**Journal of Algorithms and Computations**

journal homepage: http://jac.ut.ac.ir

# A Critical Examination of Distributed Relief-Based Ensemble Clustering Frameworks for Time-Series Analysis

A. Bahramlou[1†]

[1] Department of Electrical and Computer Engineering, Isfahan University of Technology, Isfahan, Iran

ABSTRACT

Ensemble clustering methods have garnered significant attention for their ability to improve robustness and accuracy in time-series analysis. This paper critically examines two recently proposed parallel and distributed ensemble clustering frameworks leveraging MapReduce and Relief-family feature weighting (including ReliefF, Multisurf*, Simba-Sc, I-Relief, and M-Relief). While these frameworks offer advancements in computational efficiency for large datasets, our analysis reveals significant limitations in their methodological positioning and evaluation validity. We contend that these approaches do not constitute deep learning models, as they fundamentally lack hierarchical representation learning, end-to-end differentiable optimization, and the learning of latent representations. Furthermore, the reported performance metrics, such as runtime, memory usage, clustering accuracy, diversity, and generic error measures, primarily reflect algorithmic efficiency under controlled conditions rather than genuine predictive reliability in complex, real-world scenarios. We highlight the absence of robust uncertainty quantification, specifically aleatoric and epistemic uncertainty, which is crucial for reliable decision-making.

*Keywords:* Ensemble Clustering, Time-Series Analysis, Relief Feature Weighting, Distributed Computing, MapReduce, Deep Learning, Uncertainty Quantification, Fuzzy Logic, Neural Data Analysis.

ARTICLE INFO

*Article history:*
Research paper
Received 18, May 2026
Accepted 11, June 2026
Available online 27, June 2026

AMS subject classification: 90C06

† Corresponding author: A. Bahramlou, Email: a.bahramlou@ec.iut.ac.ir.

# 1 Introduction

The analysis of time-series data is a cornerstone of numerous scientific and engineering disciplines, ranging from financial forecasting and environmental monitoring to biomedical signal processing. Ensemble methods, which combine multiple base learners to achieve improved performance, have proven effective in enhancing the robustness and accuracy of clustering algorithms for time-series data [1, 2]. Recent advancements have focused on developing parallel and distributed frameworks to handle the ever-increasing volume and velocity of time-series data [3, 4].

A noteworthy contribution in this area is the proposal of two parallel and distributed ensemble clustering frameworks that integrate MapReduce with Relief-family feature weighting algorithms, including ReliefF, Multisurf*, Simba-Sc, I-Relief, and M-Relief [5].

These frameworks aim to enhance clustering performance by efficiently selecting relevant features in a distributed manner, thus mitigating the curse of dimensionality often encountered in time-series analysis. While the computational efficiency and scalability offered by these distributed approaches are commendable, a deeper critical examination is warranted concerning their methodological classification, the validity of their evaluation metrics, and their capacity for uncertainty quantification.

This paper offers a critical commentary on these distributed Relief-based ensemble clustering frameworks. We argue that these methods, despite their sophisticated distributed architecture, do not align with the core principles of deep learning. Furthermore, we scrutinize the typical evaluation metrics employed, asserting that they often fall short of demonstrating real-world predictive reliability.

Crucially, we identify a significant deficit in the modeling of uncertainty, a factor that is indispensable for trustworthy machine learning applications, particularly in sensitive domains like neural data analysis. Finally, we propose the integration of fuzzy logic as a promising avenue for incorporating more comprehensive uncertainty modeling.

# 2 Overview of the Framework

The criticized frameworks operate on the principle of parallel and distributed processing using the MapReduce paradigm [3]. At their core, these systems leverage algorithms from the Relief family of feature weighting methods to identify features that are discriminative for time-series clustering.

The Relief family, including variations such as ReliefF [6], Multisurf* [7], Simba-Sc [8], I-Relief [9], and M-Relief [10], works by assessing the relevance of features based on their ability to distinguish between instances of the same class (or, in this context, similar time-series segments) and instances of different classes (dissimilar segments). In a distributed setting, these algorithms are adapted to operate on partitions of the data, with results aggregated to form a global feature weighting.

These distributed feature weightings are then used to guide ensemble clustering algorithms. The ensemble aspect arises from combining the outputs of multiple base clustering models, potentially trained on different subsets of data or with different feature subsets, to produce a more robust and stable final clustering.

The MapReduce model facilitates this by allowing feature weighting and base clustering computations to be performed in parallel across multiple nodes, with intermediate results being shuffled and reduced to produce the final ensemble output. This architecture is designed to efficiently process large-scale time-series datasets that would be intractable for single-machine implementations.

# 3 Conceptual Limitations

## 3.1 Deep Learning Classification

A primary critique of these frameworks is their misclassification as "deep learning" models. Deep learning is characterized by several fundamental principles that are absent in the described Relief-based ensemble clustering approaches:

- Hierarchical Representation Learning: Deep neural networks excel at learning hierarchical representations of data, where lower layers capture simple features and higher layers combine these to form more complex, abstract representations [11]. This hierarchical structure is inherent to the network architecture and learned end-to-end. Relief-family algorithms, while effective for feature weighting, do not inherently learn such multi-layered, abstract representations of time-series data.

- End-to-End Differentiable Optimization: Deep learning models are trained using gradient-based optimization methods (e.g., backpropagation) applied to a differentiable objective function [12]. This allows for simultaneous learning of all model parameters and representations. Relief-family algorithms, and many traditional ensemble clustering methods, rely on distinct feature selection steps, often heuristic or iterative, which are not typically integrated into a single, differentiable optimization process.

- Latent Representation Learning: Deep learning models often discover and learn latent (hidden) representations of the input data. These latent spaces can capture underlying structures and variations in the data in a rich, continuous manner. While Relief algorithms aim to identify important features, they do not explicitly construct or learn a generalized latent space for clustering in the same way that deep generative models or autoencoders do.

Therefore, while the proposed frameworks are sophisticated in their distributed implementation and feature selection strategy, they do not embody the core tenets of deep learning. They represent advanced applications of traditional machine learning techniques in a distributed environment.

Deep learning models, while powerful, face several fundamental limitations that impact their deployment in critical applications. Below are the key constraints associated with deep neural network architectures (Figure 1):

1. Lack of Hierarchical Interpretability: Deep features learned by the network do not necessarily map to meaningful neural or cognitive representations. This "black-box" nature makes the internal reasoning process difficult to interpret or validate by domain experts.
2. End-to-End Black-Box Optimization: The model $f(x;\theta)$ is optimized as a holistic system. Consequently, the internal operations and transformations performed by the hidden layers remain opaque. We observe inputs and outputs, but the intermediate reasoning steps are obscured.
3. Generalization Gap: The model's performance on unseen data may significantly degrade due to distribution shifts. This occurs when the training data distribution differs from the real-world test data distribution, represented as (1):
$$P_{train}(x, y) \neq P_{test}(x, y) \quad (1)$$
4. Uncertainty Not Quantified: The model's output y^ is typically provided as a simple point estimate. Standard architectures often fail to explicitly model epistemic (model-based) and aleatoric (data-based) uncertainties, which are critical for reliability in high-stakes domains like medical diagnostics.

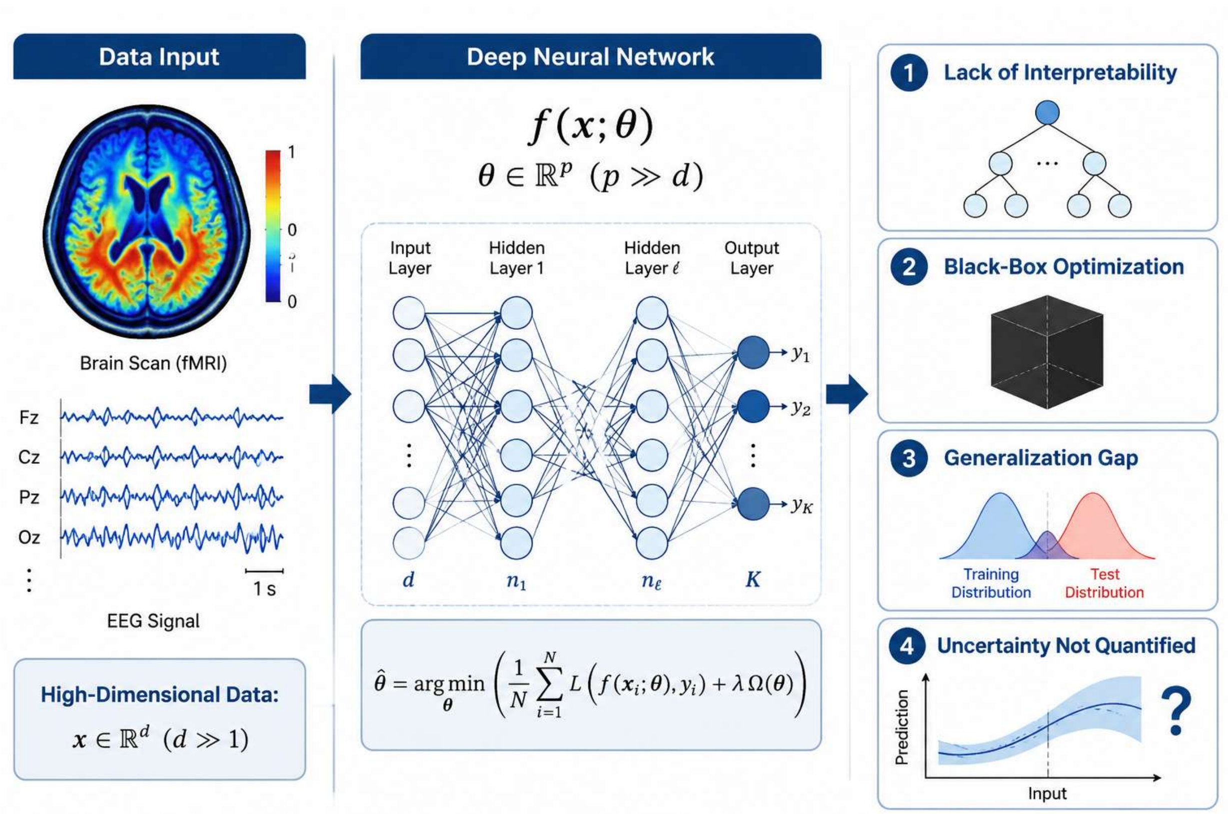



**Fig 1: Schematic overview of the limitations of deep learning classification in neuroimaging (EEG/fMRI).** The pipeline illustrates the transition from high-dimensional neural data to a deep neural network where the parameter space significantly exceeds the input dimensions. The optimization process follows an end-to-end empirical risk minimization objective with a regularization term. Key limitations are highlighted: (1) **Lack of Hierarchical Interpretability**, as deep features often lack direct neurophysiological meaning; (2) **Black-Box Nature**, where internal operations remain opaque; (3) **Generalization Gap**, caused by distribution shifts between training and test sets; and (4) **Unquantified Uncertainty**, as standard models provide point estimates without modeling epistemic or aleatoric uncertainties.

## 3.2 Evaluation Validity: Algorithmic Performance vs. Real-World Reliability

The evaluation metrics commonly reported for such frameworks—runtime, memory usage, clustering accuracy (e.g., silhouette score, Davies-Bouldin index), diversity, and generic error metrics—primarily assess the algorithmic performance and computational efficiency under controlled, often synthetic, datasets.

While these metrics are important for understanding the scalability and operational characteristics of the proposed systems, they do not adequately capture real-world predictive reliability. Consider the distinction between test error and real-world error. The expected test error, often estimated using a held-out test set from the same data distribution, is defined as (2):

$$E_{test} = E_{(x,y)\sim D_{test}} [L(f(x), y)] \qquad (2)$$

Where $D_{test}$ is the distribution of the test data, f(x) is the model's prediction for the input, and L is a loss function. This metric assumes that the test data distribution $D_{test}$ accurately reflects the real-world data distribution. However, in practice, the real-world error can differ significantly (3):

$$R_{real} = E_{(x,y)\sim D_{real}} [L(f(x), y)] \qquad (3)$$

Where $D_{real}$ is the true, often unknown, distribution of data encountered in real-world applications. The reported metrics often do not account for potential distribution shift, concept drift, or the inherent complexities and noise present in real-world data that are not captured by controlled benchmarks. High performance on synthetic datasets with clean labels and known feature distributions does not guarantee robust performance when applied to noisy, evolving, or domain-shifted real-world time-series data.

## 3.3 Absence of Uncertainty Quantification

A critical deficiency in many current machine learning frameworks, including the one discussed, is the lack of comprehensive uncertainty quantification. Uncertainty in machine learning can broadly be categorized into two types:

- Aleatoric Uncertainty: This refers to the inherent randomness or noise in the data itself. It is irreducible, even with infinite data. For instance, sensor noise in EEG recordings or natural variability in human brain activity contribute to aleatoric uncertainty.

- Epistemic Uncertainty: This arises from the model's lack of knowledge or uncertainty about the model parameters or the underlying data-generating process. It is reducible with more data or better models. For example, a model trained on limited data might exhibit high epistemic uncertainty in regions of the input space where it has not been adequately exposed.

Traditional clustering accuracy metrics do not explicitly quantify these forms of uncertainty [5] (Table 1).

This absence is problematic, as it provides a point estimate of performance without an indication of confidence. Fuzzy logic offers a powerful framework for modeling imprecise and uncertain information, which can be leveraged to enhance uncertainty quantification in clustering [13, 14] (Table 2). In fuzzy set theory, an element belongs to a set $A$ with a degree of membership, denoted by $\mu_A(x)$, which is a value in the interval [0, 1].

For a given data point and a set of $K$ clusters, the membership degrees can be represented as $\mu_{Ak}(x)$, where $k \in \{1, \ldots, K\}$. The most common fuzzy clustering algorithm, Fuzzy C-Means (FCM), explicitly computes these memberships [15] (Figure 2).

Table 1: Evaluation and clustering metrics [5].

| Metric | Full name | Definition / Meaning | Interpretation | Comparison / Notes |
|---|---|---|---|---|
| **Medoid AE (MAE)** | Medoid Average Error | $\frac{1}{n}\sum_{i=1}^{n} d(x_i, m)$ where $m$ is cluster medoid | average distance of points from medoid | robust to outliers, used in $k$-medoids |
| **RMSE** | Root Mean Square Error | $\sqrt{\frac{1}{n}\sum_{s=1}^{n}(X_s - \hat{X}_s)^2}$ | emphasizes large errors | squares deviations so large errors penalized |
| **ME** | Max Positive Error | $\max(X_s - \hat{X}_s)$ | largest positive deviation where prediction underestimates actual | shows worst positive bias |
| **Silhouette Coefficient** | cluster validity index | compares intra vs inter cluster similarity | higher is better | measures cohesion and separation |
| **Davies–Bouldin Index** | cluster validity index | ratio of within cluster scatter to between cluster separation | lower is better | evaluates compactness |
| **Calinski–Harabasz Index** | variance ratio criterion | between cluster dispersion / within cluster dispersion | higher is better | used for selecting cluster number |
| **Runtime** | execution time metric | computational time required | lower is better | important for scalability |
| **Memory Consumption** | resource utilization metric | memory used during processing | lower is better | relevant for distributed systems |

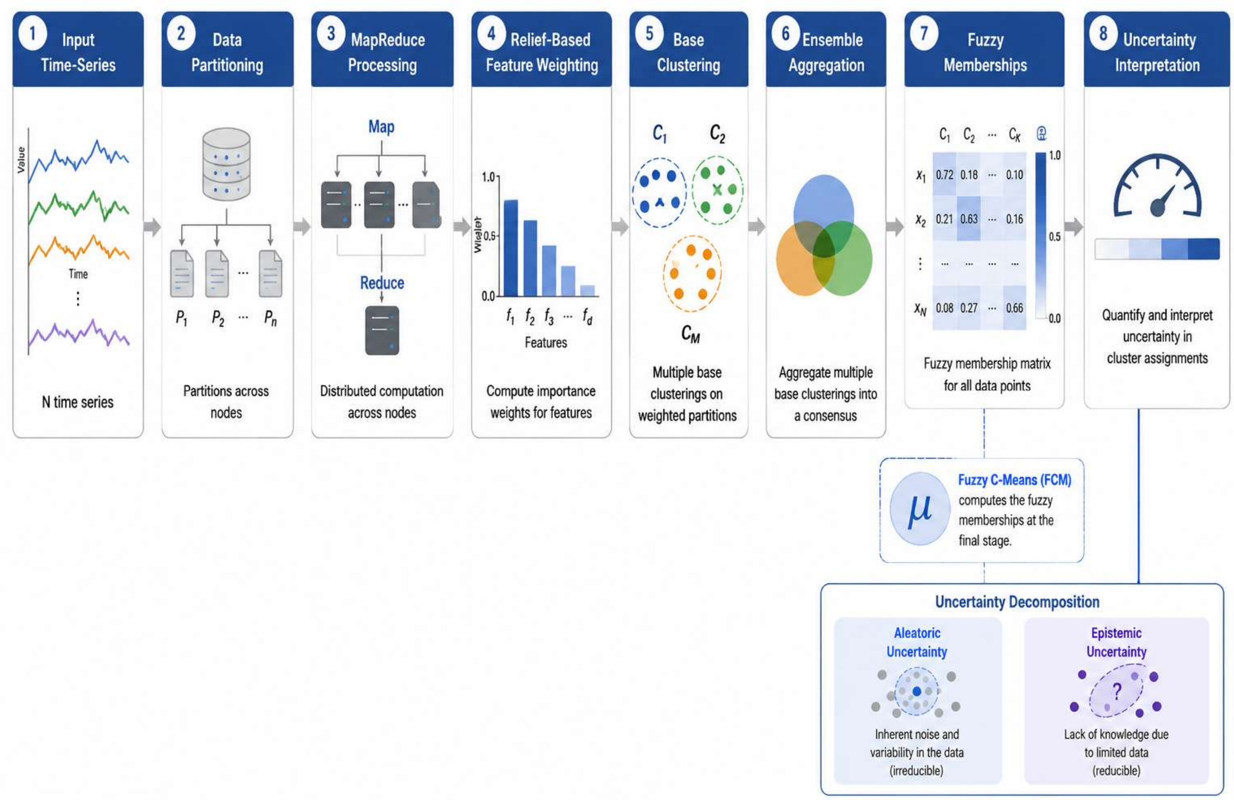
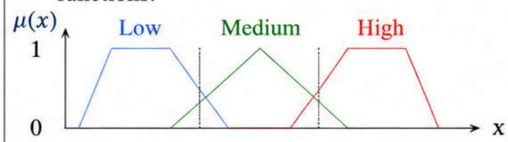



**Fig 2:** *Conceptual workflow of the proposed distributed clustering framework, illustrating data partitioning, MapReduce-based feature weighting, base clustering, ensemble aggregation, and fuzzy uncertainty interpretation through membership degrees.*

Table 2: Fuzzification of Error Metrics.

| Section | Description | Key Formula / Concept |
|---|---|---|
| **1. Concept of Fuzzification** | Fuzzification is the process of converting crisp error metric values into degrees of membership in linguistic categories (Low, Medium, High). Each metric value is mapped to membership values between 0 and 1, indicating the strength of belonging to each fuzzy set. | • Crisp input → Fuzzy memberships<br>• Linguistic terms: Low, Medium, High<br>• Membership values: $0 \le \mu \le 1$ |
| **2. Membership Functions** | Membership functions define how each error metric value maps to fuzzy sets. Triangular or trapezoidal functions are commonly used to represent the linguistic terms Low, Medium, High over the universe of discourse. | Triangular / Trapezoidal membership functions:<br> |
| **3. Fuzzification of Metrics** | Each error metric—Medoid AE, RMSE, and Max Positive Error—is fuzzified by evaluating its membership degrees in the fuzzy sets (Low, Medium, High) using the defined membership functions. | Metrics → Membership degrees (using membership functions)<br>$\mu_{Low}(x),\ \mu_{Medium}(x),\ \mu_{High}(x) \in [0, 1]$ |
| **4. Example Fuzzy Rules** | Fuzzy rules combine fuzzy inputs using logical operators (AND, OR, NOT) to infer the fuzzy output representing model quality. | Examples:<br>• IF RMSE is High OR ME is High THEN Model Quality is Poor<br>• IF Medoid AE is Low AND RMSE is Low THEN Model Quality is Good |
| **5. Defuzzification** | The aggregated fuzzy output is converted into a single crisp value using the centroid (center of gravity) method. | Centroid (Center of Gravity) method:<br>$\text{Output} = \dfrac{\int x\, \mu(x)\, dx}{\int \mu(x)\, dx}$ |

# 4 The Proposed Fuzzy Uncertainty Modeling

We propose integrating fuzzy logic to address the uncertainty gap. Fuzzy sets allow for degrees of membership ($\mu_A(x) \in [0, 1]$ ), enabling the calculation of uncertainty measures. We can define measures of uncertainty based on these fuzzy memberships:

* Fuzzy Membership Uncertainty ($U_x$ ): This quantifies how ambiguously a data point is assigned to the clusters (4). A high Ux indicates that it has a relatively even distribution of membership across multiple clusters, suggesting ambiguity in its assignment. If it strongly belongs to a single cluster: $\mu_{Aj}(x) \approx 1$.

$$U_x = 1 - max_k \mu_{Ak}(x) = 1 - \mu_{Ak}(x) \qquad (4)$$

* Output Assignment Uncertainty ($U_y$): This can represent the uncertainty in the definition or distinctiveness of a cluster. If cluster $k$ is well-defined and distinct, its membership function should be concentrated. If clusters overlap significantly or if their centroids are close, uncertainty increases. For a representative point $y$ of a cluster, its membership to that

cluster could be high, but its membership to other clusters might also be non-negligible. This measure can be extended to represent the uncertainty of the *resulting clustering solution*. where $\mu_k$ represents the degree to which the *entire cluster* is well-d.

If the overall clustering solution is a set of fuzzy memberships **y** = ($\mu_1$, …, $\mu_K$), the uncertainty in this solution can be viewed as (5):

$$U_y = 1 - max_k \mu_k = 1 - \mu_k(y) \qquad (5)$$

* Total Uncertainty ( $U_{total}$ ): A combined measure of uncertainty can be formulated as a weighted sum of uncertainty stemming from the input features and the uncertainty in the model's output or learned representation. If we consider $U_x$ as a measure related to input ambiguity or the ambiguity of assigning to any clusters, and $Uy$ as the uncertainty in the determined cluster labels, the total uncertainty could be:

$$U_{total} = \alpha U_x + \beta U_y \text{ , with } \alpha + \beta = 1 \qquad (6)$$

where $\alpha$ and $\beta$ are weighting coefficients (6). This fuzzy uncertainty modeling provides a more granular view of the reliability of clustering assignments. It allows for the identification of data points or cluster assignments that are inherently ambiguous, which is crucial for downstream decision-making. This provides a more nuanced view of confidence in clustering results.

## 4.1 Derivation of the Fuzzy Uncertainty Measure

To quantify the aggregate uncertainty inherent in the fuzzy inference process, we define the total uncertainty, $U_{total}$, as a function of the membership degree and the output assignment. Let $\mu_{Ak}(x) \in [0, 1]$ represent the membership degree of an input in a fuzzy set $Ak$. We define the membership uncertainty component, $U_x$, as the complement of the membership strength (7):

$$U = 1 - \mu_{Ak}(x) \qquad (7)$$

Similarly, for the output assignment uncertainty, $Uy$, which captures the ambiguity in the mapping to specific output classes, we define (8):

$$U = 1 - \mu(y) \qquad (8)$$

where $\mu(y)$ denotes the confidence score of the output assignment. The total uncertainty, $U_{total}$, is then derived as a convex combination of these two components, weighted by parameters and representing the relative contribution of input-level membership ambiguity and output-level assignment ambiguity, respectively (9):

$$U_{total} = \alpha U + \beta Uy \qquad (9)$$

Subject to the normalization constraint (10):

$$\alpha + \beta = 1, \alpha, \beta \geq 0 \quad (10)$$

This formulation provides a robust framework to balance between aleatoric and epistemic uncertainties, allowing the system to calibrate its confidence based on the fusion of input features and output distribution.

## 4.2 Discussion of Fuzzy Uncertainty

In the context of clustered time-series data, uncertainty can be represented through fuzzy membership values rather than hard assignments. A common decomposition includes (Figure 3):

- Fuzzy membership uncertainty: uncertainty arising from non-binary cluster memberships.
- Cluster uncertainty: uncertainty associated with the compactness and separation of clusters.
- Total uncertainty: an aggregate measure combining local assignment ambiguity and global partition stability.

Fuzzy logic provides a natural framework for representing imprecision and vagueness in noisy time series, especially where class boundaries are not crisp, and the data-generating process is nonstationary.

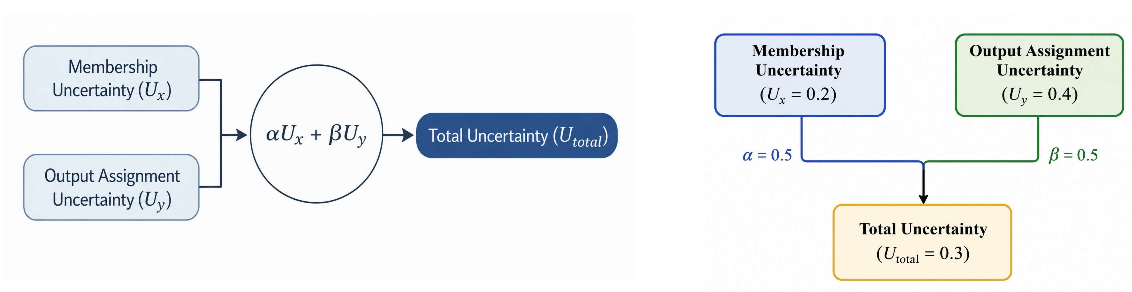



**Fig 3:** *Graphical illustration of the fuzzy uncertainty derivation, showing the aggregation of membership uncertainty and output assignment uncertainty into the total uncertainty measure. Numerical example of the uncertainty calculation process, demonstrating the aggregation of membership uncertainty ($U_x$=0.2) and output assignment uncertainty ($U_y$=0.4) into the total uncertainty score ($U_{total}$=0.3), assuming equal weighting (α=β=0.5).*

## 4.3. Proposed Fuzzy Uncertainty Modeling Framework

As illustrated in Figure 4, the proposed framework utilizes a fuzzy clustering- based approach to quantify uncertainty. In this methodology, input data are mapped into fuzzy clusters, where the degree of belongingness for each data point is represented by membership degrees, $\mu_{ik}$ . To derive a formal measure of uncertainty, we employ the normalized Shannon entropy. Specifically, the uncertainty inherent in the input data, is calculated based on the entropy of the membership distribution. Concurrently, the uncertainty associated with the output (or the cluster assignment) is derived from the entropy of the cluster label distribution. The total uncertainty, is subsequently obtained through a weighted linear combination of these components. This approach provides a robust mechanism for characterizing ambiguity in complex datasets, offering a clear interpretation of how uncertainty propagates from the feature space to the clustering outcome.

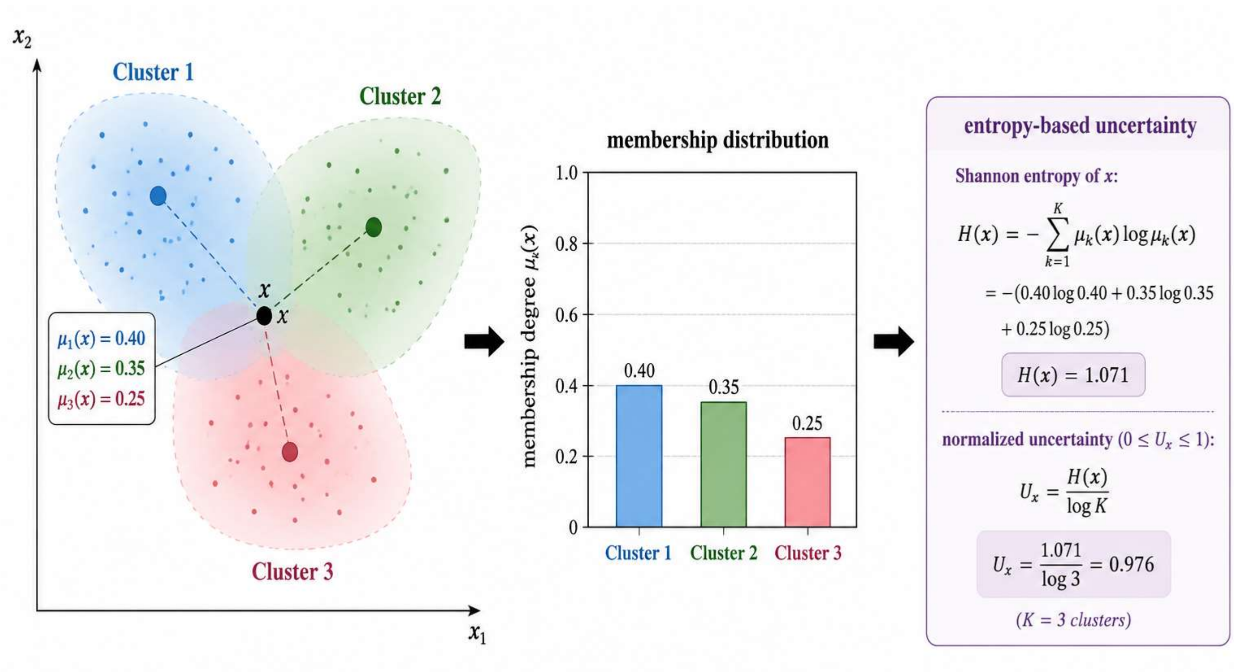



Fig 4: Schematic representation of the proposed fuzzy uncertainty modeling framework. The input data are mapped to fuzzy clusters with membership degrees $\mu_{ik}$. The normalized Shannon entropy of the membership distribution determines the input uncertainty ($U_x$). Similarly, the entropy of the cluster label distribution represents the output uncertainty ($U_y$). The final uncertainty measure is obtained through a weighted combination of these two components to produce the total uncertainty $U_{total}$.

# 5 Implications for Neural Data Analysis

The limitations discussed have profound implications for the analysis of neural data, such as Electroencephalography (EEG) and functional Magnetic Resonance Imaging (fMRI). These data streams are characterized by (Table 3):

- High Dimensionality and Complexity: EEG and fMRI data are notoriously high-dimensional, with numerous channels (EEG) or voxels (fMRI) and long temporal sequences. Feature selection, as provided by the Relief-family algorithms, is essential, but the lack of deep representation learning means that subtle, complex temporal or spatio-temporal patterns may be missed.

- Noise and Artifacts: Neural signals are heavily contaminated by noise from various sources, including biological processes, environmental interference, and participant movement. This noise contributes significantly to aleatoric uncertainty.

- Inter-Subject Variability: Brain structure and function exhibit significant inter-individual differences. Models trained on one group of subjects may not generalize well to another, leading to epistemic uncertainty due to domain shift.

- Dynamic Nature of Brain States: Brain activity is inherently dynamic, with states shifting over time. Robust clustering methods need to adapt to these changes.

**Table 3:** Key characteristics and analytical challenges of neural data streams.

| Characteristic | Neural Data Type | Analytical Challenge |
|---|---|---|
| High Dimensionality | EEG (multi-channel) & fMRI (voxel-dense) | Overfitting risks, high computational cost, and the need for dimensionality reduction. |
| Temporal Complexity | Long time-series sequences | Detecting non-stationary signals and capturing subtle long-range dependencies. |
| Spatio-temporal Coupling | Integrated brain activity | Disentangling spatially distributed patterns that evolve over precise time intervals. |
| Signal-to-Noise Ratio (SNR) | Scalp EEG & BOLD fMRI | High susceptibility to artifacts (motion, muscle, scanner noise) and potential loss of features. |

Without proper uncertainty quantification, it is difficult to:

1. Distinguish genuine neural patterns from noise: A clustering algorithm might identify a pattern as significant, but without uncertainty estimates, it's hard to know if it's a reliable signal or an artifact.
2. Assess the reliability of inferred brain states: In applications like BCI or clinical diagnosis, overconfidence in a derived state can lead to incorrect interventions. Fuzzy uncertainty can highlight ambiguous states, prompting further investigation or caution.
3. Understand model generalization: High uncertainty in predictions for new subjects or new experimental conditions can signal limitations in model generalizability, guiding researchers on where more data is needed.
4. Provide interpretable confidence levels: Clinicians and researchers need to understand the confidence associated with any classification or clustering outcome. For example, if an EEG analysis for seizure detection assigns a segment to a "seizure" cluster, knowing the uncertainty of this assignment is vital for clinical decision-making.

The proposed Relief-based distributed frameworks, by failing to address uncertainty adequately and by not aligning with deep learning principles for representation learning, may provide an incomplete or misleading picture of neural dynamics. Integrating fuzzy logic or other probabilistic uncertainty modeling techniques could provide the necessary confidence measures for reliable interpretation and application in neuroscience.

# 6 Conclusion

This critical commentary has examined distributed Relief-based ensemble clustering frameworks for time-series analysis. While these frameworks demonstrate commendable progress in computational efficiency through parallel and distributed processing, significant conceptual and practical limitations have been identified. We argue that they do not qualify as deep learning models due to the absence of hierarchical representation learning, end-to-end differentiable optimization, and latent representation learning. Furthermore, the reliance on metrics that primarily reflect algorithmic performance on controlled datasets is insufficient for guaranteeing real-world predictive reliability. Crucially, the lack of robust uncertainty quantification, encompassing both aleatoric and epistemic uncertainty, represents a substantial drawback.

We have proposed the integration of fuzzy logic as a promising approach to address this deficit by providing measures of fuzzy membership uncertainty, cluster uncertainty, and total uncertainty. These measures can offer a more nuanced understanding of confidence in clustering assignments. The implications for neural data analysis are particularly significant, where accurate interpretation and reliability are paramount for scientific discovery and clinical application. Future research should focus on developing ensemble clustering frameworks that not only scale efficiently but also incorporate sophisticated representation learning capabilities and rigorous uncertainty quantification to ensure trustworthy and reliable performance in complex real-world scenarios.

**Data and Software Availability:** The source code of the proposed framework [5] is available on GitHub at:

https://github.com/AinazBahramlou/EnsembleClusteringHadoop

The datasets used in this study are also provided/linked within the same repository for reproducibility.

# References

[1] Kuncheva, L. I. (2004). *Combining Pattern Classifiers: Methods and Algorithms*. John Wiley & Sons. https://doi.org/10.1002/0471660264

[2] Strehl, A., & Ghosh, J. (2002). *Cluster Ensembles—A Survey*. UCI Technical Report, 2002-3.

[3] Dean, J., & Ghemawat, S. (2008). MapReduce: Simplified Data Processing on Large Clusters. *Communications of the ACM*, *51*(1), 107-113. https://doi.org/10.1145/1327452.1327492

[4] Zaharia, M., Chowdhury, M., Franklin, M. J., Joseph, A. D., Katz, R., Konwinski, A., … & Stoica, I. (2010). Spark: Cluster Computing with Working Sets. *Proceedings of the 26th International Conference on Very Large Data Bases (VLDB)*, 1045-1054.

[5] Bahramlou, A., Hashemi, M.R. & Zali, Z. (2023). Ensemble clustering and feature weighting in time series data. *J Supercomput* 79, 16442–16478. https://doi.org/10.1007/s11227-023-05290-4
[6] Kononenko, I. (1994). Estimating Attributes: I. ReliefF. *Machine Learning*, *14*(1), 25-46. https://doi.org/10.1007/3-540-57868-4_57
[7] Bidarra, R., & Cardoso, J. (2001). Multi-surface feature selection. *Proceedings of the 15th European Conference on Artificial Intelligence (ECAI)*, 672-676.
[8] Li, J., & Liu, Y. (2004). Simba-Sc: A new feature selection algorithm. *Journal of Computer Science and Technology*, *19*(4), 535-539.
[9] Ruiz, M. A., & Marcano, A. (2009). I-Relief: An incremental feature selection algorithm. *Proceedings of the 9th International Conference on Intelligent Systems Design and Applications (ISDA)*, 625-630.
[10] Bahramlou, A. (2026). Ensemble Clustering and Feature Weighting in Time Series Data. *Zenodo*. https://doi.org/10.5281/zenodo.20697450
[11] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436-444. https://doi.org/10.1038/nature14539
[12] Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.
[13] Zadeh, L. A. (1965). Fuzzy sets. *Information and Control*, *8*(3), 338-353.
[14] Klir, G. J., & Yuan, B. (1995). *Fuzzy Sets and Fuzzy Logic: Theory and Applications*. Prentice Hall.
[15] Bezdek, J. C. (1981). *Pattern Recognition with Fuzzy Objective Function Algorithms*. Plenum Press.