International Journal of Combinatorial Optimization Problems and Informatics, 17(2), 2026, 256-274. ISSN: 2007-1558. https://doi.org/10.61467/2007.1558.2026.v17i2.1271

www.editada.org

Pretrained Generative Transformer (PGT) for the estimation of the response variable using causal relationships with fast convergence

Roberto Baeza Serrato[1]

[1] Department of Multidisciplinary Studies, Engineering Division, Campus Irapuato-Salamanca, University of Guanajuato, Yuriria 38944, Guanajuato, Mexico.
r.baeza@ugto.mx

Abstract. Transformer architecture has been successfully adapted for use in vision systems and, more recently, for estimating remaining useful life across various industrial fields. The training patterns of an artificial neural network have no relationship among the input variables. They are used to estimate response variables using a training algorithm, such as backpropagation or its variants. In the present research, an accurate estimation transformer structure is proposed to develop causal relationships between input variables and their respective training patterns and identify the importance and relevance of causal input-output relationships. The complete encoder-decoder structure is used. Input training patterns are entered into the encoder and decoder. In this way, the input data is processed by three self-attention blocks: one in the encoder and two in the decoder, allowing the model to capture dependencies between them and strengthening the base of training patterns. The system's output represents the required estimate.

Keywords: PGT; estimation; encoder, decoder, multilayer neural network, self-attention, dependencies.

Article Info
Received Dec 12, 2024
Accepted Dec 12, 2025

1 Introduction

Estimating response variables in any productive sector is based on the analysis of causal input-output relationships, using mathematical, statistical, or machine-learning methods such as artificial neural networks. Recent macroeconomic research, which examines input-output analysis for the effects of green transition interventions, confirms that these interventions generate benefits for growth and employment on a large scale (Marinos et al., 2025). Predicting response variables with parametric models can be limited when complex nonlinear relationships are present. By contrast, the use of non-parametric models to make estimates, such as artificial neural networks, introduces complications when data overmatch (Chen at al., 2028). A review of the state of the art revealed a significant trend towards the application of artificial intelligence (AI), with artificial neural network approaches being the most popular among various AI methods. Socio-economic, demographic and climatic variables generally have non-linear behaviour and are used to estimate energy demand based on machine learning models (Rahman et al., 2022). To effectively overcome the limitations of traditional methods of statistical analysis, deep learning-based artificial intelligence (AI) models are used for prediction of prey deformations (Huang et al., 2025). Many researchers use artificial neural networks (ANNs) to estimate complex systems with nonlinear behavior that is difficult to represent by mathematical or statistical models. The use of attention mechanisms has produced a significant change in the structure of neural networks, known as the transformer, in which input variables are related and provide information relevant to linguistic translation problems and time series estimation systems. Rende et al., (2025) mentioned that the dot-product attention mechanism cleverly captures the semantic relationships between pairs of words in sentences by calculating the overlap between queries and keys, and was initially designed for natural language processing tasks, and is a cornerstone of modern Transformers.

The PGT proposal for the estimation of response variables is conceptualized in describing self-attention as a mechanism of attention, that allows for relating different positions of a multiple sequence, as it is an array (nxm) of input variables and values of each of them to calculate a strengthened dependency representation of the multiple sequence, which allows for a better estimate. The self-attention mechanism assigns each input item a weight based on its importance in producing output. The transformer architecture integrates a self-attention mechanism to model dependencies between input sequences. The causal relationship described in this research is based on the dependence obtained between the input variables when using the dot-product operation, for the complete training base in the encoder, decoder and the union of both at the proposed transformer structure. The Transformer model based on self-attention mechanics is a deep learning architecture (Choudhary et al., 2024). The transformer architecture has been used very successfully in translating text into various languages. The architecture of transformers has been widely applied in the field of machine translation systems, question answering, and natural language processing for tasks (Zhang et al., 2024). Additionally, transformer architectures have been successfully adapted for use in vision systems. Recent computer vision research has had successful results using transformer-based performing methods (Dosovitskiy, 2020). The ability of transformers to handle nonlinear relationships and parallel calculations makes them suitable for complex causal relationship data modeling. Recent maintenance techniques, such as transformers, have shown promise in estimating the useful life of maintenance programs. Zhou et al., (2025) have analyzed time series using transformer model architecture, due to its success in parallel computing and long-term dependency modelling. Researchers use the structure of transformers to estimate time series as sequential data and capture correlations within the sequence and complex long-term patterns. These investigations have validated the potential of the transformer to predict long sequences in time (Zhou et al., 2025).

Recent research has been using transformer structures for remaining useful life (RUL) estimation as a crucial technology in health forecasting and management, such as Mo et al., (2021) proposed RUL estimation, based on the structure of a transformer due to its great success in sequence learning, to capture short- and long-term dependencies in a time sequence. In addition, it integrates a closed convolutional unit that incorporates local contexts at every time step. The Transformer makes high-level output features insensitive to local contexts. Chen et al., (2022) designed a neural network based on the Transformer. They integrated an automatic noise elimination encoder. Then, a reconstructed sequence was fed into a network of transformers to capture temporal information and learn functional characteristics. With the increase of mass data for monitoring, Wang et al., (2021) propose a modification to the structure of a transformer that consists of two main parts: the encoder, which uses the attention mechanism to extract dependencies across distances in time series, and the temporal convolution neural network to correct the insensitivity of the self-attention mechanism to local characteristics.

Zhang et al., (2022) propose a dual self-attention mechanism within the transformer architecture, consisting of an encoder-decoder structure. The proposed structure consists of two encoders that operate in parallel to extract characteristics from different sensors and time steps simultaneously. Self-attention coders are more effective at processing long data streams and can learn adaptively to focus on more important parts of the input. Ding et al., (2021) propose a novel convolutional transformer that combines local dependency modeling of the convolutional operation with global context capture of the attention mechanism. An entratable class token was added to improve the extraction of degradation-related features, and a multi-scale convolutional module was added to the Transformer architecture, with a new Swish activation function. Liu et al., (2022) propose a double-attention transformer structure for aircraft engine RUL forecasting. A convolutional CNN neural network with channel attention was used to assign greater weight to more significant features. Additionally, Guo et al., (2024) proposed a precision remaining-life prediction model based on encoder-decoder architecture with a multi-head care mechanism for further feature exploration. The model uses a one-dimensional convolutional neural network (CNN) to compress input characteristics along the time dimension. In the same way, Wang et al., (2021) used a variational autoencoder to fuse RNA expression and DNA methylation data across 32 tumor types, then uses the Hilbert curve to visualize the results, proposing a deep learning-based prediction model for tumor types.

In summary, transformer architecture has achieved success in language translation, vision systems, and time series estimation; however, there is a lack of research using the transformer as a mechanism for causal input-output relationships estimation. The present research proposes a structure for estimating the response variable based on causal relationships with input variables, achieving 100% reliability. Also, the presented structure identifies the relative importance of the input variables and the importance of the values in each variable, thereby making a scientific contribution to the field of artificial intelligence based on transformer architectures. The major contributions of this paper are:

- Present a transformer structure as a prediction model of causal input-output relationships.
- Strengthening the input variables and training patterns by determining a triple dependency between them when using the proposed transformer structure.
- The proposed encoder-decoder structure identifies the importance of input variables and the importance of training pattern order as a main contribution to the development of neural network structures called transformers.
- Rapid convergence in only two epochs using the learning algorithm Levenberg-Marquardt.

The rest of this paper is organized as follows. Section 2 describes materials and methods. Section 3 presents simulation results of PGT for accurate estimation, and Section 4 discusses and concludes the proposed method.

2 Material and Methods

2.1. Materials

The encoder-decoder architecture is used by most competitive neural sequence transduction models (Bahdanau et al., 2014, Cho et al., 2024, Sutskever et al., 2024). In the proposed architecture, the encoder and decoder map a multiple sequence of causal input variables, represented in order matrix form (nxm). Given the matrix (nxm), the decoder generates a matrix with triple dependencies of order (nxm) to feed a neural network and estimate the response variable, specified in an order matrix (nx1). In the final part of the structure, two linear transformations are presented to identify input variables importance and generate an accurate estimate.

The input-output ratio in a neural network establishes training patterns for the estimation of the variable of interest. The proposed transformer architecture uses input-output training patterns. The input feeds the encoder and decoder, which aims to enter three attention blocks to identify significant relationships in the input variables. Input variables are transformed into filtered variables, which feed the neural network to perform training and obtain reliable results of the required estimate. Two linear transformations are then applied to identify the importance of the input variables, as well as the importance of the values for each of them. Simulations are performed by varying the number of entries in a range of 50-100, and training patterns also vary in a range of 50-100. Results show 100% reliable estimates, and the importance of variables and their data are determined in a sequence to identify which variable, and which patterns are most significant for accurate and reliable estimation results.

2.2. Methods

The proposed transformer structure is shown in Figure 1. It consists of an encoder and a decoder. The input variables feed each one of them, which are related in three blocks of attention. The structure has three exits. The first step is the estimation of the target values, which subsequently yields two sequences: the first provides the sequence of importance of the input variables, and the second provides the sequence of importance of the training patterns. In this way, the analysis is strengthened to identify the most relevant variables and data from the training base, which is one of the scientific contributions of the proposed research.

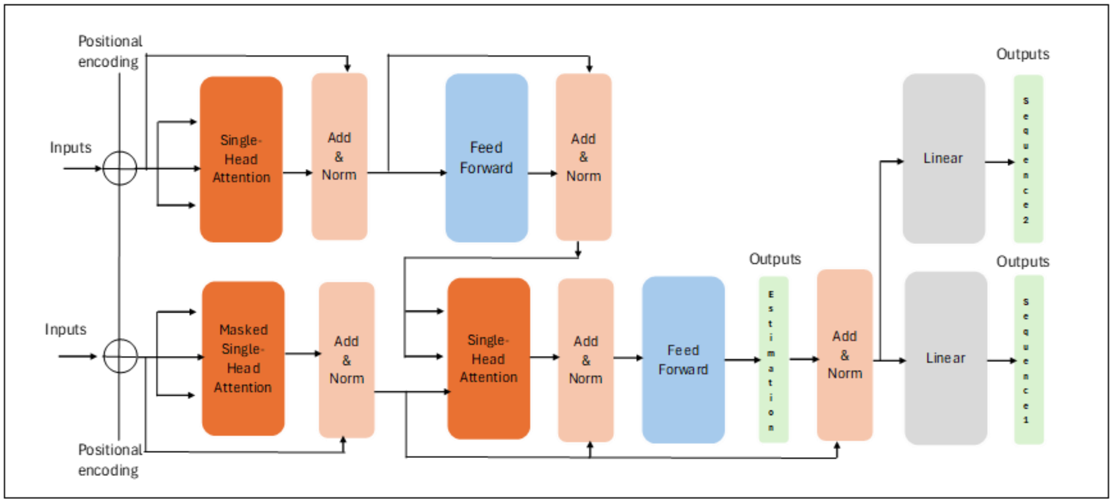


Fig. 1. Estimation-Sequence-GTP proposed.

2.2.1 Classification Embedding Vector

The embedding vector is based on the input matrix (nxm), where n is the number of data points per variable and m is the number of input variables. The positional encoding in each variable utilizes a sin-cos encoding strategy to determine the sequence of their importance, which is then output from the proposed transformer structure (Vaswwani et al., 2017). See Eq. (1,2).

$$PE_{(pos,2i)} = \sin\left(\frac{pos}{10000^{\frac{21}{d_{model}}}}\right) \quad (1)$$

$$PE_{(pos,2i+1)} = \cos\left(\frac{pos}{10000^{\frac{21}{d_{model}}}}\right) \tag{2}$$

2.2.2 Transformer block

The transformer block consists of an encoder and a decoder. The Transformer block module includes two core components: self-attention mechanisms, and feed-forward neural networks (FFN). The self-attention and FFN submodules of the Transformer blocks were constructed according to the standard Transformer architecture [10]. For self-attention, learnable weighted matrices were used to project the BFN into three representation matrices (Query, Key, and Value), and we obtained the weights for each ROI using Query and Key. The updated embedding, including the ROIs' values and weights, is obtained as shown in Eq. (3) (Vaswwani, et al., 2017).

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

where Q, K, V denote Query, Key, and Value, dk represents the dimension of connectivity vectors, and softmax (•) denotes the softmax function. And then, a single-head self-attention mechanism was adopted. The input vector was fed into encoder-decoder for further enhancing un-linear expressivity. The representation vector from the output decoder is forwarded to the multilayer perceptron (MLP) to generate an estimate of the response variable.

2.2.3 MLP

MLP: The obtained representation filtered input vector is forwarded into MLP for estimation, two linear transformations are added with softmax as the final layer for generating prediction probabilities and sequences of importance for the input variables and importance in the data order of each variable. MLP is a multi-layered feedforward neural network. The main features of this network are forward signal propagation and backward error propagation. The neurons in the layers are linked by synaptic weights. These weights can be determined with the use of the learning process (Poczeta et al., 2022).

The neuron processes input data and assigns it to the output. The neuron performs two operations: a weighted sum of inputs and weights plus the neuron threshold and then uses an activation function to map the output of the hidden Y_j and output O_k layer. See Eq. (4). In the formula, x represents the value of the input sample processed by the weight and threshold. The function f (•) represents the mapping of input values to output values (Zhang et al., 2021).

$$f(x) = \frac{1}{1+e^{-x}} \tag{4}$$

Output calculation of the hidden layer: the neural network computes the output layer based on the connection weights and thresholds of the input and hidden layers. In equation (4), x is the value of the input sample. In the equation (5) w_{jk} and a_j are the connection weight and threshold of the input and output respectively, and Y_j is the output of the hidden layer. Where, j is the number of neurons in the hidden layer, and f is the excitation function of the hidden layer. See Eq. (5).

$$Y_j = f\left(\sum_{i=1}^{n} w_{ij}x_i + a_j\right) \tag{5}$$

Output calculation of the output layer: according to the output of the hidden layer, connect the weight w_{jk} and the threshold b_k to calculate the pre-output of the BP neural network O_k. See Eq. (6).

$$O_k = f\left(\sum_{i=1}^{n} Y_j w_{jk} + b_k\right) \tag{6}$$

Calculate the error between the predicted value and the actual value: Calculate the network prediction error e by comparing the neural network forecast output O_k with the actual expected output value Z. See Eq. (7).

$$e_k = Z - O_k \tag{7}$$

Weight and threshold update: If the error is less than the set error value, the iteration will end. Otherwise, the neural network's weights and thresholds are updated, and a new iteration is performed. Updating of weights and thresholds with Eq. (8-11).

$$w_{jk} = w_{jk} + \eta e_k O_k (1 - O_k) Y_j \tag{8}$$

$$w_{ij} = w_{ij} + \eta Y_j (1 - Y_j) x(i) \sum_{k=1}^{m} w_{jk} e_k O_k (1 - O_k) \tag{9}$$

$$b_k = b_k + \eta e_k O_k (1 - O_k) \tag{10}$$

$$a_j = w_{ij} + \eta Y_j (1 - Y_j) \sum_{k=1}^{m} w_{jk} e_k O_k (1 - O_k) \tag{11}$$

The Levenberg-Marquardt algorithm provides a solution based on minimizing nonlinear least-squares. See Eq. (12).

$$f(x) = \frac{1}{2} \sum_{j=1}^{m} r_j^2(x) \tag{12}$$

The derivatives of f (x) can be written using the Jacobian matrix J. See Eq. (13-14).

$$\nabla f(x) = \sum_{j=1}^{m} r_j(x) \Delta r_j(x) = J(x)^T r(x) \tag{13}$$

$$\nabla^2 f(x) = J(x)^T J(x) + \sum_{j=1}^{m} r_j(x) \nabla^2 r_j(x) \tag{14}$$

The Hessian matrix can be simplified to Eq. 15.

$$\nabla^2 f(x) = J(x)^T J(x) \tag{15}$$

The most straightforward and intuitive technique for finding minimum of a function is vanilla gradient descent. See Eq. (16).

$$x_{i+1} = x_i - \lambda \nabla f \tag{16}$$

The behaviour and information of the gradient can be improved by using curvature when using second derivatives. The update rule for Newton's method is based on not using higher-order terms and on determining the minimum by setting the left-hand side term to 0. See Eq. (17).

$$x_{i+1} = x_i - \left(\nabla^2 f(x_i)\right)^{-1} \nabla f(x_i) \tag{17}$$

In Newton's method, the convergence rate depends on the linearity of the function near the initial point. Levenberg proposed an upgrade rule that combines descending gradient algorithms with Newton's method. See Eq. (18).

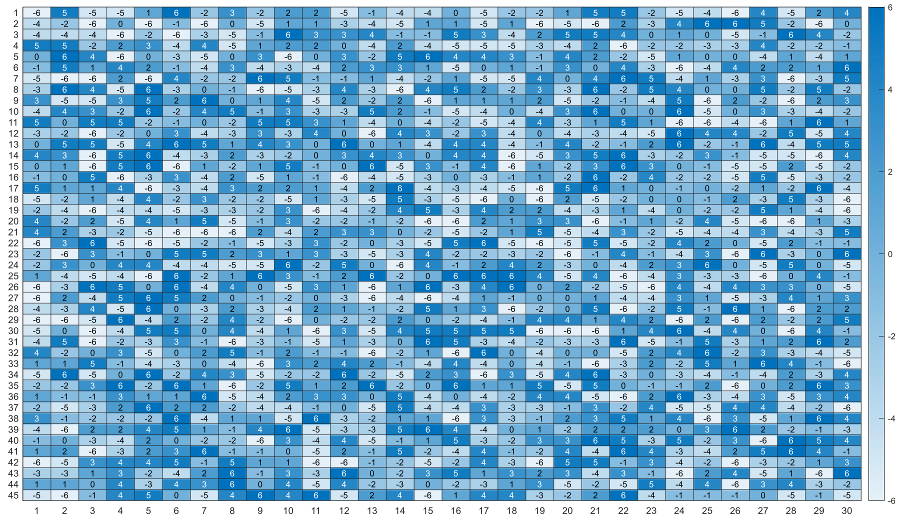
$$x_{i+1} = x_i - (H + \lambda I)^{-1} \tag{18}$$

Levenberg-Marquardt replaced the identity matrix in (18) with the diagonal of the Hessian matrix. See Eq. (19).

$$x_{i+1} = x_i - (H + \lambda diag[H])^{-1} \nabla f(x_i) \tag{19}$$

3 Results

The training patterns for Simulation 1 are shown in Figure 2. The input data to the transformer is provided as a matrix of size (45x30) in a range of [-5,5]. There are 30 variables and 45 training patterns.



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
1	-6	5	-5	-5	1	6	-2	3	-2	2	2	-5	-1	-4	-4	0	-5	-2	-2	1	5	5	-2	-5	-4	-6	4	-5	2	4
2	-4	-2	-6	0	-6	-1	-6	0	-5	1	1	-3	-4	-5	1	1	-5	1	-6	-5	-6	2	-3	4	6	6	5	-2	-6	0
3	-4	-4	-4	-6	-2	-6	-3	-5	-1	6	3	3	4	-1	-1	5	3	-4	2	5	5	4	0	1	0	-5	-1	6	4	-2
4	5	5	-2	2	3	-4	4	-5	1	2	2	0	-4	2	-4	-5	-5	-5	-3	-4	2	-6	-2	-2	-3	-3	4	-2	-2	-1
5	0	6	4	-6	0	-3	-5	0	3	-6	0	3	-2	5	6	4	4	3	-1	4	2	-2	-5	1	0	0	-4	1	-4	1
6	-1	5	1	4	2	-1	-4	3	-4	-3	-4	2	3	3	1	-5	0	1	-1	3	0	4	-3	-6	-4	4	2	2	1	6
7	-5	-6	-6	2	-6	4	-2	-2	6	5	-1	-1	1	-4	-2	1	-5	-5	4	0	4	6	5	-4	1	-3	3	-6	-3	5
8	-3	6	4	-5	6	-3	0	-1	-6	-5	-3	4	-3	-6	4	5	2	-2	3	-3	6	-2	5	4	0	0	5	-2	5	-2
9	3	-5	-5	3	5	2	6	0	1	4	-5	2	-2	2	-6	1	1	1	2	-5	-2	-1	-4	5	-6	2	-2	-6	2	3
10	-4	4	3	-2	6	-2	4	5	-1	3	-3	-3	5	2	-1	-5	-4	0	-4	3	6	0	0	6	-5	0	3	-3	-4	-2
11	5	0	5	5	-2	-1	0	-2	5	5	3	1	-4	0	-4	-2	-5	-4	4	-3	1	5	1	-6	-6	-4	-6	1	6	1
12	-3	-2	-6	-2	0	3	-4	-3	3	-3	4	0	-6	4	3	-2	3	-4	0	-4	-3	-4	-5	6	4	4	-2	5	-5	4
13	0	5	5	-5	4	6	5	1	4	3	0	6	0	1	-4	4	4	-4	-1	4	-2	-1	2	6	-2	-1	6	-4	5	5
14	4	3	-6	5	6	-4	-3	2	-3	-2	0	3	4	3	0	4	4	-6	-5	3	5	6	-3	-2	3	-1	-5	-5	-6	4
15	0	1	-6	5	6	-6	1	-2	1	5	-1	0	6	-5	3	-1	4	-6	1	-2	3	6	3	0	-1	-5	-5	2	-5	-2
16	-1	0	5	-6	-3	3	-4	2	-5	1	-1	-6	-4	-5	-3	0	-3	-1	1	-2	6	-2	4	-2	-2	-5	5	-5	-3	-2
17	5	1	1	4	-6	-3	-4	3	2	2	1	-4	2	6	-4	-3	-4	-5	-6	5	6	1	0	-1	0	-2	1	-2	6	-4
18	-5	-2	1	-4	4	-2	3	-2	-2	-5	1	-3	-5	5	-3	-5	-6	0	-6	2	-5	-2	0	0	-1	2	-3	5	-3	-6
19	-2	-4	-6	-4	-4	-5	-3	-3	-2	3	-6	-4	-2	4	5	-3	4	2	2	-4	-3	1	-4	0	-2	-2	5	1	-4	-6
20	4	-2	2	-5	4	1	5	-5	-1	3	-2	-2	-1	-2	-6	-6	2	1	3	3	-6	-1	1	-2	4	-5	-6	-6	1	-3
21	4	2	-3	-2	-5	-6	-6	-6	2	-4	2	3	3	0	-2	-5	-2	1	5	-5	-4	3	-2	-5	-4	-4	3	-4	-3	5
22	-6	3	6	-5	-6	-5	-2	-1	-5	-3	3	-2	0	-3	-5	5	6	-5	-6	-5	5	-5	-2	4	2	0	-5	2	-1	-1
23	-2	-6	3	-1	0	5	5	2	3	1	3	-3	-5	-3	4	-2	-2	-3	-2	-6	-1	4	-1	-4	3	-6	6	-3	0	6
24	-2	3	0	4	4	-4	-4	-5	-5	6	-2	5	0	-6	4	-1	2	4	2	-3	0	-4	2	3	6	0	-5	5	0	-5
25	1	-4	-5	-4	-6	6	-2	1	6	3	-1	2	6	-2	0	6	6	6	4	-5	4	-6	-4	3	-3	-3	-6	0	4	-1
26	-6	-3	6	5	0	6	-4	4	0	-5	3	1	-6	1	6	-3	4	6	0	2	-2	-5	-6	4	-4	4	3	3	0	-5
27	-6	2	-4	5	6	5	2	0	-1	-2	0	-3	-6	-4	-6	-4	1	-1	0	0	1	-4	-4	3	1	-5	-3	4	1	3
28	-4	-3	4	-5	6	0	-3	2	-3	-4	2	1	-1	-2	5	1	3	-6	-2	0	5	-6	-2	5	-1	6	1	-6	2	2
29	-6	-6	-5	6	-4	2	-2	4	-2	-6	0	-2	-2	2	0	-2	-4	-1	4	4	1	4	2	-6	2	-6	2	-2	2	5
30	-5	0	-6	-4	5	5	0	4	-4	1	-6	3	-5	4	5	5	5	5	-6	-6	-6	1	4	6	-4	4	0	-6	4	-1
31	-4	5	-6	-2	-3	3	-1	-6	-3	-1	-5	1	-3	0	6	5	-3	-4	-2	-3	-3	6	-5	-1	5	-3	1	2	6	2
32	4	-2	0	3	-5	0	2	5	-1	2	-1	-1	-6	-2	1	-6	6	0	-4	0	0	-5	2	4	6	-2	3	-3	-4	-5
33	1	1	5	-1	-4	-3	0	-4	-6	3	2	4	2	-1	-4	4	-4	0	-4	-1	-6	-3	2	-2	5	1	6	4	-1	-6
34	-5	6	-5	0	6	-2	4	3	-5	-2	-2	6	-2	-5	2	3	-6	3	-5	4	6	-3	0	-3	-4	-1	-4	2	-3	4
35	-2	-2	3	6	-2	6	1	-6	-2	5	1	2	6	-2	0	6	1	1	5	-5	5	0	-1	-1	2	-6	0	2	6	3
36	1	-1	-1	3	1	1	6	-5	-4	2	3	3	0	5	-4	0	-4	-2	4	4	-5	-6	2	6	-3	-4	3	-5	3	4
37	-2	-5	-3	2	6	2	2	-2	-4	-4	-1	0	-5	5	-4	-4	3	-3	-3	-1	3	-2	4	-5	-5	4	4	-4	-2	-6
38	3	-1	-2	-2	-2	6	-4	1	1	-5	6	-3	-2	1	1	-6	3	-3	-1	2	3	5	1	4	-6	3	-5	1	6	4
39	-4	-6	2	2	4	5	1	-1	4	6	-5	-3	-3	5	6	4	-4	0	1	-2	2	2	2	0	3	6	2	-2	-1	-3
40	-1	0	-3	-4	2	0	-2	-2	-6	3	-4	4	-5	-1	1	5	-3	-2	3	3	6	5	-3	5	-2	3	-6	6	5	4
41	1	2	-6	-3	2	3	6	-1	-1	0	-5	2	-1	5	-2	-4	4	-1	-2	4	-4	6	4	-3	-4	2	5	6	4	-1
42	-6	-5	3	4	4	5	-1	5	1	1	-6	-6	-1	-2	-5	-2	4	-3	-6	5	5	-1	3	-4	-2	-6	-3	-2	1	3
43	-3	-3	1	3	-2	-4	2	6	-1	3	-3	6	0	-2	3	5	1	3	2	3	-4	3	-1	-6	2	4	-5	1	-6	6
44	1	1	0	4	-3	4	3	6	0	4	-5	4	-2	-3	0	-2	-3	1	3	-5	-2	-5	5	-4	4	-6	3	4	-3	-2
45	-5	-6	-1	4	5	0	-5	4	6	4	6	-5	2	4	-6	1	4	4	-3	-2	2	6	-4	-1	-1	-1	0	-5	-1	-5

Figure 2. Training patterns.

The results in Figure 3 show the estimation using a multilayer neural network using training patterns without any relationship between them. Ten neurons in the hidden layer were used. The Levenberg Marquardt algorithm was used for network training for two epochs.

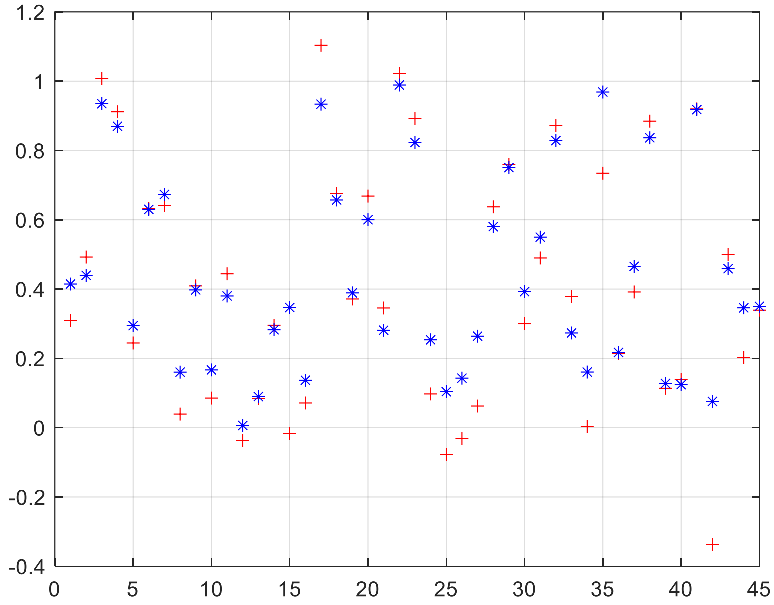


Figure 3. Estimation using multilayer neural network.

Figure 4 shows the input data processed in the encoder first block, which feeds into the decoder second block. The matrix represents the **first** dependency between the input variables of the proposed estimation transformer structure. The dot-product between the training base of the input variables is used for the **first** time in the encoder, finding the causal relationships between them, strengthening the training base to have a faster convergence in the accurate estimation of the response variable.

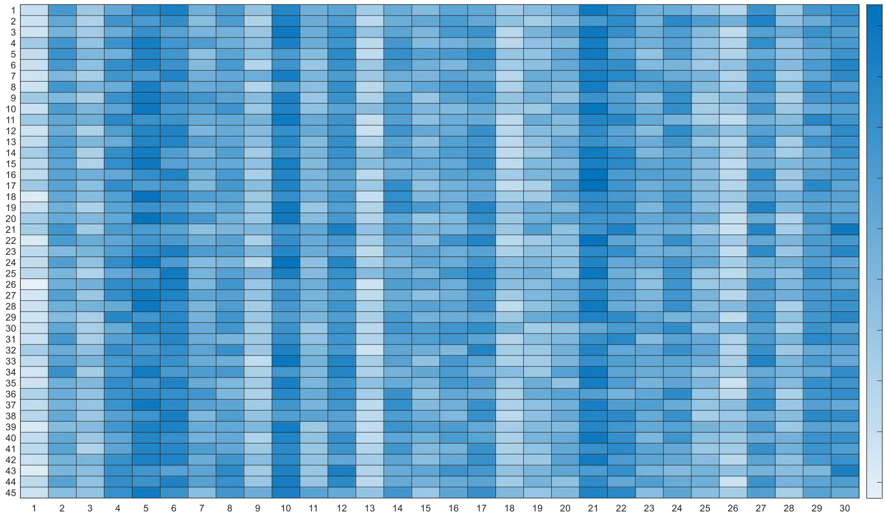


Figure 4. Encoder attention mechanism.

Figure 5 shows the input data processed in the decoder first block, which is then fed into the decoder second block. The matrix represents the **second** dependency between the input variables of the proposed estimation transformer structure. The dot-product between the training base of the input variables is used for the **second** time in the decoder with a masking self-attention mechanism, finding the causal relationships between them, strengthening the training base to have a faster convergence in the accurate estimation of the response variable.

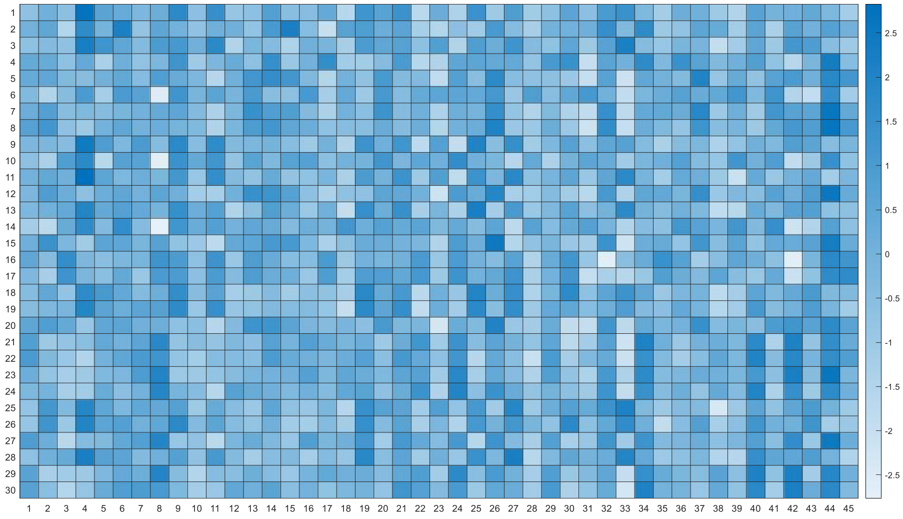


Figure 5. Decoder first block attention mechanism.

Figure 6 shows the input data processed in the decoder second block, which is then fed into the feed-forward network to produce estimates of the response variable. The matrix represents the **third** dependency between the input variables of the proposed estimation transformer structure. The dot-product is used for the **third** time between the previously related training bases of the input variables in the encoder and decoder at the decoder second block, finding the causal relationships between them, strengthening the training base to have a faster convergence in the accurate estimation of the response variable.

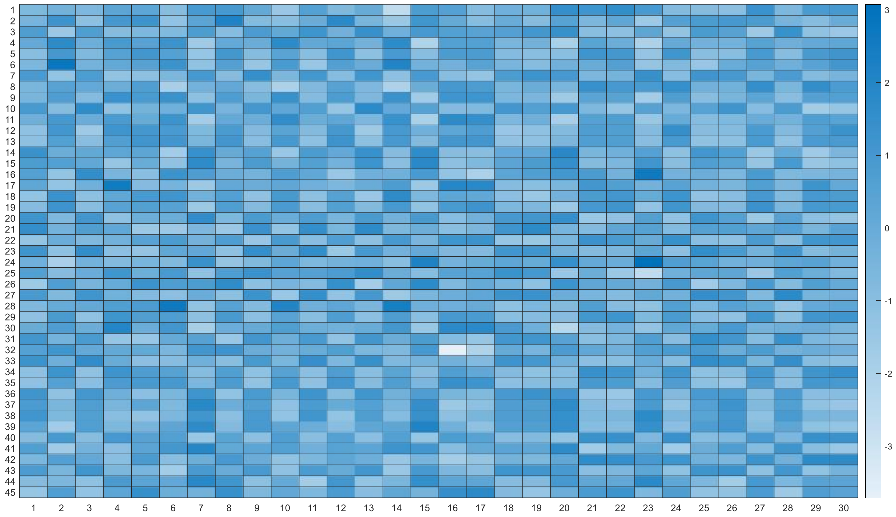


Figure 6. Decoder second block attention mechanism.

Figure 7 shows the comparison of the transformer estimate and the target set with 45 values. The results show 100% reliability in the accurate transformer structure estimates proposed in this research. Ten neurons in the hidden layer were used. The Levenberg-Marquardt algorithm was used to optimize the proposed transformer structure for two epochs.

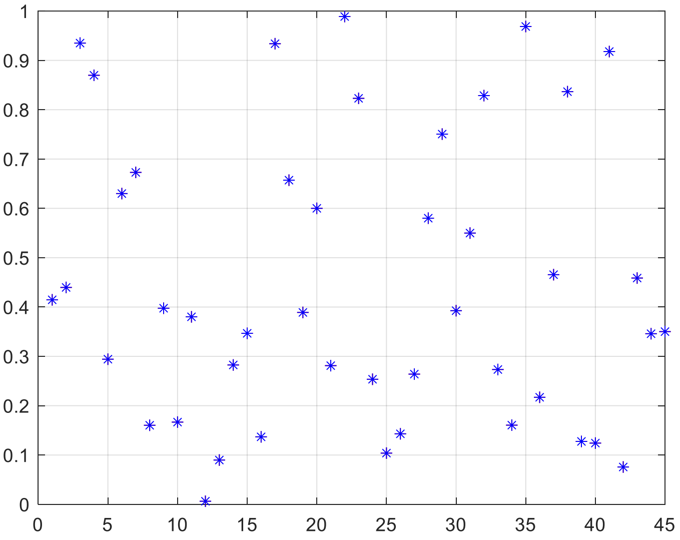


Figure 7. Encoder-Decoder estimation.

By using the PGT proposed structure of the estimation transformer, we are strengthening the training base of the input variables, finding a triple dependence between them, by using the dot-product operation on three occasions: first in the encoder block, second in the first decoder block and third in the second decoder block. Having strengthened the training base, at the end of the proposed transformer structure a multilayer neural network is used for estimation. This action is the reason for comparing results with a multi-layered neural network, using the same Levenberg-Marquardt training algorithm and the same number of epochs.

The structure proposed at the end uses a multi-layer neural network to perform the estimation. The difference lies in the use of a triply related training base between input variables, using three blocks of attention and applying three times the point product operation, allowing rapid convergence of the algorithm with 100% reliability in accuracy. Figure 8 shows a symmetric matrix of the order (30x30) of the relative importance of input variables for accurate estimates of the response variable in ascending order. The most important variable is input variable 4, with a probability of 0.35266, and the second most important variable is variable 18, with a probability of 0.1734. This approach is one of the main contributions of research, as there is a lack of tools to assess the importance of input variables, and no analysis exists of their importance in neural networks or transformers.

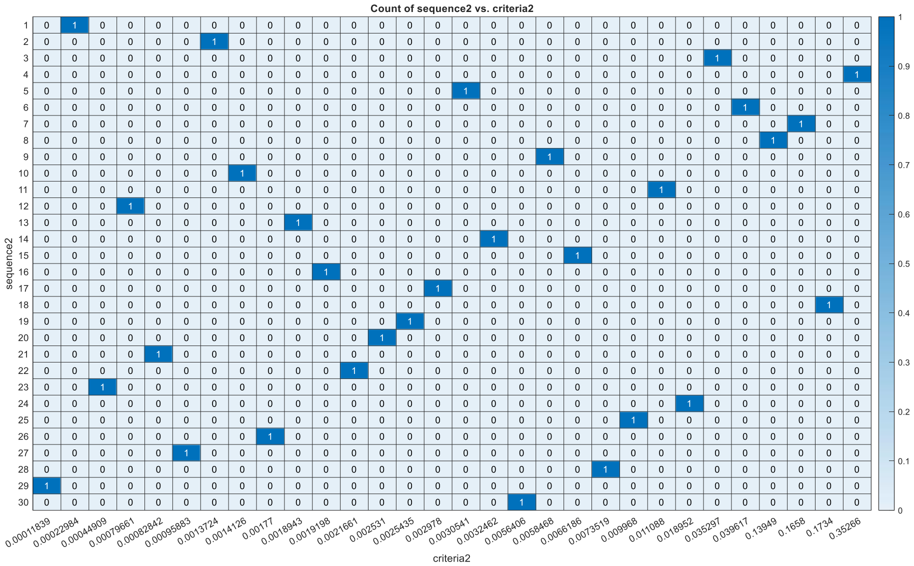


Figure 8. Sequence of importance input variables.

Figure 9 shows a symmetric matrix of the order (45x45) of the relative importance of training patterns for accurate estimates of the response variable in ascending order. The most important is the training pattern number 30, with a probability of 0.1395, and the second most important is the training pattern number 17, with a probability of 0.083431. There is a shortage of tools to assess the importance of training patterns, and the analysis of the order of their importance in a neural network or transformer does not exist.

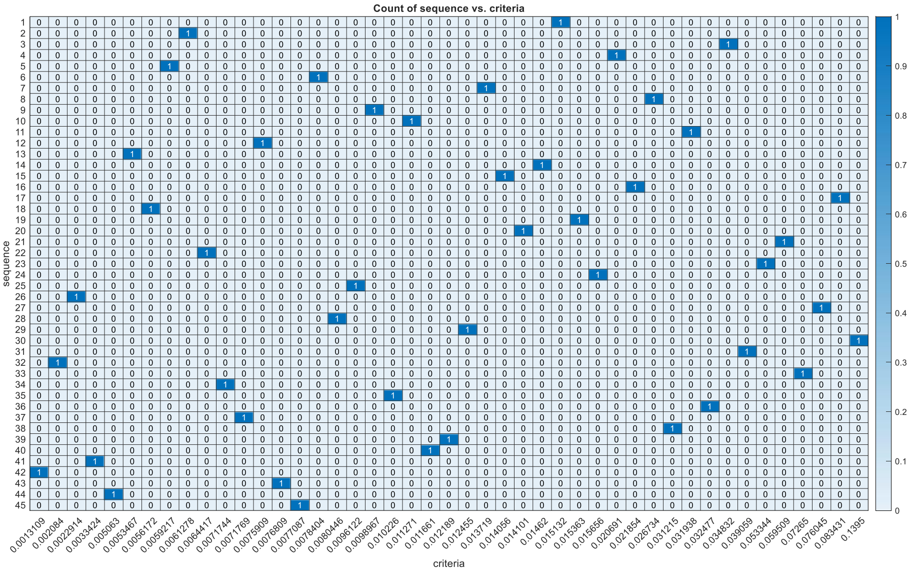


Figure 9. Sequence of importance training data.

In the validation stage of the estimation TMP, the range of values for the input variables training matrix was extended to a random order of (100, 200) for rows and columns with reliable, accurate and rapidly converging results.

The training patterns for Simulation 2 are shown in Figure 10. The input data to the transformer is provided as a matrix of size (108x158) in a range of [-6,6]. There are 158 variables and 108 training patterns.

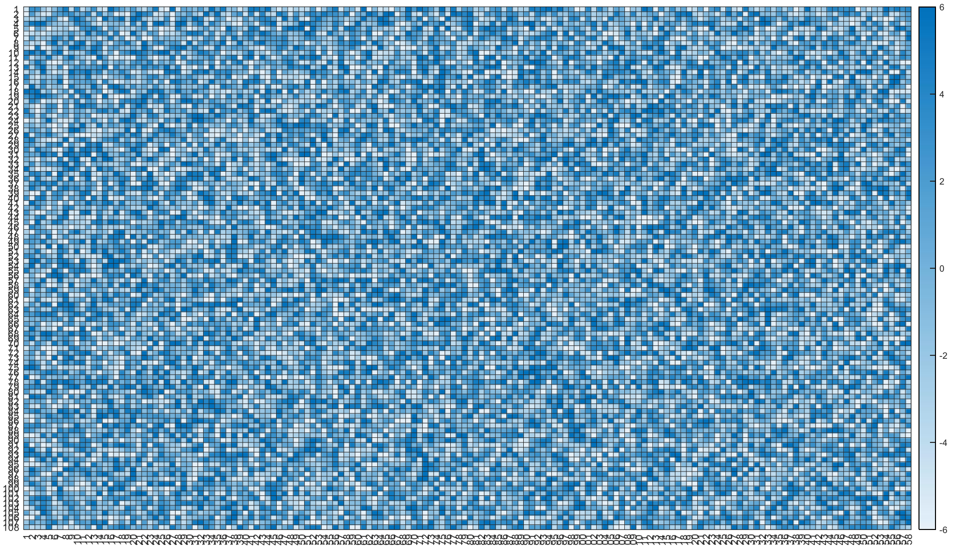


Figure 10. Training patterns.

The results in Figure 11 show the estimation using a multilayer neural network using training patterns without any relationship between them. Ten neurons in the hidden layer were used. The Levenberg Marquardt algorithm was used for network training for two epochs.

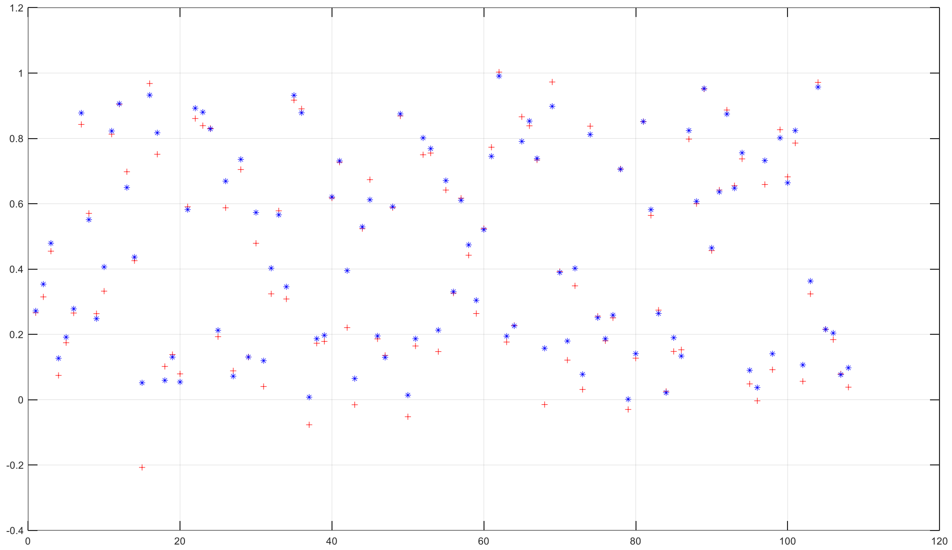


Figure 11. Estimation using multilayer neural network.

Figure 12 shows the input data processed in the encoder first block, which feeds into the decoder second block. The dot-product between the training base of the input variables is used for the **first** time in the encoder.

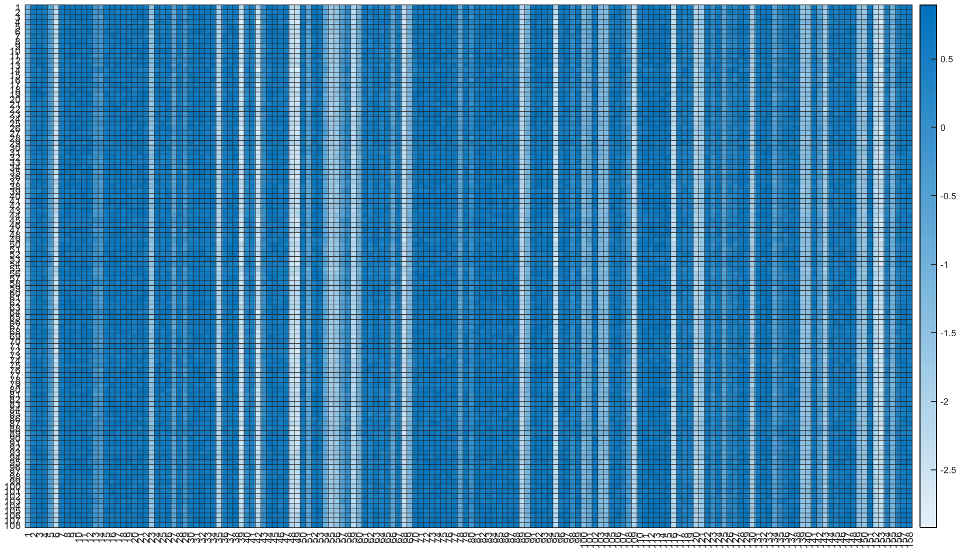


Figure 12. Encoder attention mechanism.

Figure 13 shows the input data processed in the decoder first block, which is then fed into the decoder second block. The dot-product between the training base of the input variables is used for the **second** time in the first decoder block with a masking attention mechanism.

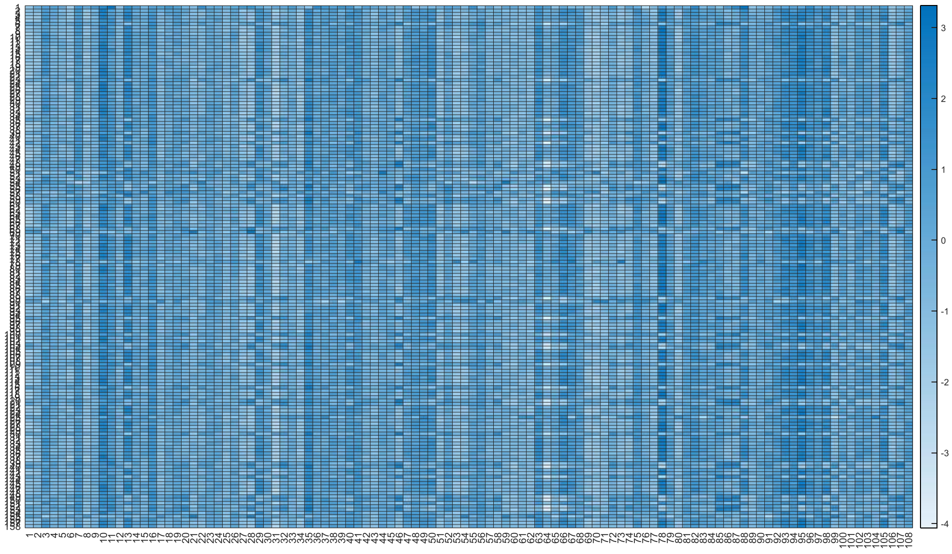


Figure 13. Decoder first block attention mechanism.

Figure 14 shows the input data processed in the decoder second block, which is then fed into the feed-forward network to produce estimates of the response variable. The dot-product between the training base of the input variables is used for the **third** time in the second decoder block, obtaining a robust matrix of input variables by having a triple dependency between them, which allows for reliable, accurate and fast convergence estimation.

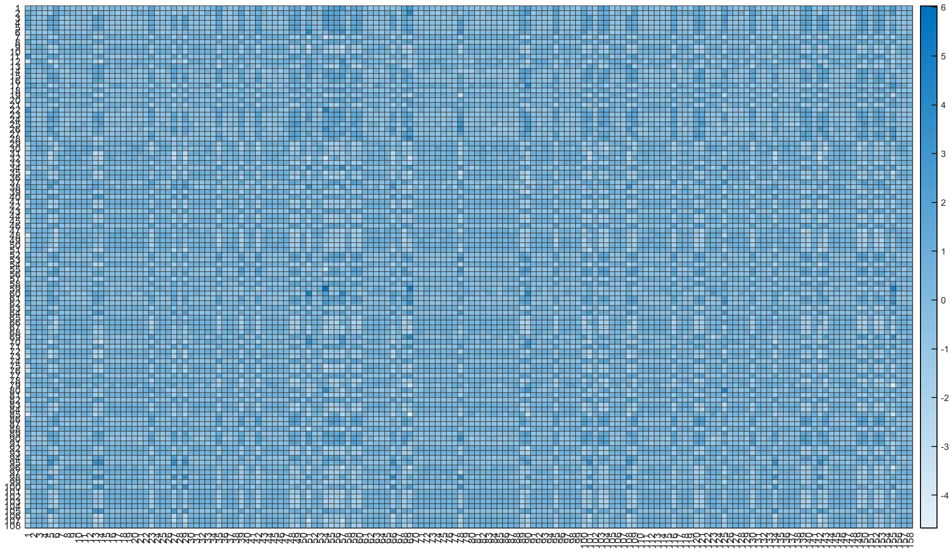


Figure 14. Decoder second block attention mechanism.

Figure 15 shows the comparison of the transformer estimate and the target set with 108 values. The results show 100% reliability in the accurate transformer structure estimates proposed in this research. Ten neurons in the hidden layer were used. The Levenberg-Marquardt algorithm was used to optimize the proposed transformer structure for two epochs.

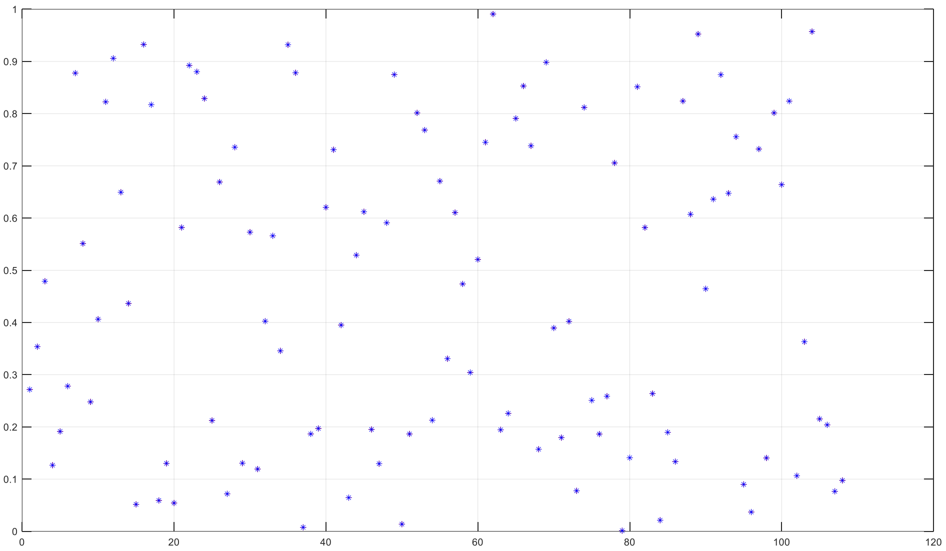


Figure 15. Encoder-Decoder estimation.

Figure 16 shows a symmetric matrix of the order (158x158) of the relative importance of input variables for accurate estimates of the response variable in ascending order. The most important variable is input variable 145, with a probability of 0.49848, and the second most important variable is variable 45, with a probability of 0.36685. This approach is one of the main contributions of research, as there is a lack of tools to assess the importance of input variables, and no analysis exists of their importance in neural networks or transformers.

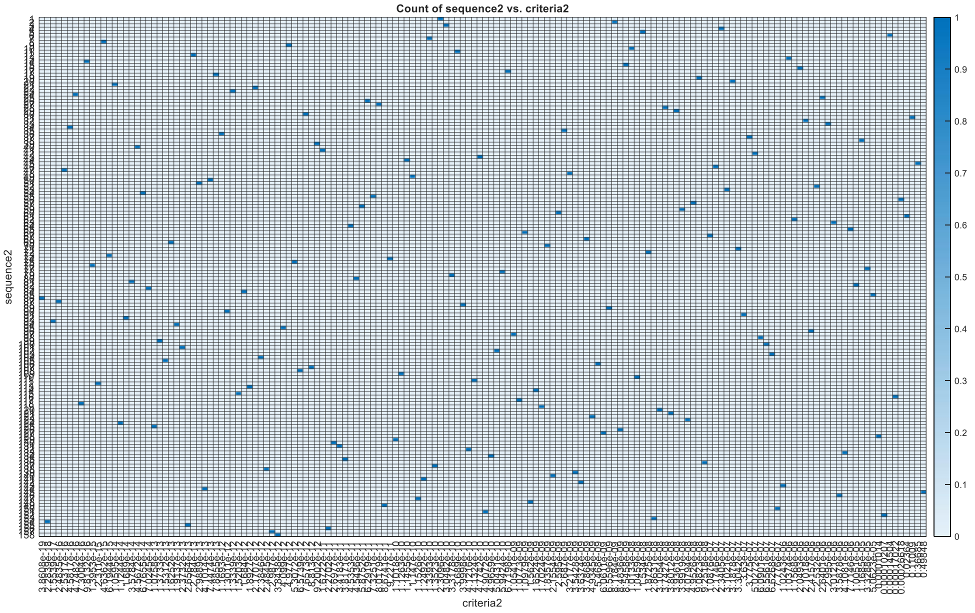


Figure 16. Sequence of importance input variables.

Figure 17 shows a symmetric matrix of the order (108x108) of the relative importance of training patterns for accurate estimates of the response variable in ascending order. The most important is the training pattern number 12, with a probability of 0.66149, and the second most important is the training pattern number 60, with a probability of 0.073598.

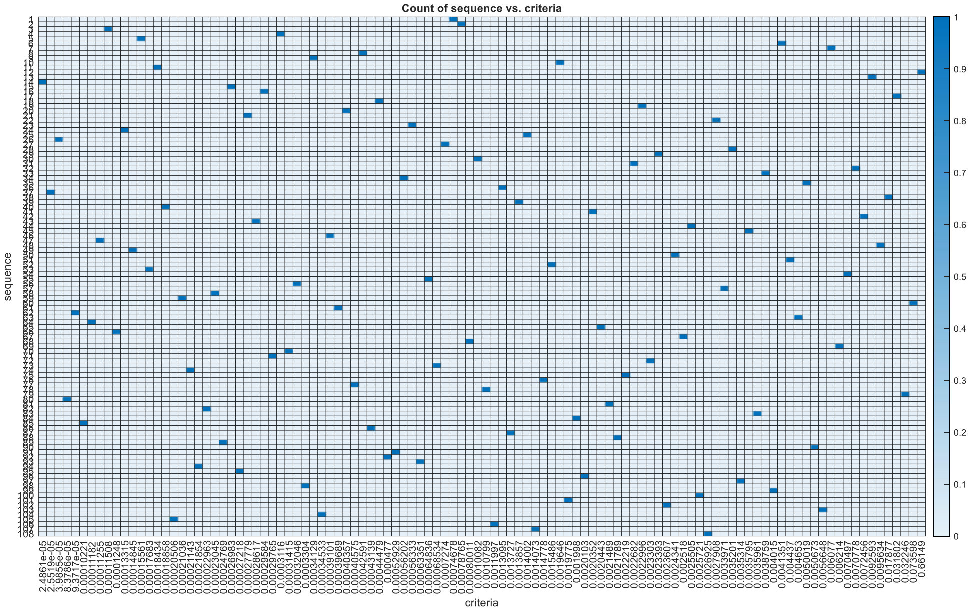


Figure 17. Sequence of importance training data.

The training patterns for Simulation 3 are shown in Figure 18. The input data to the transformer is provided as a matrix of size (164x140) in a range of [-6,6]. There are 140 variables and 164 training patterns.

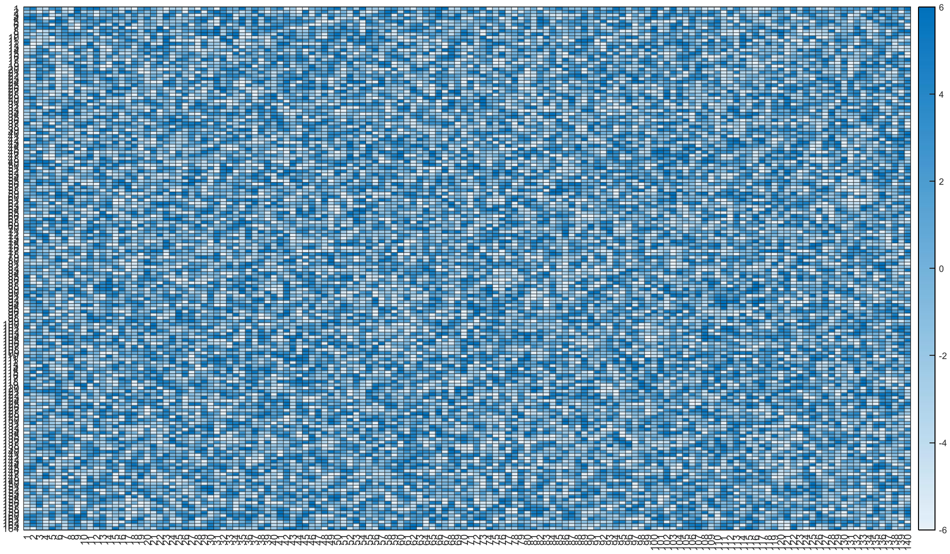


Figure 18. Training patterns.

The results in Figure 19 show the estimation using a multilayer neural network using training patterns without any relationship between them. Ten neurons in the hidden layer were used. The Levenberg Marquardt algorithm was used for network training for two epochs.

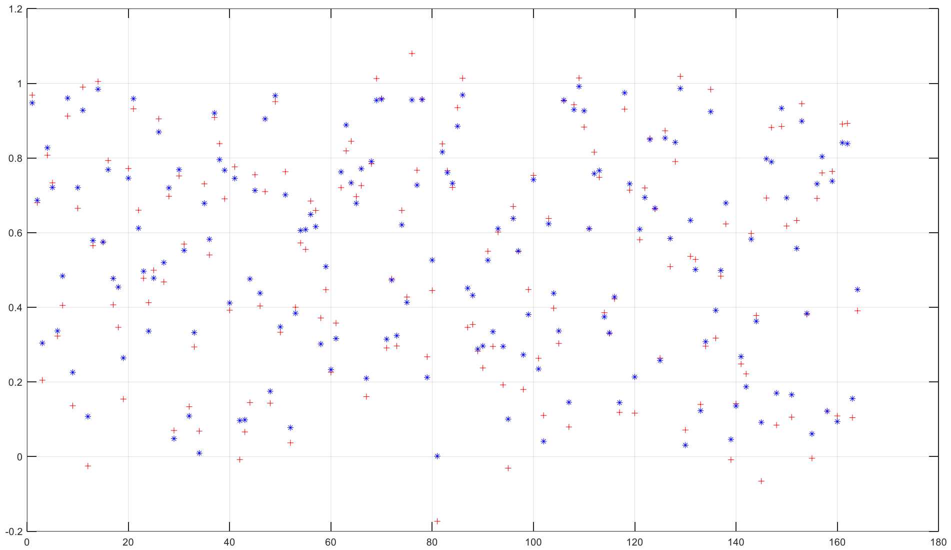


Figure 19. Estimation using multilayer neural network.

Figure 20 shows the input data processed in the encoder first block, which feeds into the decoder second block. The dot-product between the training base of the input variables is used for the **first** time in the encoder.

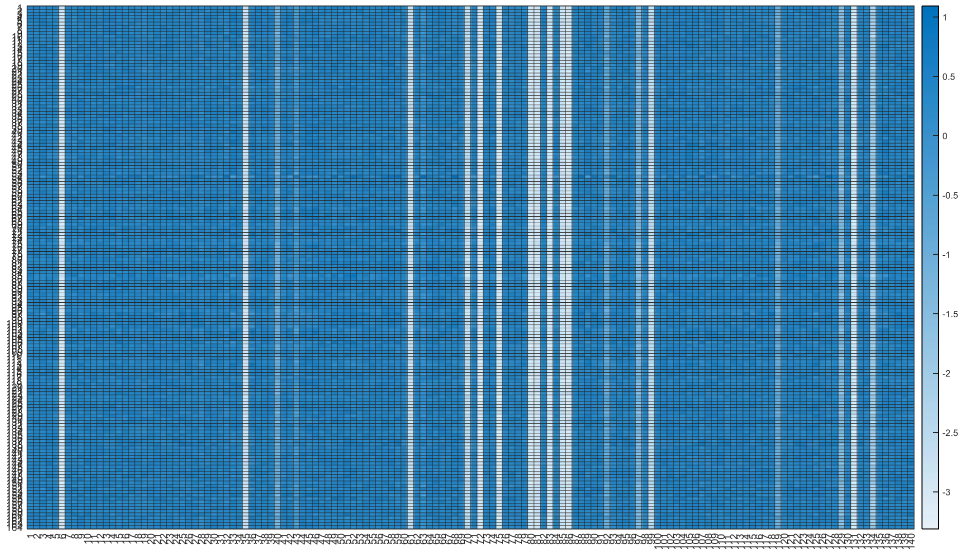


Figure 20. Encoder attention mechanism.

Figure 21 shows the input data processed in the decoder first block, which is then fed into the decoder second block. The dot-product between the training base of the input variables is used for the **second** time in the first decoder block with a masking attention mechanism.

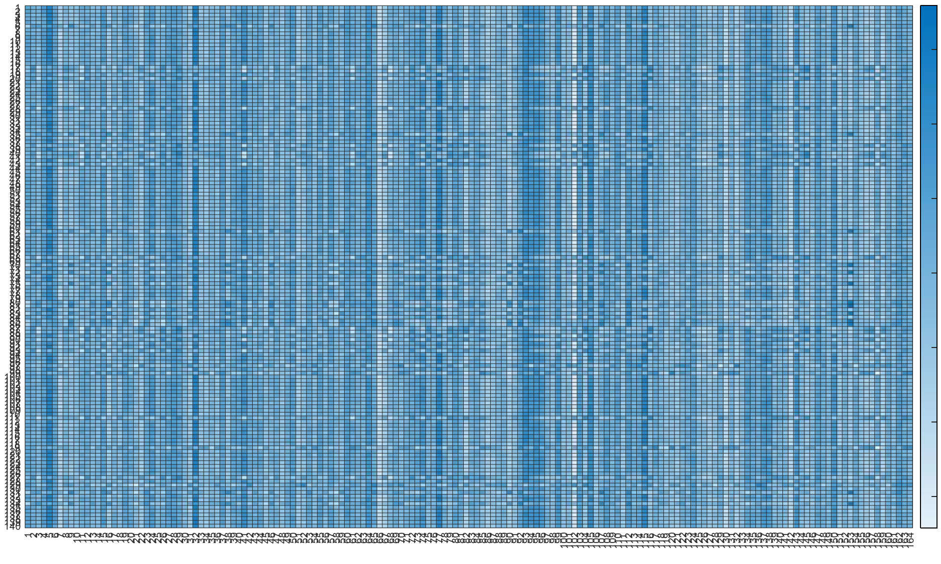


Figure 21. Decoder first block attention mechanism.

Figure 22 shows the input data processed in the decoder second block, which is then fed into the feed-forward network to produce estimates of the response variable. The dot-product between the training base of the input variables is used for the **third** time in the second decoder block, obtaining a robust matrix of input variables by having a triple dependency between them, which allows for reliable, accurate and fast convergence estimation.

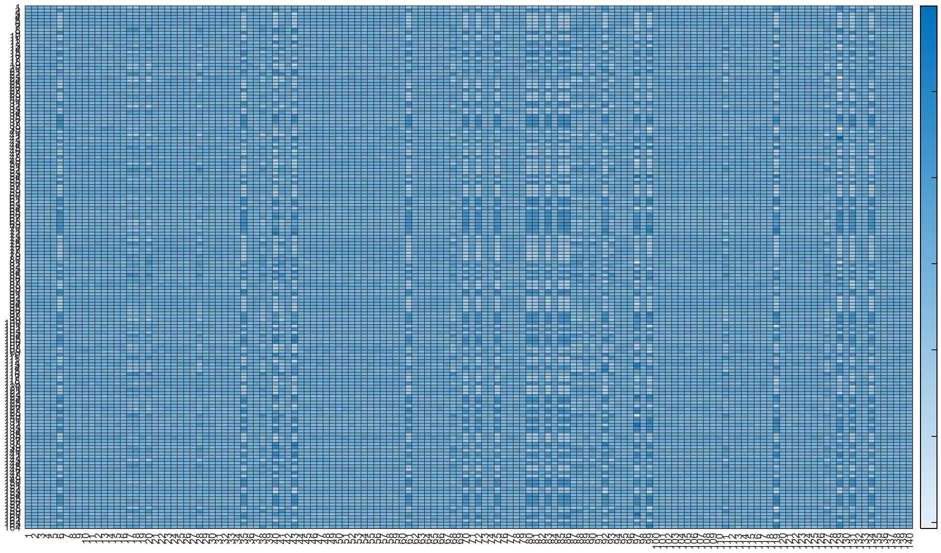


Figure 22. Decoder second block attention mechanism.

Figure 23 shows the comparison of the transformer estimate and the target set with 164 values. The results show 100% reliability in the accurate transformer structure estimates proposed in this research. Ten neurons in the hidden layer were used. The Levenberg-Marquardt algorithm was used to optimize the proposed transformer structure for two epochs.

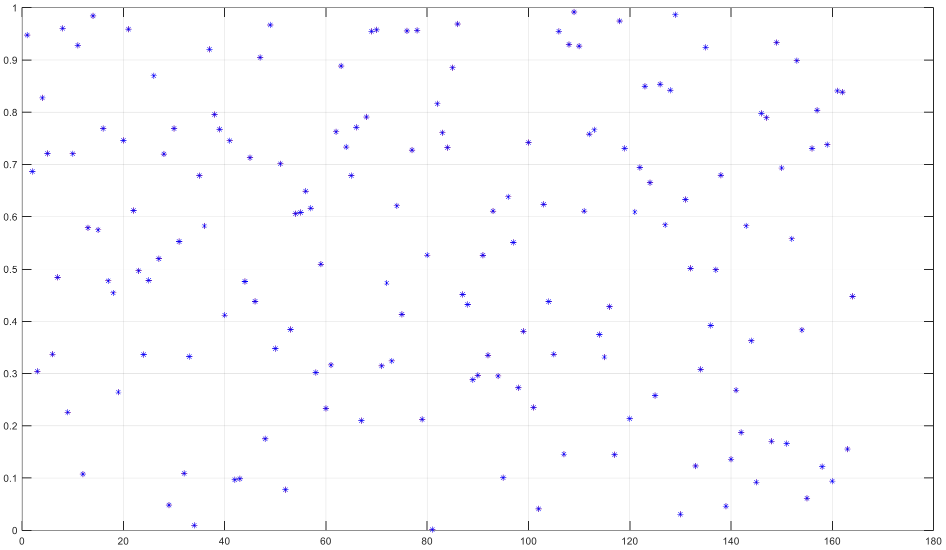


Figure 23. Encoder-Decoder estimation.

Figure 24 shows a symmetric matrix of the order (140x140) of the relative importance of input variables for accurate estimates of the response variable in ascending order. The most important variable is input variable 72, with a probability of 1, and the second most important variable is variable 54, with a probability of 5.9542e^{-10}. This approach is one of the main contributions of research, as there is a lack of tools to assess the importance of input variables, and no analysis exists of their importance in neural networks or transformers.

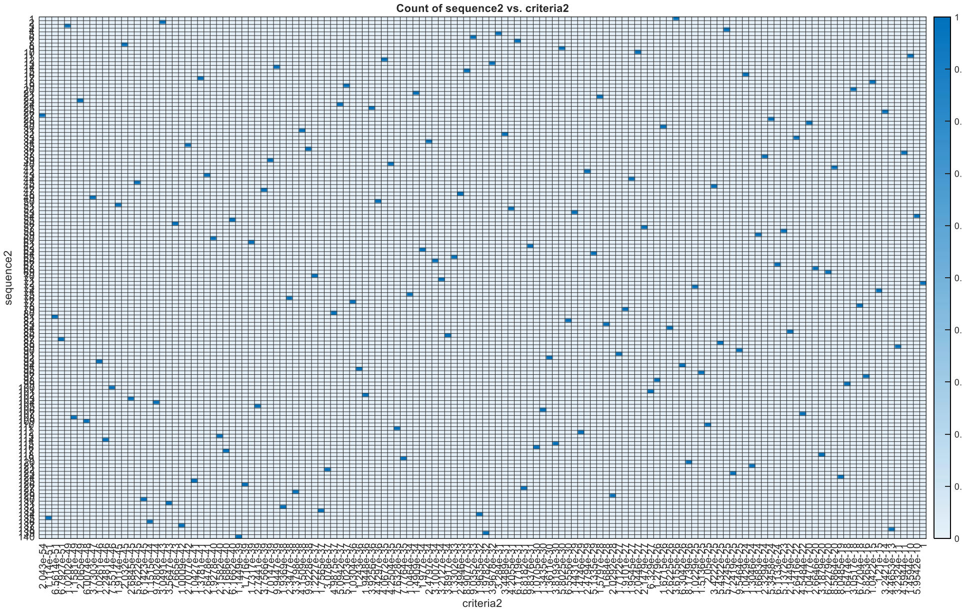


Figure 24. Sequence of importance input variables.

Figure 25 shows a symmetric matrix of the order (164x164) of the relative importance of training patterns for accurate estimates of the response variable in ascending order. The most important is the training pattern number 129, with a probability of 0.24021, and the second most important is the training pattern number 81, with a probability of 0.23495.

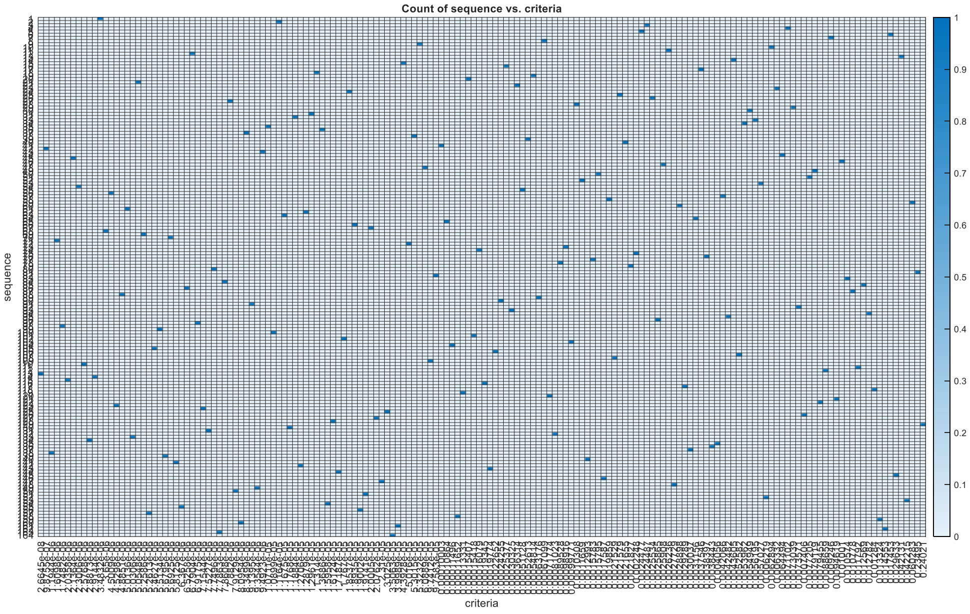


Figure 25. Sequence of importance training data.

4 Conclusions

In the present research, a structure of a pre-trained generative transformer is proposed to make accurate estimates based on input-output causal relationships. Estimates are made using input variables and training patterns in matrices, with a variable range of [100,200] for rows and columns and coded input values in a range of [-6 6]. The matrix feeds the encoder and decoder, using three blocks of attention mechanisms, to leverage a neural network and estimate the response variable. The training patterns are processed in 3 blocks of attention, with related input variables, allowing for a more precise estimation of the response variable. By using the PGT proposed structure of the estimation transformer, we are strengthening the training base of the input variables, finding a triple dependence between them, by using the dot-product operation on three occasions: first in the encoder block, second in the first decoder block and third in the second decoder block.

Having strengthened the training base, at the end of the proposed transformer structure a multilayer neural network is used for estimation. This action is the reason for comparing results with a multi-layered neural network, using the same Levenberg-Marquardt training algorithm and the same number of epochs. The structure proposed at the end uses a multi-layer neural network to perform the estimation. The difference lies in the use of a triply related training base between input variables, using three blocks of attention and applying three times the point product operation, allowing fast convergence in only two epochs with 100% reliability in accuracy.

In addition, two linear combinations are used to obtain the sequence of importance of the input variables, highlighting which are the most important for making accurate estimates, as well as the importance of the training patterns used. The results show 100% reliability in the simulated estimates, validating the proposal as a robust and reliable alternative for performing causal type estimates across any production sector. The proposed approach for case studies with input variable matrices higher than (200x200), PGT for estimation of response variable needs more than two epochs for convergence.

Future work will use the theory of vision systems to segment the matrices of input variables and training patterns, extract characteristics to feed the pretrained generative transformer, and estimate the variable response.

References

Bahdanau, D., Cho, K., & Bengio, Y. (2014). *Neural machine translation by jointly learning to align and translate* (arXiv:1409.0473). arXiv. https://arxiv.org/abs/1409.0473

Chen, D., Hong, W., & Zhou, X. (2022). Transformer network for remaining useful life prediction of lithium-ion batteries. *IEEE Access, 10*, 19621–19628. https://doi.org/10.1109/ACCESS.2022.3151975

Chen, Q., Meng, Z., Liu, X., Jin, Q., & Su, R. (2018). Decision variants for the automatic determination of optimal feature subset in RF-RFE. *Genes, 9*(6), 301. https://doi.org/10.3390/genes9060301

Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). *Learning phrase representations using RNN encoder–decoder for statistical machine translation* (arXiv:1406.1078). arXiv. https://arxiv.org/abs/1406.1078

Choudhary, A., & Arora, A. (2024). Assessment of bidirectional transformer encoder model and attention-based bidirectional LSTM language models for fake news detection. *Journal of Retailing and Consumer Services, 76*, 103545. https://doi.org/10.1016/j.jretconser.2023.103545

Ding, Y., & Jia, M. (2021). A convolutional transformer architecture for remaining useful life estimation. In *2021 Global Reliability and Prognostics and Health Management (PHM-Nanjing)* (pp. 1–7). IEEE. https://doi.org/10.1109/PHM-Nanjing52125.2021.9612814

Dosovitskiy, A., Beyer, L., Kolesnikov, A., et al. (2020). *An image is worth 16×16 words: Transformers for image recognition at scale* (arXiv:2010.11929). arXiv. https://arxiv.org/abs/2010.11929

Guo, F., Niu, H., & Li, M. (2024). Remaining useful life prediction of precision bearing based on multi-head attention mechanism. *Journal of Physics: Conference Series, 2762*(1), 012053. https://doi.org/10.1088/1742-6596/2762/1/012053

Huang, J., Liu, T., Zhan, Y., Chen, Z., Xiao, X., Wu, Q., Zheng, Y., Liu, R., & Su, Y. (2025). Prediction model of dam deformation based on attention mechanism. *Journal of Physics: Conference Series, 3005*(1), 012021. https://doi.org/10.1088/1742-6596/3005/1/012021

Hu, Q., Zhao, Y., & Ren, L. (2023). Novel transformer-based fusion models for aero-engine remaining useful life estimation. *IEEE Access, 11*, 52668–52685. https://doi.org/10.1109/ACCESS.2023.3277730

Liu, L., Song, X., & Zhou, Z. (2022). Aircraft engine remaining useful life estimation via a double attention-based data-driven architecture. *Reliability Engineering & System Safety, 221*, 108330. https://doi.org/10.1016/j.ress.2022.108330

Marinos, T., Markaki, M., Sarafidis, Y., Georgopoulou, E., & Mirasgedis, S. (2025). The economic effects of the green transition of the Greek economy: An input–output analysis. *Energies, 18*, 4177. https://doi.org/10.3390/en18154177

Mo, Y., Wu, Q., Li, X., & Huang, B. (2021). Remaining useful life estimation via transformer encoder enhanced by a gated convolutional unit. *Journal of Intelligent Manufacturing, 32*(7), 1997–2006. https://doi.org/10.1007/s10845-020-01698-4

Poczeta, K., & Papageorgiou, E. I. (2022). Energy use forecasting with the use of a nested structure based on fuzzy cognitive maps and artificial neural networks. *Energies, 15*, 7542. https://doi.org/10.3390/en15207542

Rahman, M. M., Rahman, S. M., Shafiullah, M., Hasan, M. A., Gazder, U., Al Mamun, A., Mansoor, U., Kashifi, M. T., Reshi, O., & Arifuzzaman, M. (2022). Energy demand of the road transport sector of Saudi Arabia—Application of a causality-based machine learning model to ensure sustainable environment. *Sustainability, 14*(23), 16064. https://doi.org/10.3390/su142316064

Rende, R., & Viteritti, L. (2025). Are queries and keys always relevant? A case study on transformer wave functions. *Machine Learning: Science and Technology, 6*, 010501. https://doi.org/10.1088/2632-2153/ada1a0

Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems, 27*, 3104–3112.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). *Attention is all you need* (arXiv:1706.03762). arXiv. https://arxiv.org/abs/1706.03762

Wang, H.-K., Cheng, Y., & Song, K. (2021). Remaining useful life estimation of aircraft engines using a joint deep learning model based on TCNN and transformer. *Computational Intelligence and Neuroscience, 2021*, 5185938. https://doi.org/10.1155/2021/5185938

Wang, Y. H., Tie, L., Qi, L., Wang, F., & Wang, L. (2021). Tumor type prediction based on residual attention model. *Journal of Physics: Conference Series, 1914*(1), 012029. https://doi.org/10.1088/1742-6596/1914/1/012029

Zhang, H., & Shafiq, M. O. (2024). Survey of transformers and towards ensemble learning using transformers for natural language processing. *Journal of Big Data, 11*, 25. https://doi.org/10.1186/s40537-024-00878-9

Zhang, L., & Zeng, X. (2021). Research on transformer fault diagnosis based on genetic algorithm optimized neural network. *Journal of Physics: Conference Series, 1848*(1), 012004. https://doi.org/10.1088/1742-6596/1848/1/012004

Zhang, Z., Song, W., & Li, Q. (2022). Dual-aspect self-attention based on transformer for remaining useful life prediction. *IEEE Transactions on Instrumentation and Measurement, 71*, 1–11. https://doi.org/10.1109/TIM.2022.3160561

Zhou, T., Fu, C., Liu, Y., & Xiang, L. (2025). Groundwater level estimation using improved transformer model: A case study of the Yellow River Basin. *Water, 17*, 2318. https://doi.org/10.3390/w17152318