, vol. 270, no. 2, pp. 654–669, Oct. 2018.

[6] S. PrasadDas and S. Padhy, "Support vector machines for prediction of futures prices in indian stock market," *Int. J. Comput. Appl.*, vol. 41, no. 3, pp. 22–26, Mar. 2012.

[7] W. Huang, Y. Nakamori, and S.-Y. Wang, "Forecasting stock market movement direction with support vector machine," *Comput. Oper. Res.*, vol. 32, no. 10, pp. 2513–2522, Oct. 2005.

[8] 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, Jul. 2017, Art. no. e0180944.

[9] S. Haykin and N. Network, "A comprehensive foundation," *Neural Netw.*, vol. 2, p. 41, Feb. 2004.

[10] T. Xiong, C. Li, Y. Bao, Z. Hu, and L. Zhang, "A combination method for interval forecasting of agricultural commodity futures prices," *Knowl.-Based Syst.*, vol. 77, pp. 92–102, Mar. 2015.

[11] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1798–1828, Aug. 2013.

[12] A. B. L. Larsen, S. K. Sønderby, H. Larochelle, and O. Winther, "Autoencoding beyond pixels using a learned similarity metric," 2015, *arXiv:1512.09300*. [Online]. Available: http://arxiv.org/abs/1512.09300

[13] J. Bouvrie, "Notes on convolutional neural networks," Dept. Brain Cogn. Sci., Center Biol. Comput. Learn., Massachusetts Inst. Technol., Cambridge, MA, USA, Tech. Rep. 5869, 2006. [Online]. Available: http://cogprints.org/5869/

[14] D. H. Hubel and T. N. Wiesel, "Receptive fields and functional architecture of monkey striate cortex," *J. Physiol.*, vol. 195, no. 1, pp. 215–243, Mar. 1968.

[15] F. Murtagh, J. L. Starck, and O. Renaud, "On neuro-wavelet modeling," *Decis. Support Syst.*, vol. 37, no. 4, pp. 475–484, Sep. 2004.

[16] H. Sak, A. W. Senior, and F. Beaufays, "Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition," *ArXiv*, vol. abs/1402.1128, 2014.

[17] P. Fryzlewicz, S. Van Bellegem, and R. von Sachs, "Forecasting non-stationary time series by wavelet process modelling," *Ann. Inst. Stat. Math.*, vol. 55, no. 4, pp. 737–764, Dec. 2003.

[18] K. Greff, R. K. Srivastava, J. Koutnik, B. R. Steunebrink, and J. Schmidhuber, "LSTM: A search space odyssey," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 10, pp. 2222–2232, Oct. 2017.

[19] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," *Found. Trends Theor. Comput. Sci.*, vol. 9, nos. 3–4, pp. 211–407, 2014.

[20] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Proc. Theory Cryptogr. Conf.* Springer, 2006, pp. 265–284.

[21] Y. Guo, W. Liao, Q. Wang, L. Yu, T. Ji, and P. Li, "Multidimensional time series anomaly detection: A gru-based Gaussian mixture variational autoencoder approach," in *Proc. Asian Conf. Mach. Learn.*, vol. 2018, pp. 97–112.

[22] K. Cho, B. van Merrienboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," 2014, *arXiv:1409.1259*. [Online]. Available: http://arxiv.org/abs/1409.1259

[23] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," 2014, *arXiv:1412.3555*. [Online]. Available: http://arxiv.org/abs/1412.3555

[24] C. Dwork, "Differential privacy: A survey of results," in *Proc. Int. Conf. Theory Appl. Models Comput.* Springer, 2008, pp. 1–19.

[25] A. Siraj Rakin, Z. He, and D. Fan, "Parametric noise injection: Trainable randomness to improve deep neural network robustness against adversarial attack," 2018, *arXiv:1811.09310*. [Online]. Available: http://arxiv.org/abs/1811.09310

[26] M. Lecuyer, V. Atlidakis, R. Geambasu, D. Hsu, and S. Jana, "Certified robustness to adversarial examples with differential privacy," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2019, pp. 656–672.

[27] F. Bliemel, "Theil's forecast accuracy coefficient: A clarification," *J. Marketing Res.*, vol. 10, no. 4, pp. 444–446, 1973. [Online]. Available: http://www.jstor.org/stable/3149394

[28] T.-J. Hsieh, H.-F. Hsiao, and W.-C. Yeh, "Forecasting stock markets using wavelet transforms and recurrent neural networks: An integrated system based on artificial bee colony algorithm," *Appl. Soft Comput.*, vol. 11, no. 2, pp. 2510–2525, Mar. 2011.

[29] Y. Zhang and L. Wu, "Stock market prediction of S&P 500 via combination of improved BCO approach and BP neural network," *Expert Syst. Appl.*, vol. 36, no. 5, pp. 8849–8854, 2009.

[30] J.-Z. Wang, J.-J. Wang, Z.-G. Zhang, and S.-P. Guo, "Forecasting stock indices with back propagation neural network," *Expert Syst. Appl.*, vol. 38, no. 11, pp. 14346–14355, May 2011.

[31] R. Akita, A. Yoshihara, T. Matsubara, and K. Uehara, "Deep learning for stock prediction using numerical and textual information," in *Proc. IEEE/ACIS 15th Int. Conf. Comput. Inf. Sci. (ICIS)*, Jun. 2016, pp. 1–6.

[32] D. M. Q. Nelson, A. C. M. Pereira, and R. A. de Oliveira, "Stock market's price movement prediction with LSTM neural networks," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, May 2017, pp. 1419–1426.

[33] J.-F. Chen, W.-L. Chen, C.-P. Huang, S.-H. Huang, and A.-P. Chen, "Financial time-series data analysis using deep convolutional neural networks," in *Proc. 7th Int. Conf. Cloud Comput. Big Data (CCBD)*, Nov. 2016, pp. 87–92.

[34] J. L. Ba, K. Swersky, S. Fidler, and R. Salakhutdinov, "Predicting deep zero-shot convolutional neural networks using textual descriptions," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 4247–4255.

[35] R. P. Schumaker and H. Chen, "A quantitative stock prediction system based on financial news," *Inf. Process. Manage.*, vol. 45, no. 5, pp. 571–583, Sep. 2009.

[36] B. Weng, M. A. Ahmed, and F. M. Megahed, "Stock market one-day ahead movement prediction using disparate data sources," *Expert Syst. Appl.*, vol. 79, pp. 153–163, Aug. 2017.

[37] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner, "β-VAE: Learning basic visual concepts with a constrained variational framework," in *Proc. Int. Conf. Learn. Represent.*, vol. 3, 2017.

HUI LI is currently a Lecturer with the School of Labor Economics, Capital University of Economics and Business. Her main research interests include the areas of international trade, insurance, and agricultural economics.

YUNPENG CUI is currently the Director of the Cognitive Computing Lab, Chinese Academy of Agricultural Sciences. His research topics are machine learning, big data analytics, and NLP.

SHUO WANG (Member, IEEE) is currently a Joint Research Fellow with the Faculty of Information Technology, Monash University, and CSIRO's Data61. Before joining Monash, his Ph.D. and earlier research work was at the School of Computing and Information Systems, University of Melbourne. His main research interests include the areas of machine learning, artificial intelligence, and computer security and privacy, including security and privacy issues in systems, networking, and databases.

JUAN LIU is currently a Deputy Research Director with the Cognitive Computing Lab, Chinese Academy of Agricultural Sciences. Her research topics are machine learning, big data analytics, and economics.

JINYUAN QIN is currently pursuing the Ph.D. degree with the University of Melbourne, Melbourne, Australia. His research topics are machine learning and economics.

YILIN YANG is currently a Research Student with the Faculty of Computer Science, The University of British Columbia, BC, Canada. His research topics are deep learning and computer science.