



Article

Energy Consumption Prediction of Injection Molding Process Based on Rolling Learning Informer Model

Jianfeng Huang , Yi Li, Xinyuan Li, Yucheng Ding, Fenglian Hong and Shitong Peng 

Department of Mechanical Engineering, Shantou University, Shantou 515063, China; jfhuang@stu.edu.cn (J.H.); 19yli6@stu.edu.cn (Y.L.); 22xyli3@stu.edu.cn (X.L.); 19ycding@stu.edu.cn (Y.D.); 21flhong@alumni.stu.edu.cn (F.H.)

* Correspondence: shtpeng@stu.edu.cn

Abstract: Accurate energy consumption prediction in the injection molding process is crucial for optimizing energy efficiency in polymer processing. Traditional parameter optimization methods face challenges in achieving optimal energy prediction due to complex energy transmission. In this study, a data-driven approach based on the Rolling Learning Informer model is proposed to enhance the accuracy and adaptability of energy consumption forecasting. The Informer model addresses the limitations of long-sequence prediction with sparse attention mechanisms, self-attention distillation, and generative decoder techniques. Rolling learning prediction is incorporated to enable continuous updating of the model to reflect new data trends. Experimental results demonstrate that the RL-Informer model achieves a normalized root mean square error of 0.1301, a root mean square error of 0.0758, a mean absolute error of 0.0562, and a coefficient of determination of 0.9831 in energy consumption forecasting, outperforming other counterpart models like Gated Recurrent Unit, Temporal Convolutional Networks, Long Short-Term Memory, and two variants of the pure Informer models without Rolling Learning. It is of great potential for practical engineering applications.

Keywords: injection molding process; green manufacturing; data-driven; energy consumption prediction; RL-Informer



Citation: Huang, J.; Li, Y.; Li, X.; Ding, Y.; Hong, F.; Peng, S. Energy Consumption Prediction of Injection Molding Process Based on Rolling Learning Informer Model. *Polymers* **2024**, *16*, 3097. <https://doi.org/10.3390/polym16213097>

Academic Editor: Bo Li

Received: 10 October 2024

Revised: 30 October 2024

Accepted: 1 November 2024

Published: 2 November 2024



Copyright: © 2024 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

Plastic is one of the most versatile materials in modern life, integral to numerous industries and consumer products, with global production exceeding 400 million metric tons in 2024 [1]. Within the plastic industry, injection molding is a critical process that is not only energy-intensive but also marked by large-scale production and relatively low energy efficiency, often requiring between 4 and 5 kWh of energy per kilogram of plastic produced [2]. The accurate prediction of energy consumption in injection molding enables timely adjustments to operational parameters and process optimization [3], leading to reduced energy consumption and significant improvements in both energy efficiency and production output.

The injection molding process is typically divided into four key stages: filling, packing, cooling, and ejection [4]. Research has shown that each of these stages involves multiple factors that significantly affect energy consumption (see Figure 1), such as filling speed and mold temperature during the filling phase [5]. Meekers et al. identified that different factors have varying degrees of impact on energy consumption, with cycle time having the most pronounced effect, followed by nozzle temperature [6].

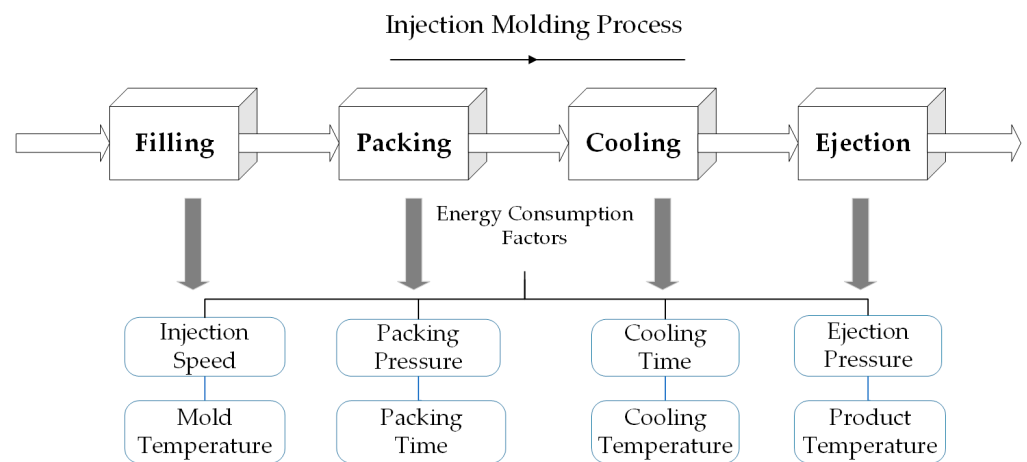


Figure 1. Injection molding steps and energy consumption influencing factors.

To date, numerous researchers have investigated the factors influencing energy consumption in injection molding, utilizing statistical models, machine learning techniques, and simulation modeling to predict energy usage in this process. For example, Pacella et al. employed a regression mixed model based on time series clustering to classify energy consumption patterns and construct virtual energy profiles for predictive purposes [7]. Takasaki et al. incorporated actual operational parameters into a simulated injection molding framework, enabling forecasts under various conditions [8]. However, due to the complex, nonlinear relationships between parameters like temperature, pressure, and cooling time [9], traditional models often fall short in accuracy, while high-fidelity simulations require substantial computational resources and data. This highlights an urgent need for more efficient analytical methods to develop predictive energy models for injection molding. To address these challenges, researchers have increasingly employed machine learning models to uncover latent relationships and predict energy consumption. For example, Bahij et al. evaluated the performance of machine learning techniques, such as artificial neural networks (ANNs) and support vector regression (SVR), in comparison with traditional methods like rule-based systems in industrial settings. Their results demonstrated that machine learning approaches significantly outperform traditional methods [10], both in terms of accuracy and efficiency, particularly when modeling complex energy consumption patterns and managing the nonlinear dynamics inherent to industrial energy use.

In the injection molding process, variables that influence energy consumption usually show complex correlations that machine learning models can effectively capture [11]. For example, Willenbacher et al. utilized the random forest algorithm to identify key factors affecting energy consumption in the production of plastic parts, enabling accurate predictions of energy usage and the determination of optimal machine settings to minimize energy consumption [12]. Similarly, Wu et al. employed an active machine learning approach to optimize the process parameters of injection molding machines, achieving a multi-objective optimization that balances energy consumption and part quality. Their research demonstrated that the trained energy consumption prediction model possesses robust predictive capabilities [13].

However, industrial data are often complex, nonlinear, and multidimensional challenging machine learning models like random forests, which require large datasets and perform poorly with noisy data or time series characteristics, thus limiting prediction accuracy. Neural networks, however, can effectively model nonlinear relationships and long-term dependencies, adapting to data variations to predict energy consumption accurately across different production settings. For instance, Nazir et al. demonstrated that the TFT achieved superior predictive performance with a symmetric mean absolute percentage error of 26.46%, outperforming LSTM (29.78%), interpretable LSTM (31.10%), and TCN (36.42%) [14].

The self-attention mechanism inherent in the Transformer model enables it to focus on relevant segments of the input sequence, effectively managing long-distance dependencies in time series data, which is particularly advantageous for energy prediction over extended periods [15]. However, traditional Transformer models exhibit high computational complexity and low inference efficiency [16] when addressing multivariate long-sequence predictions. To address these efficiency challenges, the Informer model incorporates advanced techniques [17], including a probabilistic self-attention mechanism, self-attention distillation, and a generative decoder. These enhancements result in significant improvements in both the accuracy and inference efficiency of multivariate long-sequence predictions, thereby increasing its practical applicability in energy forecasting.

To summarize, research issues that motivate the presents study are (1) Traditional machine learning methods can hardly capture the complex temporal features associated with energy consumption predictions in the injection molding process. (2) Simulation modeling requires substantial amounts of practical data, making it challenging to predict dynamic changes in energy consumption. (3) Transformer models are good at capturing temporal relationships but suffer from computation complexity for long sequences, which makes them difficult for practical deployment in the injection molding process. To bridge the research gap and enable more accurate energy prediction in the injection molding process, we propose the following contributions: We first conducted injection molding experiments to identify multiple groups of variables that may influence energy consumption. We then performed a correlation analysis on the experimental variables, selecting those with high correlations to energy consumption in order to reduce model complexity. Finally, we proposed an improved RL-Informer model to predict energy consumption during the injection molding process, which excels at capturing the intricate temporal dependencies within the data while maintaining a low computational complexity. A comparative analysis was performed between the improved RL-Informer model and other models, including LSTM and GRU, highlighting the advantages of the enhanced Informer model. Through this study, we aim to optimize production strategies, facilitate real-time monitoring, and enable predictive maintenance, ultimately contributing to more efficient and environmentally sustainable manufacturing practices.

The structure of this study is organized as follows: Section 2 describes the data collection and preprocessing methods employed in this research. Section 3 elaborates on the sparse attention mechanism, self-attention distillation, and rolling learning techniques incorporated into our approach. Section 4 presents a detailed comparison of model performances and discusses the key results derived from the experiments. Finally, Section 5 provides a comprehensive summary of the study's findings and contributions.

2. Data Processing in Injection Molding Experiments

2.1. Data Acquisition Platform

The injection molding machine utilized in this experiment is depicted in Figure 2 and is manufactured by Rui Xiang Plastics Co., Ltd. (Shantou, China). The identification number of the machine is Si-180IV_EHEXA284950-0505, and it is classified as a servo energy-saving drive model. This machine features an innovative V-type molding device and incorporates a newly designed shooting mechanism characterized by high rigidity and flexible stability.

The injection components processed in this study are handle inner buckle parts, measuring 82 mm in length, 45.9 mm in width, and 47.3 mm in height. The injection molding experiments utilize a blend of ABS and PBT in a mixing ratio of 70:30. The operational parameters of the injection molding process can be primarily categorized into temperature, speed, time, pressure, position, voltage, and current. In this study, using the data acquisition platform integrated into the machine, we collected real-time values for key parameters, including V-P switching position, remaining position, shrinkage time, mold opening time, and mold off time. Figure 3 provides a schematic diagram of the data measurement platform within the machine.



Figure 2. Injection machine Si-180IV_EHEXA284950-0505.

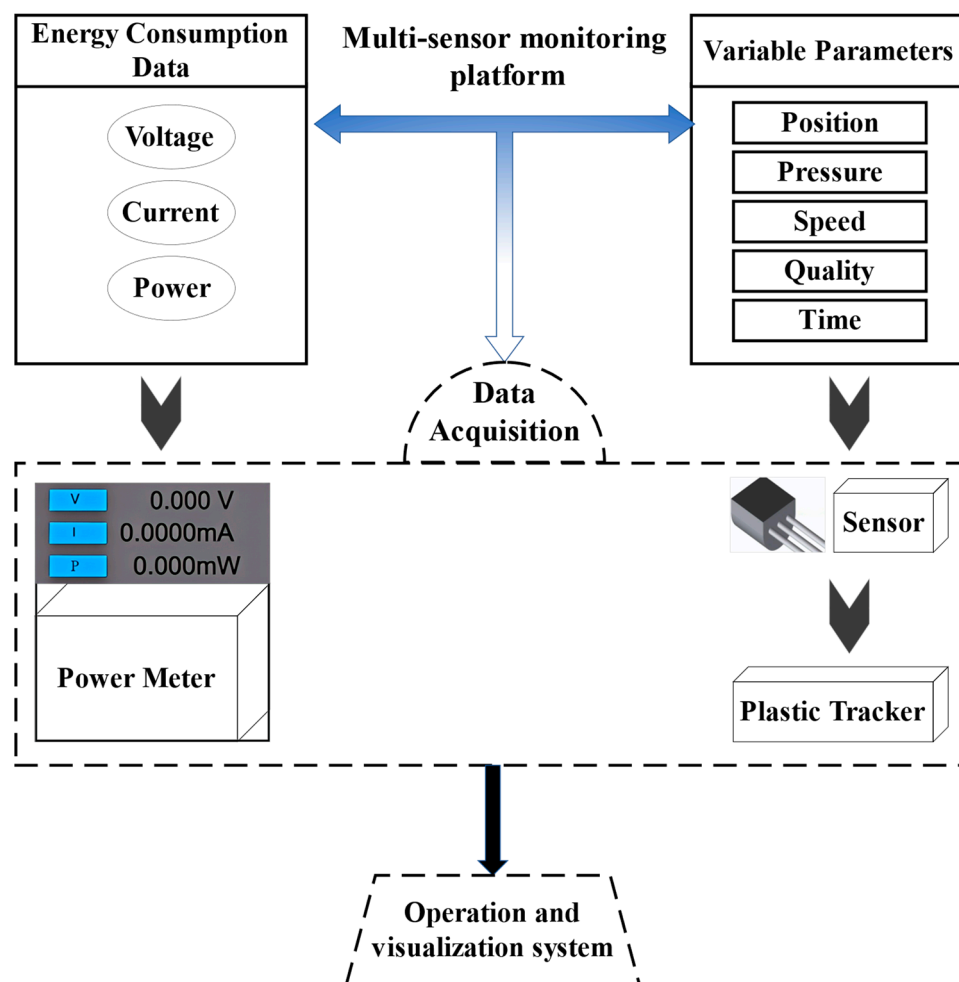


Figure 3. Schematic diagram of the multi-sensor platform.

2.2. Data Processing Procedure

During the injection molding process, the generated data typically exhibit strong continuity. However, due to interference factors such as vibrations and temperature fluctuations inherent to the molding process [18,19], the sensor-collected data often suffer from missing values or extreme outliers, etc. To ensure data integrity and continuity,

which are crucial for accurately analyzing and monitoring energy consumption, this study employs a fixed-range method to replace outliers. For instance, each component of the injection molding machine operates within a predefined range, and data exceeding these upper and lower bounds are replaced with the corresponding maximum or minimum values. In the case of power data exceeding the set limits, the outlier is substituted by the average of adjacent data points. For handling missing data, the KNN method is applied, which leverages the similarity between data points to fill in the missing values, with more accurate imputations when dealing with multiple missing values [20].

The main measured variables for the injection molding experiment are summarized in Table 1.

Table 1. Data variables of injection molding experiments.

Variable	Meaning	Unit
x_1	Time	s
x_2	Remaining position	mm
x_3	V-P switching position	mm
x_4	Periodic time	s
x_5	First-stage speed	mm/s
x_6	Second-stage speed	mm/s
x_7	Packing pressure	MPa
x_8	Maximum pressure	MPa
x_9	Maximum speed	mm/s
x_{10}	1-st injection mass	g
x_{11}	2-nd injection mass	g
x_{12}	Mass deviation	g
x_{13}	Forming label	/
x_{14}	Energy consumption	J

The relationships between various variables exhibit complexity, including linear and nonlinear characteristics, owing to the correlations of multiple influencing factors. To alleviate the computational burden of the model, enhance the operational efficiency, and effectively capture the correlations among variables, only those variables exhibiting strong effects on energy consumption are selected as model inputs. Linear relationships are evaluated using Pearson's correlation coefficient [21], while nonlinear correlations are assessed through Spearman's rank correlation coefficient [22]. In this context, a correlation coefficient closer to zero indicates a weaker association between variables.

As illustrated in Figure 4, two pairs of variables, x_7 and x_8 , as well as x_{10} and x_{11} , exhibit (a) Spearman and (b) Pearson correlation coefficients as high as 0.99. These near-perfect correlations indicate a high degree of both linear and monotonic association between the variables, suggesting that they carry almost identical information. Such high correlation coefficients are indicative of potential multicollinearity, which can negatively impact the stability and interpretability of the model. To mitigate the risk of multicollinearity and redundancy, we selected one variable from each pair that demonstrated a stronger correlation with the target variable, x_{14} , discarding the less correlated counterpart.

For example, Figure 4 highlights that variables x_3 , x_5 , x_6 , x_9 , x_{10} , and x_{11} are significantly correlated with x_{14} , either through (a) Spearman or (b) Pearson coefficients. Notably, x_{11} exhibits a marginally stronger correlation in both Pearson (0.68) and Spearman (0.63) analyses, leading to the decision to retain x_{11} and eliminate x_{10} to maintain model parsimony. Also, Figure 5b reveals that x_2 presents a Pearson correlation coefficient of -0.47 (in Figure 5, the absolute values are shown), indicating a moderate negative linear correlation with x_{14} , which suggests that, as x_2 increases, x_{14} tends to decrease, albeit not in a perfectly linear manner. However, Figure 5a shows that the Spearman correlation coefficient for x_2 is notably lower, at -0.26 , indicating a weak monotonic relationship. This discrepancy between the two correlation measures suggests that, while x_2 shares a linear relationship with x_{14} , this relationship is not uniformly consistent across the entire data range, poten-

tially due to the presence of local nonlinearities or the influence of outliers. Such behavior could indicate that x_2 and x_{14} 's relationship is sensitive to specific data points or particular regions of the data.

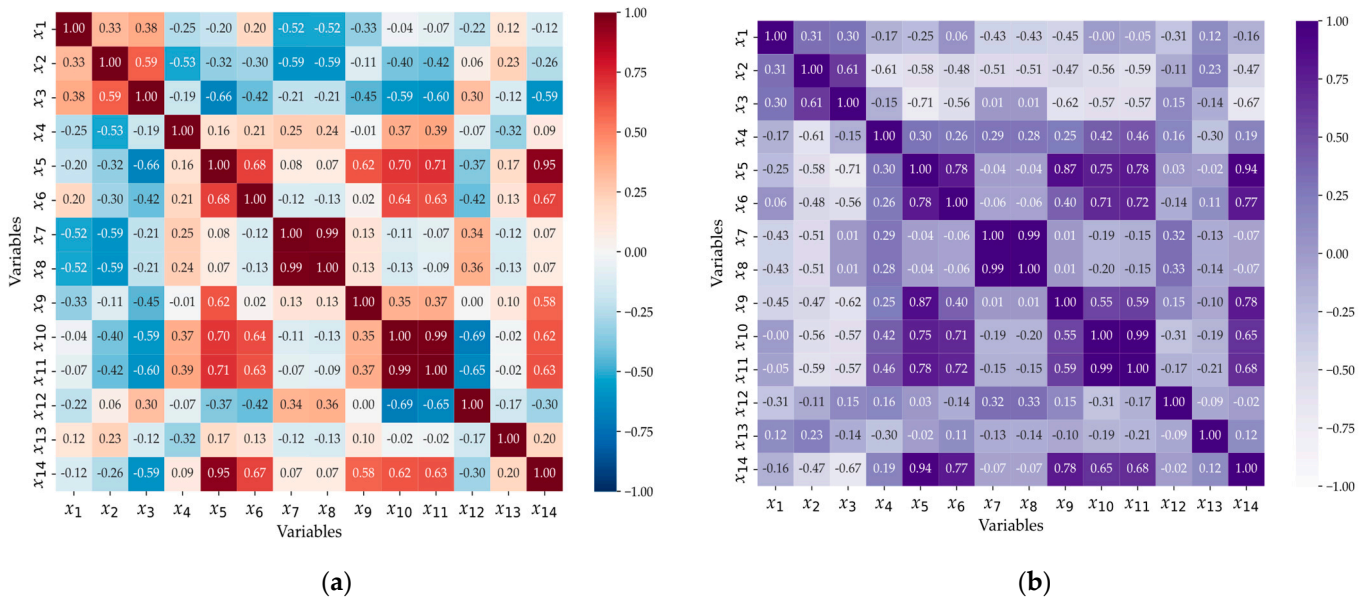


Figure 4. The coefficient heat map: (a) Spearman coefficient and (b) Pearson coefficient.

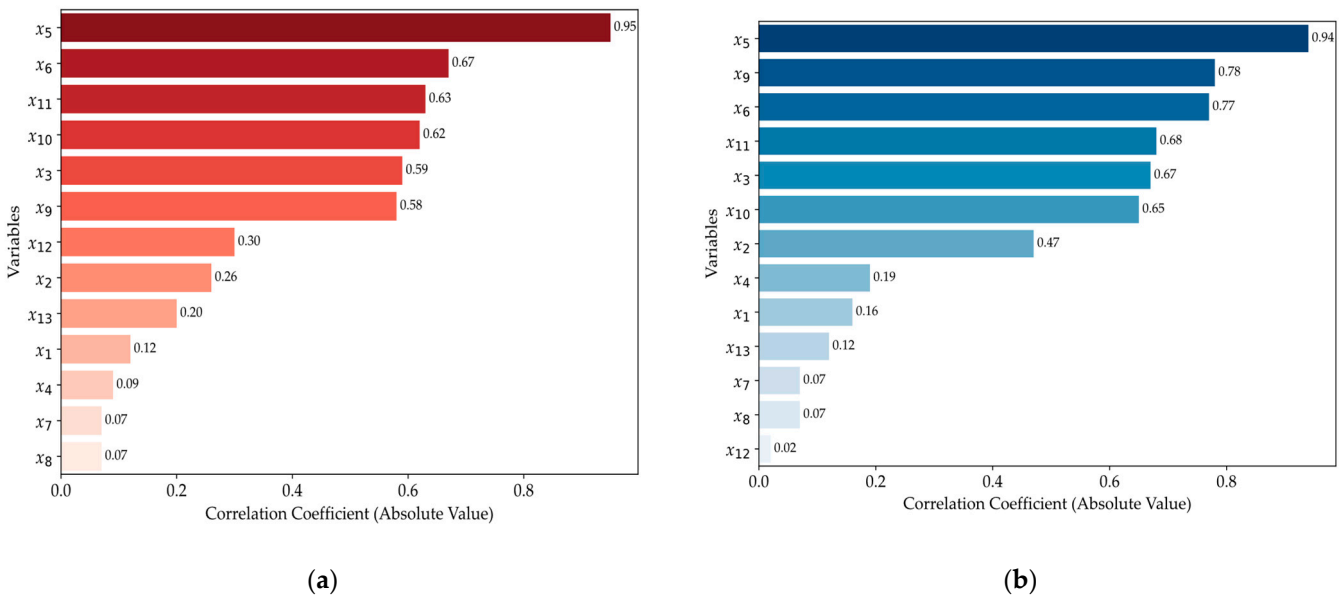


Figure 5. Correlation coefficient correlated with x_{14} : (a) Spearman coefficient and (b) Pearson coefficient.

In summary, to ensure the robustness and interpretability of the model, we retained both highly and moderately correlated variables and selected $x_2, x_3, x_5, x_6, x_9,$ and x_{11} for further analysis. This selection strategy was designed to balance the inclusion of informative variables while mitigating multicollinearity and maintaining model efficiency.

3. Informer Model and Rolling Learning Prediction

The Transformer model, introduced in 2017 [23], has gained significant attention for its effectiveness in time sequence prediction within NLP. Unlike traditional RNN [24] and CNN [25], the Transformer employs a multi-head self-attention mechanism and feed-forward networks, allowing for efficient processing of long sequences through parallel com-

puting [26]. This architecture enhances both training efficiency and task performance across various NLP applications. However, as sequence lengths increase, the Transformer exhibits limitations, including high memory consumption, inefficiencies in encoding/decoding, and increased time complexity, which restricts its application in large-scale temporal data prediction.

To address these challenges, the Informer model was introduced with several key improvements. By leveraging techniques such as model pruning, quantization, and structural innovations, the Informer significantly reduces computational and memory overhead while maintaining prediction accuracy, facilitating its deployment on embedded and edge devices [27]. Key enhancements include the sparse attention mechanism, self-attention distillation, and a generative decoder.

The sparse attention mechanism optimizes memory usage and reduces time complexity by focusing on essential key factors, improving efficiency in processing large datasets [28]. Self-attention distillation reduces input layers and parameters, effectively capturing long-term dependencies while enhancing generalization and minimizing input redundancy [29]. Finally, the generative decoder predicts entire long sequences in a single step, accelerating inference and improving efficiency for real-time applications. These innovations enable the Informer to handle long-sequence predictions more effectively, making it suitable for practical, large-scale temporal data tasks.

Figure 6 outlines the Informer model's framework, with the encoder on the left and the decoder on the right. The Informer's key improvement lies in its structural optimization of both components, enabling faster processing of long sequences. The encoder transforms the input sequence into hidden representations, leveraging sparse attention and self-attention distillation to capture critical information efficiently. Sparse attention enhances computational speed, while self-attention distillation refines feature extraction. The decoder then utilizes a generative mechanism to produce target sequence predictions by interacting with the encoder's hidden representations through attention. This interaction allows the decoder to focus on relevant parts of the input, improving prediction accuracy and stability. Together, these mechanisms make the Informer highly effective for long-sequence predictions.

3.1. Encoder Module of the Informer Model

Figure 7 is a single stack of the encoder structure in the Informer model, showing its complexity and the multi-level feature in detail. Combining the multi-head attention mechanism and the convolution operations with the maximum pooling operation and the self-attention distillation technique, the encoder can efficiently extract and integrate multilevel features of the time series data.

The primary role of the Informer model's encoder is to transform input sequences into higher-level abstract representations [30]. Each encoder layer comprises a multi-head self-attention mechanism and a feed-forward neural network. The self-attention mechanism computes correlations across different positions within the sequence, allowing the model to prioritize critical time steps and features. Simultaneously, the feed-forward network applies nonlinear transformations to enhance feature extraction and the model's representational capacity.

The core of the Informer consists of a stack of multiple encoder layers, where each layer extracts essential features and passes them to subsequent layers for deeper processing. The self-attention mechanism utilizes dot product matrices to compute positional correlations, enabling the model to focus on key time steps. The hidden representations at each step are processed and combined, eventually forming the encoder's output.

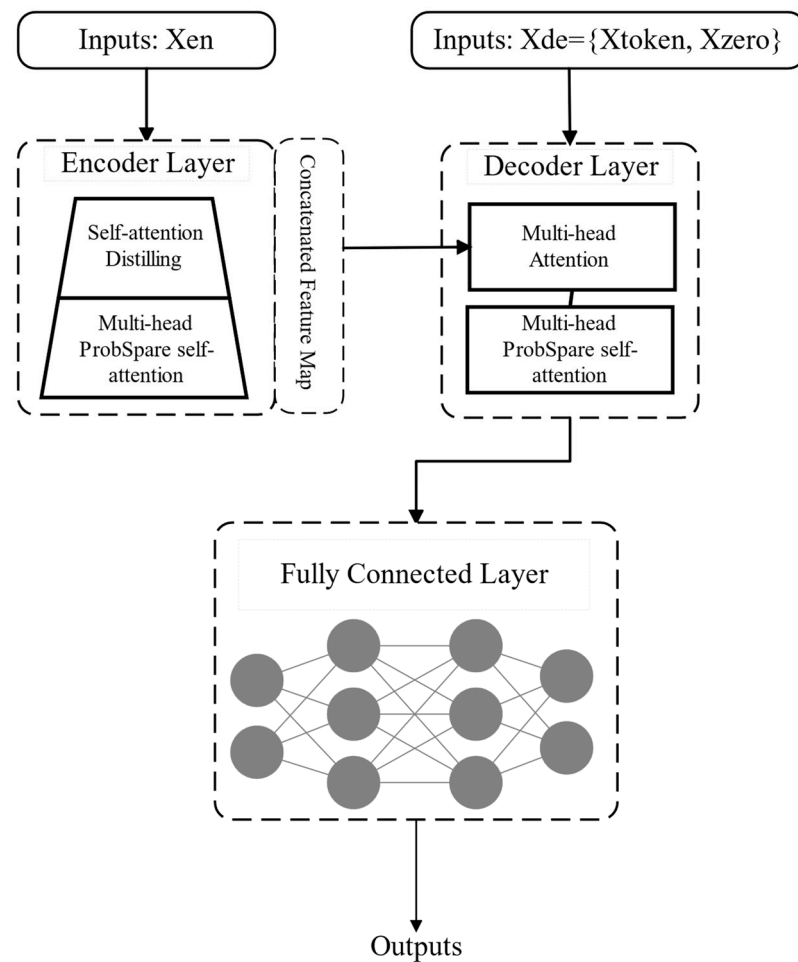


Figure 6. Structural diagram of the Informer model.

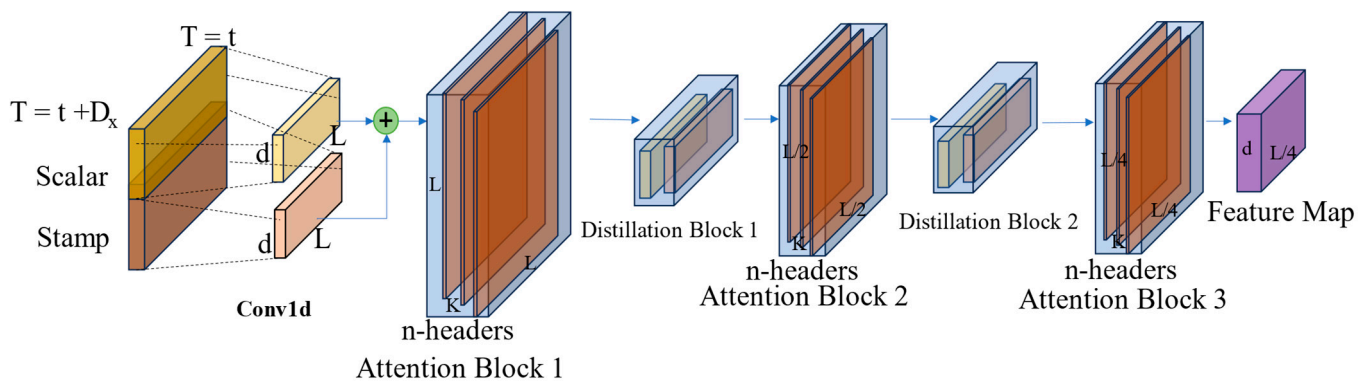


Figure 7. Encoder module of the Informer model.

A key innovation in the Informer model is its distillation operation. By applying hierarchical distillation to the encoder stack, the model simplifies its structure while improving generalization. This operation reduces the complexity of the encoder without sacrificing accuracy. The distillation operation can be represented by the following formula [17]:

$$X_{j+1}^t = \text{MaxPool} \left(\text{ELU} \left(\text{Conv1d} \left([X_j^t]_{AB} \right) \right) \right) \quad (1)$$

where $[X_j^t]_{AB}$ represents the output of the multi-headed ProbSparse self-attention layer in the previous layer, and X_{j+1}^t is the output of distillation in this layer; $Conv1d(\cdot)$ performs as a 1D convolutional filter; $ELU(\cdot)$ is used as the activation function; $MaxPool(\cdot)$ is finally applied to down-sample the processed data, retaining essential feature information while reducing data dimensionality. The distillation process enables efficient storage and processing without significant loss of critical information.

3.2. Decoder Module of the Informer Model

The decoder module is designed to generate the prediction of long sequences through a concise forward process. The model adopts the conventional decoder structure, which contains two identical multi-head attention layers. It solves the problem of high time complexity in long sequence data prediction by using a generative prediction method [31]. The input vector of the decoder is shown as follows:

$$X_{\text{feed_de}}^t = \text{Concat}(X_{\text{token}}^t, X_0^t) \in \mathbb{R}^{(L_{\text{token}}+L_y) \times d_{\text{model}}} \quad (2)$$

where X_{token}^t is the start token; L_{token} is the length of the long sequence selected from the input sequence and taken as the leading sequence that needs to be predicted. X_0^t is a placeholder of the target sequence, and it should be zero. Through introducing the masked multi-head attention into the sparse self-attention mechanism, it will avoid the autoregressive phenomenon of X_0^t , because it will stop each position from noticing the next position. In the final output phase, the model obtains the prediction results through the fully connected layer, which is able to achieve the ability of univariate or multivariate prediction. The generative structure is integrated into the encoder so that the predicted sequence can be generated at once without gradual dynamic decoding. Owing to this, it significantly reduces the decoding time and improves the prediction efficiency.

3.3. Self-Attention Mechanism of the Informer Model

The self-attention mechanism was originally defined by the three matrix tuple inputs (Q, K, and V), which perform scaled dot product operations. The initial formulas are as follows [23]:

$$A(Q, K, V) = \text{Softmax}\left(\frac{QK^T}{\sqrt{d}}\right)V \quad (3)$$

where d is the input dimension and $Q \in \mathbb{R}^{L_Q \times d}$, $K \in \mathbb{R}^{L_K \times d}$, and $V \in \mathbb{R}^{L_V \times d}$ represent query, key, and value, respectively. Both the key and the value are paired, and the query can get the corresponding value using the key. To further analyze the self-attention mechanism, the attention of the i -th query is defined as a kernel smoother in probabilistic form, which is defined as follows [17]:

$$A(q_i, K, V) = \sum_j \frac{k(q_i, k_j)}{\sum_l k(q_i, k_l)} v_j = E_{p(k_j | q_i)}[V_j] \quad (4)$$

where $p(k_j | q_i) = \frac{k(q_i, k_j)}{\sum_l k(q_i, k_l)}$, $k(q_i, k_j) = \exp\left(\frac{q_i k_j^T}{\sqrt{d}}\right)$ are the asymmetric indices. This kind of self-attention mechanism calculates the probability $p(k_j | q_i)$ through the dot product operation, reaching the quadratic time complexity. Additionally, it needs the spatial complexity of $O(L_Q | L_K)$ while calculating.

By regulating self-attention through qualitative analysis, the probability distribution of the self-attention mechanism has the potential sparsity, showing a trend of long-tailed distribution. It means that a very small amount of dot products contributes the main attention weight, so the contribution of other dot products can be ignored. In that case, the contribution of the partial query to a value can be calculated without calculation. Conse-

quently, the KL (Kullback–Leibler) divergence is introduced to compute and distinguish the important query. It is as follows [17]:

$$KL(q \parallel p) = \ln L_k \sum_{j=1}^{L_k} e^{\frac{q_i k_j^T}{\sqrt{d}}} - \frac{1}{L_k} \sum_{j=1}^{L_k} \frac{q_i k_j^T}{\sqrt{d}} - \ln L_k \quad (5)$$

Excluding the constant, the sparsity formula of the i -th query is:

$$M(q_i, K) = \ln L_k \sum_{j=1}^{L_k} e^{\frac{q_i k_j^T}{\sqrt{d}}} - \frac{1}{L_k} \sum_{j=1}^{L_k} \frac{q_i k_j^T}{\sqrt{d}} \quad (6)$$

where the first item q_i is the LogSumExp (LSE) on all keys. If the i -th query has a large value $M(q_i, K)$, it means that the corresponding probability of attention distribution has greater diversity. This shows that, in the self-attention mechanism, the chance of including the dominant dot product pair in the head domain of the long-tailed self-attention distribution is high. Based on this analysis, by allowing each key to handle only u dominant query, the ProbSparse Self-attention formula is as follows [17]:

$$A(Q, K, V) = \text{Softmax} \left(\frac{\overline{Q} K^T}{\sqrt{d}} \right) V \quad (7)$$

In the ProbSparse Self-attention, a kind of sparse matrix of the same size as the query vector q is used. This sparse matrix contains only the Top- u query under the sparse evaluation; while u is controlled by a fixed sampling factor c , let $u = c \cdot \ln L_Q$ [32]. In this way, when ProbSparse Self-attention is in each matching query key, ProbSparse Self-attention only needs to perform u dot product operations, so that the memory overhead of each layer is reduced to $O(L_K \ln L_Q)$. However, traversing all measured query $M(q_i, K)$ requires computing each dot product pair, with potential numerical stability problems. Therefore, the max-mean metric is introduced as a new indicator for assessing the sparsity of the query.

$$\overline{M}(q_i, K) = \max_j \left(\frac{q_i k_j^T}{\sqrt{d}} \right) - \frac{1}{L_K} \sum_{j=1}^{L_K} \frac{q_i k_j^T}{\sqrt{d}} \quad (8)$$

In the case of the long-tail distribution, $U = L_K \ln L$ dots are randomly selected to calculate $\overline{M}(q_i, K)$, and the other dot product pairs are filled to zero. The reason for choosing the number of Top- u to be the max operator in \overline{Q} and $\overline{M}(q_i, K)$ is that its sensitivity to zero is low and numerically stable. In practice, the input lengths of the query and key in self-attention computation are often equal, like $L_Q = L_K = L$. This reduces the temporal and spatial complexity of ProbSparse Self-attention to $O(L \ln L)$.

3.4. Rolling Learning Prediction

Rolling learning prediction enhances time series models by dynamically updating the training dataset, enabling models to capture evolving patterns and improve prediction accuracy. Liu et al. [33] applied this approach with LSTM for battery performance prediction, allowing continuous parameter updates for future state estimation. Similarly, Guo et al. [34] demonstrated that rolling learning improves MPC model accuracy, particularly in real-time energy management of plug-in hybrid electric vehicles. These studies underscore the potential of rolling learning for managing dynamic temporal data and complex nonlinear systems, achieving precise predictive control.

The original Informer model, though effective for fixed-length time series predictions, shows declining accuracy over time due to its limited adaptability to dynamic input data [35]. Rolling learning addresses this by continuously updating the model to capture

new data trends. Traditional time series models rely on static historical data, limiting their adaptability in dynamic environments. In complex processes like injection molding, where fluctuating parameters impact energy consumption, static models struggle to capture real-time variations.

Rolling learning prediction, utilizing techniques such as sliding windows, addresses this challenge by dividing time series data into overlapping subsequences. The model is trained on each subsequence and updated continuously, enhancing the adaptability to new data and shifting trends. This approach improves prediction accuracy and enables the model to handle non-stationary and abrupt changes in the data [36].

By integrating rolling learning prediction into the Informer model, an improved RL-Informer model is proposed. The steps are as follows:

1. Use the original Informer model to predict future values for the next time periods and generate the initial prediction sequence.
2. Feed the predicted values back into the model to update the input sequences.
3. Use the updated input sequences to predict the next time periods, continuously refining the model with the latest data.

3.5. Evaluation Metric

The performance of each model was assessed using several evaluation metrics, including NRMSE, RMSE, MAE, and R^2 . NRMSE calculated by Equation (9) provides a normalized measure of the prediction error, allowing for a relative assessment of model accuracy across different scales [37]. RMSE calculated by Equation (10) is a standard metric in regression analysis that calculates the average magnitude of errors between predicted and actual energy consumption data [38]. MAE defined by Equation (11), on the other hand, measures the average absolute deviation, offering an intuitive sense of prediction accuracy [39]. The R^2 metric defined by Equation (12) indicates the proportion of variance in the actual data that is explained by the model [40], reflecting its ability to capture the underlying patterns in energy consumption. Lower NRMSE, RMSE, and MAE values, combined with higher R^2 values, signify greater accuracy and robustness in the model's predictions. Equations for calculating NRMSE, RMSE, MAE, and R^2 are given below:

$$\text{NRMSE} = \frac{\sum_{i=1}^n (\hat{y}_i - y_i)^2}{\sum_{i=1}^n y_i^2} \quad (9)$$

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (10)$$

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |y_i| \quad (11)$$

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{Y})^2} \quad (12)$$

where y_1, y_2, \dots, y_n represent the actual data; $\hat{y}_1, \hat{y}_2, \dots, \hat{y}_n$ are the predicted data; \bar{Y} is the mean value of the actual data; n is the number of the samples.

4. Energy Consumption Prediction and Results Analysis

4.1. Process Structure

In Section 2, six key variables influencing energy consumption in the injection molding process were identified. To eliminate the impact of varying magnitudes and dimensions among these variables, the max–min normalization technique [41] was applied, transform-

ing the data into the $[0, 1]$ range. This ensured that each variable contributed proportionally to the model without introducing biases due to differences in scale. Following normalization, other advanced models, including LSTM, GRU, and TCN, were implemented for comparison against the RL-Informer model, providing a comprehensive performance benchmark. The overall process structure is illustrated in Figure 8. Note that, in order to highlight the benefits of incorporating rolling learning prediction, the pure Informer model and its variant, Informer_lag, are also compared. They will be detailed later.

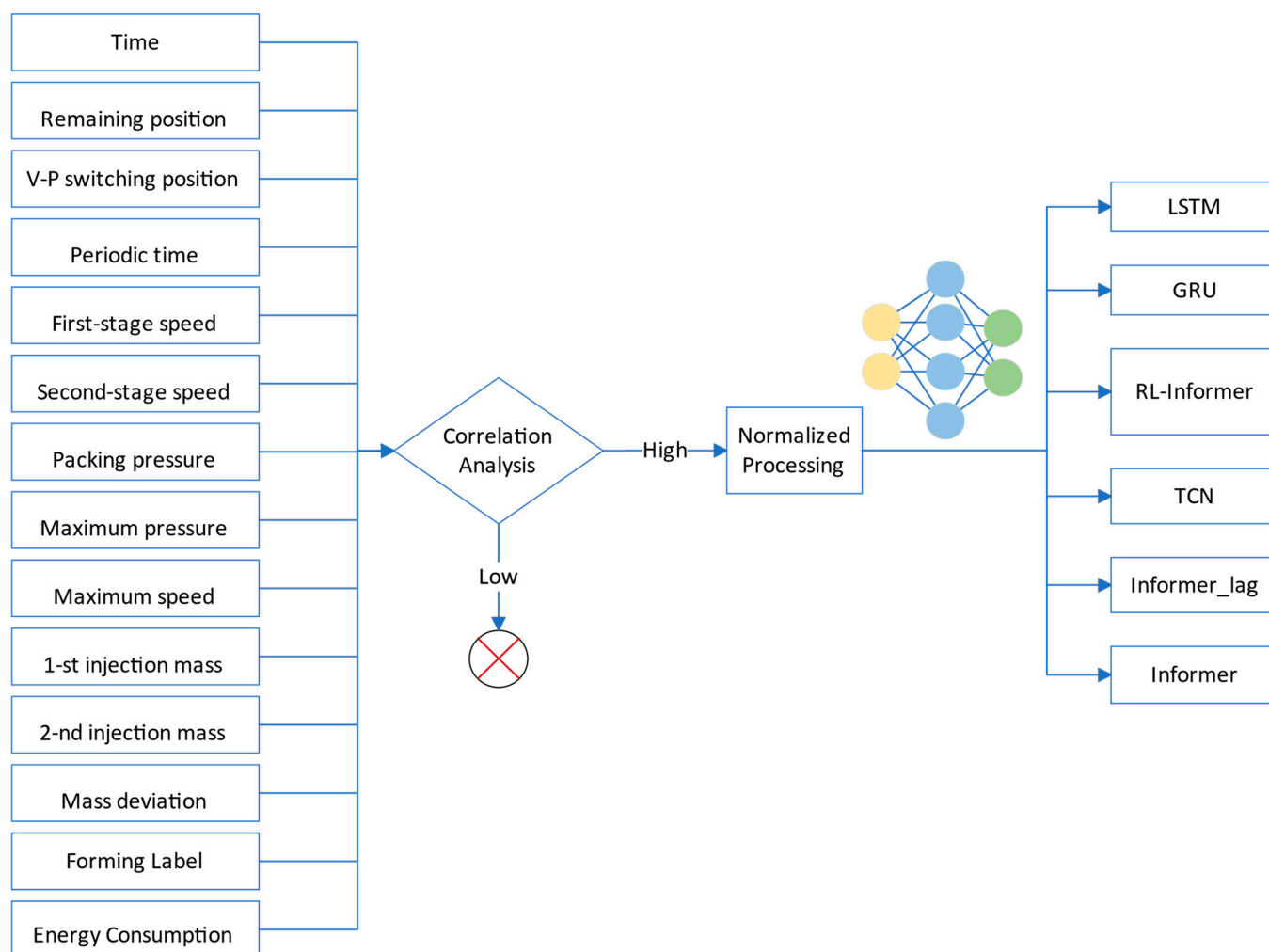


Figure 8. Structure of the energy consumption prediction.

4.2. Performance Comparison

In this experiment, data from 180 injection molding cycles were collected and divided into training, validation, and testing sets in a 7:2:1 ratio. To establish a robust comparative framework for evaluating the performance of the Informer model in predicting energy consumption in injection molding, several advanced temporal prediction models were introduced, including LSTM, GRU, and TCN. These models are well suited for handling complex time series data and capturing long-term dependencies with nonlinear features. By comparing the prediction outcomes of these models, the accuracy and generalization capabilities of the Informer and RL-Informer models can be comprehensively assessed, thereby providing a strong basis for determining their superiority in this application.

The GRU is an efficient recurrent neural network designed for capturing long-term dependencies in time series data, offering a balance between accuracy and computational cost [42]. In contrast, the TCN uses dilated convolutions to model long-range dependencies, providing faster training and improved scalability for time series forecasting [43].

The prediction curves and corresponding error plots for the GRU and TCN models are shown below.

Figure 9 shows that the GRU and TCN models capture short-term fluctuations well but struggle with larger residual errors in areas of rapid change. However, they can capture the overall change characteristics of the data more accurately. Then, we test the initial Informer model, and the predicting curve and error results are as follows.

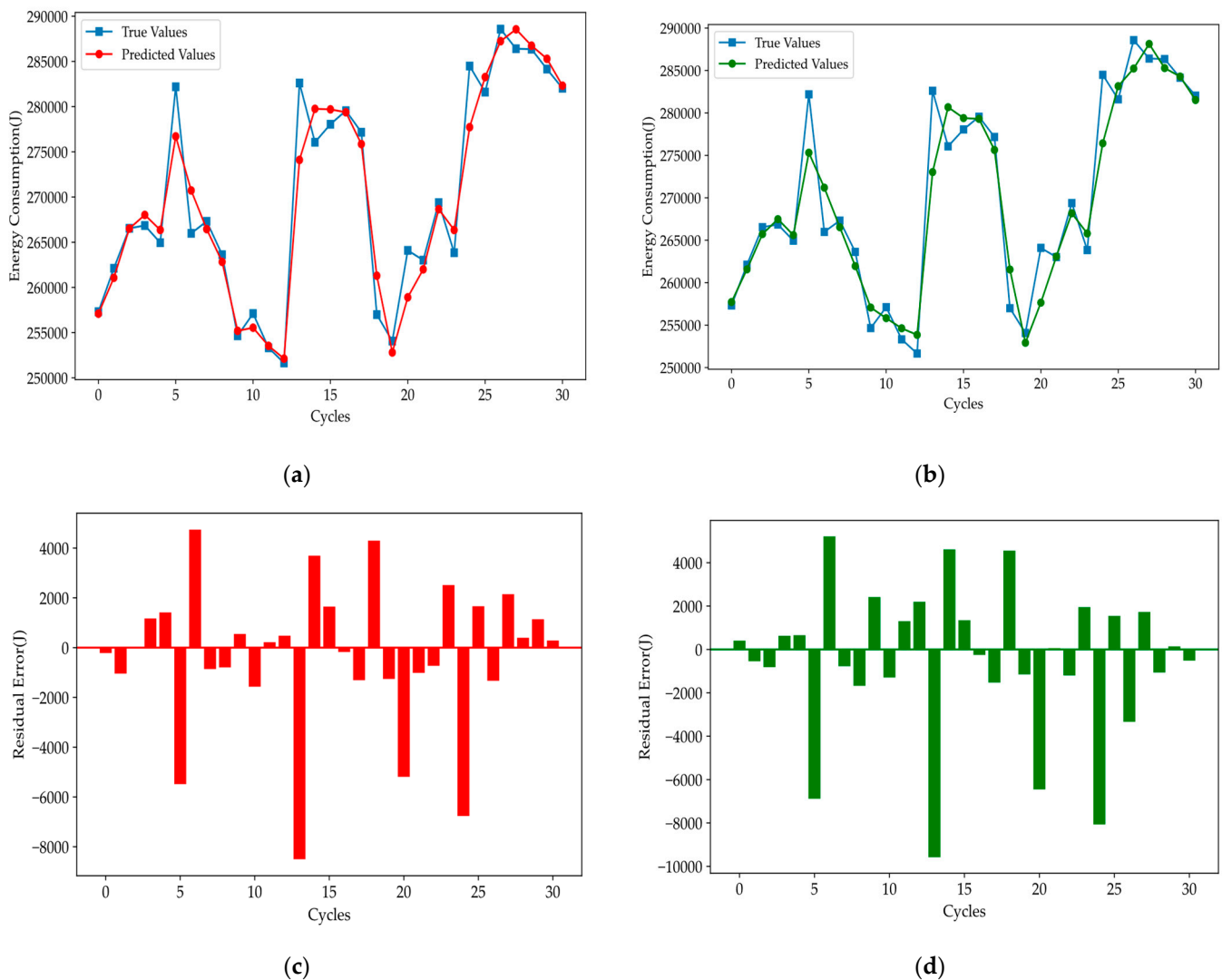
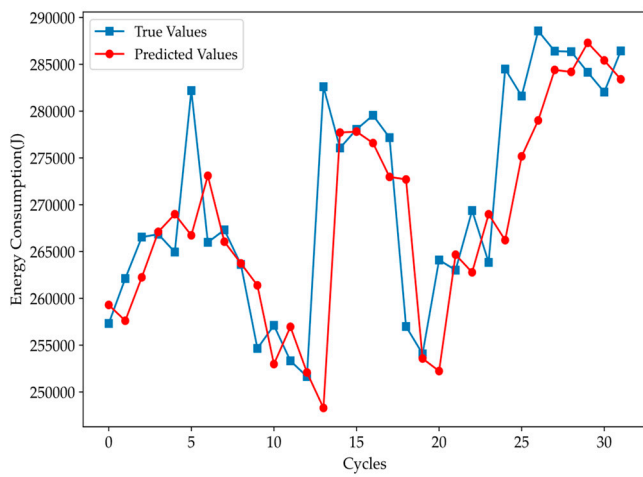
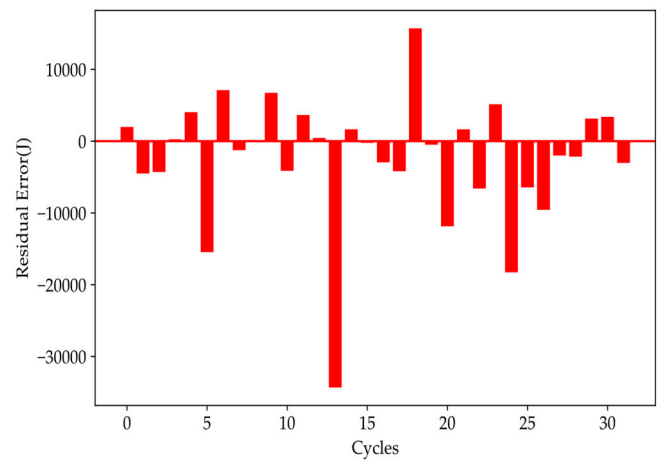


Figure 9. Prediction results for (a) GRU and (b) TCN, and residual errors for (c) GRU and (d) TCN.

Figure 10 shows that, for the Informer model, the predicted values generally track the variation trends of the true values across most cycles. However, the predicted curve exhibits an obvious lag in output, which reduces the overall prediction accuracy. To address this, output lag processing was incorporated into the Informer model, resulting in the new variant, Informer-lag. Furthermore, a comparison with the LSTM model was conducted. The results are shown in Figure 11.

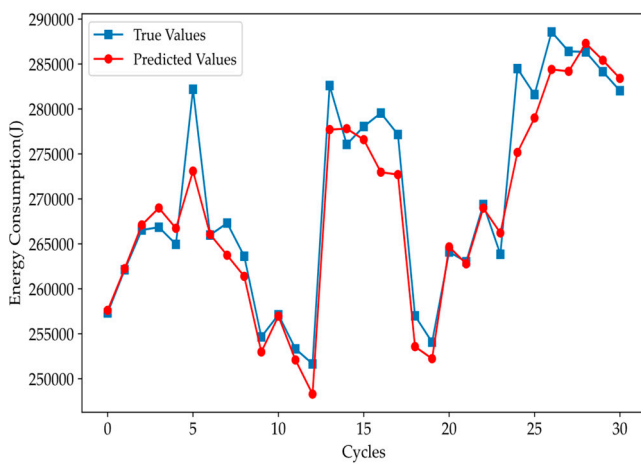


(a)

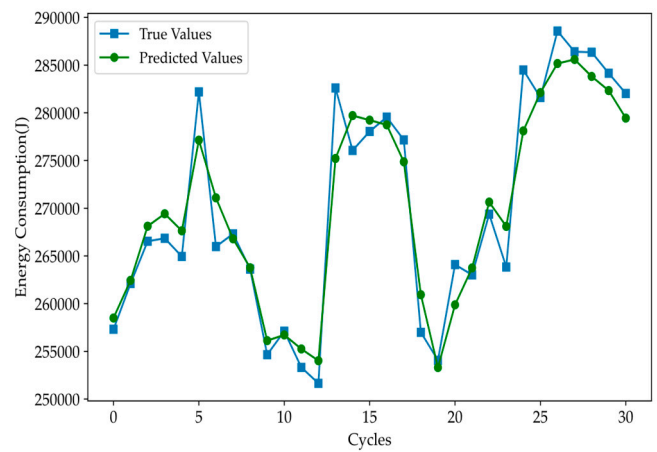


(b)

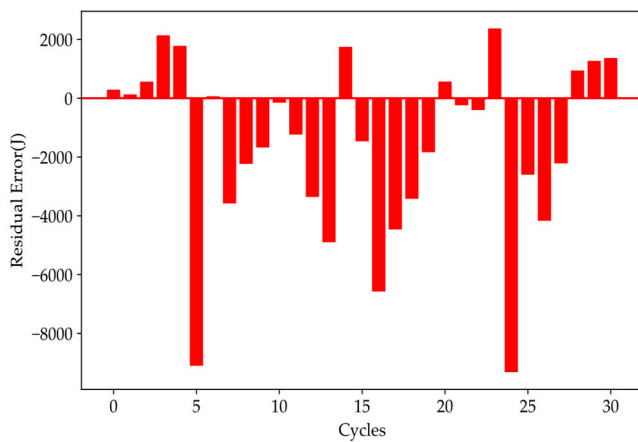
Figure 10. Prediction results: (a) comparison of true and predicted values, (b) residual error of the initial Informer model.



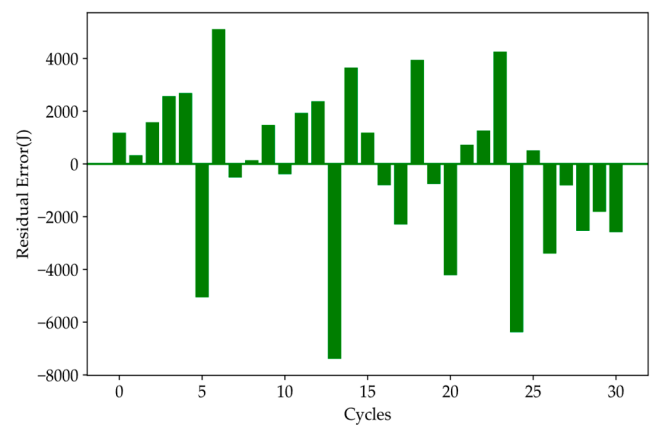
(a)



(b)



(c)



(d)

Figure 11. Prediction results for (a) Informer_lag and (b) LSTM, and residual error for (c) Informer_lag and (d) LSTM.

Figure 11 shows that the predicted curve of Informer_lag closely follows the true values, with significant reductions in both extreme values and mean errors, indicating a notable improvement in the model performance. However, when compared to the prediction curve of the LSTM model, the Informer model with anti-lag treatment still showed slightly lower tracking accuracy. To address this, we applied a rolling learning prediction technique to enhance the original Informer model. The prediction results of the improved RL-Informer model are presented in Figure 12.

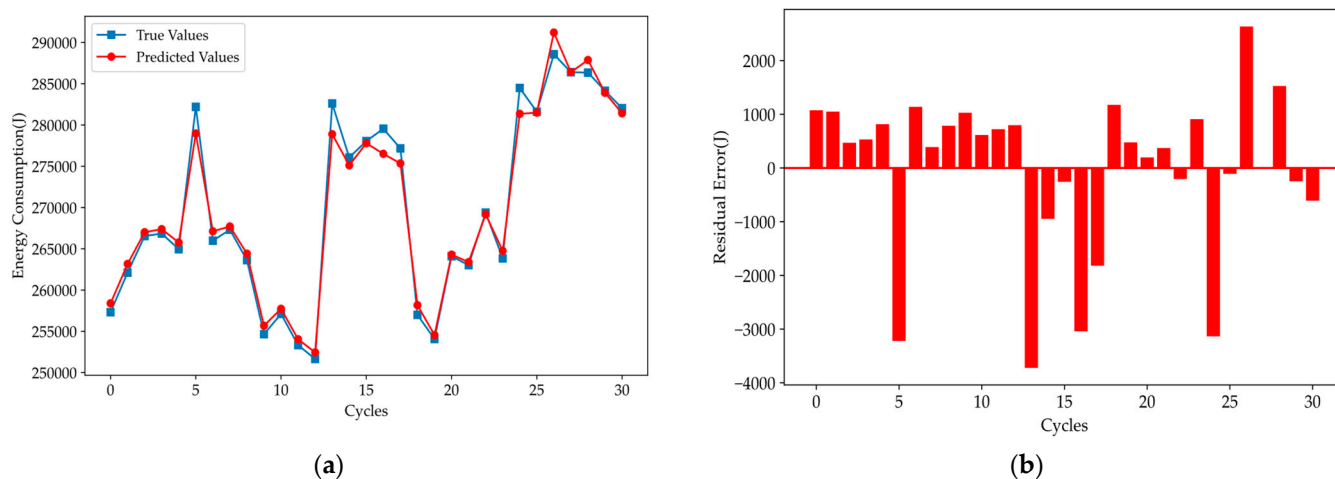


Figure 12. Prediction results: (a) comparison of true and predicted values, (b) residual error of the RL-Informer model.

As shown in Figure 12, the prediction accuracy of the improved RL-Informer model has significantly increased. The model demonstrates precise tracking of the true energy consumption trends across all cycles, with a noticeable reduction in both the extreme value and mean errors. This improvement is further highlighted by the residual error plot, where the error magnitudes are substantially lower compared to previous models.

The RL-Informer model exhibits superior performance in capturing the complex dynamics of energy consumption, effectively reducing prediction discrepancies. This enhanced accuracy, particularly in the reduction of residual errors, underscores the model's robustness and its ability to generalize well across different segments of the time series. Compared to the GRU, TCN, and LSTM models, the RL-Informer model delivers more reliable forecasts with minimized prediction errors, making it a highly effective solution for time series energy consumption forecasting.

4.3. Performance Analysis

Table 2 presents the evaluation of different models in energy consumption prediction, with RL-Informer clearly outperforming all others. It achieves the lowest root mean square error (RMSE, 0.0758) and mean absolute error (MAE, 0.0562), demonstrating the smallest prediction error, and the highest coefficient of determination (R^2 , 0.9831) not only demonstrates the model's superiority in explaining the variance within the data but also indicates the potential of the RL-Informer model to deliver accurate predictions in dynamic and complex industrial applications. This is particularly significant for energy management and optimization, as accurate predictions can help identify potential energy-saving opportunities, thereby enhancing the overall production efficiency. Additionally, its normalized root mean square error (NRMSE, 13.01%) highlights its robustness, outperforming both traditional models and pure Informer-based models across all key metrics.

Table 2. Prediction index of the timing prediction model.

Model	RMSE	MAE	R ²	NRMSE
GRU	0.1492	0.1034	0.9345	25.59%
TCN	0.1752	0.1220	0.9096	30.06%
LSTM	0.1540	0.1219	0.9302	26.42%
Informer	0.4614	0.2988	0.3923	22.04%
Informer_lag	0.1741	0.1258	0.9108	29.87%
RL-Informer	0.0758	0.0562	0.9831	13.01%

To facilitate a clearer comparison of the model performance, the evaluation metrics from Table 2 were normalized, with R² replaced by 1/R² to ensure consistency, whereby lower values indicate better performance in the models. The radar chart in Figure 13 visualizes these results, comparing six time series forecasting models—GRU, TCN, LSTM, Informer, Informer_lag, and RL-Informer—across four key metrics: RMSE, MAE, 1/R², and NRMSE. The radar chart reveals each model’s strengths and weaknesses at a glance. The Informer model notably occupies the largest area on the radar chart, with the RMSE, MAE, and 1/R² values close to the outer boundary, indicating significant errors in capturing the overall data trends and magnitudes. With the incorporation of lag-based adjustments, Informer_lag demonstrates considerable improvements in RMSE, MAE, and 1/R², indicating enhanced data fitting capabilities and increased sensitivity to trend variations. Nevertheless, the higher NRMSE observed for Informer_lag suggests limitations in accurately capturing relative errors, revealing potential weaknesses in its robustness against data variability. Despite these adjustments, the overall accuracy of the lag-enhanced Informer remains comparable to that of the TCN model while still trailing slightly behind the GRU and LSTM models in terms of overall predictive accuracy.

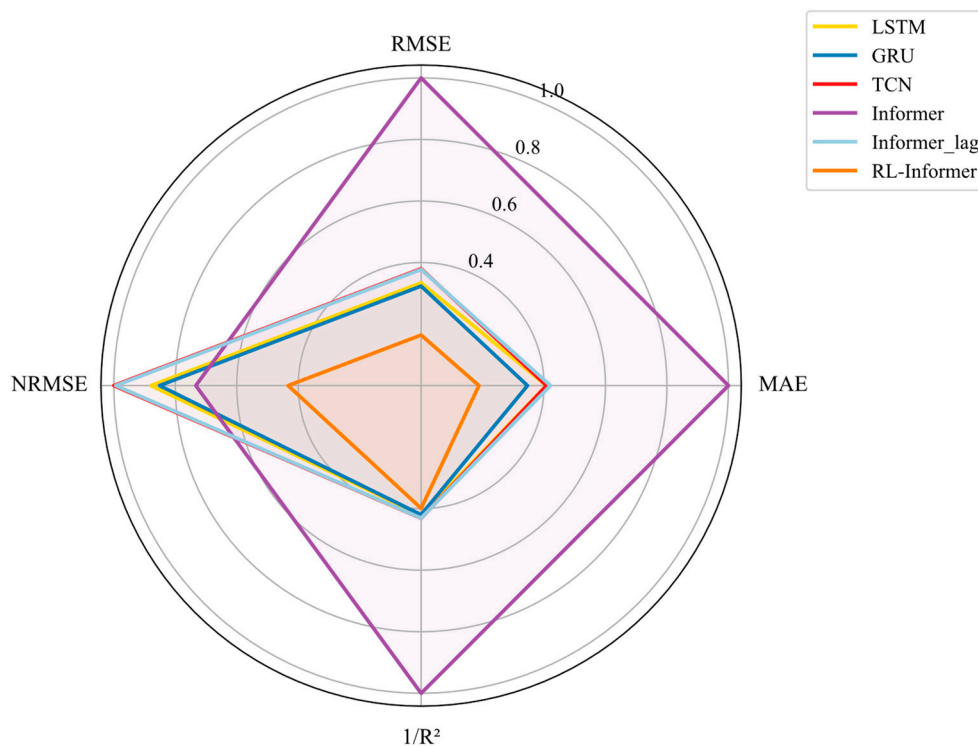


Figure 13. The model performance in the radar chart.

Further improvements were achieved by implementing rolling learning techniques, which led to substantial reductions across all the evaluated metrics, underscoring the model’s enhanced accuracy and adaptability. This trend is especially evident in RL-

Informer, which displays the smallest area on the radar chart, with all metrics converging closer to the center. These results underscore RL-Informer's superior performance, highlighting its efficacy in minimizing prediction errors comprehensively.

5. Conclusions

This study introduces the RL-Informer model as a robust solution for accurately predicting energy consumption in the injection molding process. By enhancing the baseline Informer model with output lag adjustment and integrating a rolling prediction mechanism, the RL-Informer demonstrated significantly improved predictive accuracy. Compared to conventional temporal models such as GRU, TCN, and LSTM, the RL-Informer achieved notable reductions in RMSE (0.0758), MAE (0.0562), and NRMSE (0.1301), alongside an R^2 increase to 0.9831. These results underscore the RL-Informer model's exceptional adaptability and precision, establishing it as a highly effective tool for real-time energy consumption forecasting in dynamic industrial settings.

The results confirm the reliability and accuracy of the RL-Informer model, making it a robust tool for energy consumption prediction in the injection molding process. The integration of rolling prediction has proven effective in optimizing the model's performance, providing a practical and reliable approach for energy monitoring and prediction in industrial applications.

Although the RL-Informer model demonstrated a strong performance in energy consumption prediction for injection molding processes, the dataset used was relatively small and focused on a specific machine type, which may limit the model's generalizability to other contexts. Additionally, while correlation-based variable selection reduced the computational load, it may have overlooked complex nonlinear interactions among the variables. Future work could focus on expanding the dataset and applying more advanced variable selection techniques to further enhance the model prediction accuracy and applicability across different production environments. Moreover, the current study focuses on predicting energy consumption as a single target variable. In real-world industrial processes, multiple variables such as product quality, cycle time, and machine health play critical roles in the overall performance. Future research could extend the RL-Informer model to predict multiple variables simultaneously, allowing for a more comprehensive optimization of the production process. Multi-output prediction models, including multi-task learning approaches, should be explored to simultaneously forecast energy consumption alongside other key metrics. By doing so, researchers can gain deeper insights into how various factors interact and impact the efficiency and effectiveness of the injection molding process.

Author Contributions: J.H.: Writing—Original Draft and Funding Acquisition; Y.L.: Writing—Original Draft and Methodology; X.L.: Investigation, Validation, and Resources; Y.D.: Software and Visualization; F.H.: Methodology and Data Curation; S.P.: Writing—Review and Editing, Funding Acquisition, and Formal Analysis. All authors have read and agreed to the published version of the manuscript.

Funding: This study received financial support from the Project of Guangdong Science and Technology Innovation Strategy (grant number STKJ202209065) and STU Scientific Research Foundation for Talents (grant numbers NTF20019 and NTF23037).

Institutional Review Board Statement: Not applicable.

Data Availability Statement: Data will be made available on request.

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

Abbreviations

The following abbreviations are used in this manuscript:

RL-Informer	Rolling Learning Informer
NRMSE	Normalized Root Mean Square Error
RMSE	Root Mean Square Error

MAE	Mean Absolute Error
R ²	coefficient of determination
GRU	Gated Recurrent Unit
TCN	Temporal Convolutional Networks
LSTM	Long Short-Term Memory
ABS	Acrylonitrile-Butadiene-Styrene co-polymers
PBT	Polybutylene terephthalate
KNN	K-nearest neighbor
NLP	Natural Language Processing
RNN	Recurrent Neural Networks
CNN	Convolutional Neural Networks
TFT	Temporal Fusion Transformer
MPC	Model Predictive Control

References

1. Our World in Data. Available online: <https://ourworldindata.org/grapher/global-plastics-production> (accessed on 10 June 2023).
2. Elduque, A.; Elduque, D.; Pina, C.; Clavería, I.; Javierre, C. Electricity Consumption Estimation of the Polymer Material Injection-Molding Manufacturing Process: Empirical Model and Application. *Materials* **2018**, *11*, 1740. [CrossRef] [PubMed]
3. Mianehrow, H.; Abbasian, A. Energy Monitoring of Plastic Injection Molding Process Running with Hydraulic Injection Molding Machines. *J. Clean. Prod.* **2017**, *148*, 804–810. [CrossRef]
4. Khosravani, M.R.; Nasiri, S. Injection Molding Manufacturing Process: Review of Case-Based Reasoning Applications. *J. Intell. Manuf.* **2020**, *31*, 847–864. [CrossRef]
5. Singh, G.; Verma, A. A Brief Review on Injection Moulding Manufacturing Process. *Mater. Today Proc.* **2017**, *4*, 1423–1433. [CrossRef]
6. Meekers, I.; Refalo, P.; Rochman, A. Analysis of Process Parameters Affecting Energy Consumption in Plastic Injection Moulding. *Procedia CIRP* **2018**, *69*, 342–347. [CrossRef]
7. Pacella, M.; Mangini, M.; Papadia, G. Utilizing Mixture Regression Models for Clustering Time-Series Energy Consumption of a Plastic Injection Molding Process. *Algorithms* **2023**, *16*, 524. [CrossRef]
8. Takasaki, R.; Hibino, H.; Kaifuku, K.; Nishitani, K. Simulation Method for Evaluation of Productivity and Energy Consumption Concerning Production Line for Injection Molding Machines. In Proceedings of the Advances in Production Management Systems. The Path to Intelligent, Collaborative and Sustainable Manufacturing: IFIP WG 5.7 International Conference, APMS 2017, Hamburg, Germany, 3–7 September 2017; Proceedings, Part II; Springer: Berlin/Heidelberg, Germany, 2017; pp. 496–504.
9. Otieno, S.O.; Mwema, F.M.; Mharakurwa, E.T.; Jen, T.-C. Fuzzy Assessment of Process Parameter Interactions on Warpage Defect Modeling in Plastic Injection Molding. *Int. J. Interact. Des. Manuf. IJIDeM* **2024**, 1–14. [CrossRef]
10. Bahij, M.; Labbadi, M.; Cherkaoui, M.; Chatri, C.; Lakrit, S. A Comparison Study of Machine Learning Methods for Energy Consumption Forecasting in Industry. In *Digital Technologies and Applications. ICDTA 2021*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 165–175.
11. Chen, W.-C.; Tai, P.-H.; Wang, M.-W.; Deng, W.-J.; Chen, C.-T. A Neural Network-Based Approach for Dynamic Quality Prediction in a Plastic Injection Molding Process. *Expert Syst. Appl.* **2008**, *35*, 843–849. [CrossRef]
12. Willenbacher, M.; Scholten, J.; Wohlgemuth, V. Machine Learning for Optimization of Energy and Plastic Consumption in the Production of Thermoplastic Parts in SME. *Sustainability* **2021**, *13*, 6800. [CrossRef]
13. Wu, Y.; Feng, Y.; Peng, S.; Mao, Z.; Chen, B. Generative Machine Learning-Based Multi-Objective Process Parameter Optimization towards Energy and Quality of Injection Molding. *Environ. Sci. Pollut. Res.* **2023**, *30*, 51518–51530. [CrossRef]
14. Nazir, A.; Shaikh, A.K.; Shah, A.S.; Khalil, A. Forecasting Energy Consumption Demand of Customers in Smart Grid Using Temporal Fusion Transformer (TFT). *Results Eng.* **2023**, *17*, 100888. [CrossRef]
15. Li, X.; Zhong, Y.; Shang, W.; Zhang, X.; Shan, B.; Wang, X. Total Electricity Consumption Forecasting Based on Transformer Time Series Models. *Procedia Comput. Sci.* **2022**, *214*, 312–320. [CrossRef]
16. Wang, Y.; Peng, J.; Wang, X.; Zhang, Z.; Duan, J. Replacing Self-Attentions with Convolutional Layers in Multivariate Long Sequence Time-Series Forecasting. *Appl. Intell.* **2024**, *54*, 522–543. [CrossRef]
17. Zhou, H.; Zhang, S.; Peng, J.; Zhang, S.; Li, J.; Xiong, H.; Zhang, W. Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtual, 2–9 February 2021; Volume 35, pp. 11106–11115.
18. Kien, H.T.; Minh, P.S.; Duc, T.N.M.; Nhan, D.N. Study of Vibration-Assisted Injection Molding: System Design. In Proceedings of the 2nd Annual International Conference on Material, Machines and Methods for Sustainable Development (MMMS2020), Nha Trang, Vietnam, 10–13 November 2020; Springer: Berlin/Heidelberg, Germany, 2021; pp. 45–51.
19. Baum, M.; Anders, D.; Reinicke, T. Enhancing Injection Molding Simulation Accuracy: A Comparative Evaluation of Rheological Model Performance. *Appl. Sci.* **2024**, *14*, 8468. [CrossRef]

20. Gond, V.K.; Dubey, A.; Rasool, A.; Khare, N. Missing Value Imputation Using Weighted KNN and Genetic Algorithm. In *ICT Analysis and Applications: Proceedings of ICT4SD 2022*; Springer: Berlin/Heidelberg, Germany, 2022; pp. 161–169.
21. Cohen, I.; Huang, Y.; Chen, J.; Benesty, J.; Benesty, J.; Chen, J.; Huang, Y.; Cohen, I. Pearson Correlation Coefficient. In *Noise Reduction in Speech Processing*; Springer: Berlin/Heidelberg, Germany, 2009; pp. 1–4.
22. Sedgwick, P. Spearman's Rank Correlation Coefficient. *BMJ* **2014**, *349*, g7327. [[CrossRef](#)]
23. Vaswani, A. Attention Is All You Need. In Proceedings of the Advances in Neural Information Processing Systems 30 (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017.
24. Elman, J.L. Finding Structure in Time. *Cogn. Sci.* **1990**, *14*, 179–211. [[CrossRef](#)]
25. Zhao, B.; Lu, H.; Chen, S.; Liu, J.; Wu, D. Convolutional Neural Networks for Time Series Classification. *J. Syst. Eng. Electron.* **2017**, *28*, 162–169. [[CrossRef](#)]
26. Adamowski, J.F. River Flow Forecasting Using Wavelet and Cross-Wavelet Transform Models. *Hydrol. Process. Int. J.* **2008**, *22*, 4877–4891. [[CrossRef](#)]
27. Yang, Z.; Liu, L.; Li, N.; Tian, J. Time Series Forecasting of Motor Bearing Vibration Based on Informer. *Sensors* **2022**, *22*, 5858. [[CrossRef](#)]
28. Child, R.; Gray, S.; Radford, A.; Sutskever, I. Generating Long Sequences with Sparse Transformers. *arXiv Prepr.* **2019**, arXiv:1904.10509.
29. Peng, G.; Huang, Y.; Li, H.; Wu, Z.; Wang, D. LSDNet: A Lightweight Self-Attentional Distillation Network for Visual Place Recognition. In Proceedings of the 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Kyoto, Japan, 23–27 October 2022; pp. 6608–6613.
30. Wang, Y.; Yao, H.; Zhao, S. Auto-Encoder Based Dimensionality Reduction. *Neurocomputing* **2016**, *184*, 232–242. [[CrossRef](#)]
31. Varoquaux, G.; Raamana, P.R.; Engemann, D.A.; Hoyos-Idrobo, A.; Schwartz, Y.; Thirion, B. Assessing and Tuning Brain Decoders: Cross-Validation, Caveats, and Guidelines. *NeuroImage* **2017**, *145*, 166–179. [[CrossRef](#)] [[PubMed](#)]
32. Wang, X.; Sun, S.; Xie, L.; Ma, L. Efficient Conformer with Prob-Sparse Attention Mechanism for End-to-Endspeech Recognition. *arXiv Prepr.* **2021**, arXiv:2106.09236.
33. Liu, Y.; Shu, X.; Yu, H.; Shen, J.; Zhang, Y.; Liu, Y.; Chen, Z. State of Charge Prediction Framework for Lithium-Ion Batteries Incorporating Long Short-Term Memory Network and Transfer Learning. *J. Energy Storage* **2021**, *37*, 102494. [[CrossRef](#)]
34. Guo, N.; Zhang, X.; Zou, Y.; Guo, L.; Du, G. Real-Time Predictive Energy Management of Plug-in Hybrid Electric Vehicles for Coordination of Fuel Economy and Battery Degradation. *Energy* **2021**, *214*, 119070. [[CrossRef](#)]
35. Alam, S.; Liu, L.; Yan, M.; Zhang, M. Fedrolex: Model-Heterogeneous Federated Learning with Rolling Sub-Model Extraction. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 29677–29690.
36. Lim, B.; Zohren, S. Time-Series Forecasting with Deep Learning: A Survey. *Philos. Trans. R. Soc. A* **2021**, *379*, 20200209. [[CrossRef](#)]
37. Mentaschi, L.; Besio, G.; Cassola, F.; Mazzino, A. Problems in RMSE-Based Wave Model Validations. *Ocean Model.* **2013**, *72*, 53–58. [[CrossRef](#)]
38. Chai, T.; Draxler, R.R. Root Mean Square Error (RMSE) or Mean Absolute Error (MAE)?—Arguments against Avoiding RMSE in the Literature. *Geosci. Model Dev.* **2014**, *7*, 1247–1250. [[CrossRef](#)]
39. Willmott, C.J.; Matsuura, K. Advantages of the Mean Absolute Error (MAE) over the Root Mean Square Error (RMSE) in Assessing Average Model Performance. *Clim. Res.* **2005**, *30*, 79–82. [[CrossRef](#)]
40. Nagelkerke, N.J. A Note on a General Definition of the Coefficient of Determination. *Biometrika* **1991**, *78*, 691–692. [[CrossRef](#)]
41. Yu, L.; Pan, Y.; Wu, Y. Research on Data Normalization Methods in Multi-Attribute Evaluation. In Proceedings of the 2009 International Conference on Computational Intelligence and Software Engineering, Wuhan, China, 11–13 December 2009; pp. 1–5.
42. Cho, K. Learning Phrase Representations Using RNN Encoder-Decoder for Statistical Machine Translation. *arXiv Prepr.* **2014**, arXiv:1406.1078.
43. Lara-Benítez, P.; Carranza-García, M.; Luna-Romera, J.M.; Riquelme, J.C. Temporal Convolutional Networks Applied to Energy-Related Time Series Forecasting. *Appl. Sci.* **2020**, *10*, 2322. [[CrossRef](#)]

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.