Scientific Research Publishing

Engineering, 2025, 17(3), 241-258
https://www.scirp.org/journal/eng
ISSN Online: 1947-394X
ISSN Print: 1947-3931

Transformer with Sparse Mixture of Experts for Time-Series Data Prediction in Industrial IoT Systems

Feng Shi, Bolin Li*, Weidong Zhang

Northwest Oil Field Company, Sinopec Corp., Urumqi, China
Email: shif.xbsj@sinopec.com, *libl.xbsj@sinopec.com, zhangwd.xbsj@sinopec.com

How to cite this paper: Shi, F., Li, B.L. and Zhang, W.D. (2025) Transformer with Sparse Mixture of Experts for Time-Series Data Prediction in Industrial IoT Systems. *Engineering*, **17**, 241-258.
https://doi.org/10.4236/eng.2025.173015

Received: December 23, 2024
Accepted: March 23, 2025
Published: March 26, 2025

Copyright © 2025 by author(s) and Scientific Research Publishing Inc.
This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).
http://creativecommons.org/licenses/by/4.0/

Open Access

Abstract

With the development of the Industrial Internet of Things (IIoT) and cloud computing technologies, intelligent sensors in the field that can generate large volumes of time-series data continuously have emerged. Due to the lack of equipment and network impacts, highly distributed industrial applications cannot capture and transfer all production data to a distant cloud server in real time. Consequently, a portion of critical production data is lost, which poses the significant challenge of the timely replenishment of missing data. Employing deep learning in the cloud center for data trend prediction based on relevant data can solve this problem. The objective of this study was to develop a time-series prediction model that combines a Transformer model with a sparse Mixture of Experts (MoE). The model is designed specifically for an IIoT system that is used in oil-well operations. The proposed TransMoE prediction model combines the advantages of the MoE and the Transformer model. The MoE can effectively handle multiple subtasks while the Transformer algorithm can reflect the long-range dependency of the input data series. The proposed model was used to predict oil-well yields, and the predicted outcomes were compared with those obtained using a CNN-GRU and CNN-LSTM models, as well as the actual recorded data. The experimental results indicated that the proposed TransMoE model can significantly increase the efficiency and accuracy of oil well production sequence data prediction, with an average relative error of 6.26%, which can satisfy the requirements of enterprise data usage.

Keywords

Transformer Model, Industrial Internet of Things, Multivariate Time-Series Prediction, Self-Attention Mechanism, Mixture of Experts

1. Introduction

Oil and gas industry operations from mining and extraction to transportation require efficient and reliable techniques to handle critical situations in oil fields for more effective recovery and higher yields. For example, a wellhead, which is requisite for the reliability and efficiency of oil and gas wells, is used to manage the extraction of hydrocarbons from underground formations and prevent pressure-induced oil or gas leakages from the well. Real-time monitoring of the oil wellhead production status is crucial for safe operation and productivity and for prolonging the production life of the well. Owing to the vast distances between oil wells and their remote locations, it is not feasible to manually monitor all the production data in real time, as such an undertaking would be labor-intensive, inefficient, and time-consuming. Moreover, the conventional manual method of onsite data acquisition is inevitably subject to human errors and data forgery [1] [2]. Thus, forecasting oil well production has attracted increasing attention in the oil and gas industry to support petroleum engineers in analyzing the features of crude oil wells. However, accurately forecasting oil and gas production can be challenging and requires a large amount of data and advanced technologies [3].

The Internet of Things (IoT) is a result of the convergence of various technologies, such as wireless communication and smart sensors [4] [5]. This convergence has led to the development of the industrial IoT (IIoT), which is a promising technology that can significantly reduce costs and increase efficiency. IIoT technology provides a good alternative for monitoring and controlling various processes and operations, and it has been widely used in the oil and gas industry. An online IoT-based real-time monitoring system contributes to the timely acquisition of accurate production parameters, such as oil wellhead status data, critical oil-well production data, and data related to onsite operational activities [6] [7], while minimizing the safety risks in the oil and gas industry [8]-[10]. The vast amounts of time-series data continuously generated by smart sensors are essential for the real-time monitoring of production, as well as for intelligent analysis and decision-making [11] [12]. For example, the future trend can be predicted through the periodic pattern of the data, and sudden changes in time-series data suggest anomalies in the actual production line [13]. Moreover, the loss of temporal data due to the absence of critical metering equipment, sensor failure, or network congestion can be detected, and the data can be replaced.

Deep learning algorithms are innovative data analysis techniques that utilize advanced neural networks to analyze and predict data according to large volumes of correlated time-series data. A typical method for increasing the accuracy and efficiency of predictions is to process real-time and historical data for relational trend prediction and missing data complementation [14]-[17]. From the perspective of deep learning algorithms, various Transformer-based models have been proposed for long sequence prediction. These models effectively reduce the complexity of time and space by incorporating a sparse attention mechanism [18] [19]. Lepikhin *et al*. introduced a Transformer model with conditional computations

by replacing every other feed forward layer with a MoE layer to enhance the model capacity [20]. Zoph *et al*. addressed training process instabilities by scaling a sparse MoE layer [21]. These investigations have illustrated the potential of the Transformer model in data analysis, owing to its ability to effectively capture the extensive interrelationships between the multiple input variables [22] [23]. However, few reports exist on the use of a Transformer model with MoE block (referred to as TransMoE) for the missing data trend prediction in multivariate data.

This study proposes a TransMoE algorithm designed for a cloud-assisted IIoT system. The algorithm utilizes correlated multivariate time-series datasets to forecast petroleum production and replaces missing critical production data. This is the first hybrid TransMoE algorithm to be proposed for predicting oil well production using original time-series data and historical data sequences. The existing deep-learning algorithms based on a CNN along, gated recurrent unit (GRU), or long short-term memory (LSTM) model can only extract local features from the original data. In contrast, the TransMoE algorithm is capable of extracting long-range dependencies from raw historical data without extensive prior knowledge. It can reveal data associations over time and accurately predict future data trends without complete knowledge of the raw data. The prediction accuracy of the proposed model was evaluated using a real industrial dataset. The results indicated that the proposed model achieved time-series data prediction with an average relative error of 6.26%.

The remainder of this paper is organized as follows. Section 2 reviews previous research on time-series data prediction. Section 3 presents the cloud-assisted IIoT system. In Section 4, the proposed hybrid TransMoE model and its application to missing-data prediction based on historical data are described. The performance evaluation of the proposed model is presented in Section 5. Finally, Section 6 provides a summary of the paper and presents conclusions.

2. Related Work

This section reviews previous research that is relevant to our study. IIoT has been widely used in various applications, primarily because of the increased frequency and efficiency of data collected from the field perception layer, which allows the status of specific equipment to be monitored over time. Identifying missing data or unusual conditions monitored by sensors is often necessary. This process is designed to detect defective equipment, detect quality problems, and alert supervisors to abnormal changes [2]. Because of the massive and real-time nature of time-series data, using historical data to predict future trends and complete missing data requires a large amount of computing and storage resources and must be implemented in the cloud. Key challenges in the oil industry include integrating in situ data collected by intelligent sensors on remote terminal units (RTUs) and real-time analytics in the cloud [15] [17]. The heterogeneous sensed data are generated by different sensors and data communication protocols.

Researchers have studied the integration of heterogeneous data, *i.e*., the combi-

nation of data produced by various sensors or devices. Additionally, researchers have investigated various deep-learning algorithms for discovering hidden patterns in time-series data, predicting future trends, and completing missing data using historical data. Deep learning has considerable potential as a comprehensive data-driven analytics method for making predictions and learning temporal contextual information [20] [21]. Researchers have successfully applied neural networks in the oil and gas industry to forecast oil well production and predict time-series data [3] [14] [22].

Deep learning methods excel at learning useful representations of multidimensional data features and their nonlinear interactions from high-dimensional raw data. However, multiple linear regression algorithms [23], e.g., the CNN-LSTM [24], CNN [25], and GRU [26] models, can only focus on the short- and medium-term characteristics of the original sequence. The simulation performance for the correlation characteristics of high-dimensional data has not been fully validated. Variations in the production of a single oil well are influenced by many factors, such as reservoir properties and stimulation measures. To achieve accurate production forecasting, it is necessary to consider each element comprehensively and thoroughly and to fully exploit the data features that affect production variability. Compared with traditional production prediction methods, deep-learning methods have a solid nonlinear fitting capability and higher computational efficiency, giving them considerable potential for application in production prediction. For single-well production prediction with many influencing factors, long periods, and parallel sequences. However, when the number of features increases, it is difficult to extract high-dimensional spatial and temporal information, and the prediction accuracy is limited.

Recently, self-attention mechanisms have exhibited remarkable capabilities in sequential data, such as natural language processing and computer vision. In a supply chain, a slight modification to the original transformer was used to forecast customer sales at the day, store, and item levels according to time-series data [27]. LogTrans introduces convolutional self-attention by generating queries and keys with causal convolution and proposes sparse attention to incorporate the local context into the attention mechanism [28]. Informer extends Transformer with ProbSparse attention by halving the cascading-layer input and efficiently handles extremely long input sequence predictions [17]. Lepikhin *et al*. introduced a Transformer model with conditional computations by re-placing every other feed forward layer with a MoE layer to enhance the model capacity [18]. Zoph *et al*. addressed training process instabilities by scaling a sparse MoE layer [19]. These investigations have illustrated the potential of the Transformer deep learning algorithm in data analysis, owing to its ability to effectively capture the extensive interrelationships between the multiple input variables [20] [29].

In this paper, a new oil well yield prediction algorithm based on the TransMoE model using the self-attention mechanism is proposed. When the standard attention mechanism for time-series data is combined with the MoE, it relies heavily

on the multitasking features of the MoE operations. Multivariate trends in the raw data are extracted by using different expert processing submodules. This results in additional features when calculating attention scores that can reflect the short- and long-range dependencies of the input data sequences in the Transformer part. The proposed TransMoE algorithm can be used for sequential data prediction in cloud-assisted IIoT systems. In the TransMoE model, a multihead attention mechanism is used as a sequence encoder to calculate attention scores to obtain a more complete temporal dependence compared with standard self-attention. To evaluate the effectiveness of our model, it was compared with the CNN-GRU and CNN-LSTM models. Extensive empirical results for a real industry dataset indicated that the model outperformed the existing methods.

3. Cloud-Assisted IIoT System

In cloud-based IIoT applications, the high heterogeneity of sensed data from various intelligent sensors and the fusion between other measured data should be considered. Research on real-time data analysis for oil and gas wells requires multifunctional intelligent sensor nodes that can withstand harsh environments, which inevitably produce missing data.

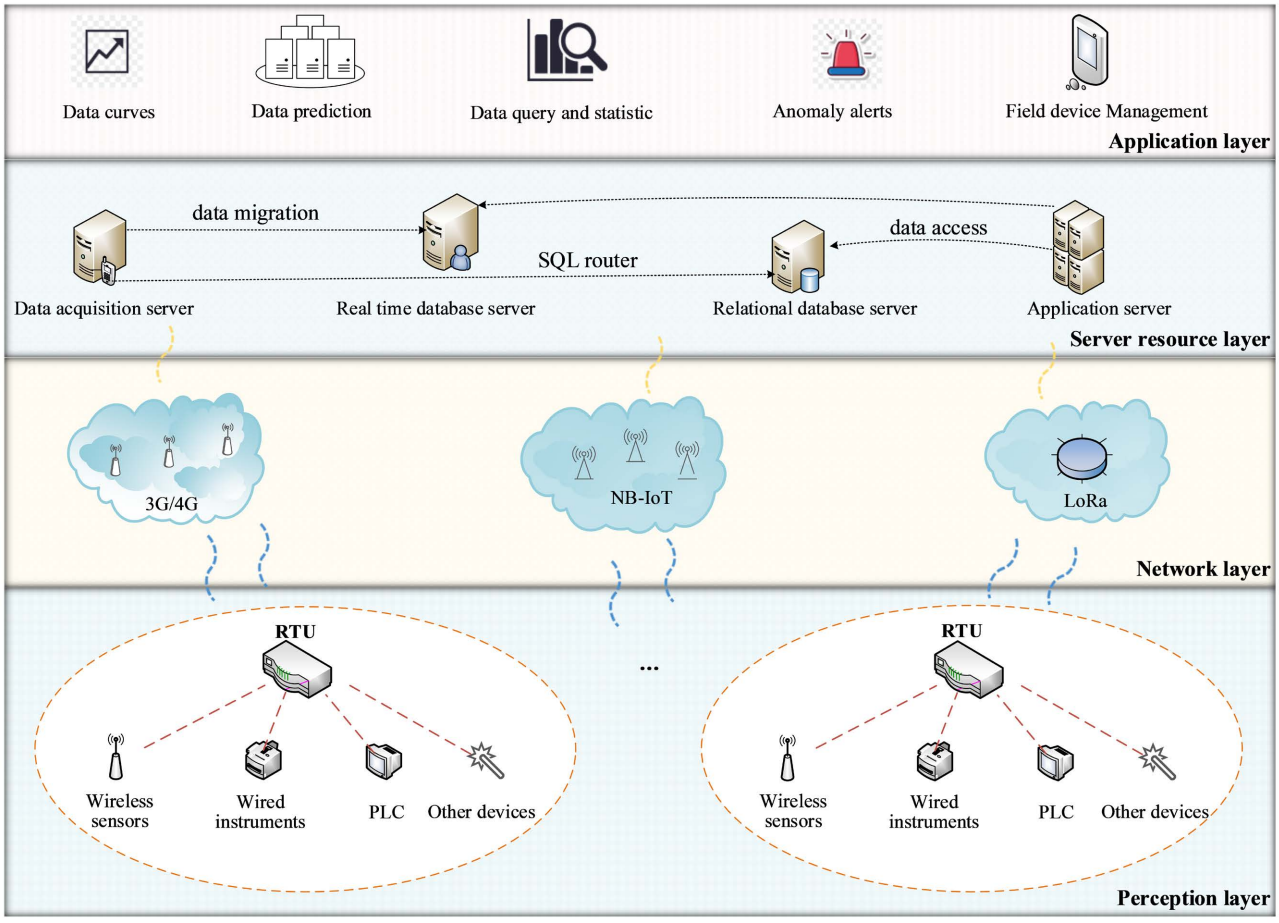

Figure 1. Architecture of the cloud-assisted IIoT system for the oil and gas industry.

Therefore, we present a system in which data are obtained from oil wells through an integrated RTU compatible with heterogeneous smart sensors and communication protocols. A deep-learning method is implemented in the application layer to detect missing sensory data, predict and complete the missing data. The developed IoT system is useful for isolated industrial applications in remote desert areas. An overview of the four-tier cloud-assisted IIoT system is presented in Figure 1. Data collected from sensors are sent to the RTU only when the request message is received. The network layer bridges the perception layer and the server resource layer. The TransMoE algorithm is embedded in the application layer for future data trend prediction and completion of missing data.

3.1. Smart Perception and Network Layer

The smart perception layer is primarily used for data collection and consists of various sensors for different parameters, as shown in Table 1. Pressure and temperature are monitored wirelessly, whereas other measurements are performed using wired instruments. The diverse data collected from various sensors are transmitted through different communication protocols in the physical layer. The network layer performs the essential functions of data transmission. A sink node is used in the network layer as a gateway that transmits the protocol packet to the remote cloud server in a transparent manner. As shown in Table 1, in the oil industry, typical heterogeneous data include pressure, temperature, and oil-well production data collected at the wellhead, along with voltage and current data from pumps.

Table 1. Monitoring data items.

Parameters	Units	Parameters	Units
Oil-well production	tons	Displacement of pump	m^3
Oil tubing pressure	MPa	Stroke of pump	m
Casing pressure	MPa	Frequency of pump	min
Pipeline pressure	MPa	Voltage	V
Oil wellhead temperature	˚C	Current (up)	A
Liquid production	tons	Current (down)	A
Gas production	m^3	Electricity consumption	kWh
Depth of pump	m	Weight of blend liquid	tons

3.2. Service Resource and Application Layer

The service resource layer composed of different servers is deployed to process and analyze sensed data. It includes a data acquisition server, a real-time database server, a relational database server, and an application server. Each data acquisition server receives real-time data from hundreds of oil wells and dump data into the relational database. Moreover, hourly data are stored in the relational database, which is used for data queries in applications with low real-time require-

ments. The application server performs real-time data processing and high-performance computing, which uses deep-learning algorithms to process, analyze, and compute data for trend prediction and automatic replenishment of abnormally missing data.

4. Proposed Transformer Model with Sparse Mixture of Experts

The proposed TransMoE model fuses a MoE block and a Transformer with a self-attention mechanism for time-series data prediction. The Transformer model based on the encoder-decoder architecture includes a multi-head attention module and feedforward neural network layer. A residual connection and layer normalization are used in the encoder and decoder sections to prevent model degradation, as shown in Figure 2. The encoder generates the vector corresponding to the input sequence, and the decoder generates the target sequence according to the output of the encoder. The encoder maps the input sequence $X=\left(X_1, X_2, \cdots, X_n\right)$ with n features to the linear embedding sequence data $Z=\left(Z_1, Z_2, \cdots, Z_{d\_model}\right)$ and then feeds it to a multi-head attention layer. The decoder generates the output sequence $Y=\left(Y_1, Y_2, \cdots, Y_n\right)$.

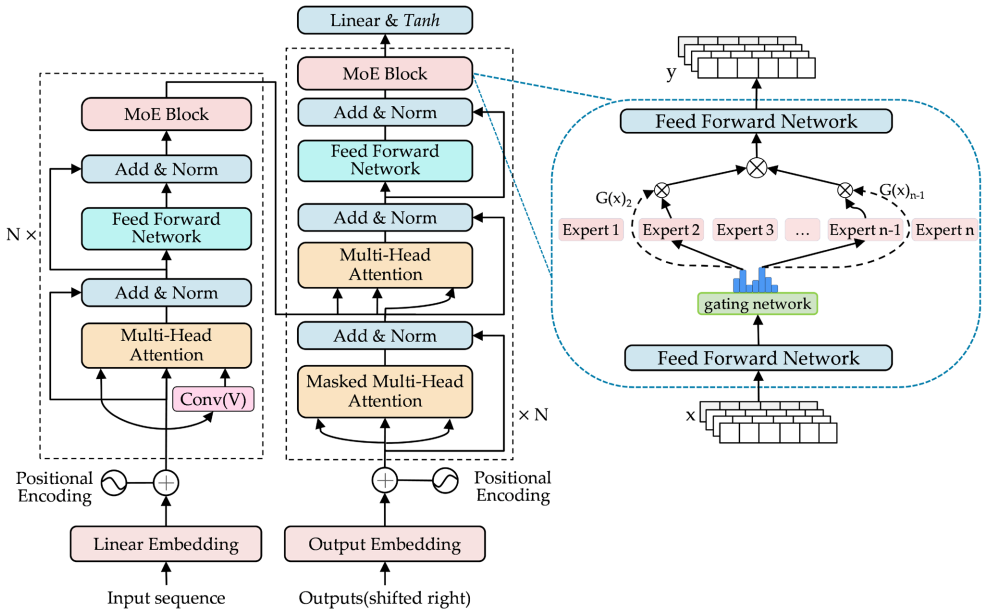

Figure 2. Architecture of the Transformer with a sparse MoE block (TransMoE).

CNN has good performance in local feature extraction, which can affect the accuracy of the predicted values. The convolution operation primarily uses feature-extraction blocks consisting of three Conv1D layers. To extract the local features of the input data, three layers of 1D convolution operations are performed on the original input data before inputting the embedding layer to enhance the data representation, with each layer having 128 neurons and a convolution kernel size of 11. However, they fail to capture long-range dependencies compared with

the Transformer model and require deeper networks with multiple layers to increase their receptive fields. Combining the efficiency and inductive prior of CNNs with the ability to capture information over long distances can yield effective architectures for sequential data applications [30] [31]. To capture the long-range dependencies of the input sequential data, the proposed model is adapted to include a sparse MoE layer to enhance the input sequence data.

The positions of the input data sequence constitute valuable information. To make the multihead attention layer aware of the sequence order, information about the relative positions of the tokens in the sequence must be injected. The fixed sine and cosine functions' positional encoding is used to identify the position information. The position encoding was implemented as follows:

$$\begin{aligned} PE_{(pos,2i)} &= \sin\left(pos/10000^{2i/d\_model}\right) \\ PE_{(pos,2i+1)} &= \cos\left(pos/10000^{2i/d\_model}\right), \end{aligned} \tag{1}$$

where pos represents the position of the embedding tokens, i is the embedding depth index in the range of $[0, d\_model]$, and d_model represents the embedding depth. The values generated by the sine and cosine functions are concatenated pairwise and added to the embedding of the input sequence.

The multihead attention layer is the main part where attention scores are calculated, and the attention scores of the input sequence are modeled using a self-attention mechanism based on three main concepts: the query vector Q, the key vector K, and the value vector V. These three pieces represent an analogy to information-retrieval systems where a query is used to search for the matching key (or the most similar one) and retrieve its value. There are many different similarity functions that can be used. The scaled dot product is a similarity function, given that it is scaled so that sequences of different lengths can be easily compared. A single sequence query searches potential relationships by finding similarities in the sequence through keys. Comparing the query and key pairs gives attention weight to the value. The interaction between the attention weights and values determines how much focus to place on other parts of the sequence while representing the current sequential data. The query, key, and value matrices are calculated by multiplying the input sequence X by three different weight matrices: W^Q, W^K, and W^V.

$$Q = XW^Q, K = XW^K, V = XW^V \tag{2}$$

In Equation (2), Q, K, and V represent the query, key, and value, respectively. The multi-head attention calculations are implemented via scaled dot-product attention, as shown in **Figure 2**. Scaled dot-product attention is expressed by Equation (3). The weight is calculated through the query and key, and it is used to obtain a weighted sum of values.

$$Attention(Q, K, V) = Softmax\left(\frac{QK^{\mathrm{T}}}{\sqrt{d_k}}\right)V \tag{3}$$

here, d_k represents the dimension of the key vector sequence. The i^{th} attention

head can be calculated as follows:

$$\begin{aligned} head_i &= Attention\left(QW_i^Q, KW_i^{QK}, VW_i^V\right) \\ MultiHead\left(Q,K,V\right) &= Concat\left(head_i,\cdots,head_n\right)W^O, \end{aligned} \tag{4}$$

where W_i^Q, $W_{,i}^K$, and W_i^V denote the linear transformations of Q, K, and V, respectively, of the i^{th} attention head. The parametric matrices $W_i^Q \in \mathbb{R}^{d\_model\times d_k}$, $W_i^K \in \mathbb{R}^{d\_model\times d_k}$, $W_i^V \in \mathbb{R}^{d\_model\times d_k}$, and $W_i^O \in \mathbb{R}^{nd_v\times d\_model}$ are linear projection parameters, and n represents the number of heads in the multi-head attention. d_k and d_v represent the dimensions of the key and query, respectively. *Concat* denotes the concatenation operation, and W^O denotes the linear transformation of concatenated outputs. The multi-head attention model can focus on the different representation spaces and simulate multiple levels of detailed information.

The feedforward layer comprises two linear transformations and a rectified linear unit activation function. The computation of the feedforward network is positionwise, and the weights of the linear transformations of the different steps are identical. The formula of the feedforward network is as follows:

$$FFN\left(x\right) = \max\left(0, xW_1 + b_1\right)W_2 + b_2. \tag{5}$$

The last two layers in the decoder are an affine transformation layer, which consists of a learnable scaling factor and a bias factor, and a multilayer perceptron, which consists of two linear transformations. The affine transformation layer and the multilayer perceptron are sequentially used for the trend prediction of the sequential data. The corresponding formulas are as follows:

$$\begin{aligned} Aff_{\alpha,\beta}\left(x\right) &= Diag\left(\alpha\right)x + \beta \\ MLP\left(x\right) &= \max\left(0, xW_1 + b_1\right)W_2 + b_2 \end{aligned} \tag{6}$$

where W_i, $Diag\left(\alpha\right)$, b_i, and β are linearly learnable weights and biases, respectively.

The last layer in the decoder is a sparse MoE block, which contains a set of n expert networks, $E_1,\cdots,E_n$, and a gating network, G whose output is a sparse n-dimensional vector. The experts are neural networks in which the experts accept the same-sized inputs and produce the same-sized outputs, each with their own parameters. For a given input x, the output of the gating network is denoted as $G\left(x\right)$, and the output of the i-th expert network is denoted as $E_i\left(x\right)$. The output y of the MoE module can be written as follows in (7) and (8).

$$\begin{aligned} y &= \sum_{i=1}^{n} G\left(x\right)_i E_i\left(x\right) \\ G\left(x\right) &= Softmax\left(KeepTopK\left(H\left(x\right),k\right)\right) \\ H\left(x\right)_i &= \left(x\cdot W_g\right)_i + SoftmaxNormal\left(\ \right)\cdot Softplus\left(\left(x\cdot W_{noise}\right)_i\right) \end{aligned} \tag{7}$$

$$KeepTopK\left(v,k\right)_i = \begin{cases} v_i & \text{if } v_i \text{ is in the top } k \text{ } e; \text{element of } v \\ -\infty & \text{otherwise} \end{cases} \tag{8}$$

The *Softmax* gating network has two components: sparsity and noise. Before

taking the *Softmax* function, we add tunable Gaussian noise and then retain only the top k values, setting the rest to minus infinity. Sparsity serves to save computations, while this form of sparsity creates discontinuities in the output of the gating function. The noise term helps with load balancing.

5. Evaluation and Discussion

We evaluated the performance of missing-data prediction based on the TransMoE model and the generated dataset. The prediction results were compared with those of the CNN-GRU and CNN-LSTM models and the actual data to evaluate their accuracy. The two baseline prediction models, *i.e.*, CNN-GRU and CNN-LSTM, included convolutional blocks, each consisting of a 3D convolutional layer with 128 hidden units and a kernel size of 11. In the comparison experiments, the models utilized the same time-series oil wellhead production data from the oil and gas industry.

5.1. Hybrid CNN-GRU Model

We developed a deep-learning model incorporating a multilayer CNN-GRU model and used it to evaluate the missing-data prediction performance of the TransMoE model [32] [33]. The architecture of the CNN-GRU model is shown in **Figure 3**, which consists of two main neural network models. With regard to the structure and parameters, the GRU is superior to the LSTM neural network. It can optimize the computation of hidden states in recurrent neural networks, where the reset gate controls how much information from the previous state is written to the current candidate set and discards historical information irrelevant to the prediction. The update gate can control the degree to which the state information of the previous moment is brought into the current state.

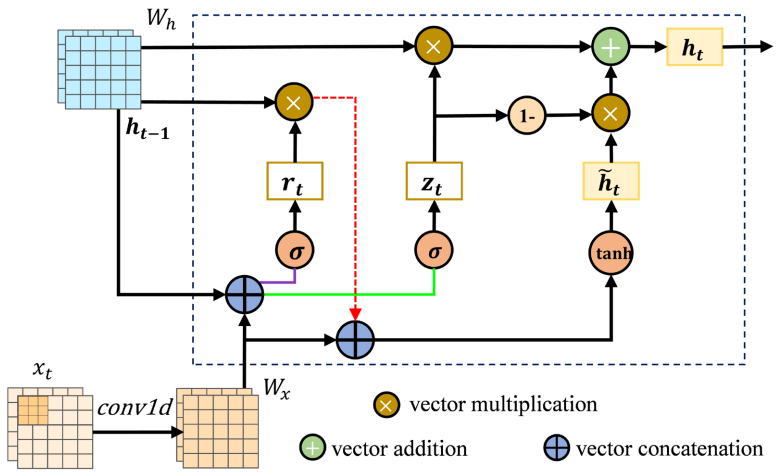

Figure 3. Proposed CNN-GRU model-based framework for evaluation of the prediction.

The GRU is used to extract the data characteristics of time-series data, and its architecture consists of a series of neuronal cells, each containing two gates, a reset

gate r_t and an update gate z_t and one hidden state layer. The hidden state layer is a crucial variable that carries information from the previous step. The gate cells in the interaction layer can partially remove the state of the previous step and add new information to the current step according to the hidden state of the previous step and the input of the current step. The corresponding formula is as follows:

$$\begin{aligned} r_t &= \sigma\left(W_r x_t + U_r h_{t-1} + b_r\right) \\ z_t &= \sigma\left(W_z x_t + U_z h_{t-1} + b_z\right) \\ \tilde{h}_t &= \tanh\left(W_h x_t + U_h\left(r_t \odot h_{t-1}\right) + b_h\right) \\ h_t &= z_t \odot h_{t-1} + \left(1 - z_t\right) \odot \tilde{h}_t \end{aligned} \tag{9}$$

$$\sigma(x) = \frac{1}{1+\mathrm{e}^{-x}}. \tag{10}$$

In Equation (9), σ is a sigmoid function given by Equation (10). x_t denotes the feature vector at timestep t. h_{t-1} denotes the previous state at time $t-1$. W_r, W_z, W_h, U_r, U_z, and U_h and b_r, b_z, and b_h are weight matrices and deviation vectors, respectively, which contribute to the linear transformations of x_t and h_{t-1}. The update gate z_t adds this linearly transformed information and inputs it to the sigmoid activation function, which can compress the result between 0 and 1. The other gate (r_t) is the reset gate, which is used to determine how much the candidate state at the current moment will inherit from the previous moment. $\otimes$ denotes element multiplication, h_t denotes the hidden state, and $\tilde{h}_t$ denotes the candidate hidden state.

5.2. Dataset

To evaluate the efficiency and accuracy of the proposed TransMoE model for time-series data prediction, we selected 16 critical parameters from oil wells located in the most remote areas for testbed scenarios. Sensor nodes collected these data at different points in the perception layer in the cloud-assisted IIoT system. The dataset was recorded with a 5-min resolution over approximately 45 d. It included 16 parameters related to pressure, temperature, pump information, liquid production, oil-well production, and voltage values. All 16 parameters were normalized with respect to the minimum and maximum values to generate the graph shown in **Figure 4**. The bottom half of this figure presents a comprehensive graph of the last data series of length 256. Consequently, the collected data included a large number of records. However, most of the nodes in the dataset had numerous missing or corrupted records. **Figure 4** shows the profiles of the selected parameters over a span of 45 d.

As mentioned in Section 2, oil well yields are closely related to the production parameters, such as the pressure, temperature, pump information, liquid production, and voltage values. Accordingly, the input data included 15 parameters, such as pressure, temperature, and pump information, and the output data were the oil-well yields. The dataset was divided into a training set, validation set, and test set according to the input and output sequences. One stride of raw data was used

to generate a series of sequential datasets of 256 data points; thus, 2722 slices of data were created, with final data dimensions of (2722, 256, 16). A three-dimensional (3D) tensor was used to store the oil well production data with dimensions of (sample, sequence, and features). Here, "sample" refers to the sample size of the dataset, which contained 2722 observations. Because the objective of this study was to forecast the oil well production for each oil well over a certain period, we set the sequence length to 256.

To predict the oil well yields, we resampled the real-time data and processed them into daily data. Thus, we selected a data sequence without missing data to train the model for trend prediction and missing-data complementation. Details regarding the selected data are presented in Table 1. To increase the correction accuracy of the measured data, a large convolution kernel of size 11 was applied to the original data sequentially to extract the local features. The input data sequence was first used as the input to three consecutive convolutional layers. Therefore, the low-level features and distinctions among variables under the context of temporal effects were acquired through the convolution operations of filters with different properties and nonlinear activation of neurons. The obtained feature map was then passed to the transformer or GRU layer, where the multihead attention mechanism thoroughly learns the complex long-range dependencies or the examination of three effective reset and update gates. Finally, the information was projected to the output space, and the prediction results were produced.

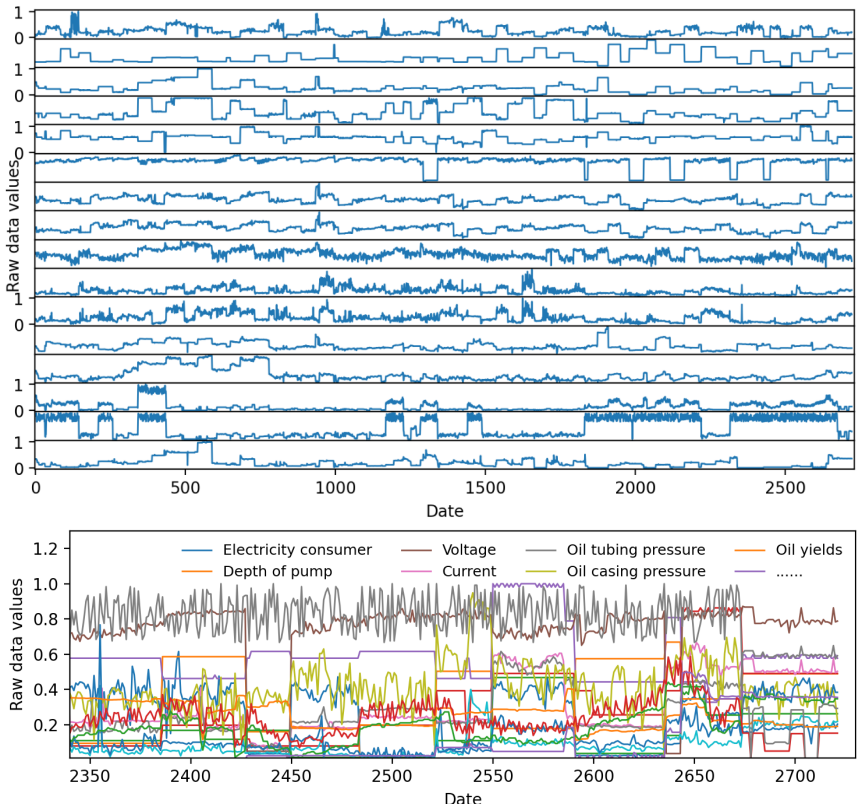

Figure 4. Raw data curves for the 16 selected parameters of the dataset.

5.2. Model Training

The proposed deep learning model using the TransMoE algorithm was implemented using Python 3.7 and PyTorch 1.9.1 and was run on a Tesla P100 GPU server. The data were split into three sets for training, validation, and testing. These datasets were normalized with respect to the minimum and maximum values in the range of 0 - 1. The first 2256 data sequences served as the training set, the intermediate 210 sequences served as the validation set, and the final sequence served as the test set. A blank 256 data sequence length between the validation and test sets ensured that it was not used in the training and testing process. The batch size was set as 50, with a total of 300 training epochs. For each training epoch, approximately 45 groups of training data were randomly selected. To achieve optimal performance, the parameters were tuned for each prediction model.

5.3. Results and Discussion

To evaluate the proposed model, three deep-learning models, *i.e.*, CNN-GRU, CNN-LSTM, and TransMoE, were employed for predicting oil-well production over a span of time. One sequence of time-series data including 256 samples (several oil well time-series data) was selected from the test dataset. The prediction results are presented in Figure 6, where the red, brown, and light green curves indicate the oil-well yields predicted by TransMoE, CNN-LSTM, and CNN-GRU, respectively. As shown, all the deep learning models can predict future trends. Moreover, the prediction results are beneficial for solving the problem of missing data due to device failure or network interruptions. The prediction performance was assessed using the bias, mean absolute error (MAE), and mean absolute percentage error (MAPE) metrics defined in Equation (11). The MAE measured the absolute difference between the actual and predicted values for the deep-learning models.

$$\begin{aligned} &\text{Bias} = y - \hat{y} \\ &\text{MAE} = \frac{1}{n}\sum_1^n |y - \hat{y}| \\ &\text{MAPE} = \frac{1}{n}\sum_1^n |y - \hat{y}| / y \times 100\% \end{aligned} \tag{11}$$

here, y represents the actual value, $\hat{y}$ represents the predicted value, and n represents the total number of samples. The MAPE indicator of the error metrics was introduced because different models have arbitrarily different error levels. The usage of MAPE normalizes the error distribution and makes the whole prediction easier to measure.

Table 2. Performance of the three deep learning models.

Model	Minimum bias	Maximum bias	MAE	MAPE	Computation time (s)
TransMoE	0.03	1.96	0.84	6.26%	25867
CNN-GRU	−1.39	2.17	0.85	7.00%	6135
CNN-LSTM	−2.39	3.09	0.50	6.84%	9212

Compared with the other two models, the TransMoE model exhibited better performance with regard to the curve morphological characteristics of the predicted values and the calculated bias, MAE, and MAPE values. For instance, the minimum and maximum biases, MAE, and MAPE of the TransMoE model were 0.03, 1.96, 0.84, and 6.26%, respectively, which were better than those of CNN-GRU and CNN-LSTM, as shown in Table 2. In particular, the CNN-LSTM model had larger minimum and maximum deviations, resulting in larger deviations in the predicted values at local points.

For the CNN-GRU and CNN-LSTM models, the minimum and maximum biases resulted in poor performance, which may have been related to the network structure. These models had three CNN layers and three GRU layers. The hidden neurons in the CNN layer extract local features from the input sequence and feed them to GRU or LSTM layers, which store information from the previous step. The hidden nodes in the GRU or LSTM layer transmit information forward, and it eventually reaches the output layer.

Figure 5 shows the distribution of bias errors for eight different oil wells. The errors for the different wells are shown side-by-side to illustrate how they vary across the oil wells. As indicated by the results, the TransMoE model had the smallest bias variation range, along with the smallest local minimum and maximum values. This implies that the TransMoE model achieved the best fit to the actual data and most accurately forecasted the data's long-term trend. Although the CNN-GRU and CNN-LSTM models accurately approximated the actual values in certain periods, there were large deviations in the predicted values between different oil well data variations. Moreover, the bias in the predicted values was large, suggesting that these two models can simulate the short-term characteristics of the data but have difficulty accurately predicting long-term trends because of the limitations of the algorithms.

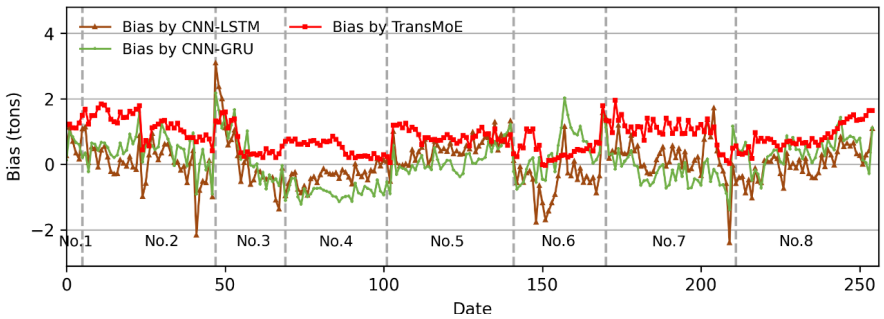

Figure 5. Distribution of prediction errors at different times for the three models and for the eight different wells.

Figure 6 presents a comparison of the actual values and the predicted values of the three deep learning models. As shown, the TransMoE model had the highest prediction accuracy among the models. Comparing the three models horizontally and vertically revealed that the prediction accuracies of the TransMoE model were higher than those of the CNN-LSTM and CNN-GRU models.

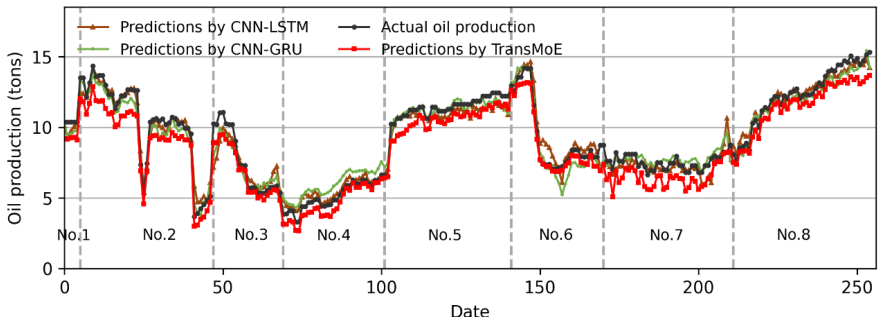

Figure 6. Predicted values of the three deep-learning models and the actual values.

A visual analysis of the bias between the predicted oil well production data and the actual data revealed that the proposed model can accurately predict oil well production levels for different wells. The predicted data changed in accordance with the actual oil well production data in a predicted data series, and the local maximum and minimum values were perfectly fitted. Larger variations in the actual data corresponded to a higher accuracy of the predicted trend. This indicates the capability of the TransMoE model for capturing the long-term dependencies of time-series data. In comparison, CNN-GRU and CNN-LSTM were slightly inferior in fitting the long-term trend patterns of the sequential data.

The findings indicate that the proposed algorithm is promising for solving time-series problems—particularly in long-term forecasting tasks. This is because it accurately characterizes long-interval time-series datasets better than conventional methods. The proposed model is useful for the oil and gas industry because of its ability to handle local features as well as long-term dependencies caused by the characteristics of time-series data, despite having a higher computational cost than CNN-GRU and CNN-LSTM.

Our simulation results indicated that the CNN-GRU model performed at a comparable level to the CNN-LSTM model for the test set. The gated neural network model can only focus on the short-term characteristics of the data and cannot learn the long-term dependencies of the data, hindering the algorithm's ability to predict long-term trends.

6. Conclusion

Forecasting production for a single well can be challenging, as oil production lacks consistency and tends to fluctuate even on consecutive days. We developed a TransMoE model for predicting oil well yields in the oil and gas industry. The model was tested using real industrial datasets in an IIoT system. As the input data were nonlinear, they were normalized using a standard min-max scalar before being fed into various training processes. Several deep learning models were investigated, and the Transformer deep learning model with a sparse MoE block was optimized. The proposed model outperformed other deep-learning models, indicating that it can be used in practical applications to predict time-series data. To assess the prediction accuracy, the differences between the predicted and ac-

tual data were evaluated using the bias, MAE, and MRE. The experimental results indicated that deep learning algorithms can achieve high prediction accuracy and can be used in practical applications.

Acknowledgements

We would like to thank Editage (http://www.editage.cn/) for English language editing.

Conflicts of Interest

The authors declare that they have no conflicts of interest in this work. We declare that we do not have any commercial or associative interest that represents a conflict of interest in connection with the work submitted.

References

[1] Obodoeze, F.C. (2012) Wireless Sensor Network in Niger Delta Oil and Gas Field Monitoring: The Security Challenges and Countermeasures. *International Journal of Distributed and Parallel systems*, **3**, 65-77. https://doi.org/10.5121/ijdps.2012.3606

[2] Shi, F., Yan, L., Zhao, X. and Xian-Ke Gao, R. (2022) Machine Learning-Based Time-Series Data Analysis in Edge-Cloud-Assisted Oil Industrial IoT System. *Mobile Information Systems*, **2022**, Article ID: 5988164. https://doi.org/10.1155/2022/5988164

[3] Al-Shabandar, R., Jaddoa, A., Liatsis, P. and Hussain, A.J. (2021) A Deep Gated Recurrent Neural Network for Petroleum Production Forecasting. *Machine Learning with Applications*, **3**, Article ID: 100013. https://doi.org/10.1016/j.mlwa.2020.100013

[4] Zhang, F., Liu, M., Zhou, Z. and Shen, W. (2016) An Iot-Based Online Monitoring System for Continuous Steel Casting. *IEEE Internet of Things Journal*, **3**, 1355-1363. https://doi.org/10.1109/jiot.2016.2600630

[5] Wanasinghe, T.R., Gosine, R.G., James, L.A., Mann, G.K.I., de Silva, O. and Warrian, P.J. (2020) The Internet of Things in the Oil and Gas Industry: A Systematic Review. *IEEE Internet of Things Journal*, **7**, 8654-8673. https://doi.org/10.1109/jiot.2020.2995617

[6] Aalsalem, M.Y., Khan, W.Z., Gharibi, W. and Armi, N. (2017) An Intelligent Oil and Gas Well Monitoring System Based on Internet of Things. 2017 *International Conference on Radar, Antenna, Microwave, Electronics, and Telecommunications* (*ICRAMET*), Jakarta, 23-24 October 2017, 124-127. https://doi.org/10.1109/icramet.2017.8253159

[7] Gao, M.J., Xu, J., Tian, J.W. and Zhang, F. (2008) Zigbee Wireless Mesh Networks for Remote Monitoring System of Pumping Unit. 2008 7*th World Congress on Intelligent Control and Automation*, Chongqing, 25-27 June 2008, 5901-5905. https://doi.org/10.1109/wcica.2008.4592834

[8] Trappey, A.J.C., Trappey, C.V., Hareesh Govindarajan, U., Chuang, A.C. and Sun, J.J. (2017) A Review of Essential Standards and Patent Landscapes for the Internet of Things: A Key Enabler for Industry 4.0. *Advanced Engineering Informatics*, **33**, 208-229. https://doi.org/10.1016/j.aei.2016.11.007

[9] Gubbi, J., Buyya, R., Marusic, S. and Palaniswami, M. (2013) Internet of Things (IoT): A Vision, Architectural Elements, and Future Directions. *Future Generation Computer Systems*, **29**, 1645-1660. https://doi.org/10.1016/j.future.2013.01.010

[10] Borgia, E. (2014) The Internet of Things Vision: Key Features, Applications and Open

Issues. *Computer Communications*, **54**, 1-31.
https://doi.org/10.1016/j.comcom.2014.09.008

[11] Wu, Y., Liu, Y., Ahmed, S.H., Peng, J. and Abd El-Latif, A.A. (2020) Dominant Data Set Selection Algorithms for Electricity Consumption Time-Series Data Analysis Based on Affine Transformation. *IEEE Internet of Things Journal*, **7**, 4347-4360. https://doi.org/10.1109/jiot.2019.2946753

[12] Heidari Kapourchali, M. and Banerjee, B. (2018) Unsupervised Feature Learning from Time-Series Data Using Linear Models. *IEEE Internet of Things Journal*, **5**, 3918-3926. https://doi.org/10.1109/jiot.2018.2845340

[13] Akbar, A., Khan, A., Carrez, F. and Moessner, K. (2017) Predictive Analytics for Complex IoT Data Streams. *IEEE Internet of Things Journal*, **4**, 1571-1582. https://doi.org/10.1109/jiot.2017.2712672

[14] Sagheer, A. and Kotb, M. (2019) Time Series Forecasting of Petroleum Production Using Deep LSTM Recurrent Networks. *Neurocomputing*, **323**, 203-213. https://doi.org/10.1016/j.neucom.2018.09.082

[15] Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., *et al*. (2021) Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting. *Proceedings of the AAAI Conference on Artificial Intelligence*, **35**, 11106-11115. https://doi.org/10.1609/aaai.v35i12.17325

[16] Wu, H., Xiao, B., Codella, N., Liu, M., Dai, X., Yuan, L., *et al*. (2021) CvT: Introducing Convolutions to Vision Transformers. 2021 *IEEE/CVF International Conference on Computer Vision* (*ICCV*), Montreal, 10-17 October 2021, 22-31. https://doi.org/10.1109/iccv48922.2021.00009

[17] Lepikhin, D., Lee, H.J., Xu, Y., *et al*. (2020) GShard: Scaling Giant Models with Conditional Computation and Automatic Sharding. arXiv: 2006.16668.

[18] Barret, Z., Bello, Z., Kumar, S., *et al*. (2022) ST-MoE: Designing Stable and Transferable Sparse Expert Models. arXiv: 2202.08906.

[19] Stassin, S., Corduant, V., Mahmoudi, S.A. and Siebert, X. (2024) Explainability and Evaluation of Vision Transformers: An In-Depth Experimental Study. *Electronics*, **13**, Article 175. https://doi.org/10.3390/electronics13010175

[20] Yan, F., Li, S., Zhou, Z. and Shi, Y. (2024) A Residual Network with Efficient Transformer for Lightweight Image Super-Resolution. *Electronics*, **13**, Article 94. https://doi.org/10.3390/electronics13010194

[21] Minoli, D., Sohraby, K. and Occhiogrosso, B. (2017) IoT Considerations, Requirements, and Architectures for Smart Buildings—Energy Optimization and Next-Generation Building Management Systems. *IEEE Internet of Things Journal*, **4**, 269-283. https://doi.org/10.1109/jiot.2017.2647881

[22] Verma, S., Kawamoto, Y., Fadlullah, Z.M., Nishiyama, H. and Kato, N. (2017) A Survey on Network Methodologies for Real-Time Analytics of Massive IoT Data and Open Research Issues. *IEEE Communications Surveys & Tutorials*, **19**, 1457-1477. https://doi.org/10.1109/comst.2017.2694469

[23] Seren, H.R., Buzi, E., Al-Maghrabi, L., Ham, G., Bernero, G. and Deffenbaugh, M. (2018) An Untethered Sensor Platform for Logging Vertical Wells. *IEEE Transactions on Instrumentation and Measurement*, **67**, 798-803. https://doi.org/10.1109/tim.2017.2774183

[24] Akbar, A., Kousiouris, G., Pervaiz, H., Sancho, J., Ta-Shma, P., Carrez, F., *et al*. (2018) Real-time Probabilistic Data Fusion for Large-Scale IoT Applications. *IEEE Access*, **6**, 10015-10027. https://doi.org/10.1109/access.2018.2804623

[25] Wang, Y., Shen, Y., Mao, S., Chen, X. and Zou, H. (2019) LASSO and LSTM Integrated Temporal Model for Short-Term Solar Intensity Forecasting. *IEEE Internet of Things Journal*, **6**, 2933-2944. https://doi.org/10.1109/jiot.2018.2877510

[26] Zhang, Y., Thorburn, P.J., Xiang, W. and Fitch, P. (2019) SSIM—A Deep Learning Approach for Recovering Missing Time Series Sensor Data. *IEEE Internet of Things Journal*, **6**, 6618-6628. https://doi.org/10.1109/jiot.2019.2909038

[27] Wang, X., Pi, D., Zhang, X., Liu, H. and Guo, C. (2022) Variational Transformer-Based Anomaly Detection Approach for Multivariate Time Series. *Measurement*, **191**, Article ID: 110791. https://doi.org/10.1016/j.measurement.2022.110791

[28] Morid, M.A., Sheng, O.R.L. and Dunbar, J. (2023) Time Series Prediction Using Deep Learning Methods in Healthcare. *ACM Transactions on Management Information Systems*, **14**, 1-29. https://doi.org/10.1145/3531326

[29] Basha, E., Jurdak, R. and Rus, D. (2014) In-network Distributed Solar Current Prediction. *ACM Transactions on Sensor Networks*, **11**, 1-28. https://doi.org/10.1145/2629593

[30] Piccialli, F., Giampaolo, F., Prezioso, E., Crisci, D. and Cuomo, S. (2021) Predictive Analytics for Smart Parking: A Deep Learning Approach in Forecasting of IoT Data. *ACM Transactions on Internet Technology*, **21**, 1-21. https://doi.org/10.1145/3412842

[31] Chang, X., Li, G., Xing, G., Zhu, K. and Tu, L. (2021) DeepHeart: A Deep Learning Approach for Accurate Heart Rate Estimation from PPG Signals. *ACM Transactions on Sensor Networks*, **17**, 1-18. https://doi.org/10.1145/3441626

[32] Fang, H., Liu, Y., Chen, C. and Hwang, F. (2022) Travel Time Prediction Method Based on Spatial-Feature-Based Hierarchical Clustering and Deep Multi-Input Gated Recurrent Unit. *ACM Transactions on Sensor Networks*, **19**, 1-21. https://doi.org/10.1145/3544976

[33] Vallés-Pérez, I., Soria-Olivas, E., Martínez-Sober, M., Serrano-López, A.J., Gómez-Sanchís, J. and Mateo, F. (2022) Approaching Sales Forecasting Using Recurrent Neural Networks and Transformers. *Expert Systems with Applications*, **201**, Article ID: 116993. https://doi.org/10.1016/j.eswa.2022.116993