

Autoformer and Transformer-Based Neural Networks for State Estimation of Third-Order Liquid Level Dynamics in Chemical Processes

Abstract

One of the important variables in chemical process industries is the liquid level. The challenges of estimating the third-order liquid level system are: the system exhibits non-linear behavior, the system has coupled states where changes in one tank directly affect others, uncertainty where there are unknown disturbances, and prediction complexity. Yet, with advances in artificial intelligence (AI), modeling and estimating the liquid third-order levels has become a reality. The present work attempts to construct an accurate AI model to estimate the levels of third-order liquid level tanks utilizing the available inlet liquid flow rate and the bottom level. In order to estimate the dynamic level responses as a function of time for three liquid level tanks connected in series, this article presents the training of several types of artificial neural networks (ANNs): basic ANNs with different settings, autoformer, transformer regression, and attentive interpretable tabular learning (TabNet). These ANNs are trained to estimate the two unmeasured top and middle levels using the simulated response data of the bottom tank's level (measured level) for different step changes in the inlet water feed flow rate ranges from 0.1 to 3 liters/min. The findings, using the coefficient of determination (R^2), demonstrated that autoformer produces the best closed-fit responses; $R^2 = 99.77\%$ for the top-level estimation and $R^2 = 99.86\%$ for the middle-level estimation. Advanced AI techniques act as highly computationally efficient surrogate models for conventional complex models based on nonlinear conservation without the need for heavy computing power for process control and optimization.

Keywords: Artificial intelligence (AI); level estimation; artificial neural networks (ANN); transformers; machine learning

1. Introduction

AI was applied for modeling and control of a wide range of chemical engineering applications, such as 1) modeling of the heat exchanger for steady-state outlet temperature predictions [1], 2) modeling the non-linear relationship between the acid-base concentrations and the pH of solution [2], 3) modeling and predictive control of two well-mixed non-isothermal chemical reactors [3], 4) applying the MPC-ANN technique for the level control of interacted coupled tanks [4], and 5) many other applications [5]. The techniques of AI for engineering processes have another application for predicting variables that do not have online sensors on which they develop models to control the process quality. Many process variables such as temperature, pressure, level, pH, and mass flow rate can be easily measured online. On the other hand, other variables are challenging or do not exist, depending on an offline laboratory analysis. Therefore, an artificial neural network (ANN) was used to estimate readings (called virtual sensors) for the non-available measurement and hence improve the process control quality in the air-fuel system [6]. As observed by the literature. There is less work to deal with the application of neural networks and AI for modeling and estimating levels for control purposes.

AI is an exciting, powerful mathematical tool with a wide range of applications in science and engineering fields. AI applies statistics to produce excellent estimations of complex functions. AI is considered a linear and/or non-linear framework based on the human brain's functions through interconnected neurons, which can learn tasks from training experience [7]. One of the recent advancements in AI is developing a complicated ANN that is known as a transformer [8]. The transforms are another form of deep learning model utilized to solve complex problems, especially in natural language processing (NLP). They are also adopted in other fields, such as time-series prediction/forecasting and computer vision. The difference between transformers and other deep learning algorithms, such as recurrent neural networks (RNNs), is that transformers depend mainly on self-attention mechanisms where the entire data can be processed in parallel. Parallelism is significant because it solves the problem of long-range dependencies presented in traditional deep-learning algorithms that process the data sequentially, avoiding vanishing gradient problems. In addition, transformers assign weight to the input sequence, which allows them to capture the contextual relationships across the entire input, which makes them effective in tasks where the relationships between distant elements are essential, e.g., time series and sequential state estimation problems (the problem in hand). Islam, et.al. [8] presented a comprehensive AI complex models survey and some of their applications. They surveyed the work from 2017 to 2022 and identified over six hundred models based on the transformer architecture. The study attempted to categorize these models for different applications. They concluded that models such as BERT, GPT, and T5 are suitable for handling large NLP data, whereas, for computer vision, vision transformers (ViTs) are great for addressing the limitations of convolutional neural networks (CNNs). The study also considered other applications, such as audio and speech processing.

AI has potential uses in engineering and science applications, including solving the following: multivariable and non-linear processes, models prediction, biomedical engineering, food, solar and energy systems, communication, materials, and coating, prediction of mechanical properties, material consumption rate, prediction control, reaction kinetics and thermodynamics, process analysis and optimization, computer security, etc. [8-20]. For instance, Khadom [18] tested different ANN designs to predict zinc consumption rates. The authors utilized the STATISTICA 10 software to implement the ANN models, which include a linear regression model, ANN, and radial basis functions (RBF). Their experiments concluded that the best-performing model is the ANN, which had a structure of four inputs, one hidden of seven nodes, and one output, i.e., for the predicted single value. The study indicated that the best model achieves a correlation coefficient of 99.46%.

Many advanced control techniques assume that process variables are available as measurements. In practice, however, it is usually impossible to measure all the variables because of the cost and difficulties of obtaining reliable measurements. Thus, of very great importance is the estimation of critical physical variables within a process from more easily available measurements. Examples are the estimation of biomass in the fermentation process, gas, and chemical compositions, and temperature inside the large metal blocks. The techniques that estimate the system's variables are State Observer and Extended Kalman Filter [21]. However, both techniques require developing mathematical models and complex mathematical derivations. One of the main advantages of AI is that it does not require the development of mathematical models. It learns system input and output data and the sequence detected patterns. In addition, because of its great capacity to handle

discrete data and its high capacity to store data in memory, it has earned outstanding ratings for modeling adoption [18].

The present work is an attempt to construct an accurate AI model to estimate the levels of third-order liquid level tanks utilizing the available inlet liquid flow rate and the bottom level. The study compares several ANN models, including basic ANN designs and a few transformer models. Then, it compares and evaluates them to find the best estimating model. According to the author, no previous works have been carried out in applying AI in predictions of large-order liquid-level systems. Such estimation can be advantageous in the use of advanced control techniques.

2. Methodology

A third-order liquid level system, shown in Figure 1, is considered for level estimations using AI models. Water supplied is fed to the top vessel, and there are two available measurements for the level at the bottom liquid level and water feed flow rate. The mathematical model of the system is given in Equation (1) as follows:

$$\begin{pmatrix} \frac{dWL1}{dt} \\ \frac{dWL2}{dt} \\ \frac{dWL3}{dt} \end{pmatrix} = \begin{pmatrix} -0.631 & 0 & 0 \\ 0.615 & -0.615 & 0 \\ 0 & 0.615 & -0.615 \end{pmatrix} \begin{pmatrix} WL1 \\ WL2 \\ WL3 \end{pmatrix} + \begin{pmatrix} 3.19 \\ 0 \\ 0 \end{pmatrix} u \quad (1)$$

where WL1; WL2; WL3 are the levels; top, middle, bottom and u is the water feed flow rate respectively. The constant coefficients shown in Equation (1) (e.g., -0.631, 0.615, 3.19) are lumped parameters that model the relationship between the tank cross sectional areas and the valve resistance coefficients. The data set was generated from a steady state condition where all levels were set to zero. The differential equations were solved for the simulation time using a fixed integration step size of $\Delta t = 0.01$ seconds with a numerical integration method (e.g., Runge-Kutta 4th order).

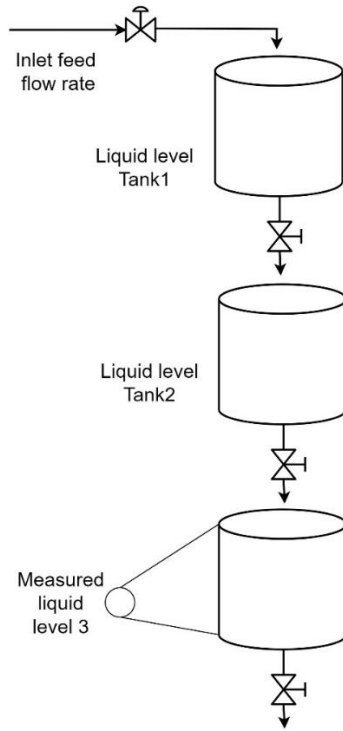


Figure 1. Schematic diagram of the third-order liquid level system

2.2. Basic Artificial Neural Network (ANN)

The first experiment to predict the first two water levels (i.e., WL1 and WL2) using the third water level (i.e., WL3) and the water feed flow rate (i.e., u) to use different designs of the basic ANN. In this experiment, three different ANN architectures are trained and tested.

Figures 2 illustrate the three architectures of basic ANN. Figure 2 (a) shows the first architecture (A1) where only one hidden layer is used. On the other hand, Figure 2 (b) gives the second architecture (A2), which includes three hidden layers. A2 is considered deeper, and that is designed to see the effectiveness of adding hidden layers to the design. Lastly, the third architecture (A3) is given in Figure 2 (c). A3 is the deepest design with seven hidden layers. These three architectures have different settings, such as the optimization algorithm. These settings are listed in Table 1.

Table 1. Parameter settings for the three ANN architectures (A1, A2, and A3)

Settings	A1	A2	A3
Data normalization method	Min-max normalization	Min-max normalization	Robust Scaling
Optimization algorithm	Adam	Stochastic gradient descent (SGD)	Adam
Error function	Mean Square Error (MSE)	Mean Square Error (MSE)	Mean Square Error (MSE)
Learning rate	0.001	0.001	0.001
Epochs	1000	1000	2000

The hyperparameters in Table 1 were not selected randomly – they were actually chosen. Instead, they are the individually optimized settings that must be used to converge each particular network depth. For example, Robust Scaling is needed for network A3 to avoid vanishing gradients and a higher learning rate is needed.

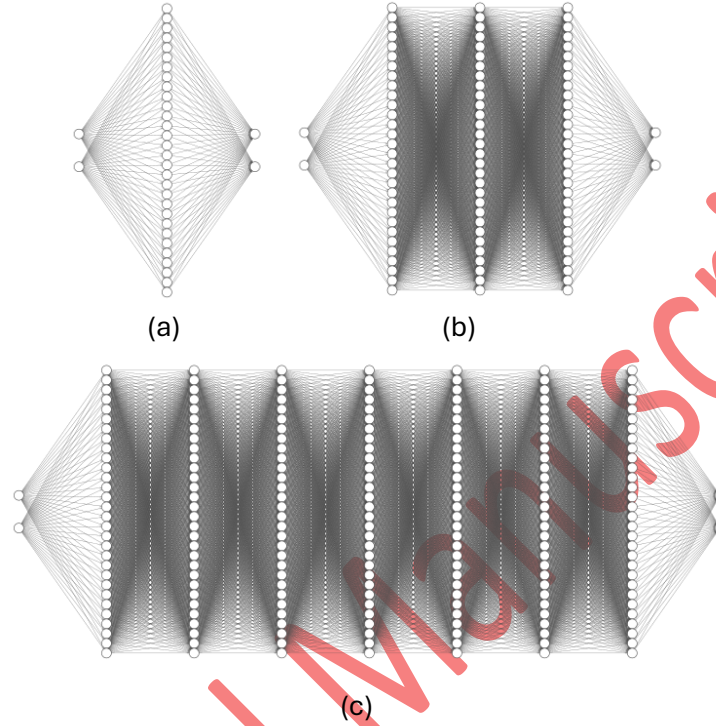


Figure 2. Three ANN architectures: (a) A1 with one hidden layer, (b) A2 with three hidden layers, and (c) A3 with seven hidden layers. First layer is for the inputs, i.e., WL3 and u , whereas the last layer is the outputs, i.e., WL1 and WL2

The different architectures are used to see how different settings affect the outcome of the predictions. In the results section, they will be compared to see if they are sufficient to solve the problem of this study. The results of this test will be graphical (showing the results in a graphical form) and measured with the coefficient of determination regression score function ($r2_score$). The $r2_score$ function is calculated using Equation (2) as follows:

$$R^2 = 1 - \frac{RSS}{TSS} \quad (2)$$

Where R^2 is the coefficient of determination, RSS denotes the sum of squares of residuals and TSS gives the total sum of squares.

2.3. Transformer Neural Networks

This section explains the transformer models tested and compared in this study, starting with the autoformer, then the transformer regression, and then the attentive interpretable tabular learning (TabNet).

2.3.1. Autoformer

The autoformer [22] is an adaptation of transformer architecture, especially transformers used for time series forecasting. Although traditional transformers are powerful, they might not be suited for time series forecasting because they primarily focus on all input features simultaneously. This focus might lead to overfitting and inefficiency when handling continuous time series data. The autoformer, on the other hand, separates the input into trends and seasonal components in blocks. This allows the model to capture the temporal patterns effectively. In terms of liquid level dynamics, the "trend" is the long-term tank volume change, which is a steady process at low frequencies. The "seasonal" component represents the high frequency, immediate, transient and coupled dynamic responses which happen on a moment-to-moment basis after a step-change in the feed valve.

The autoformer structure, as shown in Figure 3, is similar to original transformers, where it consists of an encoder and decoder; however, the autoformer is modified to handle sequential data. Each encoder contains self-attention and a feedforward network to handle the sequential input. At the same time, the decoder layer is used for cross-attention to integrate what comes from the encoder and predict future values. The decoder consists of corporate the auto-correlation and series decomposition blocks, as well as feed-forward layers.

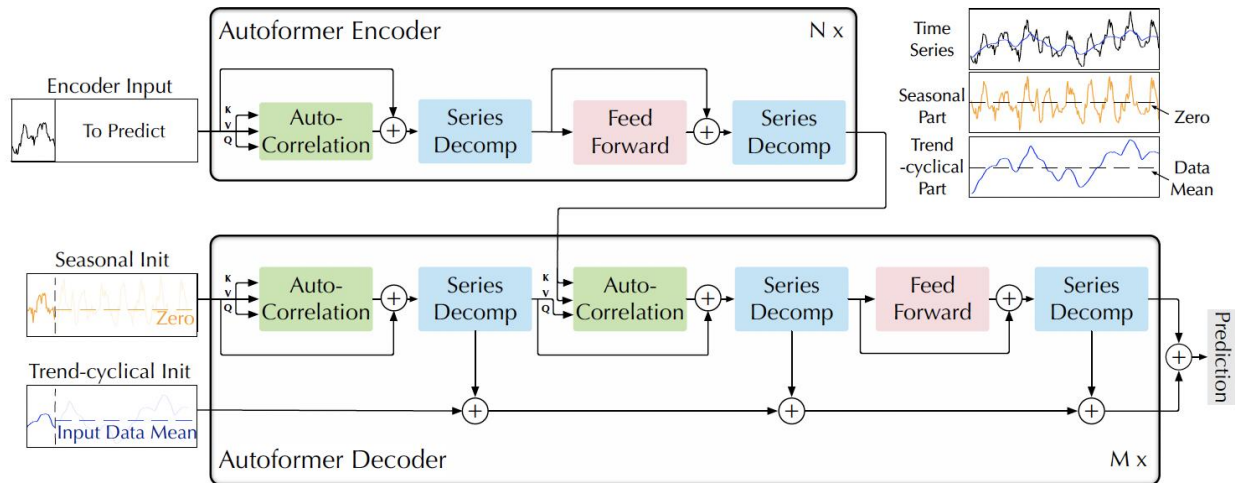


Figure 3. Autoformer structure [22]

Based on the structure given in Figure 3, the autoformer is well suited for the problem in hand with the time series given u and WL3 as input and predicts the future sequence of WL1 and WL2. By decomposing the data into trends and seasonal parts, the model can easily learn generalized data representations and accurate predictions.

2.3.2. Transformer Regression

As can be seen in Figure 4, the transformer regression model implements a transformer-based regression architecture proposed by [23], and it is adopted for what is called sequence-to-sequence learning tasks. It fits problems in certain input sequences and predicts some output sequences. The transformer uses self-attention layers for dependencies in sequential data. Again, the transformer handles the sequences in parallel, which will be able to improve the training and focus on the long-

range dependencies. The proposed model consists of an encoder and a decoder. The encoder plays an important role when it processes the input through multiple layers; those layers utilize self-attention and feedback networks in which they allow each element to attend to each other. However, the decoder is responsible for generating predictions out of the encoder output as well as the previous steps of the target sequence. A multi-head attention model is also utilized to attend to different parts of the input sequence. This allows the transformer to capture complex patterns and relationships. The self-attention mechanism allows handling sequential data, which could be useful for time series forecasting and structured prediction.

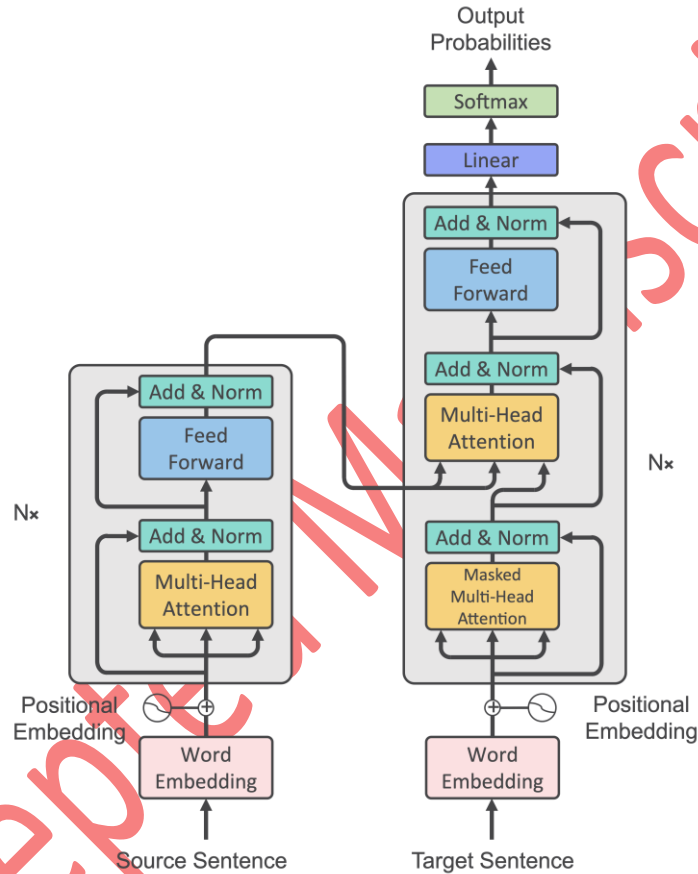


Figure 4. Transformer regression structure [23]

4.3. Attentive Interpretable Tabular Learning (TabNet)

The TabNet model, shown in Figure 5 [24], is similar in structure to the transformers employing the deep learning architecture designed for tabular data. The model combines interpretability with high performance. It utilizes a sequential attention mechanism to be able to focus on different features within each decision step. Therefore, it provides feature selection interoperability through multiple stages. Based on this architecture, the model is suitable for structured data, like the problem at hand. With its differentiable structure, the TabNet model enables integration with deep learning frameworks such as PyTorch. This also makes it adaptable to regression and classification. Many decision steps are included in the TabNet structure, where each step uses a learned attention mechanism for processing a subset of features. Therefore, with repeated processes, the model

builds a comprehensive understanding of the data patterns. Furthermore, the model has sparse feature transforms for the purpose of developing complex non-linear relationships.

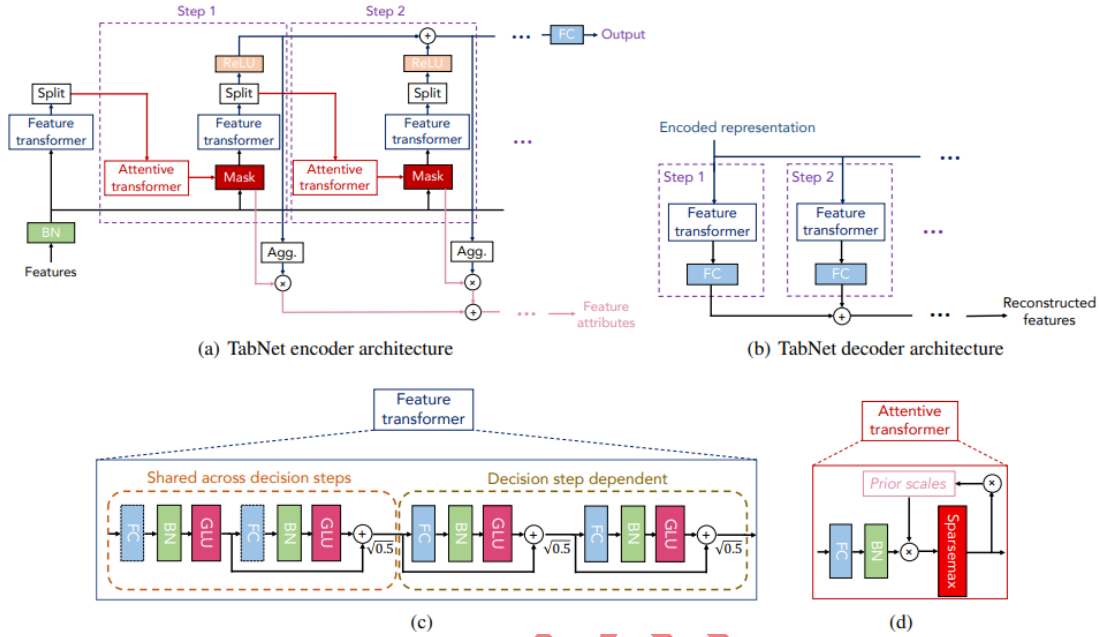


Figure 5. TabNet architecture, a) the encoder architecture, b) the decoder architecture, c) feature transform components, and d) attentive transform components [14]

3. Results and discussion

This section explains the dataset used, shows the results of using the basic ANN models, and compares those results against the complex models explained in the previous section.

3.1. Dataset

The dataset used for training and testing the AI techniques is generated using Equation (1). Since all this data comes from a numerical simulation, and not a physical experimental configuration, all subsequent evaluations show how well the AI can serve as a surrogate model for the underlying math. Generally, AI techniques require large data to be trained more accurately, and for that reason, different values of u have been used to generate as much data as possible. Specifically, the value of u started from 0.1 and is incremented by 0.1 accumulatively until it reaches 3.0. Each change in value generates data instances of more than a hundred records. As a total, 4110 instances are collected and stored as the training and testing data of the AI methods. Table 2 gives statistics on the generated dataset. The 4,110 rows are along sequential step changes; hence the points are autocorrelated. These models were chosen, in particular, for their capability to capture such temporal correlations and long-range dependencies, such as Transformers and deep ANNs.

Table 2. Generated dataset for AI training and testing.

Item	Value
Number of instances	4110
Number of columns	5
Columns names	Time, WL1, WL2, WL3, and u

Inputs to be used	WL3 and u
Outputs to be predicted	WL1 and WL2
WL1 (std, min, max)*	4,363, 0, 140, 10,166
WL2 (std, min, max)*	4,403, 0, 0.19, 10,310
WL2 (std, min, max)*	4,740, 0, 0.2, 16,110
u (std, min, max)*	0,860, 0, 100, 3,000

* std, min, and max denotes standard deviation, minimum value, and maximum value.

3.2. The Three ANN Architectures and Evaluation Results

After executing the three explained ANN designs, results are compiled and illustrated in Figure 6.

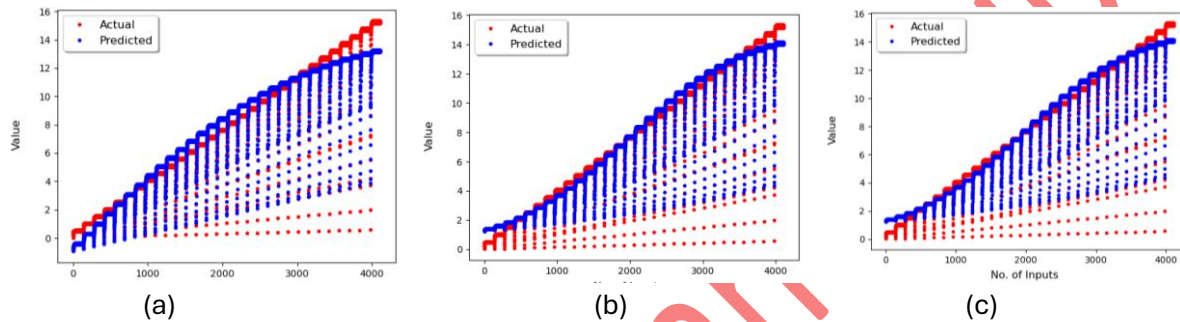


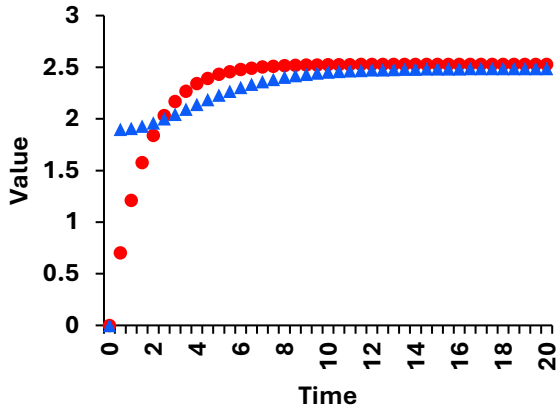
Figure 6. Results of the three ANN designs, (a) results of A1, (b) results of A2, and (c) results of A3

According to the demonstration given in Figure 6, all designs have slightly similar results. The red color dots are for the actual data, whereas the blue color dots are for the predicted values. For this comparison, both predicted values (WL1 and WL2) are under the same color. This is also true for the input values (WL3 and u). However, after careful observation, the results show that A3 is the best compared to the other two after using $r2_score$ function. The coefficient of determination (R^2) for each design across each of the outputs is listed in Table 3. Among the models, model A3 had the best accuracy score, while model A2 (3 hidden layers, $R^2 = 94.90\%$ for WL1) had a lower accuracy score than model A1 (1 hidden layer, $R^2 = 96.18\%$ for WL1). This is suggesting that depth alone is not enough to perform well, most probably because of optimization problems (e.g. using SGD rather than Adam for A2). Thus, the deeper network with a proper tuning gives optimal performance, if the depth of the architecture is matched with the optimal solvers and learning rates.

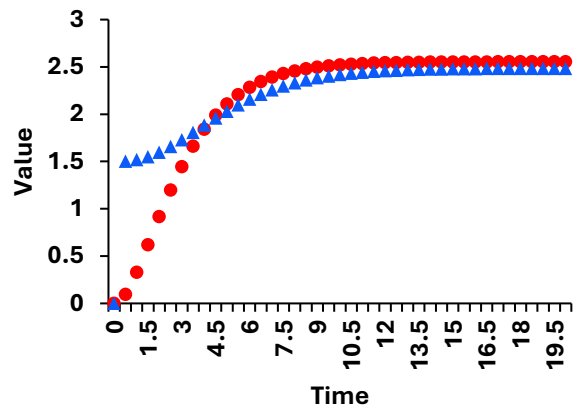
Table 3. Comparison of different ANN designs in terms of R^2 for each predicted value

ANN Design	R^2 for WL1	R^2 for WL2
A1	96,18%	96,68%
A2	94,90%	96,34%
A3	98,23%	98,40%

In order to see the performance of the A3 model, the prediction results for WL1 (left) and WL2 (right) are given in Figures 7, 8, and 9. Figures 7, 8, and 9 respectively show the predictions for u set as 0.5, 1.5, and 3.0. The model's prediction performance seems to be less accurate early on, but then it becomes closer to the actual value in higher values.

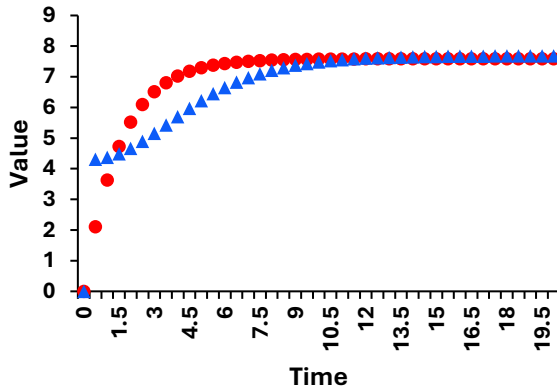


● Actual WL1 ▲ Predicted WL1

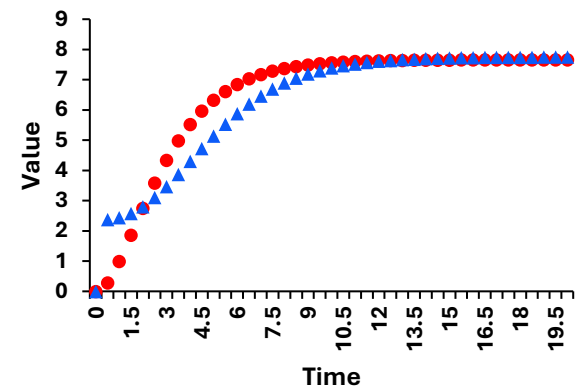


● Actual WL2 ▲ Predicted WL2

Figure 7. ANN (i.e., A3 model) predictions where $u = 0.5$.

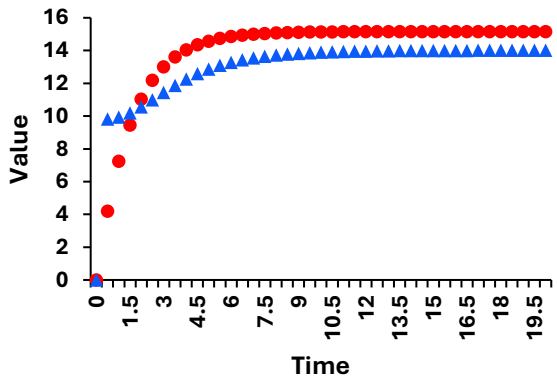


● Actual WL1 ▲ Predicted WL1

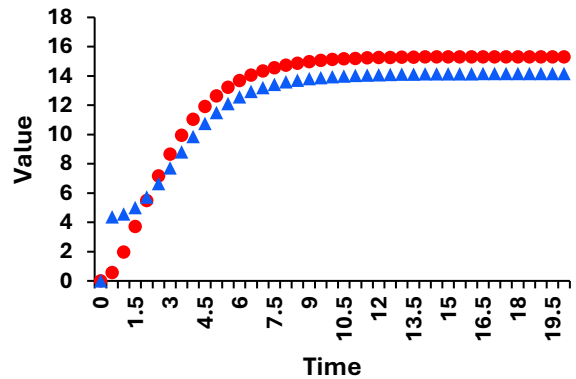


● Actual WL2 ▲ Predicted WL2

Figure 8. ANN (i.e., A3 model) predictions where $u = 1.5$.



● Actual WL1 ▲ Predicted WL1



● Actual WL2 ▲ Predicted WL2

Figure 9. ANN (i.e., A3 model) predictions where $u = 3.0$.

Based on the results explained in this section, a more complex model may be needed for the data of this research.

3.3. Transformers' Settings and Evaluation Results

The autoformer model used in this paper contains many hyperparameters to control the model performance. For instance, the input dimension (input_dim = 2) where the input contains two parameters (u and WL3). Also, the output dimension (output_dim = 2) is set to 2 due to two inputs (WL1 and WL2). At the same time, the embedding dimension (d_model = 128) is set to 128, representing the space size the input and output features could be projected. The embedding dimension is used based on the experimental results where a larger d_model increases the computational cost.

In addition, the attention heads (head = 4) are set to 4, enhancing the model's capability to capture the temporal patterns and focus on different aspects of the input sequence simultaneously. Both the encoder and decoder are designed with two layers to balance between the model complexity and computation. Similarly, the feedforward network was designed with a dimension of 256 (dim_feedforward = 256). Consequently, the drop rate is set to 0.1 (dropout = 0.1) to reduce overfitting and prevent the model from depending on a single feature.

Also, in transformer regression, the parameters are set as follows: the input and output dimensions are set to 2, matching the input and output features found in the data. The hidden dimensions are also set to 128 with 4 attention heads in addition to two layers for the encoder and decoder, respectively. Moreover, the feed-forward network dimension is set to 256 with a dropout rate 0.1 for overfitting prevention. The number of epochs is set to 100 using Adam optimizer with a learning rate of 0.001. The decoder input is zero-initialized for the training process, which is usually suitable for sequence prediction models and target variables that are unknown in the first step.

In TabNet, the pytorch_tabnet library is utilized, where it is designed for tabular data. The training data represents 80% of the data, and the testing data is considered 20% of the data. Features are also standardized with StandardScaler, preventing any feature from dominating the model. The dimensionality of the decision prediction layer and the attention layer are both set to 8, while the number of decision steps is set to 3. The relation factor and the number of independent Gated Linear Units (GLU) layers at each step are set to 1.3 and 2, respectively. In addition, the number of shared Gated Linear Units (GLU) at each step is set to 2. Moreover, the model is trained with 100 epochs, and the batch size is set to 32 with a virtual batch size of 16.

The transformer models, i.e., autoformer, transformer regression, and TabNet are compared the same way the ANN models are compared. Figure 10 illustrates the difference between the predicted values and actual values for both water levels. Figure 10 (a) shows the autoformer, (b) illustrates the transformer regression performance, and (c) gives the predictions for the TabNet.

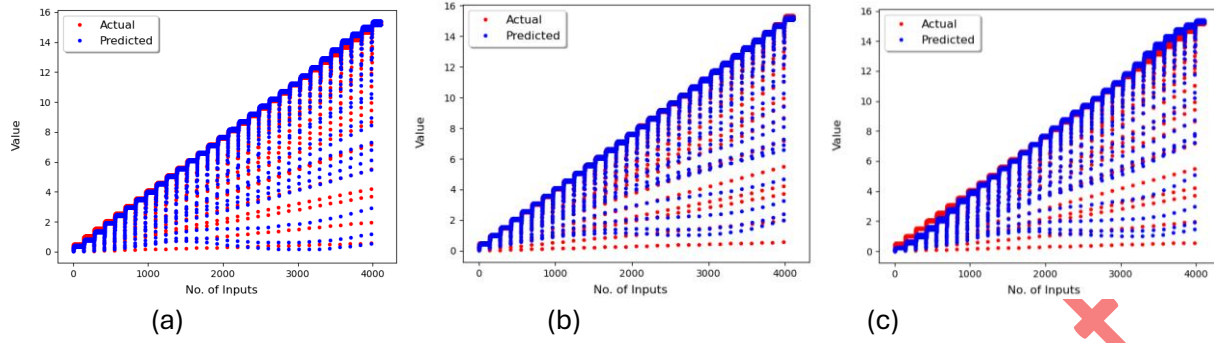


Figure 10. Results of the three transformer models, (a) results of autoformer, (b) results of transformer regression, and (c) results of TabNet

The R^2 for each transformer model across each of the predicted outputs are given in Table 4. This comparison is used to find which of the transformer models is the best.

Table 4 comparison of different transformer models in terms of R^2 for each predicted value

Transformer Model	R^2 for WL1	R^2 for WL2
Autoformer	99.77%	99.81%
transformer regression	96.88%	87.93%
TabNet	99.31%	99.02%

According to the results of Table 4, it can be concluded that the autoformer achieves the best performance. Thus, it will be studied further by checking the predictions across different u values.

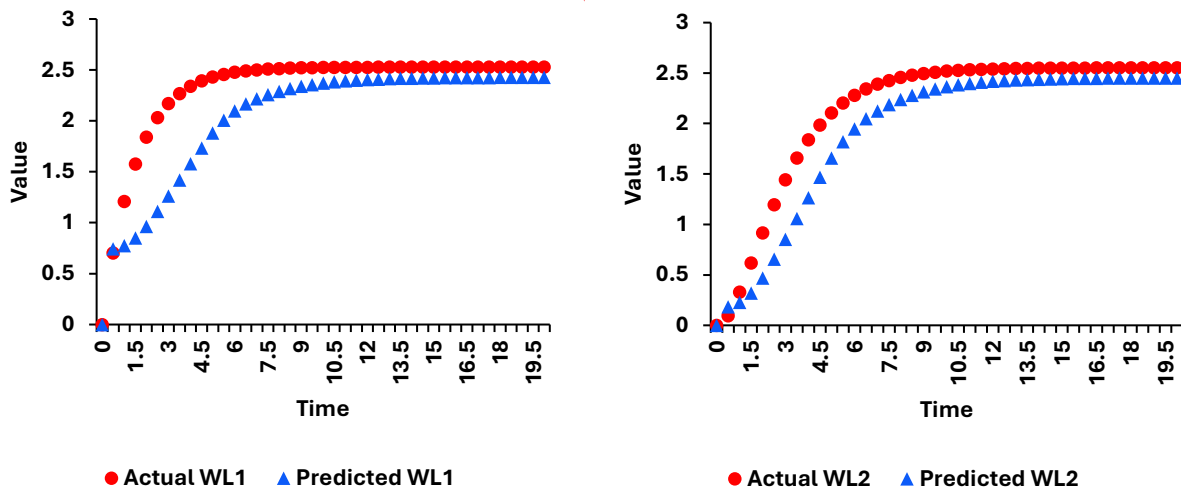


Figure 11. Autoformer predictions where $u = 0.5$.

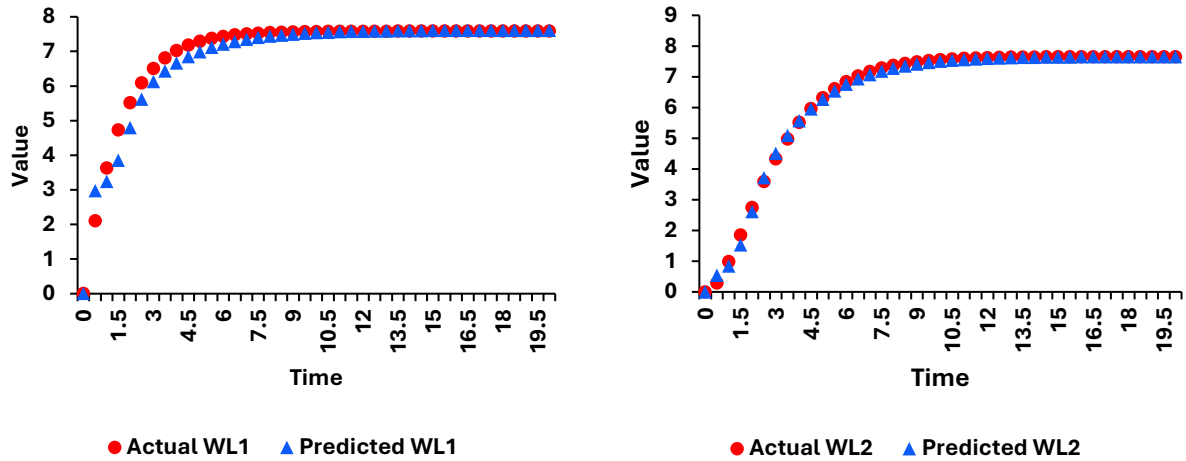


Figure 12. transformer regression predictions where $u = 1.5$.

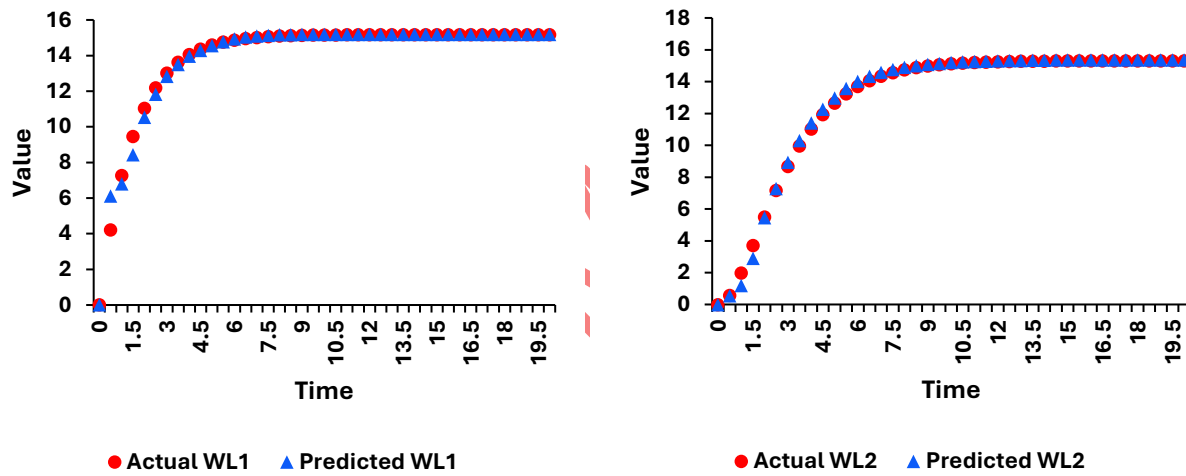


Figure 13. TabNet predictions where $u = 3.0$.

The results of the autoformer for both WL1 (left) and WL2 (right) predictions are illustrated in Figures 11, 12, and 13. In order to illustrate the performance in predicting the water level values, different values of u are given, i.e., 0.5, 1.5, and 3.0. Figure 11 shows the predictions made compared to the actual values for u being 0.5. Figure 12 gives the same illustration but for u being 1.5. As for a large value of the u , the result is shown in Figure 13.

Figure 14 illustrates the difference between using the best basic ANN models (i.e., A3) and the transformer model that achieved the best results.

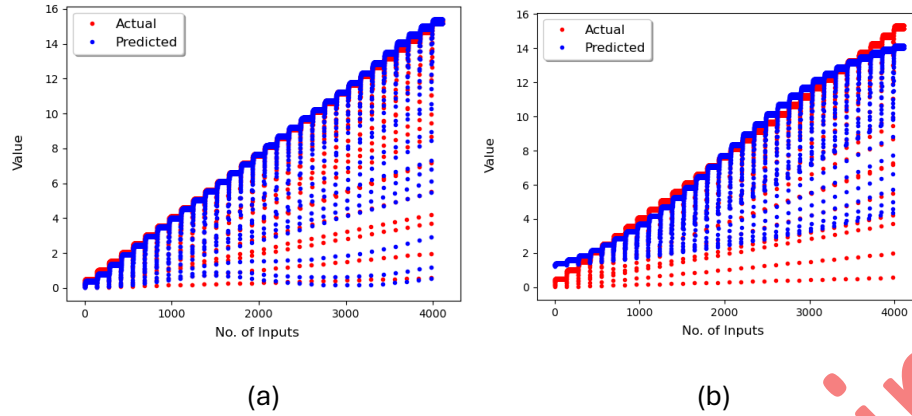


Figure 14 Difference between the best transformer model (i.e., autoformer) given in (a) and the best basic ANN model (i.e., A3) given in (b)

The comparison shown in Figure 14 clearly shows the advantage of using a complicated AI model such as the transformer over a basic deep ANN model such as A3. Also, after calculating the R^2 for this transformer model shown in Figure 14 (a), 99.77% is achieved for predicting WL1 and 99.86% is the result of predicting WL2.

In addition, considering the physical consistency of the AI models (Figures 7-13), indicates that the forecasted tank levels are consistent with the known physical properties of the system, as they are always non-negative and have the right scale for the inlet flow rate, which shows their ability to learn the system's capacity limits and boundary conditions from the training data distribution. In terms of computation, after they are trained, the forward pass (inference) of these neural networks requires simple matrix multiplications that only take milliseconds. These AI surrogate models are particularly well suited for online state estimation in advanced process control because of this fast execution, which is much more efficient than the execution of complex numerical differential equations (NDEs) online.

4. Conclusion

This paper compared three different artificial intelligence (AI) strategies: autoformer, transformer regression, and attentive interpretable tabular learning (TabNet) for model prediction. The main motivation was to investigate nonlinear behavior, an unavailable measurement that makes control design difficult. Different AI techniques have been used in this study and investigated to estimate the levels of a third-order liquid-level system. The available simulated response data of one level were used to train the ANNs to estimate the other unmeasured levels. The comparisons between the predicted and actual levels of responses showed excellent agreement with the generated data using an autoformer model with a coefficient of determination (R^2) ranging 99.77% - 99.86%. Based on the results, it can be used as a computationally efficient surrogate model for process control and optimization, to approximate complicated and nonlinear mathematics with little online computational cost. The future work of this paper is to try different types of complex levels to predict with more input variables and multiple outputs.

Author Contributions: Conceptualization, modeling, and writing—original draft preparation, S. Al-Dawery; methodology, AI programming, software, and writing—review and editing, M. Al-Daweri and R. Ramadan.

All authors have read and agreed to the published version of the manuscript.

Funding: This research did not receive any funds to support the work of this study.

Data Availability Statement: Not applicable.

Acknowledgments: Not applicable.

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

Nomenclature

- R^2 : Coefficient of determination
- RSS : Sum of squares of residuals
- t : Time
- TSS : Total sum of squares
- u : Water feed flow rate
- $WL1$: Water top level
- $WL2$: Water middle level
- $WL3$: Water bottom level

Declaration of Generative AI and AI-assisted technologies in the writing process: AI tool was utilized in order to proofread the manuscript and improve the English language. All authors reviewed the changes and approved the final content.

References

- [1]. N. M. Sundaram, S. N. Sivanandam, and V. Renupriya, "Artificial Neural Network Approach for Dynamic Modelling of Heat Exchanger for Data Prediction," *Indian Journal of Science and Technology*, vol. 9, pp. 1-7, 2016. <https://doi.org/10.17485/ijst/2016/v9iS1/86189>
- [2]. B. Thangagiri, "The Artificial Neural Network (ANN) Based Computational Modelling for the Prediction of the Acid-Base PH Metric Titration Value," *International Journal of Current Research and Modern Education*, vol. 1, no. 1, pp. 55-64, 2016.
- [3]. Z. Wu, D. Rincon, and P. D. Christofides, "Process structure-based recurrent neural network modeling for model predictive control of nonlinear processes," *Journal of Process Control*, vol. 89, pp. 74-84, 2020. <https://doi.org/10.1016/j.jprocont.2020.03.013>
- [4]. B. S. Sousa, F. V. Silva, and A. M. F. Fileti, "Level Control of Coupled Tank System Based on Neural Network Techniques," *De Gruyter*, vol. 20190086, 2010. <https://doi.org/10.1515/cppm-2019-0086>
- [5]. R. Verma and C. S. Besta, "Modelling of chemical processes using artificial neural network," *Indian Chemical Engineer*, vol. 66, no. 1, pp. 84-105, 2023. <https://doi.org/10.1080/00194506.2023.2280655>

- [6]. S. Kamat and K. P. Madhavan, "Developing ANN based virtual/soft sensors for industrial problems," *IFAC-PapersOnLine*, vol. 49, no. 1, pp. 100-105, 2016.
<https://doi.org/10.1016/j.ifacol.2016.03.036>
- [7]. S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 3rd ed., Pearson Education, Inc., USA, 2010.
- [8]. S. Islam, H. Elmekki, A. Elsebai, J. Bentahar, N. Drawel, G. Rjoub, and W. Pedrycz, "A comprehensive survey on applications of transformers for deep learning tasks," *arXiv preprint arXiv:2306.07303*, 2023. [Online]. Available:
<https://arxiv.org/html/2306.07303>
- [9]. P. Mikulskis, A. Hook, A. A. Dundas, D. Irvine, O. Sanni, D. Anderson, R. Langer, M. R. Alexander, P. Williams, and D. A. Winkler, "Prediction of Broad-Spectrum Pathogen Attachment to Coating Materials for Biomedical Devices," *ACS Applied Materials & Interfaces*, vol. 10, no. 1, pp. 139-149, 2018. <https://doi.org/10.1021/acsami.7b14197>
- [10]. Gonzalez-Fernandez, M. A. Iglesias-Otero, M. Esteki, O. A. Moldes, J. C. Mejuto, and J. Simal-Gandara, "A critical review on the use of artificial neural networks in olive oil production, characterization and authentication," *Critical Reviews in Food Science and Nutrition*, vol. 59, no. 12, pp. 1913-1926, 2019.
<https://doi.org/10.1080/10408398.2018.1433628>
- [11]. Y. Lohmari, Y. Timoumi, and A. Messadi, "Performance comparison of two global solar radiation models for spatial interpolation purposes," *Renewable and Sustainable Energy Reviews*, vol. 82, pp. 837-844, 2018. <https://doi.org/10.1016/j.rser.2017.09.092>
- [12]. H. K. Ghritlahre and R. K. Prasad, "Application of ANN technique to predict the performance of solar collector systems – A review," *Renewable and Sustainable Energy Reviews*, vol. 84, pp. 75-88, 2018. <https://doi.org/10.1016/j.rser.2018.01.001>
- [13]. Y. Liu, S. Yu, Y. Zhu, D. Wang, and J. Liu, "Modeling, planning, application and management of energy systems for isolated areas: A review," *Renewable and Sustainable Energy Reviews*, vol. 82, pp. 460-470, 2018. <https://doi.org/10.1016/j.rser.2017.09.063>
- [14]. Y. Li, S. Hu, X. Sun, and M. Stan, "A review: Applications of the phase field method in predicting microstructure and property evolution of irradiated nuclear materials," *NPJ Computational Materials*, vol. 3, no. 1, 2017. <https://doi.org/10.1038/s41524-017-0018-y>
- [15]. N. Lin, F. Xie, J. Zou, H. Wang, and B. Tang, "Application of artificial neural network in predicting the thickness of chromizing coatings on P110 steel," *Journal of Wuhan University of Technology-Mater. Sci. Ed.*, vol. 28, no. 1, pp. 196-201, 2013.
<https://doi.org/10.1007/s11595-013-0664-y>
- [16]. N. Haghdadi, A. Zarei-Hanzaki, A. R. Khaledian, and H. R. Abedi, "Artificial neural network modeling to predict the hot deformation behavior of an A356 aluminum alloy," *Materials & Design*, vol. 49, pp. 386-391, 2013. <https://doi.org/10.1016/j.matdes.2012.12.082>
- [17]. P. L. Mazzeo and P. Spagnolo, "Deep Learning Applications," in *Application of Artificial Neural Networks to Chemical and Process Engineering*, F. M. Cavalcanti, C. E. Kozonoe, K. A. Pacheco, and R. M. de Brito Alves, Eds. IntechOpen, 2021.
<https://doi.org/10.5772/intechopen.91576>
- [18]. A. Khadom, "Artificial neural network for prediction of zinc consumption rate of cathodic protection of copper in saline water: A short communication," *Results in Chemistry*, vol. 4, pp. 100370, 2022. <https://doi.org/10.1016/j.rechem.2022.100370>
- [19]. Y. J. Qin, Q. L. Pan, Y. B. He, W. B. Li, X. Y. Liu, and X. Fan, "Artificial neural network modeling to evaluate and predict the deformation behavior of ZK60 magnesium alloy during hot compression," *Materials and Manufacturing Processes*, vol. 25, no. 7, pp. 539-545, 2012. <https://doi.org/10.1080/10426991.2012.689944>

- [20]. M. S. Al Daweri, S. Abdullah, and K. A. Zainol Ariffin, "An adaptive method and a new dataset, UKM-IDS20, for the network intrusion detection system," *Computer Communications*, vol. 180, pp. 57–76, 2021.
<https://doi.org/10.1016/j.comcom.2021.09.007>
- [21]. S. Al-Dawery, "State estimation for control of chemical process," *Eng. & Technology*, vol. 11, no. 1, pp. 107–112, 1991.
- [22]. H. Wu, J. Xu, J. Wang, and M. Long, "Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting," in *Advances in Neural Information Processing Systems*, vol. 34, pp. 22419–22430, 2021.
<https://doi.org/10.48550/arXiv.2106.13008>
- [23]. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, vol. 30, pp. 5998–6008, 2017. <https://doi.org/10.48550/arXiv.1706.03762>
- [24]. S. Ö. Arik and T. Pfister, "Tabnet: Attentive interpretable tabular learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 8, pp. 6679–6687, May 2021.
<https://doi.org/10.1609/aaai.v35i8.16826>