STATISTICS, OPTIMIZATION AND INFORMATION COMPUTING
Stat., Optim. Inf. Comput., Vol. **15**, February 2026, pp 1357–1369.
Published online in International Academic Press (www.IAPress.org)

Resolution of linear interval systems using neural networks and their application to the Leontief economic model

Benhari Mohamed Amine [1,*], Kaicer Mohammed [1], Bennis Driss [2]

[1]*Department of Mathematics, Faculty of Sciences, Ibn Tofail University, Kenitra, Morocco*
[2]*Department of Mathematics, Faculty of Sciences, Mohammed V University, Rabat, Morocco*

Abstract Linear systems with interval coefficients are a class of mathematical modeling problems whose coefficients do not have exact values, but are multi-valued sets of possible values. This feature characterizes many processes in the real world, especially in economics. The solution of these systems is essential for making decisions under uncertainty. In machine learning, neural networks represent an excellent tool for solving this problem. The performance of neural networks is largely due to their flexibility; they can learn multifaceted dependencies without any prior assumptions about the distribution of input data. In addition, it is important when the original data are measured inaccurately and/or noised. Such data are common in economic forecasting. In addition, their ability to learn data structures enables them to make accurate forecasts based on new data, which is decisive for risk management and strategic decision-making.
One possibility is to model a concrete application on the Leontief model, which describes the flow between individual sectors of the economy. This way, when neural networks are integrated into linear systems with interval coefficients, it is possible to forecast the impact of variations in demand and supply over the whole economy, reducing the uncertainties of economic activity forecasts. In conclusion, the new methodology of neural networks in the resolution of linear systems with interval coefficients may, at some point, prove essential progress in managing economic uncertainty, allowing companies and decision-makers to navigate more confidently in a complex and unpredictable environment.

Keywords Arithmetic Interval, Neural Networks, System Of Interval Linear Equations, Leontief Model

DOI: 10.19139/soic-2310-5070-2778

1. Introduction

Solving linear systems is an essential task in the applied sciences, and several methods have been developed to deal with these systems, whether they are defined precisely (real values) or uncertainly (interval). When the coefficients of the matrices are exact real values, methods such as Gauss elimination, the determinant method, and Cholesky decomposition offer efficient and stable solutions. However, in real applications, the coefficients of systems are often uncertain or subject to measurement errors, leading to the study of linear systems with interval coefficients. This type of system, where each coefficient is represented by an interval of possible values, has been studied to model uncertainties more realistically.
Among the classic methods for solving linear systems with interval coefficients is Gauss elimination [2], a standard method that consists of transforming a matrix into an upper triangular form, making it easier to solve by backward substitution. This method is widely used for its simplicity and efficiency in systems with real coefficients. However, when applied to matrices with interval coefficients, this method suffers from a major drawback: the amplification of uncertainties as the elimination steps progress. Each subtraction and division step tends to widen the solution intervals[1]. They showed that this propagation of uncertainties can make the solution intervals too wide to be

[*]Correspondence to: Benhari Mohamed Amine (Email: mohamedamine.benhari@uit.ac.ma). Department of Mathematics, Ibn Tofail University. University campus, Kenitra, BP 133, Morocco.

ISSN 2310-5070 (online) ISSN 2311-004X (print)

useful, particularly when the coefficients of the matrix are highly variable. To overcome this limitation, techniques such as partial pivoting have been proposed to reduce the effect of interval amplification, but these solutions remain limited in systems with high uncertainties.
Another commonly used method is the determinant method [9], which is used to test the invertibility of a real matrix A. If the determinant of matrix A is different from zero, then the matrix is invertible, guaranteeing a unique solution for the system $AX = B$. However, for matrices with interval coefficients, calculating the determinant becomes more complex. The determinant of an interval matrix is itself an interval, the bounds of which must be calculated. If these bounds include zero, the matrix may not be invertible in certain configurations of the coefficients, making it more difficult to interpret the results. In addition, calculating the bounds of the determinant can lead to overly conservative results, i.e. The possible solutions are often overestimated to ensure that all configurations are covered. This can lead to solution intervals that are too wide and imprecise.
Another approach has been based on the de Cholesky decomposition [8] for Generalised Interval Matrices to obtain exact algebraic solutions for triangular linear systems of intervals, it offers significant improvements in terms of accuracy and modelling capability of linear systems with interval coefficients, however the method relies on certain assumptions, such as the symmetry of interval matrices and the positivity of diagonal elements, in addition to challenges of computational complexity, which may limit its applicability to certain types of linear interval systems.
Classical methods for solving linear systems with interval coefficients, although effective in certain configurations, suffer from a number of limitations such as complexity, execution time, and the propagation of interval errors, which can render solutions unusable.
Given these limitations, neural networks offer a promising alternative for solving linear systems with interval coefficients. Unlike traditional techniques, neural networks can be trained to capture the complex relationships between interval coefficients and the corresponding solutions, and thanks to their learning capacity and flexibility, can handle larger intervals without compromising the accuracy of the solutions, offering a robust approach in the face of uncertainty.
In this paper, we explore this approach by examining how neural networks can be adapted to efficiently solve linear systems with interval coefficients. We begin with a presentation of the necessary theoretical background, including the basic concepts of linear systems with interval coefficients and neural networks. Next, we detail our proposed methodology, explaining how we have adapted neural networks to solve these systems. We then present the results of our experiments, demonstrating the effectiveness of our approach compared with existing methods. Finally, we conclude with an application of an economic model known as the Leontief model.
This study contributes to opening new perspectives in the solution of linear systems with interval coefficients, offering potentially accurate and efficient solutions for applications in various scientific and technical fields.

2. Interval arithmetic

Interval arithmetic, an extension of classical mathematics, aims to manipulate sets of values rather than single real numbers. It is advantageous in calculations where the parameters or data contain uncertainties. An interval is defined as a set containing all possible values between a lower bound and an upper bound, usually represented as $[a_1; a_2]$, where a_1 is the lower bound and a_2 the upper bound.

2.1. Arithmetical operations on intervals

Arithmetic operations on intervals require a redefinition of each elementary operation (addition, subtraction, multiplication, and division) to ensure that all possible combinations of the values included in these intervals are considered. The operations therefore do not produce a single result, but a new interval that encompasses all possible values [6].
Mathematically speaking [1] :
The addition between two intervals $\widehat{a} = [a_1; a_2]$ with $a_1 \leq a_2$ and $\widehat{b} = [b_1; b_2]$ with $b_1 \leq b_2$ is defined as the sum of

the bounds of these intervals, creating a new interval which contains all the possible sums between the values of the two intervals.

$$\widehat{a}+\widehat{b}=[a_1+b_1;a_2+b_2]$$

Interval subtraction is similar to addition, except that the limits of the intervals are subtracted from each other.

$$\widehat{a}-\widehat{b}=[a_1-b_2;a_2-b_1]$$

Multiplication of the intervals takes into account all the possible combinations between the limits of the intervals to ensure that the product covers all the possible values.

$$\widehat{a}\times\widehat{b}=[min(a_1b_1,a_1b_2,a_2b_1,a_2b_2);max(a_1b_1,a_1b_2,a_2b_1,a_2b_2)]$$

The division is defined in a similar way to multiplication but imposes additional restrictions. Division by an interval containing zero is not permitted, as this would create indeterminacy.
If $0\notin\widehat{b}$ then :

$$\widehat{a}\times\widehat{b}=[min(a_1/b_1,a_1/b_2,a_2/b_1,a_2/b_2);max(a_1/b_1,a_1/b_2,a_2/b_1,a_2/b_2)]$$

Also if $\widehat{a}=\widehat{b}$ then $\widehat{a}/\widehat{b}=[1;1]$

2.2. Interval matrix operations

When linear systems involve matrices with interval coefficients, classical matrix operations, such as the sum, scalar product, or multiplication of matrices, must be redefined in terms of intervals. A matrix with interval coefficients is a generalization of classical matrices in which each element is an interval instead of a precise real number.
Formally, a square matrix with interval coefficients $\widehat{A}_{n,n}$ [7] is defined as a matrix in which each element $[\widehat{a}_{i,j}]$ is a real interval.

$$\widehat{A}_{n,n}=\begin{pmatrix}[\widehat{a}_{1,1}] & [\widehat{a}_{1,2}] & \cdots & [\widehat{a}_{1,n}]\\ [\widehat{a}_{2,1}] & [\widehat{a}_{2,2}] & \cdots & [\widehat{a}_{2,n}]\\ \vdots & \vdots & \ddots & \vdots\\ [\widehat{a}_{n,1}] & [\widehat{a}_{n,2}] & \cdots & [\widehat{a}_{n,n}]\end{pmatrix}$$

Operations on these matrices follow the same principles as for operations on real numbers, but applied to each element individually.
For two matrices with interval coefficients $\widehat{A}$ and $\widehat{B}$, the sum is obtained by adding the corresponding intervals element by element and The product of a matrix with interval coefficients by a scalar is obtained by multiplying each element of the matrix by this scalar.
The matrix product of two matrices with interval coefficients is obtained by applying the interval multiplication rules for each combination of matrix elements.
In other words:
If $\widehat{A}_{n,n}$ and $\widehat{B}_{n,n}$ are two interval matrices and $\lambda\in\mathbb{R}$, then :
For any $(1\leq i\leq n$ and $1\leq j\leq n)$:
- $\lambda\widehat{A}_{n,n}=\lambda[\widehat{a}_{i,j}]$
- $\widehat{A}_{n,n}+\widehat{B}_{n,n}=[\widehat{a}_{i,j}+\widehat{b}_{i,j}]$
- $\widehat{A}_{n,n}-\widehat{B}_{n,n}$= $[\widehat{a}_{i,j}-\widehat{b}_{i,j}]$ if $\widehat{A}\neq\widehat{B}$, if not $\widehat{A}_{n,n}-\widehat{B}_{n,n}$=[0; 0]
- $\widehat{A}_{n,n}.\widehat{B}_{n,n}=\left[\sum_{k=1}^{n}\widehat{a}_{i,k}.\widehat{b}_{k,j}\right]$

Some properties of classical matrices also generalise to matrices with interval coefficients. For example :

The determinant of a square matrix $\widehat{A}_{n,n}$ with interval coefficients is a generalization of the classical determinant denoted $det(\widehat{A}_{n,n})$ formally,

$$det(\widehat{A}_{n,n}) = \sum_{i=1}^{n} (-1)^{i+j} \widehat{A}_{i,j}.\widehat{A}_{cof(i,j)}$$

Where $\widehat{A}_{cof(i,j)}$ is the cofactor of the matrix associated with element $\widehat{A}_{i,j}$. Also if $0 \notin det(A)$, then the matrix $\widehat{A}_{n,n}$ is invertible.

2.3. Algebraic solution of an interval linear system

Definition : Linear systems with interval coefficients [2] are a generalization of classical linear systems, in which the coefficients of matrices and vectors are represented not by exact numbers, but by intervals.
A linear system with interval coefficients can be formally expressed as : $\widehat{A}\widehat{X} = \widehat{B}$, or :

- $\widehat{A}$: Is an interval square matrix of order $n \times n$.
- $\widehat{B}$: Is an interval coefficient vector of order n.
- $\widehat{X}$: Is the interval solution vector.

The objective of solving this type of system is to determine the smallest interval containing all possible solutions X_i for each real matrix $A_i \in \widehat{A}$ and each real vector $B_i \in \widehat{B}$.

3. Resolution of linear systems with interval coefficients using neural networks

Artificial neural networks, inspired by human brain processes, are computational structures capable of processing and generalizing complex data. They are made up of layers of interconnected neurons, each connection being defined by an adjustable weight. The main advantages of neural networks include their ability to model complex non-linear relationships, their robustness to variations in input data, and their ability to adapt and learn from the processes underlying the data without requiring explicit programming of the rules. These characteristics make them particularly effective for tasks such as image recognition, natural language processing, and time series prediction.
In the context of linear systems with interval coefficients, neural networks offer several advantages. Firstly, they can approximate complex functions and handle uncertain or noisy data. These networks can be trained on data sets where the coefficients of matrices and vectors are represented by intervals, enabling them to estimate possible solutions within these intervals.
This approach not only makes it possible to find approximate solutions, but also to bypass certain restrictive conditions on the input matrices, which are often present in traditional iterative methods. In addition, neural networks offer the possibility of dealing with large-scale systems while guaranteeing flexible modeling of uncertainties in the coefficients, which is not always evident with traditional approaches.
In this section, we present an algorithmic approach for solving a linear system with interval coefficients $\widehat{A}\widehat{X} = \widehat{B}$, using neural networks.

3.1. Proposed methodology

The objective is to predict the values of the vector $\widehat{X}$ using a neural network model trained on input-output examples represented by intervals. This model aims to generalize the relationship between the coefficients of $\widehat{A}$ and $\widehat{B}$, both defined by intervals and to provide an estimate of the solution vector $\widehat{X}$ that encompasses all possible solutions within these intervals.

3.2. Generation of training data using Monte Carlo simulation techniques

The primary step in this methodology is generating a set of linear systems with interval coefficients. The subsequent steps in the procedure are as follows:
1. Defining the coefficients of the matrices and interval vectors: Each coefficient of the interval matrix A and the interval vector B is defined by a lower and upper bound, denoted respectively by $[a_{ij}^-; a_{ij}^+]$ and $[b_{ij}^-; b_{ij}^+]$.
2. Monte Carlo simulation: Many samples A^k and B^k are drawn randomly by selecting each coefficient uniformly within its interval. Each realisation corresponds to a deterministic system in the form: $A^k X^k = B^k$.
3. Building solutions: For each solution X_i, we collect all the solutions obtained on the samples, then we define:

$$X_i^- = \min_k x_i^k \text{ and } x_i^+ = \max_k x_i^k$$

It should be noted that the coefficients of the solution vector are, by definition, guaranteed to contain all possible solutions.
4. Training and testing: $\widehat{A}$, $\widehat{B}$, and $\widehat{X}$ obtained serve as a database, subdivided into a training set and a validation/test set.
The number of samples generated by Monte Carlo was set at **10,000**. This choice ensures adequate coverage of interval variability, particularly for higher-dimensional systems ($n = 5$ and $n = 10$). Smaller sizes led to unstable results, while increasing beyond 10,000 did not bring any noticeable improvement but increased the training time. This compromise ensures the robustness and reproducibility of the proposed method.

3.3. Interval neural model and adapted loss function

The proposed model is based on a fully connected multilayer perceptron, comprising two hidden layers of 256 neurons with ReLU activation, followed by an output layer.The proposed model is based on a fully connected multilayer perceptron, comprising two hidden layers of 256 neurons with ReLU activation, followed by an output layer. However, the size of the hidden layers was set to 256 neurons after several comparative tests (64, 128, 256, 512). Smaller sizes resulted in under-approximation of intervals (increased MAE and low coverage), while larger sizes caused overfitting and calibration instability.
The latter is transformed via an Interval Head, which separates the output vector into two parts. The centres c_i represent the predicted mean value of each variable, and the widths w_i are constrained to be positive via a softplus function to ensure the consistency of the intervals.
The cost function plays an essential role in the stability of the output intervals. We have adopted a hybrid loss combining several complementary terms:
1. **MSE**(c): a quadratic error on the centres, which ensures the accuracy of the central predictions, improves the coefficient of determination R^2 and reduces the MAE.
With:

$$\mathcal{L}_c = MSE(c_{true}; c_{pred})$$

2. **MSE**(w): a quadratic error on widths, which prevents their collapse (a common case where the network learns to reduce intervals to zero artificially).
With:

$$\mathcal{L}_w = MSE(w_{true}; w_{pred})$$

3. **Hausdorff loss**: a term measuring the maximum difference between true and predicted bounds, to ensure that both ends of the interval are correctly positioned.
With:

$$\mathcal{L}_H = max(|Low_{true} - Low_{pred}|; |Up_{true} - Up_{pred}|)$$

4. **Coverage**: a penalty for cases where the predicted interval does not contain the reference interval. This term pushes predictions to encompass true solutions, while avoiding excessive expansion.
With:

$$\mathcal{L}_{cov} = ReLU(Low_{true} - Low_{pred}) + ReLU(Up_{true} - Up_{pred})$$

From these four terms, we can define an overall loss:

$$\mathcal{L} = \mathcal{L}_c + \mathcal{L}_w + \lambda_u.\mathcal{L}_{cov} + \lambda_H.\mathcal{L}_H$$

With λ_u and λ_H are weighting coefficients chosen experimentally to balance accuracy and coverage.

3.4. Post-hoc calibration by dimension

To boost the reliability of the neural network's predictions, a post-hoc calibration step was applied for each dimension. This process separately adjusts the width of each output interval using a multiplier factor m_j, determined to achieve a target coverage level (e.g., 90%). In concrete terms, the predicted centres of the intervals are retained, while the half-widths are expanded or contracted so that the proportion of actual solutions contained in the predicted intervals reaches the set threshold. Unlike uniform calibration, this approach allows each variable to be finely tuned to its own variability, thus avoiding unnecessary over-expansion in the most stable dimensions. This strategy ensures a balanced compromise between accuracy (low MAE and good R^2), controlled interval width, and high coverage.

3.5. Algorithm linear system solution

Algorithm 1: An algorithm proposed for neural network solving

```
Input: Number of unknowns $n$, number of samples $N$, intervals of $A$ and $B$, target coverage rate $\alpha$.
Output: Lower and upper bounds $\hat{X}^-, \hat{X}^+$ of the solutions.
Step 1: Data generation (Monte Carlo)
    1. Sample several values of matrix $A$ and vector $B$ within their intervals.
    2. Solve each system $A^{(k)}X^{(k)} = B^{(k)}$.
    3. Construct the reference intervals of $X$ by taking the minimum and maximum of the solutions.
Step 2: Preparing the dataset
    1. Construct the inputs by concatenating the boundaries of $A$ and $B$.
    2. Construct the target outputs as the centre and width of the intervals of $X$.
Step 3: Neural network training
    1. Create a dense network with two hidden layers (256 neurons, ReLU activation).
    2. The network output comprises the centre and a positive width (softplus activation).
    3. Optimise with Adam and a hybrid loss function combining:
    - quadratic error on centres,
    - error on widths,
    - a penalty for intervals that do not cover the true solution,
    - Hausdorff distance between intervals.
Steps 4: Post-hoc calibration by dimension
    1. For each dimension $j$, find a factor $m_j$ that slightly increases the predicted width.
    2. Adjust the predicted intervals to achieve approximately $\alpha = 90\%$ coverage.
Step 5: Evaluation and prediction
    1. Evaluate the model on test data using the following metrics: MAE, $R^2$, mean width, coverage, and
      Hausdorff distance.
    2. Apply the calibrated model to the reference system to obtain the predicted intervals.
```

4. Numerical results and comparative analysis

This section presents the results obtained by applying the proposed neural model to linear systems of dimensions $n = 2$, $n = 3$, $n = 5$, and $n = 10$. Performance is evaluated using standard metrics: mean absolute error (MAE), coefficient of determination (R^2), mean interval width, Hausdorff distance, and coverage rate. Results are reported before and after post-hoc calibration by dimension.

4.1. Comparative tables and analysis

The following tables summarise the results for both approaches (classical MSE loss and stabilised hybrid loss).

Table 1. Results for $n = 2$

Method	MAE	R^2	Width (modified/actual)	Hausdorff	Coverage	Coverage (cal.)
MSE	0.0550	0.9986	2.35 / 2.35	0.1094	25.50%	89.40%
Hybrid	0.0593	0.9984	2.14 / 2.36	0.1689	1.35%	92.25%

Table 2. Results for $n = 3$

Method	MAE	R^2	Width (modified/actual))	Hausdorff	Coverage	Coverage (cal.)
MSE	0.0916	0.9964	2.36 / 2.36	0.1623	20.40%	90.40%
Hybrid	0.1019	0.9955	2.02 / 2.37	0.2783	0.80%	90.23%

Table 3. Results for $n = 5$

Method	MAE	R^2	Width (modified/actual))	Hausdorff	Coverage	Coverage (cal.)
MSE	0.1385	0.9923	2.35 / 2.36	0.2248	16.10%	88.84%
Hybrid	0.1510	0.9908	1.85 / 2.36	0.4091	0.68%	90.36%

Table 4. Results for $n = 10$

Method	MAE	R^2	Width (modified/actual))	Hausdorff	Coverage	Coverage (cal.)
MSE	0.2155	0.9809	2.36 / 2.36	0.3122	12.95%	89.82%
Hybride	0.2422	0.9765	1.66 / 2.37	0.6035	0.72%	89.97%

The results show that:

- Both approaches (MSE and Hybrid) achieve very low MAE (< 0.25) even for $n = 10$, proving the strength of the model.
- The R^2 remains above 0.97 for all cases, confirming the precision of the predicted centres.
- Before calibration, coverage is low (from 1% to 25%), confirming a typical underestimation of intervals.
- After post-hoc calibration by dimension, coverage consistently reaches $\approx 90\%$, with adjusted widths close to the true widths.
- The Hybrid method slightly reduces the widths, but at the cost of very low gross coverage; calibration compensates for this weakness.

4.2. Visualisations

Figures 1–4 illustrate the distributions of predicted and actual widths (histograms) as well as the correspondence between predicted centres and actual centres (scatter plots).

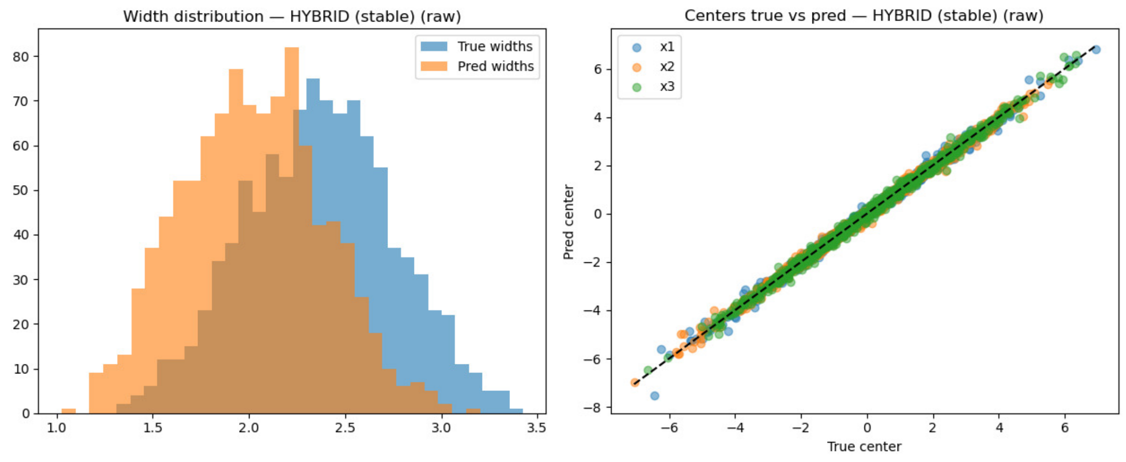

Figure 1. Hybrid results for $n = 3$ (before calibration).

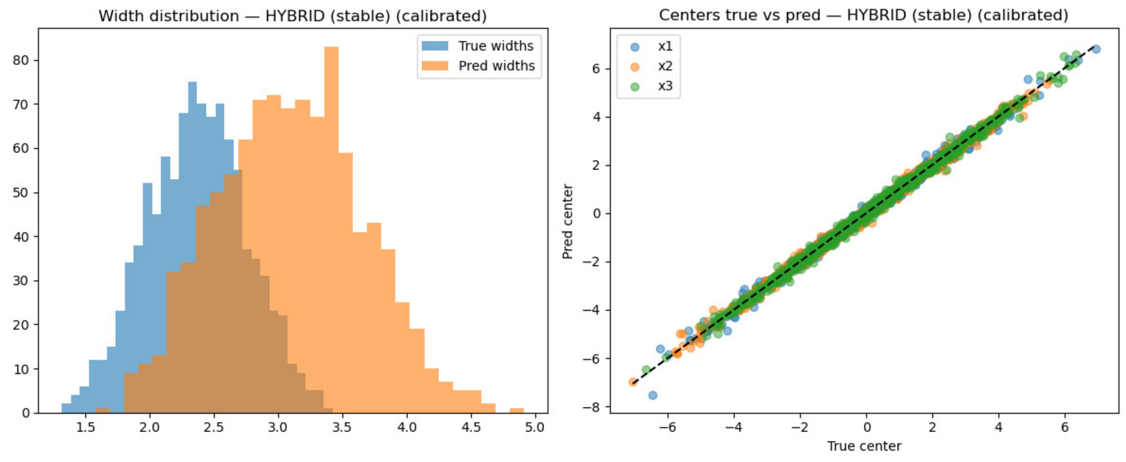

Figure 2. Hybrid results for $n = 3$ (after calibration).

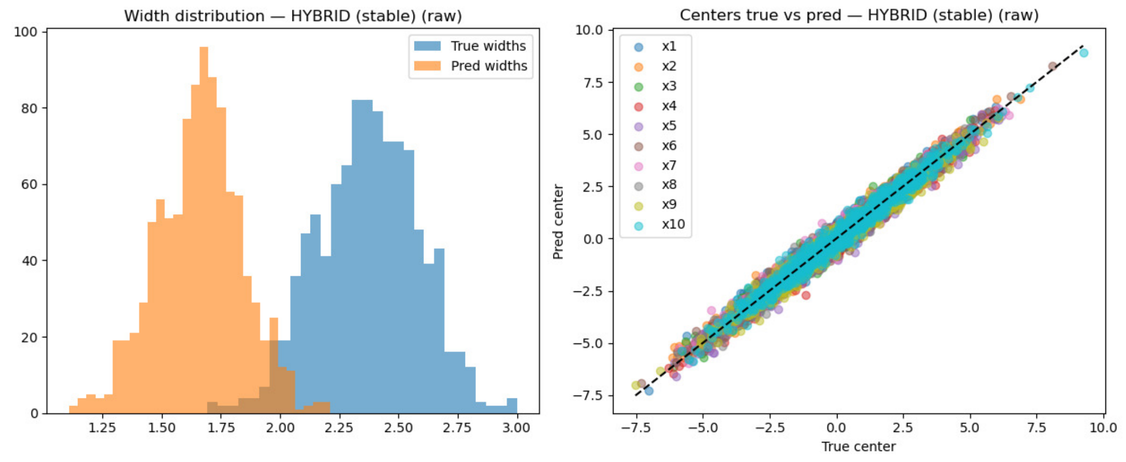

Figure 3. Hybrid results for $n = 10$ (before calibration).

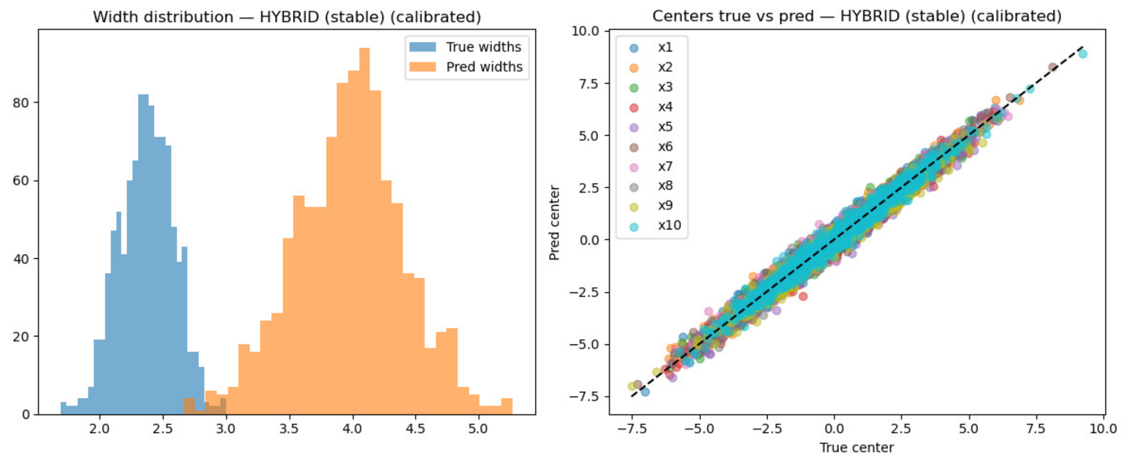

Figure 4. Hybrid results for $n = 10$ (after calibration).

In general, the results highlight several key points:

- The centres of the intervals (c) are predicted with very high accuracy, regardless of the dimension ($R^2 > 0.97$).
- The widths (w) are underestimated before calibration, resulting in very low coverage ($< 25\%$) .
- Post-hoc calibration by dimension corrects this bias and achieves coverage of approximately 90%, without significantly degrading the accuracy of the centres.
- As the dimension increases ($n = 10$), the variance of the calibrated intervals increases, which can introduce a slight over-approximation. Nevertheless, the results remain stable and demonstrate the robustness of the neural model.

These results show that the proposed approach combines accurate prediction of centres and, thanks to calibration, a reliable envelope of intervals. This is a significant advance over conventional methods, which often suffer from excessive widening of intervals.

4.3. Application on a 3D system

To illustrate the effectiveness of the proposed methodology, we consider a linear system with coefficient matrices of dimensions 3×3, and we compare the results obtained by our algorithm with those obtained using traditional iterative methods such as the Jacobi and Gauss-Seidel methods[5].
We take the example from [12]:

$$\widehat{A} = \begin{pmatrix} [3.7; 4.3] & [-1.5; -0.5] & [0; 0] \\ [-1.5; -0.5] & [3.7; 4.3] & [-1.5; -0.5] \\ [0; 0] & [-1.5; -0.5] & [3.7; 4.3] \end{pmatrix} \text{ and } \widehat{B} = \begin{pmatrix} [-14; 0] \\ [-9; 0] \\ [-3; 0] \end{pmatrix}$$

Since $det(\widehat{A}) = [36.5; 75.4]$, then $\widehat{A}$ is invertible and the system has a solution.

Table 5. Comparison of solutions for a 3D system

Method	Vector solution
Baseline (calibrated)	$\begin{pmatrix}[-3.89;-0.03]\\ [-2.99;-0.83]\\ [-1.23;-0.35]\end{pmatrix}$
Hybrid (calibrated)	$\begin{pmatrix}[-4.42;0.09]\\ [-2.41;-0.88]\\ [-1.25;-0.63]\end{pmatrix}$
Gauss-Seidel	$\begin{pmatrix}[-4.03;0.86]\\ [-2.73;1.32]\\ [-1.12;0.65]\end{pmatrix}$
Gauss elimination (Ning et al.)	$\begin{pmatrix}[-6.38;0]\\ [-6.40;1.32]\\ [-3.40;0]\end{pmatrix}$
Hansen's method (Ning et al.)	$\begin{pmatrix}[-6.38;1.12]\\ [-6.40;1.54]\\ [-3.40;1.40]\end{pmatrix}$

The results highlight the effectiveness of neural networks for solving linear systems with interval coefficients, compared with classical methods such as Gauss elimination, Gauss-Seidel, and Hansen's method.
Firstly, in terms of accuracy, neural networks have made it possible to significantly reduce the width of solutions, as observed in the results for the two-dimensional and three-dimensional systems.
Our results also show that the neural network approach offers several distinct advantages. Most importantly, it significantly reduces computation time. Traditional iterative methods often require many iterations to converge to a solution, particularly for systems with highly uncertain coefficients. In contrast, the neural network, once trained, offers rapid convergence and does not require iterative recalculations, making it more suitable for real-time or large-scale applications and it can provide solutions almost instantaneously.
In addition, the neural network approach is more scalable and adaptable. As system limits increase, the neural network can be retrained with new data, whereas iterative methods may require substantial adaptation and recalibration.
The inflexibility and efficiency of the neural network system make it a compelling option for working on linear systems with interval coefficients, offering better performance and ease of perpetuation.

5. Application on the Leontief model

The Leontief model [3] is an profitable tool used to assayinter-industry overflows and their effects on the global economy. Traditionally, this model uses matrices with fixed coefficients. Still, in a environment where coefficients can vary due to economic uncertainties, interval coefficients come more applicable. We can acclimatize Leontief's model to use the neural network system to solve systems of equations with interval coefficients.

The Leontief model is represented by the following equation $\widehat{X} = \widehat{A}\widehat{X} + \widehat{D}$ with :

- $\widehat{X}$ is the vector of sector production, it represents the total production of each industry.
- $\widehat{D}$ is the final demand vector, Each element $\widehat{d_i}$ of $\widehat{D}$ indicates the total demand for the product of industry i.
- $\widehat{A}$ is the technical coefficients matrix, each element $\widehat{a_{ij}}$ of $\widehat{A}$ indicates how many units of product from industry j are needed to produce one unit of product from industry i.

The Input-Output table is an essential tool for representing inter-industry relations in an economy. It is used to visualize and analyze production and demand flows between different sectors.

In this section, we apply our neural approach to Leontief's input-output model, defined by the system:

$$(\widehat{I} - \widehat{A})\widehat{X} = \widehat{D},$$

Where A is the matrix of technical coefficients and $\widehat{D}$ is the final demand vector.

5.1. Data used

The matrix $\widehat{A}$ considered is reported by [11]. :

$$\widehat{A} = \begin{bmatrix} [0.1389, 0.1396] & [0.0804, 0.0806] & [0.0033, 0.0036] & [0.0001, 0.0001] & [0.0321, 0.0327] & [0.0052, 0.0054] \\ [0.1565, 0.1571] & [0.5043, 0.5047] & [0.5634, 0.5643] & [0.3401, 0.3421] & [0.2405, 0.2411] & [0.2642, 0.2654] \\ [0.0001, 0.0002] & [0.0004, 0.0005] & [0.0067, 0.0069] & [0.0013, 0.0015] & [0.0090, 0.0090] & [0.0145, 0.0149] \\ [0.0110, 0.0113] & [0.0178, 0.0178] & [0.0296, 0.0298] & [0.0140, 0.0146] & [0.1103, 0.1110] & [0.0413, 0.0417] \\ [0.0214, 0.0216] & [0.0749, 0.0750] & [0.0917, 0.0917] & [0.0490, 0.0490] & [0.0400, 0.0402] & [0.0454, 0.0458] \\ [0.0268, 0.0271] & [0.0284, 0.0407] & [0.0142, 0.0144] & [0.0358, 0.0363] & [0.1086, 0.1090] & [0.0981, 0.0987] \end{bmatrix}.$$

The final demand vector is given by:

$$\widehat{D} = \begin{bmatrix} [5000, 5200] \\ [91000, 92000] \\ [5200, 5500] \\ [1100, 1300] \\ [3400, 3600] \\ [5900, 6180] \end{bmatrix}.$$

The corresponding $\widehat{I} - \widehat{A}$ matrix was calculated to ensure the feasibility of the system.

5.2. Results obtained by the neural approach

Using our neural network trained on Monte Carlo data, with post-hoc calibration by dimension, we obtain the following interval vector:

$$\widehat{X}_{cal}^{NN} = \begin{bmatrix} [27920.83, 34535.00] \\ [76416.44, 87783.62] \\ [28527.79, 35684.97] \\ [25431.15, 31696.46] \\ [28652.21, 34578.26] \\ [26557.17, 33612.30] \end{bmatrix}.$$

These results comply with the economic constraints of the Leontief model ($\widehat{X}_i \geq 0$) and ensure 90% coverage after calibration.

5.3. Comparative analysis and economic interpretation

Table 6 presents a comparison between our approach and the main classical methods reported by [10].

The results obtained from our calibrated neural network approach show wider solution intervals than those produced by conventional methods (Gauss, Jacobi, Gauss-Seidel). At first glance, this observation may seem unfavourable, as a wider interval is often associated with lower accuracy. However, a thorough analysis reveals several factors that transform this apparent limitation into a methodological advantage. Conventional methods generate relatively narrow bounds, reflecting an 'optimistic' estimate of solutions. Nevertheless, they do not

Table 6. Comparison of interval solutions obtained by different methods [10].

Sector	Gauss	Gauss–Jordan	Jacobi	Seidel	NN (calibrated)
X_1	[28294,28555]	[27957,28896]	[28291,28558]	[28291,28558]	[27920,34535]
X_2	[228367,230179]	[226315,232261]	[228307,230246]	[228307,230246]	[76416,87783]
X_3	[5933,6106]	[5817,6223]	[5927,6112]	[5927,6112]	[28527,35685]
X_4	[9304,9507]	[9087,9727]	[9308,9504]	[9308,9504]	[25431,31696]
X_5	[24009,24345]	[23689,24670]	[24047,24308]	[24047,24308]	[28652,34578]
X_6	[19198,20256]	[17850,21631]	[18830,20631]	[18830,20631]	[26557,33612]

guarantee that the actual solution will always lie within these bounds. In practice, this can lead to underestimations of uncertainty and economically unrealistic solutions.
In contrast, our neural model, thanks to post-hoc calibration, achieves coverage in line with the set target ($\approx 90\%$). This property is essential because it ensures that the intervals generated do not underestimate the actual variability of the system. In contexts such as the Leontief model, where economic decisions are made on the basis of forecasts, having reliable bounds is more important than having artificially narrow intervals.
From an economic perspective, wider intervals can be interpreted as a decision-making safety strategy. In production planning, underestimating needs can lead to costly shortages or public policy errors. Conversely, cautious forecasts (even wide ones) reduce the risk of misallocation and provide room for manoeuvre to absorb uncertainty. The neural network approach is therefore a complementary tool to traditional methods.

6. Advantages and limits

Using neural networks for solving linear systems with interval coefficients has several notable advantages. On the one hand, the flexibility of machine learning makes it possible to process large matrices (e.g. n=10) with reduced computation time compared to traditional iterative methods. Furthermore, combining a hybrid loss function and post-hoc calibration by dimension offers a good compromise between centre accuracy and bound reliability, consistently achieving coverage close to 90% after adjustment. Another advantage is the model's ability to generalise to different types of Monte Carlo-generated data, which increases its robustness in the face of uncertainty.
However, some limitations remain. The main one concerns the widening of intervals: although calibration improves coverage, it sometimes leads to broader solutions than those obtained by conventional analytical methods, which can reduce economic accuracy. Furthermore, the methodology remains sensitive to the choice of hyperparameters and the distribution used in the Monte Carlo simulation. Finally, the approach relies on supervised training requiring a large volume of artificially generated data, which could limit its direct application to real economic data if such data is scarce.

7. Conclusion

In this work, we proposed an approach based on neural networks for solving linear systems with interval coefficients, with a particular application to the Leontief economic model. Unlike traditional methods (Gauss-Seidel, Jacobi, Hansen, etc.), our approach exploits the flexibility of neural networks to better manage uncertainty propagation and ensure more representative predictive intervals.
The numerical results obtained on systems of dimensions (n=2, 3, 5, 10) show that the proposed model achieves excellent accuracy while offering, after post-hoc calibration, coverage close to 90% of the actual solutions.
The integration of a stabilised hybrid loss function, combining MSE, Hausdorff, and under-coverage penalty, proved decisive in balancing accuracy and reliability. Furthermore, application to the Leontief model demonstrated the economic relevance of this approach, providing production intervals consistent with the model's constraints and

interpretable for sectoral policy analysis. However, some limitations remain, including the sometimes excessive widening of intervals compared to certain specialised methods, as well as the dependence on synthetic data for training. On the other hand, the development of neural networks specifically designed for interval calculations aims to reduce the width of the bounds while maintaining high coverage.

Ultimately, this research shows that neural networks are a promising and scalable alternative for processing linear interval systems in complex contexts such as economics and that they offer a new avenue of research for combining artificial intelligence and interval calculations.

REFERENCES

1. K. Ganesan, and P. Veeramani, *On arithmetic operations of interval numbers*, International Journal of Uncertainty, Fuzziness and Knowledge - Based Systems, vol. 13, No. 6, pp. 619–631, 2005.
2. N. Karkar, K. Benmohamed, and B. Arres *Solving Linear Systems Using Interval Arithmetic Approach*, International Journal of Science and Engineering Investigations, vol. 1, no. 1, pp. 29–33, 2012.
3. Wassily Leontief, *Input-output analysis, in Input-output economics*, Oxford university press, 2nd ed, pp 19–40, 1986.
4. M. Dehghani, and Madiseh, *Inner and outer estimations of the generalized solution sets and an application in economic*, Journal of Mathematical Modeling, vol. 8, no. 4, pp. 345–361, 2020.
5. S. P. Shary, *Interval Gauss-Seidel method for generalized solution sets to interval linear systems*, Reliable Computing, vol. 7, pp. 141–155, 2001.
6. R. E. Moore, R. B. Kearfott, and M. J. Cloud, *Introduction to interval analysis*, SIAM, vol. 110, pp. 85–103, 2009.
7. L. Jaulin, M. Kieffer, O. Didrit, and E. Walter, *Interval analysis*, Applied interval analysis, Springer, London, pp. 25–27, 2001.
8. M. A. Benhari , and M. Kaicer, *Resolution of linear systems using interval arithmetic and Cholesky decomposition*, Mathematics and Statistics, Vol. 11, No. 5, pp. 840–844, 2023.
9. M. A. Benhari, and M. Kaicer, *Using Interval Arithmetic on the Leontief Model*, I(eds) Advanced Intelligent Systems for Sustainable Development (AI2SD'2020), vol. 1, Springer, pp. 1079–1085, 2022.
10. L. Dymova, P. Sevastjanov and M. Pilarek, *A method for solving systems of linear interval equations applied to the Leontief input–output model of economics*, Expert Systems with Applications, vol. 40, Issue 1, pp. 222–230, 2013.
11. Li, Q-X and Liu, *The foundation of the grey matrix and the grey input– output analysis*, Applied Mathematical Modelling, vol. 32, Issue 1, pp. 267–291, 2008.
12. Mohamed Amine, B and Mohammed, K, *Analysis of uncertainty in the Leontief model by interval arithmetic*, Statistics, Optimization & Information Computing, vol. 13, Issue 5, pp. 2011–2026, 2025.