Journal of Science Research and Reviews

PRINT ISSN: 1595-9074
E-ISSN: 1595-8329
DOI: https://doi.org/10.70882/josrar.2025.v2i5.188
Homepage: https://josrar.esrgngr.org

Review Article

Systematic Review of Learning Algorithms for Large-Scale Fuzzy Cognitive Maps

***Echobu, Faith O., Olanrewaju, Oyenike M. and Zaharaddeen, Sani**

Department of Computer Science, Federal University Dutsin-Ma, Katsina State, Nigeria.
*Corresponding Author's email: fadebiyi@fudutsinma.edu.ng

KEYWORDS
Fuzzy Cognitive Maps,
Fuzzy Cognitive Maps,
Large-Scale Maps,
Learning Algorithms.

CITATION
Echobu, F. O., Olanrewaju, O. M., & Zaharaddeen, S. (2025). Systematic Review of Learning Algorithms for Large-Scale Fuzzy Cognitive Maps. *Journal of Science Research and Reviews*, *2*(5), 175-183. https://doi.org/10.70882/josrar.2025.v2i5.188

ABSTRACT
Fuzzy Cognitive Maps (FCMs) are an important and versatile tool for modeling systems with complex dynamics in various domains like medicine, engineering, environmental monitoring, financial systems, among others to aid in decision-making. These domains usually datasets with large number of nodes and connections leading to large-scale maps. This paper reviews learning algorithms for FCMs under Hebbian, Population-based and hybrid classifications. While Hebbian-based algorithms suffer from local optima and generalization issues, population-based algorithms are generally computationally prohibitive from large-scale exploration, which also affects the global search component of hybrid algorithms. A number of algorithms have been developed specifically for large-scale FCMs by adopting problem decomposition, parallelization, sparsity inducement, and multi-agent-based techniques. Considering the great potential of FCMs as a modeling tool in domains with large, complex systems, research on algorithms tailored to large-scale systems remain limited and needs to be explored further.

INTRODUCTION

The increasing complexity of data-driven environments has driven significant interest in robust modeling methodologies capable of capturing the dynamic behavior of intricate systems. Fuzzy Cognitive Maps represent a prominent knowledge-based methodology introduced to model complex systems through causal reasoning. Rooted in the cognitive maps originally proposed by political scientist Robert Axelrod in the 1970s to represent social scientific knowledge, Fuzzy Cognitive Maps (FCMs) were formalized by Kosko in 1986 by integrating fuzzy logic to accommodate degrees of causality between hazy concepts. Since their inception, these FCMs have been successfully applied across diverse scientific domains. Applications range from medical diagnosis and radiotherapy treatment planning (Amirkhani et al., 2018; Farahani et al., 2025), to industrial process control (Salmeron & Papageorgiou, 2014), pattern classification (Nápoles et al., 2018), engineering (Mazzuto et al., 2023), agriculture (Kokhan et al., 2026) and socio-economic forecasting (Obiedat & Samarasinghe, 2022).

Despite their widespread applicability, traditional Fuzzy Cognitive Maps face critical challenges that limit their utility in large-scale deployments. The main challenge lies in their heavy reliance on human experts for model construction. Human experts are inherently subjective and possess cognitive limitations that restrict their ability to manually define relationships for systems involving dozens or hundreds of interacting concepts (Nápoles et al., 2026). As the number of concepts and relations increases, the ability of a human to maintain consistency decreases. When experts attempt to model highly complex systems, the resulting cognitive maps often suffer from inaccuracies, conflicting opinions, and an inability to reliably converge to desired steady states (Barisic, 2025; Sarmiento et al., 2024). Researchers have thus sought automated computational methods to generate these models directly from historical data. The purpose of this review is to systematically examine the learning algorithms developed to automate the calibration of Fuzzy Cognitive Maps, with a specific focus on advanced techniques designed to overcome the computational and structural challenges of large-scale networks.

Theoretical Foundations of Fuzzy Cognitive Maps

A Fuzzy Cognitive Map is a signed directed graph equipped with feedback mechanisms. The semantics behind a standard model can be defined by (C, W, A, f). The component $C = \{C_1, C_2, ..., C_N\}$ represents the family of N concepts (nodes) modeled after fuzzy sets, which depict the variables, states, or characteristics of the physical system under investigation (Felix et al., 2019). These nodes are interconnected by directed edges assigned with specific causal weights. The weight matrix W contains the causal weight w_{ij} assigned to each pair of concepts (C_i, C_j). The value of w_{ij} takes a value within the interval [-1, 1] denoting the strength and direction of the causal influence. A positive weight ($w_{ij} > 0$) indicates a promoting effect where an increase in the cause concept leads to an increase in the effect concept, whereas a negative weight ($w_{ij} < 0$) indicates an inhibiting effect. A weight of zero signifies the absence of a causal relationship between two given concepts (Dong et al., 2023).

The dynamic behavior of an FCM is governed by a recurrent mathematical inference process. During simulation, the system produces a state vector $A(t) = [A_1(t), A_2(t), ..., A_N(t)]$ at each discrete time step t which gathers the activation values of all concepts. The activation value of each concept $A_1(t + 1)$ is calculated by aggregating the weighted inputs from the previous time step (Papageorgiou, 2012).

The standard inference rule (Eqn. *1*), originally proposed by Kosko (1986), is mathematically expressed as:

$$A_i^{(t+1)} = f\left(\sum_{\substack{j=1 \\ i \neq j}}^{N} w_{ji} \cdot A_j^{(t)}\right) \tag{1}$$

However, a modified reasoning rule in Eqn. (*2*) by Stylios & Groumpos (2004) is frequently used to introduce memory into the system, thereby allowing concepts to take into account their own past activation value apart from the incoming influence from other concepts. This is represented by adding the previous state to the aggregated sum:

$$A_i^{(t+1)} = f\left(\sum_{\substack{j=1 \\ i \neq j}}^{N} w_{ji} \cdot A_j^{(t)} + A_i^{(t)}\right) \tag{2}$$

The aggregated sum of these weighted inputs is passed through a non-linear transformation function $f(\cdot)$ to squash the resulting value into a normalized range. The choice of activation function provides the model with its nonlinear capabilities and fundamentally affects convergence (Apostolopoulos et al., 2024; Nápoles et al., 2023). The most widely used transfer functions include:

1. Bivalent Function (Dickerson & Kosko, 1993; Tsadiras, 2008)

This is a binary function (Eqn. *3*) that restricts nodes to rigid active or inactive states. While it supports qualitative modelling it does not allow nuanced outputs.

$$f(x) = \begin{cases} 0, & x \leq 0 \\ 1, & x > 0 \end{cases}. \tag{3}$$

2. Sigmoid Function(Dickerson & Kosko, 1993; Tsadiras, 2008)

It is a continuous function that squashes the output into the range [0, 1] and allows for a smooth transition between states which enhances the interpretability of causuality. The sigmoid function (Eqn. 4) is preferred for its high predictive capacity.

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

Here, λ determines the steepness of the continuous function.

3. Hyperbolic Tangent Function (Stylios & Groumpos, 2000)

This function (Eqn. 5) maps values into the [-1, 1] interval making it suitable for bipolar concepts. It offers finer granularity allowing concepts to represent negative activation states.

$$f(x) = \tanh(x) = \frac{e^{\lambda x} - e^{-\lambda x}}{e^{\lambda x} + e^{-\lambda x}}. \tag{5}$$

As the network iterates through this reasoning process, the system will eventually exhibit specific dynamic behaviors. It may converge to a stable fixed-point attractor, settle into a periodic limit cycle, or diverge into a chaotic attractor where state vectors fluctuate unpredictably (Edalatpanah, 2024). Mathematical bounds on the weight matrix and the steepness parameter λ have been established to guarantee that the map acts as a contraction mapping, ensuring convergence to a unique fixed-point attractor. Stable convergence to a fixed point is generally required for reliable decision-making and pattern classification (Glykas, 2010).

FCM Development Strategies

The construction of a Fuzzy Cognitive Map entails defining the relevant concepts of a system and establishing the weighted causal relationships between them. Because systems vary drastically in data availability and complexity, researchers utilize three primary methodologies to develop these models: expert-driven, data-driven, and hybrid approaches.

Expert-driven approach

The expert-driven approach, also known as deductive modeling, relies entirely on human knowledge and domain expertise to construct the map. The development process typically follows three fundamental steps: identifying the key domain concepts, determining the causal relationships among these concepts, and estimating the strengths of these causal relationships (Glykas, 2010).

To gather this knowledge, researchers employ several elicitation techniques. These methods frequently involve open-ended probing interviews, collaborative group sessions, purposefully prepared questionnaires, or the documentary coding of written texts and historical archives. Because humans struggle to assign precise numerical values to causal strengths, experts usually describe the relationships using linguistic variables, such as "weak", "moderate", or "very strong". These linguistic terms are subsequently transformed into crisp numerical weights within the normalized interval of [-1, 1] through fuzzy logic defuzzification methods, such as the Center of Gravity approach (Malek, 2017).

To increase the reliability of the model and mitigate individual ignorance, researchers often aggregate the knowledge of multiple experts. Individual maps are mathematically combined, frequently using weighted averages based on each expert's credibility score, which helps to cancel out conflicting opinions and produce a consensus model (Sarmiento et al., 2024). However, models developed through this approach remain highly susceptible to human subjectivity and cognitive biases. Furthermore, because the number of possible interconnections grows quadratically with the number of concepts, human experts find it exceptionally difficult to manually construct and accurately weight large-scale systems (Stach et al., 2010b).

Data-driven approach (Inductive modeling)

Data-driven methodologies were developed to overcome the subjective limitations and scalability issues of manual mapping. These fully automated methods rely entirely on historical observations, such as multivariate time series data, to generate the weight matrices without any human intervention (Stach, 2010). The primary advantage of data-driven modeling is its independence from the application domain. By extracting knowledge directly from empirical data, these computational algorithms generate models that are free from human bias and assumptions (Stach et al., 2008). Data-driven approaches are broadly categorized into unsupervised Hebbian-based methods and population-based methods.

Hybrid approaches

Hybrid approaches combine the qualitative insights of expert knowledge with the quantitative power of data-driven learning to enhance overall model accuracy and robustness. This methodology effectively fuses subjective human expertise with objective historical data patterns, creating a cooperative modeling environment (Giabbanelli & Nápoles, 2024). In a typical hybrid framework, the model construction occurs in two complementary stages. Domain experts might first define the initial network topology, identify specific input concepts, and establish the foundational causal weights based on physical system constraints. Once this expert-derived framework is established, a machine learning algorithm utilizes available historical data to compute the remaining unknown weights or to fine-tune the overall network (Nápoles et al., 2020).

Learning algorithms for FCMs

Learning algorithms are fundamentally designed to compute a weight matrix *W* that best mimics available historical data or desired system behaviors. Based on the underlying learning paradigm, these methods are classified into three primary groups: Hebbian-based algorithms, population-based algorithms, and hybrid algorithms. These algorithms address FCM limitations like undesired steady states or suboptimal weights, enhancing robustness and accuracy. (Papageorgiou, 2012).

Classification of Learning Algorithms for Large-Scale FCMs

Hebbian-Based Learning Algorithms

Hebbian-based learning methods represent the earliest attempts to train fuzzy cognitive maps, drawing directly from artificial neural network principles (Felix et al., 2019). This learning paradigm translates the biological mechanism of synaptic plasticity into a framework for adjusting the strengths of causal relationships between nodes within a network. As unsupervised algorithms, they modify connection weights based on the simultaneous activation of connected concepts (Chen, 2015). When a concept node's activation consistently influences another, the weight of their connecting edge is strengthened proportionally. This process mirrors the neural reinforcement described in Hebbian theory but operates within the context of abstract systems. The weight adjustment rule (Eqn. *6*) is presented as:

$$w_{ij}^{t+1} = w_{ij}^{t} + \eta \cdot A_i^{(t)} \cdot A_j^{(t)} \tag{6}$$

Differential Hebbian Learning (DHL)

The foundational Differential Hebbian Learning (DHL) algorithm by Dickerson & Kosko (1993) updates weights by correlating the changes in concept activations. The DHL weight update (Eqn. *7*) is formalized as:

$$w_{ij}^{(t+1)} = \begin{cases} w_{ij}^{(t)} + \eta_t\left(\Delta A_i^{(t)} A_j^{(t)} - w_{ij}^{(t)}\right), & \Delta A_i^{(t)} \neq 0 \\ w_{ij}^{(t)}, & \Delta A_i^{(t)} = 0 \end{cases} \tag{7}$$

Where:

$w_{ij}^{(t+1)}$ is the updated weight between nodes i and j $t+1$

$\Delta A_i^{(t)} = A_i^{(t)} - A_i^{(t-1)}$ is the temporal difference of the activation for node *i*

$\eta_t = o.1\left[1 - \frac{t}{1.1N}\right]$ is the time-varying learning rate, with *N* as the total number of iterations

While fast, DHL suffers from poor generalization and primarily considers local node correlations rather than the overall system error (Felix et al., 2019).

Balanced Differential Algorithm (BDA)

Huerga (2002) introduced an improved variant building on the foundation DHL. While DHL only considered correlations between two local nodes, BDA is able to take into account all the concept values that change at the same time when the weights are updated. Although its applications have been limited, it is an improvement over its predecessor. Its update rule is formalized by Eqn (8):

$$w_{ij}^{(t+1)} = \begin{cases} w_{ij}^{(t)} + \eta_t \left[\frac{\Delta A_i^{(t)} \Delta A_j^{(t)}}{\sum_{k=1}^{m} \Delta A_i^{(t)} \Delta A_j^{(t)}} - w_{ij}^{(t)} \right], & \Delta A_i^{(t)} \Delta A_j^{(t)} > 0 \\ w_{ij}^{(t)} + \eta_t \left[\frac{-\Delta A_i^{(t)} \Delta A_j^{(t)}}{\sum_{k=1}^{m} \Delta A_i^{(t)} \Delta A_j^{(t)}} - w_{ij}^{(t)} \right], & \Delta A_i^{(t)} \Delta A_j^{(t)} < 0 \end{cases} \tag{8}$$

Active Hebbian Learning (AHL)

Papageorgiou et al. (2004) designed AHL to refine the causal weights of FCMs by iteratively updating them based on the activation levels of concepts. AHL updates all weights excluding self-connections unlike previous algorithms that only update non-zero weights. Domain experts select a set of active desired concepts as well as the sequence of activation concepts. This however poses a challenge if the initial map structure or data sequence is suboptimal as can lead to inconsistent results. The AHL update rule is seen in Eqn (9) where $\gamma^{(t)}$ is the weight decay term at iteration *t*, and $\breve{A}_j^{(t)}$ is the value of the *j*-th activation at iteration *t*.

$$w_{ij}^{(t+1)} = [1 - \gamma^{(t)}] \cdot w_{ij}^{(t)} + \eta^{(t)} \cdot A_i^{(t)} \cdot [A_j^{(t)} - w_{ij}^{(t)} \cdot \breve{A}_j^{(t)}] \tag{9}$$

Nonlinear Hebbian Learning (NHL)

NHL algorithm applies Oja's rule to constrain weight growth which addresses the limitation of the original Hebbian learning of unbounded weight growth. NHL depends on experts' initial determination of weights and concepts this initial structure is retained during the learning process, preserving its interpretability (Papageorgiou et al., 2003). A stopping criterion is defined based on constraints imposed on some or all of the concepts. During learning, weights are adjusted until the stopping criterion is satisfied. So far, it has been the most widely adapted and applied variant of Hebbian learning algorithms. The weight update formula (Eqn. 10) for NHL is given by:

$$w_{ji}^{(t)} = w_{ji}^{(t-1)} + \eta_t A_j (A_i^{(t)} - A_j w_{ji}^{(t-1)}) \tag{10}$$

$w_{ji}^{(k)}$ is the updated weight from node *j* to node *i* at iteration *k;* $w_{ji}^{(k-1)}$ is the weight from node *j* to node *i* from the previous iteration; η_k is the learning rate at iteration *k* which controls the magnitude of weight updates; $A_i^{(k)}$ is the activation of the target node *i* at iteration *k;* A_j is the activation of the source node *j*

Data-Driven Nonlinear Hebbian Learning (DD-NHL)

DD-NHL was developed as an improvement of NHL to replace expert-defined initial maps with random matrices relying entirely on historical sequence data (Stach et al., 2008). Although the concept of using historical data makes it superior, Papakostas et al. (2012) discovered that it performed poorly in comparison with other Hebbian-based algorithms for classification problems.

Population-Based Learning Algorithms

Population-based algorithms, also known as evolutionary methods, treat the learning process of FCMs as a global optimization problem. Real-Coded Genetic Algorithms (RCGA), Particle Swarm Optimization (PSO), and Ant Colony Optimization (ACO) are widely used examples. These algorithms eliminate the need for expert input by searching the solution space to minimize a specified cost function (or maximize a fitness function) that measures the error between simulated model outputs and historical training data (Felix et al., 2019).

The optimization objective is typically formulated using a simulation error metric, such as the Mean Squared Error (MSE) or Mean Absolute Error (MAE). For a dataset with *K* response sequences containing *T* time points, the L1-norm error function is often defined as:

$$E = \frac{1}{(K-1)N} \sum_{k=1}^{K} \sum_{t=1}^{T-1} \sum_{i=1}^{N} |A_i^{(t)} - \hat{A}_i^{(t)}| \tag{11}$$

where $A_i^{(t)}$ is the actual historical state and $\hat{A}_i^{(t)}$ is the simulated response generated by the candidate weight matrix. The fitness function is then constructed to maximize performance, commonly written as

$$Fitness = \frac{1}{\alpha \cdot E + 1} \tag{12}$$

where α is a normalization constant (Poczęta et al., 2015). Evolutionary methods excel at producing highly accurate models that simulate complex dynamics. However, because the number of parameters to optimize (N^2) grows quadratically with the number of concepts, these standard algorithms suffer from extreme computational costs and scale poorly to larger networks 66. They also tend to generate overly dense networks that lack semantic interpretability (Nápoles et al., 2020; Wu & Liu, 2022).

Stach et al. (2005) developed a variation of genetic algorithms where chromosomes are represented as vectors of real numbers rather than binary strings. Its length is determined by the number of variables in a given problem and it performs linear transformation on each variable of the solution to decode it to the desired interval. Their method exploited RCGA to find the FCM structure capable of mimicking a given input historical data. Based on input data, it can use either one or multiple sets of concepts values over successive iterations.

Hybrid Learning Algorithms

Hybrid algorithms integrate the principles of Hebbian learning with population-based optimization to balance the strengths and weaknesses of both approaches. In these frameworks, a Hebbian-type method like NHL or DD-NHL is first applied to rapidly establish a near-optimal initial weight matrix. This matrix is subsequently supplied to a global optimizer, such as Differential Evolution (DE) or a Genetic Algorithm, for fine-tuning. This cooperative strategy improves convergence speed and overall accuracy compared to using evolutionary methods in isolation, though it still inherits computational bottlenecks when applied to massive networks without structural modifications (Felix et al., 2019).

A summary of the different categories of FCM learning algorithms is presented in Table 1.

Table 1: Summary of FCM learning algorithms

Algorithm Category	Optimisation Mechanism	Mathematical Objective	Key Strength	Primary Limitations
Hebbian-based (DHL, BDA, AHL, NHL, DD-NHL)	Local weight updates driven by correlation of concept activations	Updates proportional to products of activations	Fast execution; preserves causal interpretability and expert constraints.	Prone to local optima; struggles with generalization and massive state spaces
Population-based (PSO, ACO, RCGA)	Global search heuristics driven by error minimization over populations	Minimization of distance between simulated states and raw data	Fully automated; highly accurate modeling of historical dynamics	Computationally prohibitive for large maps; produces overly dense matrices
Hybrid	Sequential application of Hebbian initialization followed by global tuning	Two-stage optimization minimizing local activation variance then global error	High accuracy; avoids random initialization pitfalls	Inherits scalability issues from the global search component

Advanced Techniques for Learning Large-Scale FCMs

As the dimensionality of a target system increases, standard evolutionary algorithms experience an exponential increase in computational complexity, rendering them incapable of efficiently learning maps with hundreds of concepts (Wu & Liu, 2022). Researchers have developed several advanced frameworks specifically tailored to mitigate these scalability barriers. These techniques can be categorized based on problem decomposition and parallelization, structured sparsity learning, and multi-agent systems.

Problem Decomposition and Parallelization Techniques

One of the most effective scalability strategies is problem decomposition. The foundational concept behind problem decomposition is that optimizing an entire dense weight matrix simultaneously is computationally inefficient. Instead of optimizing the entire $N \times N$ weight matrix simultaneously, algorithms separate the learning task into independent sub-problems (Liu et al., 2017). Because the activation value of a specific concept C_i depends exclusively on the incoming weights from other concepts, the overall optimization can be decomposed into N distinct tasks. Each sub-problem focuses on learning solely the $N \times 1$ incoming weight vector W_i for a single node (Wu & Liu, 2016). One approach to decomposition involves dividing the input data. Stach et al. (2010a) developed a divide and conquer strategy that split available data into distinct subsets. Independent submodels are then learned from each data subset using RCGA, after which, the learned submodels are merged into a single FCM. To further accelerate execution, this process can be distributed across multiple processors using a master-slave parallelization architecture (Stach et al., 2007).

A more mathematically rigorous decomposition strategy involves separating the objective function based on the network structure. Because the activation value of a specific concept depends exclusively on the incoming weights from its connected neighbors, the learning of an N-node FCM can be fully decomposed into N independent optimization tasks. Each sub-problem focuses strictly on estimating a single column of the weight matrix. This reduces the number of variables optimized in a single process from N^2 to N, massively shrinking the search space.

By isolating the learning process, algorithms such as the decomposed Multi-Agent Genetic Algorithm (dMAGA) dramatically reduce search space complexity. Furthermore, because these N sub-problems are mathematically independent, they can be distributed across multiple processors in a parallel computing architecture. This parallelization yields near-linear speedups, establishing a reliable paradigm for modeling massive networks (Stach, 2010).

Structured Sparsity Learning through Convex Optimization

Real-world complex systems generally exhibit sparse connections, averaging around 40% density. On the other hand, traditional learning algorithms artificially generate overly dense weight matrices to minimize data fitting errors, severely degrading model interpretability. Advanced techniques address this by reformulating the learning process as a sparse signal reconstruction problem and integrating convex optimization methods like the Least Absolute Shrinkage and Selection Operator (Lasso) (Wu & Liu, 2017).

In frameworks like LASSOFCM, an L1-norm regularization penalty is added directly to the objective function. The mathematical optimization (Eqn. 13) for a specific node's incoming weight vector W_i is formulated as:

$$min_{W_i}\left(\frac{1}{2M}\|Y_i - \Phi W_i\|_2^2 + \lambda\|W_i\|_1\right) \quad (13)$$

Here, Φ represents the matrix of historical concept states, Y_i represents the inverse-transformed target states, $\|Y_i - \Phi W_i\|_2^2$ is the least-squares data fitting error, and $\|W_i\|_1$ is the sparsity-inducing L1 penalty. The regularization parameter λ controls the trade-off between accuracy and sparsity; λ increases, superfluous connections are forced to exactly zero (Wu & Liu, 2016). To further enhance robustness against noise, researchers have also introduced elastic penalties (combining L1 and L2 norms) and Total Variation (TV) regularizations. Because these advanced penalties introduce non-smoothness into the objective function, accelerated proximal gradient descent and Nesterov's smoothing techniques are utilized to solve the optimization efficiently (Ding & Luo, 2021; Petalas et al., 2009).

Challenges and Future Directions for Large-Scale FCM Learning

Despite recent methodological advancements, the application of computational learning to large-scale systems continues to face complex challenges.

A primary limitation is that current state-of-the-art frameworks operate as batch learners. Algorithms like RCGA or LASSOFCM require the entirety of the historical dataset to be loaded into memory simultaneously to compute gradients and fitness evaluations. In big data environments characterized by real-time streaming data, batch processing is computationally prohibitive and fails to adapt to sudden systemic shifts (Wu et al., 2021). Establishing true online learning algorithms represents a critical future direction. Recent proposals utilizing Follow-The-Regularized-Leader (FTRL) proximal algorithms demonstrate the potential of updating the weight matrix sequentially as individual data instances arrive, enabling real-time analysis of streaming environments.

Furthermore, modeling complex systems frequently involves higher orders of uncertainty that traditional fuzzy sets cannot capture. There is a clear roadmap toward integrating advanced mathematical frameworks into large-scale optimization. For instance, Intuitionistic Fuzzy Cognitive Maps (iFCMs) introduce hesitancy parameters to evaluate the credibility of data and expert rules 88, 89. Fuzzy Grey Cognitive Maps (FGCMs) replace crisp numeric points with interval-valued grey numbers to model environments characterized by highly incomplete data (Chen et al., 2021). Expanding scalable learning algorithms to naturally optimize these higher-order uncertainty parameters without exploding computational costs remains an active area of investigation.

CONCLUSION

Fuzzy Cognitive Maps are powerful cognitive modeling tools, yet their traditional reliance on human experts fundamentally restricts their application to small-scale problems. The evolution of inductive learning algorithms has provided a necessary bridge to automate model generation and eliminate subjective biases. While Hebbian methods offer execution speed and population-based methods offer high simulation accuracy, standard formulations lack the capacity to natively scale to large-scale datasets. The introduction of advanced optimization techniques has attempted to address these limitations. Problem decomposition reduces the exponential search space by breaking matrix learning into independent vectors. Structured sparsity regularization guarantees that the generated models remain interpretable and robust against interference, while parallel computing architectures ensure execution times remain viable. Moving forward, the research trajectory must prioritize the development of online, real-time learning algorithms and the integration of higher-order uncertainty theories. By addressing these remaining computational and theoretical challenges, automated learning frameworks will solidify Fuzzy Cognitive Maps as highly scalable, transparent, and reliable tools for complex system modeling and critical decision support.

REFERENCES

Amirkhani, A., Papageorgiou, E. I., Mosavi, M. R., & Mohammadi, K. (2018). A novel medical decision support system based on fuzzy cognitive maps enhanced by intuitive and learning capabilities for modeling uncertainty. *Applied Mathematics and Computation*, *337*, 562–582. https://doi.org/10.1016/j.amc.2018.05.032

Apostolopoulos, I. D., Papandrianos, N. I., Papathanasiou, N. D., & Papageorgiou, E. I. (2024). Fuzzy Cognitive Map Applications in Medicine over the Last Two Decades: A

Review Study. *Bioengineering*, *11*(2), 139. https://doi.org/10.3390/bioengineering11020139

Axelrod, R. (1976). *Structure of Decision: The Cognitive Maps of Political Elites* (R. Axelrod, Ed.). Princeton University Press.

Barisic, D. (2025). EXPLORING THE LANDSCAPE OF EXPERT SYSTEMS: A REVIEW. *International Journal of Management Trends: Key Concepts and Research*, *4*(1), 58–68. https://doi.org/10.58898/ijmt.v4i1.58-68

Chen, J., Gao, X., Rong, J., & Gao, X. (2021). The dynamic extensions of fuzzy grey cognitive maps. *IEEE Access*, *9*, 98665–98678. https://doi.org/10.1109/ACCESS.2021.3096058

Chen, Y. (2015). *Fuzzy Cognitive Maps: Learning Algorithms and Biomedical Applications*. http://rave.ohiolink.edu/etdc/view?acc_num=ucin1423581705

Dickerson, J. A., & Kosko, B. (1993). Virtual worlds as fuzzy cognitive maps. *1993 IEEE Annual Virtual Reality International Symposium*, 471–477. https://doi.org/10.1162/PRES.1994.3.2.173

Ding, F., & Luo, C. (2021). Structured sparsity learning for large-scale fuzzy cognitive maps. *Engineering Applications of Artificial Intelligence*, *105*, 104444. https://doi.org/10.1016/j.engappai.2021.104444

Dong, Q., Huo, D., & Wang, K. (2023). Risk measurement and application of the international carbon market in the era of global conflict: A data-driven study using FCM. *Journal of Environmental Management*, *342*(March). https://doi.org/10.1016/j.jenvman.2023.118251

Edalatpanah, S. A. (2024). *A Study on Fuzzy Cognitive Maps Using Fuzzy Inference System*. (October). https://doi.org/10.22105/bdcv.2024.482326.1212

Farahani, H., Kovač, N., Fardi, H., & Watson, P. C. (2025). Modelling Pain Perception Using Fuzzy Cognitive Maps. *Journal of Pain Research*, *18*(July), 5153–5174. https://doi.org/10.2147/JPR.S525200

Felix, G., Nápoles, G., Falcon, R., Froelich, W., Vanhoof, K., & Bello, R. (2019). A review on methods and software for fuzzy cognitive maps. *Artificial Intelligence Review*, *52*(3), 1707–1737. https://doi.org/10.1007/s10462-017-9575-1

Giabbanelli, P. J., & Nápoles, G. (2024). Fuzzy Cognitive Maps: Best Practices and Modern Methods. In P. J. Giabbanelli & G. Nápoles (Eds.), *Fuzzy Cognitive Maps*. Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-48963-1

Glykas, M. (2010). Fuzzy Cognitive Maps. In M. Glykas (Ed.), *Soft Computing* (Vol. 247, Numbers 3–4). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-03220-2

Huerga, A. V. (2002). A balanced differential learning algorithm in fuzzy cognitive maps. *Proc. 16th International Workshop on Qualitative Reasoning*.

Kokhan, S., Popov, M., Alpert, S., Andreiev, A., Drozdivskyi, O., Temna, Y., Sybirtseva, O., & Dorofey, Y. (2026). Fuzzy Cognitive Maps in Corn Yield Forecast. In *Studies in Systems, Decision and Control* (Vol. 627). https://doi.org/10.1007/978-3-032-03616-2_18

Kosko, B. (1986). Fuzzy cognitive maps. *International Journal of Man-Machine Studies*, *24*(1), 65–75. https://doi.org/10.1016/S0020-7373(86)80040-2

Liu, J., Chi, Y., Zhu, C., & Jin, Y. (2017). A time series driven decomposed evolutionary optimization approach for reconstructing large-scale gene regulatory networks based on fuzzy cognitive maps. *BMC Bioinformatics*, *18*(1). https://doi.org/10.1186/s12859-017-1657-1

Malek, Ž. (2017). Fuzzy-Logic Cognitive Mapping: Introduction and Overview of the Method. In S. Gray, M. Paolisso, R. Jordan, & S. Gray (Eds.), *Environmental Modeling with Stakeholders* (pp. 127–143). Springer International Publishing. https://doi.org/10.1007/978-3-319-25053-3_7

Mazzuto, G., Carbonari, S., Bevilacqua, M., & Ciarapica, F. E. (2023). A Multiphase Liquid-Gas Plant Modelling Using Fuzzy Cognitive Maps: An Application to an Actual Experimental Plant. *2023 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, 1143–1147. https://doi.org/10.1109/IEEM58616.2023.10406673

Nápoles, G., Jastrzebska, A., Grau, I., Salgueiro, Y., & Leon, M. (2026). A Review on Fuzzy Cognitive Mapping: Recent Advances and Algorithms. *Big Data and Cognitive Computing*, *10*(1). https://doi.org/10.3390/bdcc10010022

Nápoles, G., Jastrzębska, A., Mosquera, C., & Vanhoof, K. (2020). Deterministic learning of hybrid Fuzzy Cognitive Maps and network reduction approaches. *Neural Networks*, *124*, 258–268. https://doi.org/10.1016/j.neunet.2020.01.019

Nápoles, G., Leon Espinosa, M., Grau, I., Vanhoof, K., & Bello, R. (2018). Fuzzy cognitive maps based models for pattern classification: Advances and challenges. In *Studies in Fuzziness and Soft Computing* (Vol. 360, pp. 83–98). Springer Verlag. https://doi.org/10.1007/978-3-319-64286-4_5

Nápoles, G., Ranković, N., & Salgueiro, Y. (2023). On the interpretability of Fuzzy Cognitive Maps. *Knowledge-Based Systems*, *281*, 111078. https://doi.org/10.1016/j.knosys.2023.111078

Obiedat, M., & Samarasinghe, S. (2022). *Modelling Socio-ecological Systems: Implementation of an Advanced Fuzzy Cognitive Map Framework for Policy development for addressing complex real-life challenges*. https://doi.org/https://doi.org/10.48550/arXiv.2208.05103

Papageorgiou, E. I. (2012). Learning Algorithms for Fuzzy Cognitive Maps—A Review Study. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, *42*(2), 150–163. https://doi.org/10.1109/TSMCC.2011.2138694

Papageorgiou, E. I., Stylios, C. D., & Groumpos, P. P. (2004). Active Hebbian learning algorithm to train fuzzy cognitive maps. *International Journal of Approximate Reasoning*, *37*(3), 219–249. https://doi.org/10.1016/j.ijar.2004.01.001

Papageorgiou, E., Stylios, C., & Groumpos, P. (2003). Fuzzy Cognitive Map Learning Based on Nonlinear Hebbian Rule. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (Vol. 2903, pp. 256–268). Springer-Verlag. https://doi.org/10.1007/978-3-540-24581-0_22

Papakostas, G. A., Koulouriotis, D. E., Polydoros, A. S., & Tourassis, V. D. (2012). Towards Hebbian learning of Fuzzy Cognitive Maps in pattern classification problems. *Expert Systems with Applications*, *39*(12), 10620–10629. https://doi.org/10.1016/j.eswa.2012.02.148

Petalas, Y. G., Parsopoulos, K. E., & Vrahatis, M. N. (2009). Improving fuzzy cognitive maps learning through memetic particle swarm optimization. *Soft Computing*, *13*(1), 77–94. https://doi.org/10.1007/s00500-008-0311-2

Poczęta, K., Yastrebov, A., & Papageorgiou, E. I. (2015). Learning fuzzy cognitive maps using structure optimization genetic algorithm. *Proceedings of the 2015 Federated Conference on Computer Science and Information Systems, FedCSIS 2015*, *5*, 547–554. https://doi.org/10.15439/2015F296

Salmeron, J. L., & Papageorgiou, E. I. (2014). Fuzzy grey cognitive maps and nonlinear Hebbian learning in process control. *Applied Intelligence*, *41*(1), 223–234. https://doi.org/10.1007/s10489-013-0511-z

Sarmiento, I., Cockcroft, A., Dion, A., Belaid, L., Silver, H., Pizarro, K., Pimentel, J., Tratt, E., Skerritt, L., Ghadirian, M. Z., Gagnon-Dufresne, M. C., & Andersson, N. (2024). Fuzzy cognitive mapping in participatory research and decision making: a practice review. *Archives of Public Health*, *82*(1), 1–15. https://doi.org/10.1186/s13690-024-01303-7

Stach, W. (2010). *Learning and Aggregation of Fuzzy Cognitive Maps – an Evolutionary Approach*. 1–121.

Stach, W., Kurgan, L., & Pedrycz, W. (2008). Data-driven Nonlinear Hebbian Learning method for Fuzzy Cognitive Maps. *2008 IEEE International Conference on Fuzzy Systems (IEEE World Congress on Computational Intelligence)*, 1975–1981. https://doi.org/10.1109/FUZZY.2008.4630640

Stach, W., Kurgan, L., & Pedrycz, W. (2010a). A divide and conquer method for learning large fuzzy cognitive maps. *Fuzzy Sets and Systems*, *161*(19), 2515–2532. https://doi.org/10.1016/j.fss.2010.04.008

Stach, W., Kurgan, L., & Pedrycz, W. (2010b). *Expert-Based and Computational Methods for Developing Fuzzy Cognitive Maps* (pp. 23–41). https://doi.org/10.1007/978-3-642-03220-2_2

Stach, W., Kurgan, L., Pedrycz, W., & Reformat, M. (2005). Genetic learning of fuzzy cognitive maps. *Fuzzy Sets and Systems*, *153*(3), 371–401. https://doi.org/10.1016/j.fss.2005.01.009

Stylios, C. D., & Groumpos, P. P. (2000). FUZZY COGNITIVE MAP IN MODELING SUPERVISORY CONTROL SYSTEMS. *Journal of Intelligent & Fuzzy Systems*, *8*(2), 83–98.

Stylios, C. D., & Groumpos, P. P. (2004). Modeling complex systems using fuzzy cognitive maps. *IEEE Trans Syst Man Cybernetics - Part A: Syst Hum*, *34*. https://doi.org/10.1109/TSMCA.2003.818878

Tsadiras, A. K. (2008). Comparing the inference capabilities of binary, trivalent and sigmoid fuzzy cognitive maps. *Information Sciences*, *178*(20), 3880–3894. https://doi.org/10.1016/j.ins.2008.05.015

Wu, K., & Liu, J. (2016). Robust learning of large-scale fuzzy cognitive maps via the lasso from noisy time series. *Knowledge-Based Systems*, *113*, 23–38. https://doi.org/10.1016/j.knosys.2016.09.010

Wu, K., & Liu, J. (2017). Learning of sparse fuzzy cognitive maps using evolutionary algorithm with lasso initialization. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *10593 LNCS*, 385–396. https://doi.org/10.1007/978-3-319-68759-9_32

Wu, K., & Liu, J. (2022). Learning large-scale fuzzy cognitive maps under limited resources. *Engineering Applications of Artificial Intelligence*, *116*(July), 105376. https://doi.org/10.1016/j.engappai.2022.105376

Wu, K., Liu, J., Liu, P., & Shen, F. (2021). Online Fuzzy Cognitive Map Learning. *IEEE Transactions on Fuzzy Systems*, *29*(7), 1885–1898. https://doi.org/10.1109/TFUZZ.2020.2988845