XLVI CILAMCE
VITÓRIA - ES - BRAZIL - 2025
XLVI IBERO-LATIN AMERICAN CONGRESS ON COMPUTATIONAL METHODS IN ENGENEERING

Reducing Computational Costs in Phase-Field Fracture Simulations Using Machine Learning Techniques

Adalberto Horácio de Oliveira Mendes[1], Hugo Mouro Leão[1], Saulo Silvestre de Castro[1] , Roque Luiz da Silva Pitangueira[1]

[1]*Dept. of Structural Engineering, Federal University of Minas Gerais*
Av. Antonio Carlos, 6627, Pampulha, Belo Horizonte, 31270-901, Minas Gerais, Brazil
adalbertohom@gmail.com, hugomleao@yahoo.com.br, saullo9@yahoo.com, roque@dees.ufmg.br

Abstract. The study of crack propagation is critical in Structural Engineering to prevent material failure and structural collapse. Among the various approaches available, phase-field modeling stands out by transforming the sharp crack into a smooth transition between intact and damaged material. This eliminates the need for explicit tracking of the crack surface. In this method, an additional field variable is introduced to quantify the damage level at each material point, with values ranging from zero for intact material to one for fully damaged material. Phase-field models are particularly suitable for modeling heterogeneous materials, such as concrete or composite structures, due to their ability to naturally capture complex crack patterns, including branching and merging, without the need for predefined crack paths. Despite these advantages, phase-field modeling is computationally expensive and limits its applicability in large-scale simulations or in real-time scenarios. To address this limitation, the present work proposes replacing the incremental iterative procedure used for computing the phase-field variable with a machine learning model, which has been previously trained to directly predict the phase-field. Machine learning offers a promising alternative by learning complex mappings from input fields (such as stress or strain) to damage evolution patterns, potentially bypassing the need for iterative numerical solutions. This integration of traditional numerical methods with artificial intelligence aims to reduce computational time while maintaining accuracy, enabling more efficient simulations of fracture in heterogeneous materials. All simulations will be conducted using the open-source INSANE software.

Keywords: Phase field, Machine Learning, Constitutive Model, Optimization

1 Introduction

The simulation of fracture in quasi-brittle materials, such as concrete, plays a fundamental role in predicting structural behavior under various loading conditions. The combination of the heterogeneous nature of these materials with the complex mechanisms involved in fracture, such as crack nucleation, branching and coalescence, makes modeling this phenomenon particularly challenging. In this context, computational models, especially those based on the Phase-Field formulation, have emerged as robust tools.

The Phase-Field approach regularizes discontinuities through a continuous scalar variable, ϕ, which enables modeling the transition between intact and fully fractured material. This formulation allows the capture of nonlinear behaviors associated with fracture, including in complex geometries or under multiaxial loading conditions [1]. However, one of the main limitations of such models lies in their high computational cost, as they require refined meshes and the coupled solution of nonlinear systems, often through iterative schemes.

In recent years, several studies have investigated the use of Artificial Neural Networks (ANNs) and other Machine Learning (ML) techniques as surrogate models to accelerate simulations in computational mechanics. Among these contributions, the work of [2] stands out, in which a Neural Network-based Constitutive Model (NNCM) was developed to represent the biaxial behavior of concrete slabs. Using a Multilayer Perceptron (MLP) trained on stress-strain curves obtained experimentally, the author demonstrated the potential of neural networks to capture complex constitutive relationships, even in the absence of explicit physical parameters in the model.

The present work proposes a direct extension of these approaches by implementing a Multilayer Perceptron model directly integrated into the INSANE system, replacing the iterative solution of the phase-field equation with a direct prediction of the ϕ variable.

2 Phase-field models

Phase-field models are based on two primary functions: the crack surface density function $\gamma(\phi, \nabla\phi)$, which regularizes the discontinuity by diffusing the crack over a finite region, and the energetic degradation function $g(\phi)$, which modulates the strain energy. Wu [3] proposed a generalization of $\gamma(\phi, \nabla\phi)$ by introducing a geometrical crack function $\alpha(\phi)$, a monotonically increasing function satisfying $\alpha(0) = 0$ and $\alpha(1) = 1$. The degradation function $g(\phi)$, in turn, governs the reduction of the strain energy density $\psi(\varepsilon)$, by splitting it into tensile and compressive components:

$$\psi(\varepsilon) = g(\phi)\,\psi_0^+(\varepsilon) + \psi_0^-(\varepsilon) \tag{1}$$

This decomposition is typical of anisotropic formulations. Consequently, the total energy functional governing the phase-field fracture problem is expressed as:

$$E_t = \int_\Omega \psi(\varepsilon(u), \phi)\,\mathrm{d}V + \int_\mathcal{B} G_c\,\gamma(\phi, \nabla\phi)\,\mathrm{d}V - \int_\Omega \mathbf{b}\cdot\mathbf{u}\,\mathrm{d}V - \int_{\partial\Omega} \mathbf{t}\cdot\mathbf{u}\,\mathrm{d}A \tag{2}$$

where Ω is the body domain, $\mathcal{B}$ is the degraded domain, G_c is the critical energy release rate, $\mathbf{u}$ is the displacement vector, and $\mathbf{b}$ and $\mathbf{t}$ are the body and surface forces, respectively.

In the INSANE system, the formulation of the phase-field fracture model is discretized using the Finite Element Method (FEM), with both the displacement field, the strain field, and the phase-field considered as nodally interpolated variables. The implementation supports both monolithic and staggered solvers, the latter being adopted in this work.In the staggered formulation, the displacement field ($\mathbf{u}$) is calculated using the phase-field (ϕ) from the previous iteration, and then the updated displacements are used to calculate the phase-field. The phase field, responsible for representing material degradation, is obtained by minimizing a diffuse regularized energy functional, and its evolution equation is derived from the variation of this functional. This approach, as well as its implementation in the INSANE system, is discussed in detail by Leão [4].

3 Machine Learning models

Machine Learning is a branch of Artificial Intelligence (AI) dedicated to the development of algorithms capable of learning patterns from data without the need for explicit programming. Initially defined by Samuel (1959), ML has proven particularly effective in addressing complex problems where traditional methods exhibit limitations [5–8].

This study focuses on supervised learning algorithms, which utilize labeled data to train models capable of mapping inputs to outputs. By generalizing the relationships learned during training, such models are capable of making accurate predictions even for previously unseen data.

Among the most widely used families of models in supervised learning are Artificial Neural Networks. Inspired by the structure of the human brain, ANNs are composed of interconnected processing units known as neurons. Each artificial neuron receives multiple inputs, computes a weighted sum, and applies a nonlinear activation function to produce an output. These outputs are then propagated through the network layers, enabling complex transformations of the input data.

The fundamental architecture of an ANN is the Multilayer Perceptron (MLP), also referred to as a Feedforward Neural Network. This structure consists of an input layer, one or more hidden layers, and an output layer, with neurons organized in successive levels and no intra-layer connections. Each neuron in a layer is fully connected to all neurons in the next layer, forming a fully connected network. The connections are associated with synaptic weights, which store the knowledge acquired by the model and act as filters, determining the influence one neuron has on another [9].

Mathematically, the activation v_j of a neuron j is computed as:

$$v_j = \sum_{i=1}^{m} w_{ji} x_i + b_j \tag{3}$$

where x_i are the input values, w_{ji} are the weights associated with each input, and b_j is the bias term.

The output of the neuron, y_j, is given by:

$$y_j = \varphi(v_j) \tag{4}$$

where φ denotes the activation function.

Training an MLP typically involves the Backpropagation algorithm, introduced by Rumelhart et al. (1986), which combines gradient descent optimization with the chain rule of differential calculus to minimize a loss function. This process consists of a forward pass (output computation), followed by a backward pass (gradient computation and weight update). The objective is to iteratively reduce the prediction error over the training set.

Determining the ideal stopping point for training is critical to avoid overfitting, a phenomenon where the network becomes too tailored to the training data and performs poorly on new data. A common strategy to mitigate overfitting is k-fold cross-validation, which repeatedly partitions the data into training and testing subsets, enabling a more robust assessment of the model's generalization capability [9].

In this work, the MLP is employed as a surrogate constitutive model. The network was trained to emulate the behavior of a phase-field model by learning to map the relationship between the input features—specifically, the components of the strain state and the crack driving force—and the corresponding phase-field value (ϕ). The specific pre-trained model integrated into the INSANE framework is thoroughly described by the authors in [10].

4 Implementation

INSANE is an open-source framework developed in Java, structured according to object-oriented programming (OOP) principles. This modular organization facilitates the extension and specialization of functionalities through inheritance of existing classes. Within this context, the class `PhaseFieldStandardNewtonRaphsonStaggeredSolver`, originally implemented by Leão [4], served as the foundation for the development proposed in this work.

The `PhaseFieldStandardNewtonRaphsonStaggeredSolver` class executes a standard Newton-Raphson method with staggered iterations to solve the phase-field fracture problem. Its logic is structured around two main loops: one for the displacement field, and another for the phase-field variable, which are solved sequentially at each load step. Initially, a predictor is computed (displacement and residual force vectors), followed by an iterative procedure for the convergence of displacement, using the phase-field value from the previous iteration. Once the displacements converge, the phase-field variable is updated iteratively, now using the newly computed displacements.

Each of these stages performs a local convergence check, based on a relative error criterion:

$$\text{Error} = \frac{\|\delta \bar{X}\|}{\|\bar{X}\|} \tag{5}$$

where $\delta \bar{X}$ and $\bar{X}$ represent, for example, the increments and current values of displacements or forces, depending on the context. At the end of each step, a global convergence verification ensures that displacement and phase-field are within the specified tolerances, guaranteeing consistency between mechanical and fracture effects.

To enable the replacement of the traditional convergence procedure for the phase field with a machine learning based model, a new class named `PhaseFieldMLStandardNewtonRaphsonStaggeredSolver` was created. This class extends the original implementation, preserving its structure and control flow, but intercepts the convergence step of the phase-field and redirects the prediction of phase-field to a pre-trained Python model.

The communication between the numerical core of INSANE (in Java) and the ML model (in Python) is handled through the Java Embedded Python (JEP) library. This tool allows direct invocation of Python functions from the Java environment, maintaining performance and synchronization during the simulation process.

During the convergence procedure of the phase-field, INSANE extracts the required input data for prediction—such as the components of strain ε and the crack driving force Y. These inputs are organized and extrapolated to the mesh nodes, forming a tabular `DataFrame`. This table is then sent to the Python script, which responds with the predicted phase-field values. These values are immediately incorporated into the system, replacing the results that would otherwise be obtained through iterations in the internal Newton-Raphson-based solver.

Figures 1 and 2 present the activity diagrams for the phase-field convergence procedure. Figure 1 depicts the traditional Newton-Raphson-based iterative scheme, whereas Figure 2 illustrates the proposed approach, in which the iterative process is replaced by a direct prediction using a machine learning-based model.

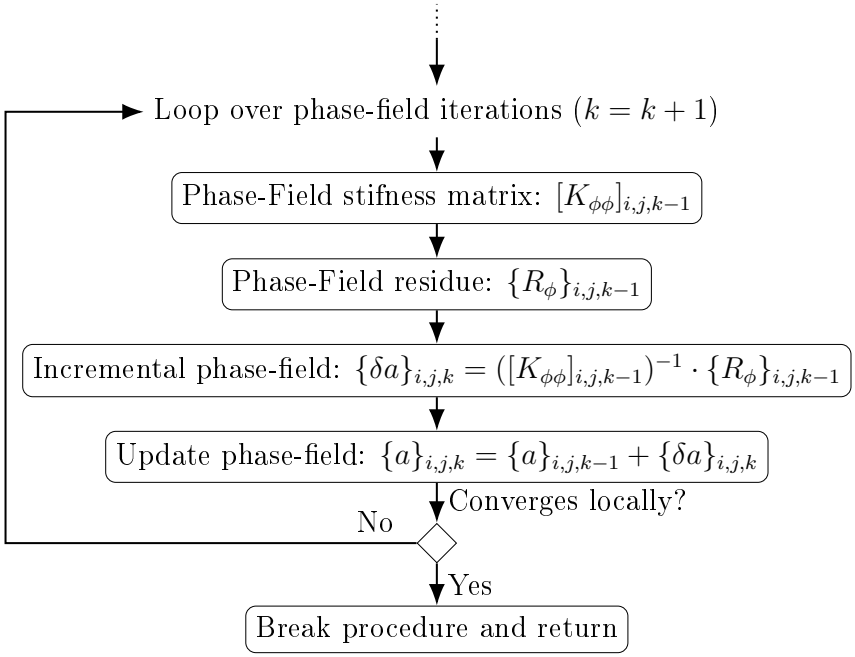


Figure 1. Activity diagram of the traditional Newton-Raphson-based phase-field convergence.

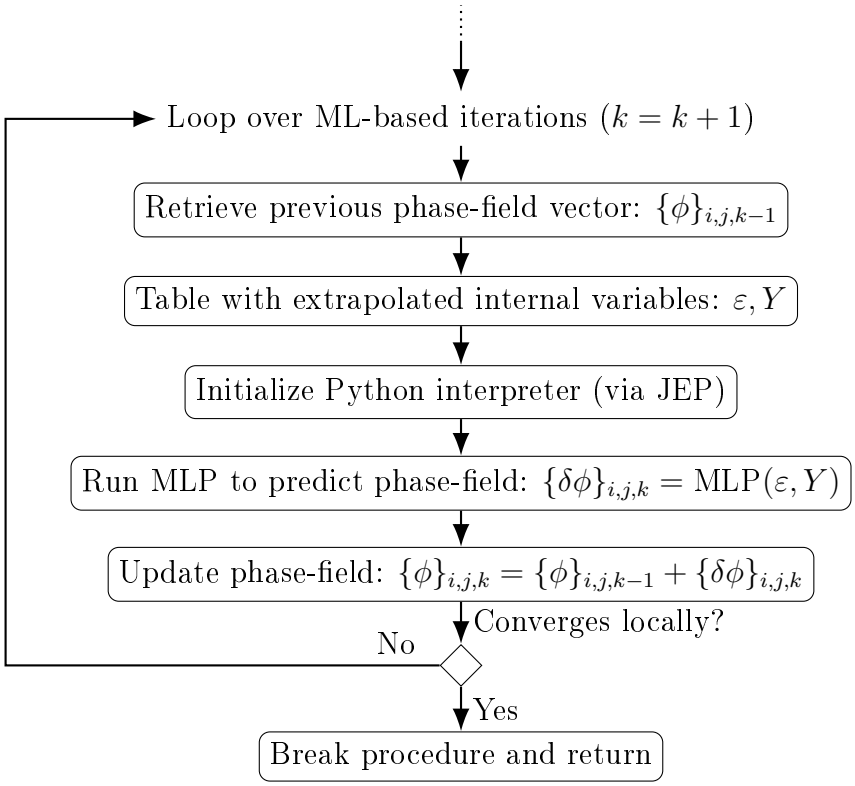


Figure 2. Activity diagram of the machine learning-based phase-field convergence.

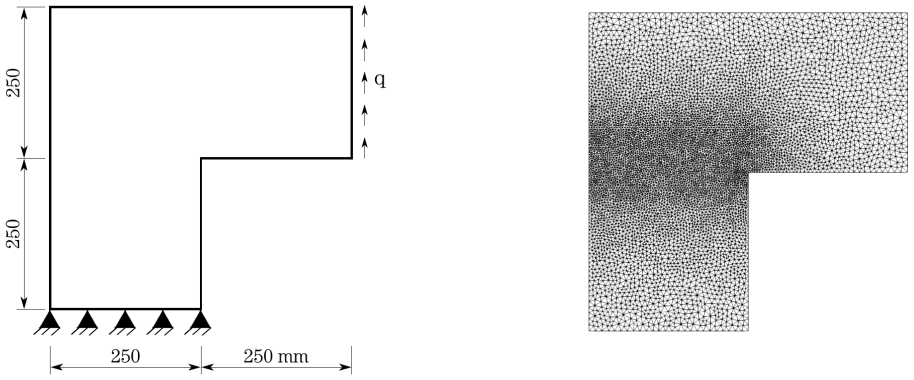
5 Implementation Validation

As discussed in the previous sections, phase-field models exhibit a strong capability to represent quasi-brittle media, albeit at considerable computational cost. To mitigate this limitation, a methodology was developed to construct a constitutive model grounded in phase-field theory and assisted by machine learning (ML) techniques. The approach consists of performing numerical simulations to assemble a database that correlates strain states with the corresponding phase-fields (damage), as well as with the effective crack driving force and its historical

maximum. Based on this database, an ML model is trained to predict the phase-field from the deformation state and associated thermodynamic quantities. The primary objective is to embed this predictor in the PhaseFieldML-StandardNewtonRaphsonStaggeredSolver (Section 4), thereby substantially reducing the computational cost of the iterative procedure required to determine the phase-field (see Section 2).

To ensure the feasibility of the proposal, the numerical models used to generate the training set must be physically realistic and duly validated in the literature.

As an initial study, we present the first simulation performed with the implementations discussed in Section 4. To this end, it was necessary to develop an ML model capable of predicting the phase-field associated with a given deformation state. In this context, the model's training dataset was obtained from a numerical simulation reproducing the tensile test of an L-shaped panel, as reported by Winkler et al. (2004), implemented in the INSANE system with the Miehe constitutive model. The problem setup—geometry, loading, and boundary conditions—is depicted in Figure 3. The material properties were: Young's modulus $E = 21876.0\,\text{kN/mm}^2$, Poisson's ratio $\nu = 0.18$, critical energy release rate $G_c = 1.8 \times 10^{-2}$, kN/mm, and length scale parameter $l_0 = 2.234\,\text{mm}$.

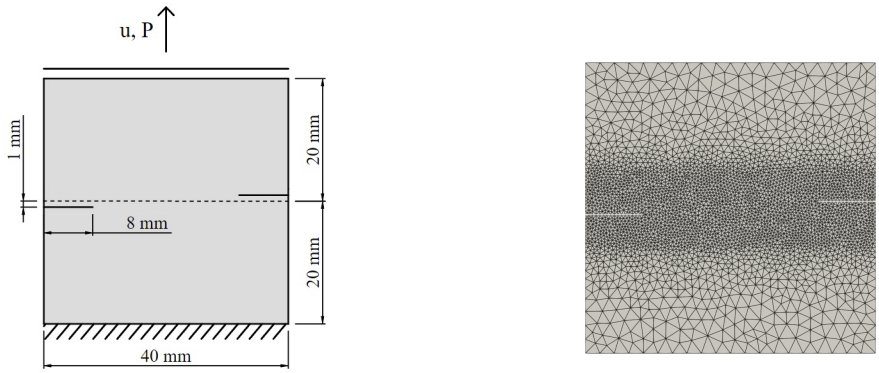


(a) (b)

Figure 3. (a) L-panel-Problem setting; (b)Mesh for phase-field model;

A multilayer perceptron (MLP) with three hidden layers containing 500, 1000, and 50 neurons, respectively, was developed as described in [10]. The ReLU activation function was used in the hidden layers, and the identity function was employed in the output layer. Training was performed for 50 epochs using the Adam optimizer with a learning rate of 0.05, employing the PyTorch library. The input variables comprise the three components of the strain state, the effective crack driving force, and the historical (maximum observed) driving force.

To validate the implementations and the ML model, a simulation was conducted reproducing the tensile test of a plate with asymmetric notches, as described by Molnar and Gravouil (2017). The simulation was run in INSANE with the Miehe model, incorporating the implementation described in Section 4. The configuration—geometry, loading, and boundary conditions—is presented in Figure 5. For the validation simulation, the same material model and parameter set used to generate the ML training dataset were employed.

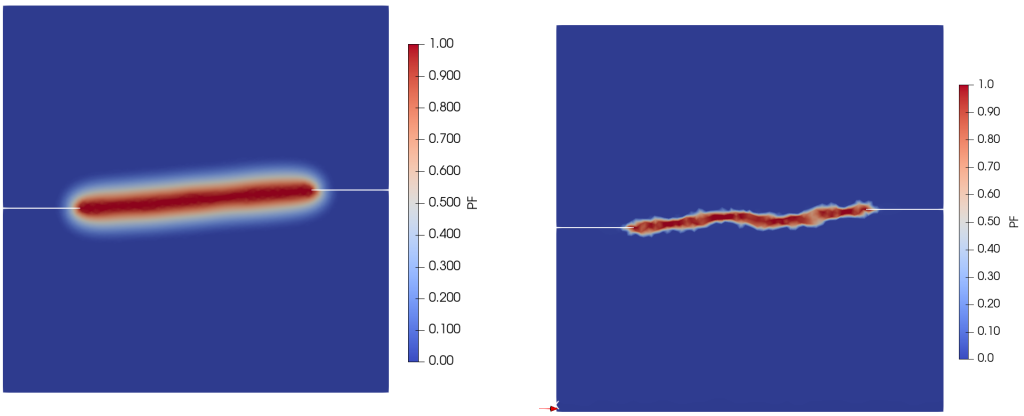


(a) (b)

Figure 4. (a) asymmetric double notched tensile test; (b) mesh asymmetric double notched tensile test;

In Figure xx, the final step of the equilibrium path for the validation example is shown, obtained both from the

FEM-based model and from the implementation proposed in Section 4. Both approaches yield a very similar crack pattern, indicating correct integration of the implementations into the INSANE system and the ability to represent the same level of material degradation.



(a) (b)

Figure 5. (Phase field result the last step - a)MEF .; (b) FEM assisted by an ML model ;

For each simulation run, we measured the time required to compute the phase field. In the conventional model—FEM with a constitutive model based on Miehe's analytical formulation—the average time per iteration was 306 ms. In the simulation employing the model proposed in Section 4, the average time per iteration was 406 ms.

Although this outcome runs counter to the expected cost reduction, the increase arises primarily from the nodal-variable extrapolation routine, which carries a substantial computational cost. Accordingly, the methodology must be refined and this stage optimized to achieve the anticipated gains. It is worth noting that the ML model's prediction, in isolation, is faster than the traditional iterative update; therefore, future efforts will focus on improving the implementations and reducing the coupling overhead.

6 Conclusions

The results presented constitute an initial study of the proposed formulation and therefore require an additional experimental campaign to corroborate its effectiveness. Although the implementation has proven compatible with the numerical core of the INSANE system, further refinements are needed to achieve the desired reduction in computational effort. Regarding the reliability of the methodology, it should be emphasized that it critically depends on the quality of the coupled ML predictor: the model must be physically consistent and representative of the material whose degradation process is to be described. To address this limitation, this research group has been conducting lines of investigation aimed at developing physically consistent ML models calibrated for different materials. In future publications, this research group will present advances in this direction, as well as their integration with the study reported here.

Acknowledgements. The authors gratefully acknowledge the support from the Brazilian research agencies CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior), FAPEMIG (Fundação de Amparo à Pesquisa do Estado de Minas Gerais; Grant PPM-00747-18), and CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico; Grant 309515/2017-3).

Authorship statement. This section is mandatory and should be positioned immediately before the References section. The text should be exactly as follows: The authors hereby confirm that they are the sole liable persons responsible for the authorship of this work, and that all material that has been herein included as part of the present paper is either the property (and authorship) of the authors, or has the permission of the owners to be included here.

References

[1] J. Y. Wu, V. P. Nguyen, C. T. Nguyen, D. Sutula, S. Sinaie, and S. Bordas. Phase-field modeling of fracture. *Advances in Applied Mechanics: Fracture Mechanics – Recent Developments and Trends*, vol. 53, pp. 1–183, 2020.
[2] A. F. Figueiredo. The challenges of concrete constitutive modeling via artificial neural networks. Master's thesis, Universidade Federal de Minas Gerais, 2023.
[3] J.-Y. Wu. A unified phase-field theory for the mechanics of damage and quasi-brittle failure. *Journal of the Mechanics and Physics of Solids*, vol. 103, pp. 72–99, 2017.
[4] H. M. Leão and others. A critical study on phase-field modelling of fracture. Master's thesis, 2021.
[5] S. Haykin. *Neural networks: a comprehensive foundation*. Prentice Hall PTR, 1994.
[6] T. M. Mitchell and others. Machine learning, 1997.
[7] M. Awad and R. Khanna. *Machine Learning*, pp. 1–18. Apress, Berkeley, CA, 2015.
[8] I. Goodfellow, Y. Bengio, and A. Courville. *Deep learning*. MIT press, 2016.
[9] S. Haykin. *Neural networks and learning machines, 3/E*. Pearson Education India, 2009.
[10] A. H. Oliveira Mendes, H. M. Leao, S. S. Castro, and R. L. Silva Pitangueira. Machine learning aided phase-field method in constitutive modeling of concrete structures. In *Ibero-Latin American Congress on Computational Methods in Engineering (CILAMCE)*, 2024.