





Article

Digital Twins Temporal Dependencies-Based on Time Series Using Multivariate Long Short-Term Memory

Abubakar Isah ¹, Hyeju Shin ¹, Seungmin Oh ¹, Sangwon Oh ¹, Ibrahim Aliyu ¹, Tai-won Um ^{2,*}
and Jinsul Kim ^{1,*}

¹ Department of ICT Convergence System Engineering, Chonnam National University, Gwangju 61186, Republic of Korea; abubakarisah@jnu.ac.kr (A.I.); sinhye102@jnu.ac.kr (H.S.); 216655@jnu.ac.kr (S.O.); osw0782@naver.com (S.O.); aliyu@jnu.ac.kr (I.A.)

² Graduate School of Data Science, Chonnam National University, Gwangju 61186, Republic of Korea

* Correspondence: stwum@jnu.ac.kr (T.-w.U.); jsworld@jnu.ac.kr (J.K.)

Abstract: Digital Twins, which are virtual representations of physical systems mirroring their behavior, enable real-time monitoring, analysis, and optimization. Understanding and identifying the temporal dependencies included in the multivariate time series data that characterize the behavior of the system are crucial for improving the effectiveness of Digital Twins. Long Short-Term Memory (LSTM) networks have been used to represent complex temporal dependencies and identify long-term links in the Industrial Internet of Things (IIoT). This paper proposed a Digital Twin temporal dependency technique using LSTM to capture the long-term dependencies in IIoT time series data, estimate the lag between the input and intended output, and handle missing data. Autocorrelation analysis showed the lagged links between variables, aiding in the discovery of temporal dependencies. The system evaluated the LSTM model by providing it with a set of previous observations and asking it to forecast the value at future time steps. We conducted a comparison between our model and six baseline models, utilizing both the Smart Water Treatment (SWaT) and Building Automation Transaction (BATADAL) datasets. Our model's effectiveness in capturing temporal dependencies was assessed through the analysis of the Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF). The results of our experiments demonstrate that our enhanced model achieved a better long-term prediction performance.

Keywords: temporal dependency; Digital Twins; LSTM; multivariate time series



Citation: Isah, A.; Shin, H.; Oh, S.; Oh, S.; Aliyu, I.; Um, T.-w.; Kim, J. Digital Twins Temporal Dependencies-Based on Time Series Using Multivariate Long Short-Term Memory. *Electronics* **2023**, *12*, 4187. <https://doi.org/10.3390/electronics12194187>

Academic Editors: Martin Reisslein and Franco Cicirelli

Received: 31 July 2023

Revised: 5 October 2023

Accepted: 7 October 2023

Published: 9 October 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Digital Twins connect the real and virtual worlds [1,2], offering simulations, projections, and insights that can be applied to decision-making, optimization, and maintenance tasks [3]. The Digital Twin can learn and capture the underlying patterns and dependencies of the dynamic system by examining the historical and real-time data of the multivariate time series and training an LSTM network with this data [4]. This enables it to generate precise simulations and predictions. Digital Twin is a technology that is still under development, but it has the potential to revolutionize the way we manage assets. Digital Twins are virtual representations of physical assets that can be used to simulate, monitor, and optimize the performance of those assets [5].

Temporal dependencies are the patterns and relationships that develop over time between the variables in multivariate time series data. These dependencies can include seasonality, trends, lagged relationships, sequential patterns, and other temporal structures. Temporal dependencies and multivariate time series analysis are crucial in many areas, including weather forecasting, industrial operations, and others [6]. To make effective decisions and maximize system performance, it is essential to have the ability to accurately analyze and forecast [7] the behavior of complex dynamics. Time series data can be utilized

for analyzing various phenomena, including household electricity usage, road occupancy rates, currency exchange rates, solar power generation, and even musical notation. Most of the time, the data collected consists of multivariate time series (MTS) data, such as the local power company monitoring the electricity consumption of numerous clients. Complex dynamic interdependencies between different series [6] can be significant but challenging to capture and analyze. As science and technology continuously advance, systems used by people are becoming increasingly complex.

Multivariate time series (MTS) and increasingly sophisticated data are required to explain complex systems [7,8]. The system generates multiple variables at any given time, resulting in a multivariate time series denoted by the matrix $X = \{X_1, X_2, X_3, \dots, X_m\}$, which records the values of these numerous variables at different time steps within the same period. In several areas, such as urban air quality forecasting [9], traffic prediction [10,11], the COVID-19 pandemic [12], and the industrial sector [13], it is essential to analyze the MTS data. Analysts frequently attempt to predict the future using historical data. Forecasting can be more precise when the interdependencies among distinct variables are effectively modeled. In general terms, we refer to the concurrent correlation between different variables in the MTS as spatial connections [12], the concurrent correlation between variables at different time points as temporal dependency correlation, and the concurrent correlation between different variables at different points in time as temporal linkage.

The concept of the Digital Twin represents one potential application of LSTM networks in the context of time series analysis. A Digital Twin is a virtual representation of a real-world system or process. By integrating real-time input from sensors and other sources with the capabilities of LSTM models, a Digital Twin can accurately replicate the behavior and dynamics of the physical system it represents. Long Short-Term Memory (LSTM) networks, a type of Recurrent Neural Network (RNN) explicitly designed to capture and leverage temporal dependencies, offer an effective approach for analyzing multivariate time series data. LSTM networks are particularly well-suited for modeling time series data due to their ability to handle sequences with long-term dependencies.

To build effective predictive models, identifying the appropriate lag order (the number of past observations used as inputs in a time series model) is crucial. The Autocorrelation Function (ACF) and the Partial Autocorrelation Function (PACF) are used for this purpose. Our methodology's primary contribution to this research lies in its utilization of statistical methods for identifying lags in time series data. These methods play a pivotal role in helping us to comprehend the relationships between data points at different time stamps, a critical aspect of identifying and modeling temporal dependencies.

Utilizing temporal dependencies in multivariate time series analysis through LSTM networks, in combination with the concept of Digital Twins, forms a powerful approach to understanding and predicting complex systems. These methodologies open up new avenues for optimization, proactive maintenance, and decision support across various industries, ultimately enhancing productivity, reliability, and overall performance.

The main contribution of this paper includes:

- We propose a Digital Twins scenario based on temporal connections in multivariate time series data. It aims to identify patterns, relationships, and lags among the time series data variables.
- To capture complex temporal connections and uncover long-term links in the data, the study utilizes Long Short-Term Memory (LSTM) networks to represent and analyze the multivariate time series data based on past observations.
- The lag orders between variables are identified using autocorrelation analysis, including ACF and PACF, which facilitate the determination of lag orders at each time step in the time series data. This analysis simplifies the comprehension of connections and dependencies among various variables within the feature.
- Metrics such as Mean Absolute Percentage Error (MAPE), Mean Absolute Error (MAE), and Root Mean Square Error (RMSE) were measured to demonstrate the robustness of the system. The model was tested for its ability to predict future values at various time

steps using historical observations as training data. This assessment sheds light on the efficiency and efficacy of the LSTM model in identifying temporal dependencies in the data.

Paper Structure

The structure of this paper is as follows: Section 2 explores related works on temporal dependencies in multivariate time series data. Section 3 defines a Digital Twin overview, temporal dependencies in multivariate LSTM, and the datasets used. Section 4 details the experimental results of the proposed system. Section 5 focuses on correlation analysis. Section 6 presents the conclusion and summarizes the results. Finally, we conclude with future work in Section 7.

2. Related Work

Recently, several industries, including manufacturing and the automobile industry [14], have chosen to make Digital Twin a cornerstone of their technology. Digital Twin offers data fusion and the ability to replicate physical systems [15].

In the field of processing time series data, the autoregressive model has traditionally been employed. This methodology assumes that the time series [16] under investigation exhibit a linear relationship with their past values. It predicts future values based on linear modeling of previous values, possibly with a constant term and random error. Autoregressive Integrated Moving Average (ARIMA) is the model [17] that most commonly employs the autoregressive model concept. It transforms a non-stationary time series into a stationary one [18], after which the ARIMA model is applied to represent the data. However, as the ARIMA model assumes a linear relationship between the projected value of the time series, past values, and noise, it can only be used for assessing stationary series and cannot effectively predict or address numerous complex time series.

As deep learning advances, more researchers are exploring the use of deep learning to model the problem of multivariate time series analysis. Recurrent Neural Networks (RNNs) and their variations serve as representative models for sequence-based deep learning. However, it can be challenging for these models to converge due to issues like the vanishing gradient and exploding gradient [19]. The vanishing gradient problem in RNNs has been partially addressed by LSTM, which is still utilized in many sequential models. The authors of [20] integrated LSTM with the conventional genetic algorithm to forecast time series. The genetic algorithm selected the optimal LSTM structure, which was subsequently successfully tested on time series data from the petroleum industry. In another study, ref. [18] employed LSTM for supply chain analysis and forecasting, achieving excellent results. LSTM was used to estimate the power load in a power compliance early warning system and assess several sets of time series [19,20] generated by power consumption. The authors successfully combined the random forest method with LSTM to predict the price of carbon. Furthermore, ref. [21] constructed a self-encoder network based on LSTM for forecasting daily precipitation time series data, while the authors of [22] applied LSTM to analyze historical oil well production data and make predictions.

Time series data are present in all aspects of daily life. We collect time series data by observing evolving variables produced by sensors over discrete time increments [21]. There are some examples prior to the work of [22] for knowledge discovery in temporal data. These methods mostly handle point-based events and only consider data as chronological series. As a result, the physical arrangement of events is relatively straightforward, and the expressiveness of using temporal relations such as “during” and “overlaps,” etc., is limited. In addition to the parallel and serial ordering of event sequences, when dealing with time series data for events that last across time, we may encounter other intriguing temporal patterns [23]. Examples of patterns that cannot be described as simple sequential ordering are “event A occurs during the time event B happens” and “event A’s occurrence time overlaps with that of event B and both of these events occur before event C appears.” However, it is suggested that temporal logic [24] be used to express temporal patterns

defined over categorical data. Temporal operators are utilized, including since, until, and next. Event A may always occur until Event B appears in our patterns [25]. Sequence data is typically processed using Recurrent Neural Networks (RNNs), which are a crucial type of neural network. However, vanishing- or exploding-gradient issues, which cannot resolve the long-term reliance problem, severely affect RNNs. Long Short-Term Memory (LSTM), a particular type of RNN, adds a gate mechanism and can prevent back-propagated errors from disappearing or blowing up [23]. In contrast to other approaches, StemGNN [26] uses a novel strategy to capture both inter-series correlations and temporal dependence simultaneously in the spectral domain.

Complex models built using Artificial Neural Networks (ANNs) and Deep Learning (DL) architectures typically struggle with issues relating to the need for large training data.

When considering the dynamic system of the Digital Twin and the Industrial Internet of Things Applications, temporal dependencies were considered to discover temporal patterns within the historical time series data based on lags and missing data. Industry 4.0 is the fourth industrial revolution, which is characterized by the integration of digital technologies into manufacturing and other industrial processes. Table 1 presents a comparative study of different methods used and their limitation between multivariate LSTM, temporal dependencies, and Digital Twins. Information serves as the vital foundation for the mass personalization concept, and cooperative, people-centered strategies form the fundamental elements for achieving a significant degree of sustainability [27]. Sensor technologies play a crucial role in Industry 4.0 (Acme Corporation, San Francisco, CA, USA) by collecting data from the physical world in real time. These data can then be used to create Digital Twins, which are virtual representations of physical systems. However, environmental factors or inherent problems may cause sensors to be faulty [28].

Some RNN, GRU, and LSTM models have been shown to be able to handle very high missing values and delays in time series data [29]. This is because they are able to learn the underlying patterns in the data even when there is a lot of missing information.

Table 1. A comparative study of multivariate LSTM, temporal dependency, and Digital Twin.

Authors	Year	Methodology	Limitations
P. G. Zhang et al. [17]	2003	The hybrid model uses the distinctive quality and power of both ANN and ARIMA models to identify various patterns.	The proposed hybrid model is not contrasted with other cutting-edge time series models in the paper.
R. Vohra et al. [19]	2015	The DBN-LSTM network is used to keep track of the temporal information.	Their work only focuses on DBN-LSTM without comparing it with other ML algorithms.
A. Sagheer et al. [20]	2019	An evolutionary algorithm is used to optimally configure DLSTM architecture.	They emphasize the significance of applying deep learning techniques to overcome the complexity and time-consuming nature of conventional forecasting methodologies.
S. Y. Shih et al. [6]	2019	Temporal Pattern Attention (TPA) is used for multivariate time series forecasting.	The method is not comprehensively compared in the paper to other state-of-the-art methodologies, which may restrict the generalizability of the findings.
P. S. Kam et al. [22]	2000	The research offers potential techniques for identifying temporal patterns in interval-based events.	The article offers fresh approaches for identifying intriguing temporal patterns in interval-based occurrences.

Table 1. *Cont.*

Authors	Year	Methodology	Limitations
K. J. Uribe et al. [21]	2020	The Unbiased Finite Impulse Response (UFIR) filtering is used for time-stamped delay and missing data.	The method proposed is based on numerical investigation.
En Fu et al. [7]	2022	A temporal attention mechanism is proposed based on (ConV-LSTM).	Since LSTM units continue to be necessary for the temporal self-attention mechanism, the model cannot be fully parallelized.
P. C. Bascones et al. [14]	2023	Digital Twin (DT) is introduced and Kernel Principal Component Analysis (KPCA) and One-Class Support Vector Machines (OCSVM) are used.	The methodology’s execution might need a lot of processing power, technical know-how, and data analysis.
Y. Lian et al. [30]	2023	MTAD-GAN (Multivariate Time Series Data Anomaly Detection with GAN).	The suggested method is not contrasted in the research with cutting-edge anomaly detection techniques created especially for multivariate time series data for large scale applications.

3. Digital Twin and Temporal Dependency on Multivariate LSTM

Let $T \in \mathbb{R}^{n \times t}$ represent the number of exogenous series n and the total length of time t . Suppose the exogenous series represents a series of data, with the k th series data denoted as $X_m = \{X_{1m}, X_{2m}, X_{3m}, \dots, X_{nm}\}$ to represent all features at time m . The total length of time t for the target series data can be expressed as

$$T \in \mathbb{R}^{n \times t}$$

where

- T is the tensor representing the exogenous and target series data, with dimensions n (number of exogenous series) \times t (total length of time).
- n is the number of exogenous series.
- t is the total length of time.

The exogenous series are represented by the tensor $x \in \mathbb{R}^{n \times t \times k}$, where k is the number of features in each exogenous series. The target series is represented by the tensor $y \in \mathbb{R}^t$.

The Digital Twin uses the historical time series data (tensor T) to analyze the pattern and their relationships with machine failures. It employs machine learning models, such as LSTM, to capture temporal dependencies and predicts based on data $(X_{1m}, X_{2m}, X_{3m}, \dots, X_{nm})$. The Digital Twin can provide early warnings when it detects abnormal patterns that indicate potential machine failures.

3.1. Notation and Problem Formulation

To anticipate the target series accurately, it is imperative to discern temporal dependencies within time series data when dealing with Digital Twins. In this context, Digital Twins serve as virtual counterparts of real-world physical systems, facilitating efficient monitoring, prediction, and control of the physical system itself. This involves capturing correlations and patterns among variables as they evolve over time.

The Digital Twin temporal dependency on multivariate LSTM is captured by the following equation:

$$y_t = f(T_{t-1}, y_{t-1}),$$

where f is the multivariate LSTM function. This equation states that the target series at time t depends on the exogenous and target series data at time $t - 1$.

In the context of multivariate time series analysis, let us consider a scenario in which we have a set of variables represented as $x_i \in \mathbb{R}^n$. The objective is to predict the value $x_{t-1} + \Delta$ at the time t , where Δ represents a constant limit that defines a distinct task. For clarity, we denote the predicted value as $y_{t-1} + \Delta$, which is expected to match the actual ground truth value $x_{t-1} + \Delta$. To achieve this prediction, we only require the input data $\{x_{t-w}, x_{t-w+1}, \dots, x_{t+1}\}$ for each task, with w representing the window size. This approach is commonly employed because it is assumed that there is no meaningful information available prior to the defined window, and as a result, the input remains fixed [31,32].

$$\hat{y}_{w+1} = f(y_1, y_2, \dots, y_w, x_1, x_2, \dots, x_w), \quad (1)$$

where $f(\cdot)$ is a nonlinear mapping function our model aims to learn.

3.2. Recurrent Neural Network (RNN)

A Recurrent Neural Network (RNN) typically defines a recurrent function, f , and calculates $h_t \in \mathbb{R}^n$ for each time step, t , as follows:

$$h_t = f(h_{t-1}, x_t) \quad (2)$$

where the kind of RNN cell utilized determines how function f is implemented.

The widely employed Long Short-Term Memory (LSTM) [33] cells have a slightly distinct recurring function:

This is defined by the equations below:

$$i_t = \text{sigmoid}(W^i H + b^i) \quad (3)$$

$$f_t = \text{sigmoid}(W^f H + b^f) \quad (4)$$

$$O_t = \text{sigmoid}(W^o H + b^o) \quad (5)$$

$$C_t = \text{sigmoid}(W^c H + b^c) \quad (6)$$

where i_t , f_t , o_t , and c_t in the equations stand for the new memory cell states, forgetting gate values, output gate values, and entry threshold values, respectively. W^i , W^f , W^o , and W^c are weight matrices and are a sigmoid function. The offset terms that correlate to b^i , b^f , b^o , and b^c are equivalent terms. M_t is the final state of the memory cell, and h_t is the final output of the memory unit.

LSTM demonstrates promising results even when dealing with noisy and incompressible input sequences. It has the ability to learn and capture temporal dependencies spanning beyond short time lags, making it a valuable tool for handling multivariate time series (MTS) data, which often present challenges for other models. Researchers have dedicated decades to studying methods for predicting the future and identifying temporal relationships among these variables. They achieve this by modeling previously observed sequences of values, which in turn aids in making more informed decisions. When it comes to predicting future occurrences, the concept of Digital Twin technology can be applied to monitor both the physical and virtual environments effectively.

3.3. Digital Twins and Model Mapping

Modern industrial management places a high priority on Digital Twins, a topic that has been extensively studied and applied in various industries. The Digital Twin is capable of providing timely and accurate simulations, among other features, allowing it to monitor, regulate, and manage the state of physical entities effectively [34]. Due to its effectiveness in industrial manufacturing, community administration, and other sectors, Digital Twin technology has recently attracted significant attention. To achieve more precise management and prediction, recent studies have introduced the concept of distributed Digital

Twins, as depicted in Figure 1. In our approach, we utilize a functional data-driven model to identify temporal dependencies in Digital Twins using time series data. Additionally, the contribution of this research lies in the utilization of statistical methods for identifying lags in time series data. These methods aid us in understanding the relationships between data points at different time stamps, which is crucial for identifying and modeling temporal dependencies.

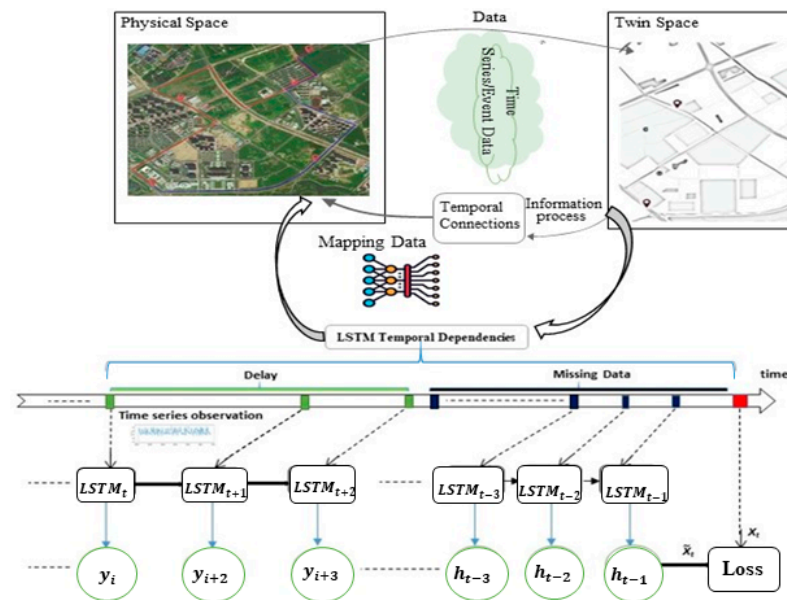


Figure 1. Digital Twin and temporal dependency LSTM framework.

Between the physical space and the twin space, there exists a connection through wired and wireless links to the industrial system. Based on their specific requirements and significance, industrial devices request the associated server to perform simulations to predict temporal connections. In this study, LSTM serves as the functional model of the Digital Twin, aiming to identify patterns, trends, and lags in the system. In this mode, the domain of Digital Twins can conduct integrated simulations in accordance with the physical connection relations of these virtual models. It can also perform correlation discovery of patterns, thus enhancing accuracy and scalability through this mechanism. In general, several relationships that are challenging to define in the real world are often easier to resolve in the virtual world.

3.4. Temporal Dependency Techniques

Temporal dependencies (TDs) are the patterns and relationships that develop over time between the variables in multivariate time series data. These dependencies can be seasonality, trends, lagged relationships, sequential patterns, and other temporal structures.

For Digital Twin applications, identifying temporal connections in multivariate time series data is essential, as shown in Figure 1. Understanding the temporal correlations between various variables enables spotting hidden patterns and future behavior. This information can be applied to Digital Twin implementations to improve system performance, anticipate problems, carry out condition monitoring, and enable environment or proactive maintenance.

Figure 2 visualizes how temporal dependency can be based on observed, trend, seasonal, and residual components in the datasets. In time series analysis, understanding temporal dependency involves decomposing a time series into its various components, which typically include observed, trend, seasonal, and residual components. Each of these components represents different aspects of the data, and they help in identifying the underlying patterns and dependencies in the time series.

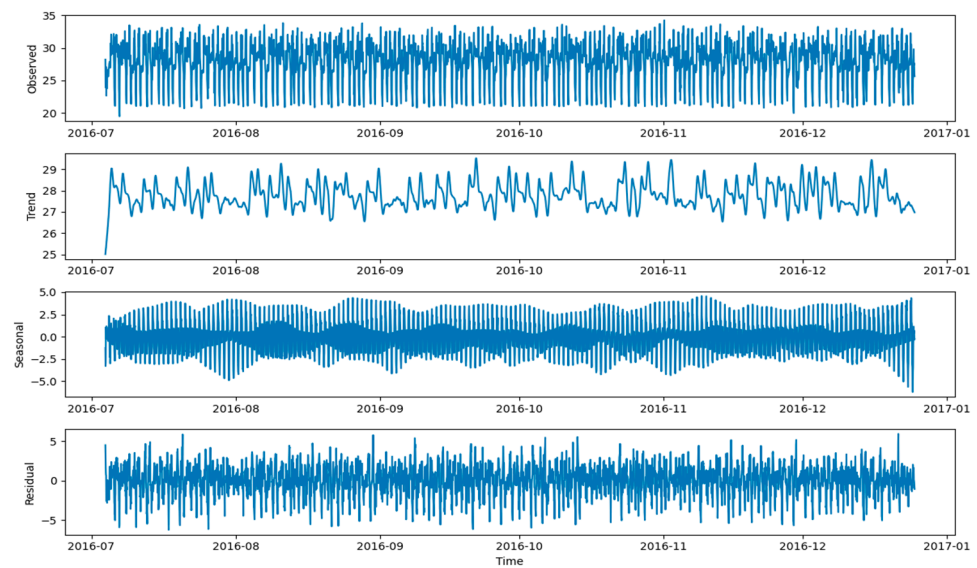


Figure 2. Representations of the temporal dependency based on trends, sequential patterns, and seasonality in time-stamped series data.

In the context of the Digital Twin, the temporal dependencies can be identified based on trends and observed data but can also be identified based on other components like residual and seasonality. In the physical space, these trends might indicate how a specific parameter, say, temperature or production output, is gradually changing over time while in the twin space, LSTM models can capture these trends by learning from the data. The Digital Twin leverages these learned dependencies to provide real-time insights and predictions. For instance, it might send an alert when the pressure exceeds safe limits or predict peak flow rates for better resource allocation.

Overall, by decomposing a time series into its constituent parts, analysts can gain insights into the different patterns and relationships that exist within the data. Temporal dependency in time series data is an important concept that can be used to comprehend, predict, and analyze time series data effectively.

3.5. Experimental Environment

This work uses the Python Jupiter notebook 6.5.2 framework to develop the appropriate LSTM machine-learning-based model for the verification of the temporal dependency model based on Digital Twins. A Windows 10 computer with a 64-bit operating system, an LG 13th-generation Intel Core i5-13400 processor, 32 GB of RAM, and an NVIDIA GeForce RTX 3060 Ti GPU Seoul, South Korea were all used in the training configuration. We employed the Adam optimizer with a batch size of 16, a learning rate of 0.0001, and a number of epochs of 300 to enhance the performance of our model. We employed multivariate time series to verify the temporal dependence of Digital Twin using SWaT [28] and BATADAL [35].

4. Experiment and Results

4.1. Data Preprocessing

In our datasets, all the input features based on our dataset were collected from different sensors and actuators, and their unit scale may be different.

The first step in dealing with the sparseness of IIoT data is to use the StandardScaler to normalize the input variables. The data are transformed using standard scaling to have a mean of 0 and a standard deviation of 1. It scales the data so that the distribution is centered around 0, and some machine learning algorithms can operate more effectively as a result. The StandardScaler is used to scale down input data ‘inputs’ to a common scale. Table 2 shows the features used for the normalization of the LSTM model.

Table 2. Dataset with features used.

Datasets	Features
SWaT	'PIT501', 'PIT502', 'PIT503', 'AIT501', 'AIT502', 'AIT503', 'AIT504', 'FIT501', 'FIT502', 'FIT503'
BATADAL	'P_J280', 'P_J269', 'P_J300', 'P_J256', 'P_J289', 'P_J415', 'P_J302', 'P_J306', 'P_J307', 'P_J317', 'P_J14', 'P_J422'

Missing values in the dataset can cause problems while training machine learning models. This is handled by the SimpleImputer. Normalizing the variables helps the optimization algorithm to converge faster during the training by bringing them to a common scale. The optimization process is generally more stable and efficient.

4.2. Training Stage of TD-LSTM

We employed the multivariate LSTM to improve our model's ability to learn by applying temporal dependency filters to the row vectors by capturing and learning the temporal relationships in the time series data. Algorithm 1 aimed to help the TD-LSTM model generate precise predictions and comprehend the dynamics of the system across the time stamps.

Algorithm 1: Training Stage of Using LSTM.

```

1 Input:
2   Time series phenomenal:  $\{X_1, X_2, \dots, X_{n-1}\}$ ; target at
3   time t:  $X_t$ ;
4   Length of lags(delays), period, trend:  $l_d, l_t$ ;
   period: p;
5 Output:
   TD-LSTM model M;
Procedure:
   //create a training example
6    $T \leftarrow \mu$ 
7   for every window of time that exists  $t(1 \leq t \leq n - 1)$  do:
8      $Z_d = [X_{t-l_d}, X_{t-(l_d-1)}, \dots, X_{t-1}]$ ;
9      $Z_p = [X_{t-l_{p,p}}, X_{t-(l_{p-1}),p}, \dots, X_{t-p}]$ ;
   put a training instance ( $\{Z_d, Z_p\}$ ) into T;
10  end for
   // Train model
11  for epoch in range (2000):
12    for batch in random batches(T): // Divide T into random mini-batches
13      update  $\mu$  using optimization approach (Adam) and batch;
   // Provide the trained TD-LSTM model M
14  M = TD-LSTM( $\mu$ );

```

4.3. Model Evaluation Criteria

Several metrics were employed to evaluate the models' efficacy, which consisted of Mean Absolute Percentage Error (MAPE), Mean Absolute Error (MAE), and Root Mean Square Error (RMSE). When using a forecasting method that often expresses prediction accuracy as a percentage, the MAPE offers a helpful way to quantify prediction accuracy. The MAE provides the average deviation between the model's predictions and the actual data. The RMSE measures the standard deviation of the model's prediction results. A lower value implies improved model performance. The three standards are described below:

$$MAPE = \left(\frac{1}{n} \sum_{i=1}^n \frac{y_i - \hat{y}_i}{y_i} \right) \times 100 \quad (7)$$

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \tag{8}$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \tag{9}$$

We used three evaluation criteria where y_i , \hat{y}_i , and y_i are respectively the i th actual value, the i th predicted value, and the average value of n samples.

4.4. Results and Discussion

As shown in Figure 3 we employed LSTM layers specifically designed to handle time series data. The first LSTM layer comprised 64 units, while the second LSTM layer consisted of 32 units. The return_sequences = True argument was utilized to instruct the model to return the output of the LSTM layer at each time step. This was imperative because we aimed to enable the model to learn temporal dependencies within the data.

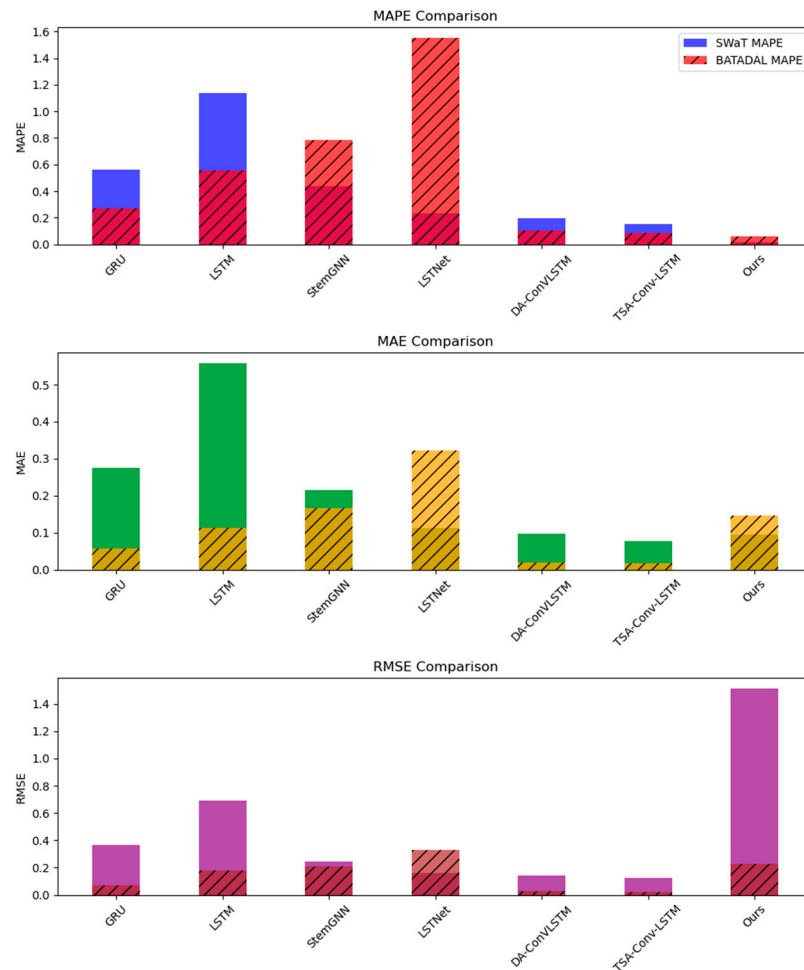


Figure 3. Error rate comparison between SWaT and BATADAL datasets.

Additionally, we introduced a dropout layer of 20% into the model. The dropout layer randomly deactivates some of the units during training, which effectively prevents the model from overfitting the training data. Furthermore, a dense layer was implemented to connect all the units within the layer, matching the number of units to the output data.

During the experiment, we specified the optimizer, loss function, and metrics employed for training and evaluating the model. We opted for the Adam optimizer to update the weights and measure the error between the model’s predictions and the ground truth

labels. Metrics were employed to monitor the model's performance throughout both the training and evaluation phases. The results in Figure 4 were remarkable, with excellent MAPE, MAE, and RMSE metrics signifying high predictive accuracy. However, in the case of the SWaT dataset, our proposed model exhibited a weakness, particularly in terms of RMSE.

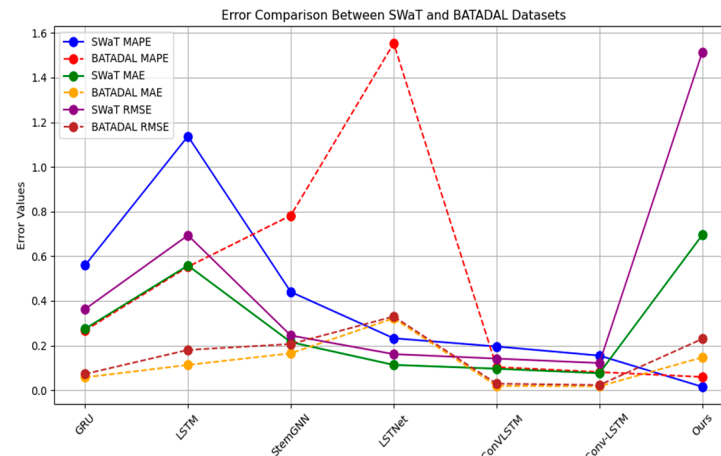


Figure 4. Error comparison based on the six baseline models.

Figures 3 and 4 display the error rate metrics for several models across the datasets. Our method was compared to six baseline studies, including GRU [29], LSTM [36], LSTNet [32], StemGNN [26], DA-ConvLSTM [31], and TSA-ConvLSTM [7].

To evaluate the LSTM network's ability to recognize temporal relationships in multivariate Digital Twins, we conducted an assessment using the SWaT and BATADAL datasets. As depicted in Figure 3, when comparing error rates between the datasets, a significant distinction becomes apparent between the baseline and our model. MAPE and MSE demonstrate promising results in reducing the error rate disparity, with differences of less than 0.2 when compared to the six baseline models. However, RMSE exhibits some weaknesses in terms of error rates between the datasets.

StemGNN [26], on the other hand, is built upon the discrete Fourier approach and incorporates a Graph Neural Network (GNN) model to extract features based on the spatial relationships within the multivariate time series. It introduces an innovative self-attention mechanism for learning the necessary graph structure for the GNN. During our training process, we adhered to the default method configuration, using sample lengths of 10 for SWaT and 12 for BATADAL.

Additional studies of LSTNet [32], designed to capture both long-term and short-term dependencies, employ a fully connected layer for data autoregression. In our experiments, we adopted the default model settings provided by the authors, and they utilized sample lengths of 15 and 20 for their datasets. LSTM [32] is a conventional sequential model, while GRU [26] represents an improved variant of LSTM with a reduced number of parameters. These models were combined with a dense layer featuring a single hidden unit.

DA-ConvLSTM [31] and TSA-ConvLSTM [7] are two recent models for multivariate time series prediction. They incorporate two attention layers within the convolutional layer to capture temporal and spatial correlations, which have been shown to produce excellent results.

In our paper, we followed the authors' training procedures in [31] and [7] but increased the number of iterations from 300 to 2000. The baseline models in the original papers used 256 LSTM units, but we used lighter units of 64 to capture temporal correlations. Our results in Table 3 were remarkable, with excellent MAPE, MAE, and RMSE metrics in Figures 3 and 4, indicating high predictive accuracy. However, our proposed model showed a weakness in the SWaT dataset, particularly in terms of RMSE.

Table 3. Comparing our methods with other methods in the literature.

Methods	SWaT			BATADAL		
	MAPE	MAE	RMSE	MAPE	MAE	RMSE
GRU [29]	0.5593	0.2749	0.3631	0.2675	0.0575	0.0725
LSTM [36]	1.1373	0.5583	0.6936	0.5529	0.1131	0.1804
StemGNN [26]	0.4401	0.2152	0.2449	0.7825	0.1652	0.2066
LSTNet [32]	0.2325	0.1133	0.1615	1.5534	0.3221	0.3302
DA-ConVLSTM [31]	0.1958	0.0962	0.1416	0.1036	0.0193	0.0291
TSA-Conv-LSTM [7]	0.1549	0.0762	0.1216	0.0812	0.0174	0.0232
Ours	0.0153	0.6952	1.5140	0.0592	0.1470	0.2296

5. Correlation Analysis

Various analysis approaches can be applied to multivariate time series data to detect temporal dependencies. Among the methods that are frequently utilized is autocorrelation analysis, where one can determine the correlation between a variable and its lagged values. The Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) can be used to determine the proper order of the moving average (MA) and autoregressive (AR) components of a time series model.

In time series analysis, identifying the appropriate lag order (the number of past observations used as inputs in a time series model) is crucial for building effective predictive models. The Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) plots seen in Figure 5a–d of the SWaT and BATADAL datasets are commonly used tools to identify the lag order. The *y*-axis represents the correlation while the *x*-axis represents the lag for both ACF and PACF.

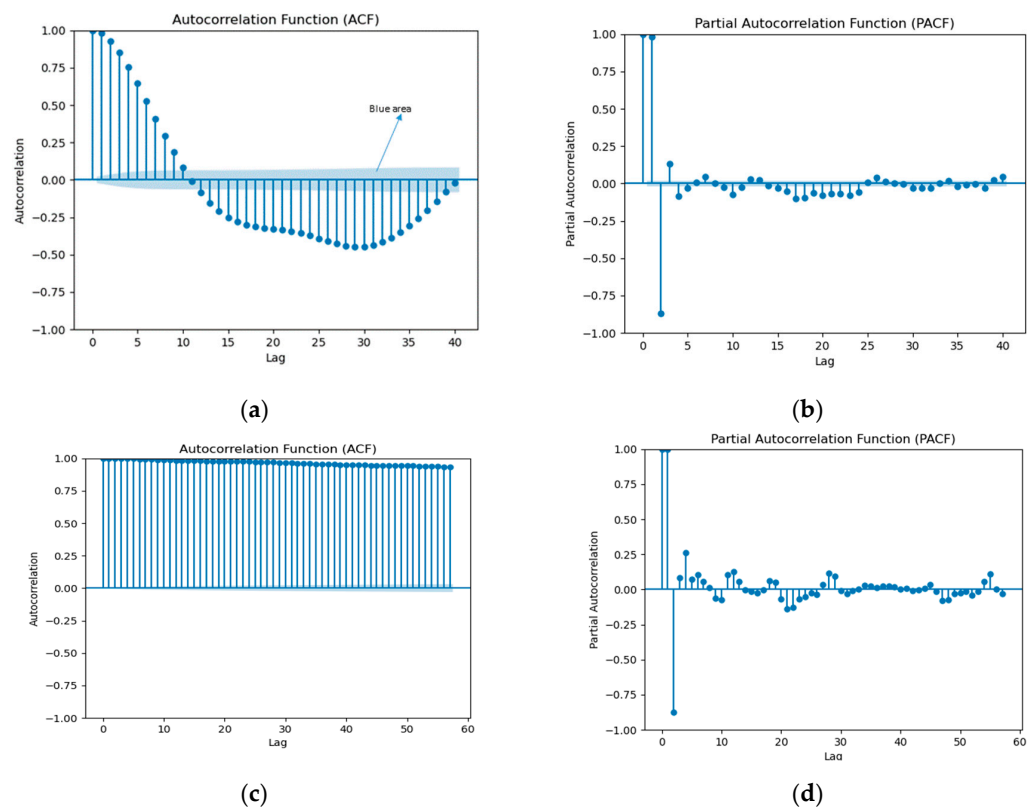


Figure 5. Representations of an autocorrelation analysis in identifying lags of the target instances: (a,b) the BATADAL dataset with ACF and PACF; (c,d) the SWaT dataset of the target variable using ACF and PACF, respectively.

The ACF plot shows the correlation between the time series and its lagged values. The PACF plot shows the correlation between the time series and its lagged values, after removing the effects of the intervening lags.

Both the ACF and PACF plots are typically plotted on the same graph, with the lag on the x -axis and the correlation coefficient on the y -axis. The blue area in the plots represents the 95% confidence interval. Any correlation coefficient that falls outside the blue area is considered to be statistically significant.

To interpret the ACF and PACF plots, we looked for the following patterns:

- A significant spike at lag 0, indicating that the time series is autocorrelated. This means that the current value of the time series is correlated with its past values.
- A gradual decay in the ACF plot, suggesting that the time series is an autoregressive (AR) process. This means that the current value of the time series can be predicted from its past values.

A sharp drop in the PACF plot after a certain lag suggests that the time series is a moving average (MA) process. This means that the current value of the time series can be predicted from the errors of its past predictions.

6. Conclusions

In this research paper, we explored the application of Digital Twins and harnessed the power of LSTM networks to uncover long-term temporal dependencies within complex multivariate time series data. Our proposed method, utilizing multivariate LSTM, effectively captured these extended temporal relationships present in time series data. Dealing with missing data is a common challenge in time series analysis, and to overcome this hurdle, we employed LSTM, a technique capable of handling missing values seamlessly. The integration of LSTM networks played a significant role in advancing Digital Twin technology for industrial use, showcasing their efficiency in capturing intricate connections and handling missing data.

Our experiments demonstrated that our approach outperformed others when assessed using key evaluation metrics such as MAPE, MAE, and RMSE. To gauge our model's performance comprehensively, we conducted thorough comparisons with six baseline models using the SWaT and BATADAL datasets.

Overall, our methodology incorporated two essential statistical techniques, ACF and PACF, which proved instrumental in identifying time lags within time series data. These statistical methods facilitated a deeper comprehension of the relationships between data points at different time intervals, a critical aspect in effectively identifying and modeling temporal dependencies within time series data.

7. Limitations and Future Work

With the aim of enhancing the efficiency of industrial systems and identifying temporal correlations, this study has made significant advancements by utilizing Digital Twins and LSTM networks with time series datasets in the Industrial Internet of Things (IIoT). However, it is essential to acknowledge that the research has several limitations.

Future work should focus on addressing the weaknesses observed in the SWaT dataset, particularly regarding the RMSE error rate comparison. This can be achieved by conducting more comprehensive hyperparameter tuning analyses and investigating methods for interpreting the temporal dependencies learned by the LSTM model. Furthermore, real-world applications would greatly benefit from considering the actual impacts and challenges associated with model deployment and integration.

In summary, while this study has made significant progress in utilizing the Digital Twins concept and LSTM networks for temporal correlation identification in time series data, it is imperative to recognize and address the research's limitations to ensure its validity and applicability in practical scenarios.

Author Contributions: Conceptualization, A.I. and H.S.; methodology, A.I.; software, S.O. (Seungmin Oh); validation, I.A. and S.O. (Sangwon Oh); formal analysis, S.O. (Seungmin Oh); investigation, H.S.; resources, S.O. (Seungmin Oh); data request, A.I.; writing—original draft preparation, A.I. and J.K.; writing—review and editing, A.I.; visualization, T.-w.U.; supervision, J.K.; project administration, T.-w.U.; funding acquisition, J.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the MSIT (Ministry of Science and ICT), Korea, under the Innovative Human Resource Development for Local Intellectualization support program (IITP-2023-RS-2022-00156287) supervised by the IITP (Institute for Information & Communications Technology Planning & Evaluation). This work was also funded by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) (NRF-2021R1I1A3060565) and was partly supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2021-0-02068, Artificial Intelligence Innovation Hub).

Data Availability Statement: You can view the SWaT and BATADAL datasets at https://itrust.sutd.edu.sg/itrust-labs_datasets/dataset_info, requested on 9 June 2023.

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

References

1. Sergeeva, M.B.; Voskobovich, V.V.; Kukharensko, A.M. Data Processing in Industrial Internet of Things (IIoT) Applications: Industrial Agility. In Proceedings of the 2022 Wave Electronics and Its Application in Information and Telecommunication Systems (WECONF), St. Petersburg, FL, USA, 29 May–2 June 2022; pp. 4–8. [\[CrossRef\]](#)
2. Shin, H.; Oh, S.; Isah, A.; Aliyu, I.; Park, J.; Kim, J. Network Traffic Prediction Model in a Data-Driven Digital Twin Network Architecture. *Electronics* **2023**, *12*, 3957. [\[CrossRef\]](#)
3. Mo, F.; Rehman, H.U.; Monetti, F.M.; Chaplin, J.C.; Sanderson, D.; Popov, A.; Maffei, A.; Ratchev, S. A framework for manufacturing system reconfiguration and optimization utilising digital twins and modular artificial intelligence. *Robot. Comput. Integr. Manuf.* **2023**, *82*, 102524. [\[CrossRef\]](#)
4. Xiao, Y.; Yin, H.; Duan, T.; Qi, H.; Zhang, Y.; Jolfaei, A.; Xia, K. An Intelligent prediction model for UCG state based on dual-source LSTM. *Int. J. Mach. Learn. Cybern.* **2021**, *12*, 3169–3178. [\[CrossRef\]](#)
5. Aheleroff, S.; Xu, X.; Zhong, R.Y.; Lu, Y. Digital Twin as a Service (DTaaS) in Industry 4.0: An Architecture Reference Model. *Adv. Eng. Informatics* **2021**, *47*, 101225. [\[CrossRef\]](#)
6. Shih, S.Y.; Sun, F.K.; Lee, H. Temporal pattern attention for multivariate time series forecasting. *Mach. Learn.* **2019**, *108*, 1421–1441. [\[CrossRef\]](#)
7. Fu, E.; Zhang, Y.; Yang, F.; Wang, S. Temporal self-attention-based Conv-LSTM network for multivariate time series prediction. *Neurocomputing* **2022**, *501*, 162–173. [\[CrossRef\]](#)
8. Aliyu, I.; Um, T.-W.; Lee, S.-J.; Gyoon Lim, C.; Kim, J. Deep Learning for Multivariate Prediction of Building Energy Performance of Residential Buildings. *Comput. Mater. Contin.* **2023**, *75*, 5947–5964. [\[CrossRef\]](#)
9. Widiputra, H.; Mailangkay, A.; Gautama, E. Multivariate CNN-LSTM Model for Multiple Parallel Financial Time-Series Prediction. *Complexity* **2021**, *2021*, 9903518. [\[CrossRef\]](#)
10. Zhang, F.; Zhu, X.; Hu, T.; Guo, W.; Chen, C.; Liu, L. Urban link travel time prediction based on a gradient boosting method considering spatiotemporal correlations. *ISPRS Int. J. Geo-Inf.* **2016**, *5*, 201. [\[CrossRef\]](#)
11. Feng, X.; Wu, J.; Wu, Y.; Li, J.; Yang, W. Blockchain and digital twin empowered trustworthy self-healing for edge-AI enabled industrial Internet of things. *Inf. Sci.* **2023**, *642*, 119169. [\[CrossRef\]](#)
12. Institute of Electrical and Electronics Engineers. Proceedings of the 2021 IEEE 29th International Conference on Network Protocols (ICNP 2021), Virtual, 1–5 November 2021; ISBN 9781665441315.
13. Klingenberg, C.O.; Borges, M.A.V.; Antunes, J.A.V. Industry 4.0 as a data-driven paradigm: A systematic literature review on technologies. *J. Manuf. Technol. Manag.* **2021**, *32*, 570–592. [\[CrossRef\]](#)
14. Calvo-Bascones, P.; Voisin, A.; Do, P.; Sanz-Bobi, M.A. A collaborative network of digital twins for anomaly detection applications of complex systems. Snitch Digital Twin concept. *Comput. Ind.* **2023**, *144*, 103767. [\[CrossRef\]](#)
15. Isah, A.; Shin, H.; Aliyu, I.; Oh, S.; Lee, S.; Park, J.; Hahn, M.; Kim, J. A Data-Driven Digital Twin Network Architecture in the Industrial Internet of Things (IIoT) Applications. In Proceedings of the 11th International Conference on Advanced Engineering and ICT-Convergence, AEICP, Jeju, Republic of Korea, 11–14 July 2023; Volume 6.
16. Oh, S.; Oh, S.; Um, T.W.; Kim, J.; Jung, Y.A. Methods of Pre-Clustering and Generating Time Series Images for Detecting Anomalies in Electric Power Usage Data. *Electronics* **2022**, *11*, 3315. [\[CrossRef\]](#)
17. Zhang, P.G. Time series forecasting using a hybrid ARIMA and neural network model. *Neurocomputing* **2003**, *50*, 159–175. [\[CrossRef\]](#)

18. Endemann, C.M.; Krause, B.M.; Nourski, K.V.; Banks, M.I.; Veen, B. Van Multivariate autoregressive model estimation for high-dimensional intracranial electrophysiological data. *Neuroimage* **2022**, *254*, 119057. [[CrossRef](#)] [[PubMed](#)]
19. Vohra, R.; Goel, K.; Sahoo, J.K. Modeling temporal dependencies in data using a DBN-LSTM. In Proceedings of the 2015 IEEE International Conference on Data Science and Advanced Analytics (DSAA), Paris, France, 19–21 October 2015; pp. 1–4. [[CrossRef](#)]
20. Sagheer, A.; Kotb, M. Time series forecasting of petroleum production using deep LSTM recurrent networks. *Neurocomputing* **2019**, *323*, 203–213. [[CrossRef](#)]
21. Uribe-Murcia, K.J.; Shmaliy, Y.S.; Ahn, C.K.; Zhao, S. Unbiased FIR Filtering for Time-Stamped Discretely Delayed and Missing Data. *IEEE Trans. Automat. Contr.* **2020**, *65*, 2155–2162. [[CrossRef](#)]
22. Kam, P.S.; Fu, A.W.C. Discovering temporal patterns for interval-based events. *Lect. Notes Comput. Sci. (Incl. Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinform.)* **2000**, *1874*, 317–326. [[CrossRef](#)]
23. Sherstinsky, A. Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network. *Phys. D Nonlinear Phenom.* **2020**, *404*, 132306. [[CrossRef](#)]
24. Lavalley, M.; Yu, T.; Evans, L.; Van Hemelrijck, M.; Bosco, C.; Golozar, A.; Asimwe, A. Evaluating the performance of temporal pattern discovery: New application using statins and rhabdomyolysis in OMOP databases. *BMC Med. Inform. Decis. Mak.* **2022**, *22*, 31. [[CrossRef](#)]
25. Randles, M.; Lamb, D.; Odat, E.; Taleb-Bendiab, A. Distributed redundancy and robustness in complex systems. *J. Comput. Syst. Sci.* **2011**, *77*, 293–304. [[CrossRef](#)]
26. Özden, C.; Bulut, M. Spectral temporal graph neural network for multivariate agricultural price forecasting. *Cienc. Rural* **2024**, *54*, e20220677. [[CrossRef](#)]
27. Aheleroff, S.; Huang, H.; Xu, X.; Zhong, R.Y. Toward sustainability and resilience with Industry 4.0 and Industry 5.0. *Front. Manuf. Technol.* **2022**, *2*, 951643. [[CrossRef](#)]
28. Kim, B.; Alawami, M.A.; Kim, E.; Oh, S.; Park, J.; Kim, H. A Comparative Study of Time Series Anomaly Detection Models for Industrial Control Systems. *Sensors* **2023**, *23*, 1310. [[CrossRef](#)] [[PubMed](#)]
29. Jin, X.B.; Zheng, W.Z.; Kong, J.L.; Wang, X.Y.; Bai, Y.T.; Su, T.L.; Lin, S. Deep-learning forecasting method for electric power load via attention-based encoder-decoder with bayesian optimization. *Energies* **2021**, *14*, 1596. [[CrossRef](#)]
30. Lian, Y.; Geng, Y.; Tian, T. Anomaly Detection Method for Multivariate Time Series Data of Oil and Gas Stations Based on Digital Twin and MTAD-GAN. *Appl. Sci.* **2023**, *13*, 1891. [[CrossRef](#)]
31. Qin, Y.; Song, D.; Cheng, H.; Cheng, W.; Jiang, G.; Cottrell, G.W. A dual-stage attention-based recurrent neural network for time series prediction. *arXiv* **2017**, arXiv:1704.02971.
32. Lai, G.; Chang, W.C.; Yang, Y.; Liu, H. Modeling long- and short-term temporal patterns with deep neural networks. In Proceedings of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, Ann Arbor, MI, USA, 8–12 July 2018; pp. 95–104. [[CrossRef](#)]
33. Aksoy, A.; Ertürk, Y.E.; Erdoğan, S.; Eyduran, E.; Tariq, M.M. Estimation of honey production in beekeeping enterprises from eastern part of Turkey through some data mining algorithms. *Pak. J. Zool.* **2018**, *50*, 2199–2207. [[CrossRef](#)]
34. Hei, X.; Yin, X.; Wang, Y.; Ren, J.; Zhu, L. A trusted feature aggregator federated learning for distributed malicious attack detection. *Comput. Secur.* **2020**, *99*, 102033. [[CrossRef](#)]
35. Zhou, W.; Kong, X.M.; Li, K.L.; Li, X.M.; Ren, L.L.; Yan, Y.; Sha, Y.; Cao, X.Y.; Liu, X.J. Attack sample generation algorithm based on data association group by GAN in industrial control dataset. *Comput. Commun.* **2021**, *173*, 206–213. [[CrossRef](#)]
36. Chimmula, V.K.R.; Zhang, L. Time series forecasting of COVID-19 transmission in Canada using LSTM networks. *Chaos Solitons Fractals* **2020**, *135*, 109864. [[CrossRef](#)] [[PubMed](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.