OPTIMIZATION OF EXPERT FUZZY SYSTEMS

Matyáš Ratislav, [*]

Department of Concrete and Masonry Structures, Faculty of Civil Engineering,
Czech Technical University in Prague, Thakurova 7/2077, 166 29 Prague 6, Czech Republic.
matyas.ratislav@fsv.cvut.cz

ABSTRAKT

Fuzzy inferenční expertní systémy (FIS) představují silný nástroj pro modelování složitých, nelineárních a částečně stochastických systémů tím, že napodobují lidské uvažování pomocí fuzzy pravidel a funkcí příslušnosti. Přestože jsou široce využívány v inženýrství i rozhodovacích systémech, jejich ladění tak, aby odrážely měřená data, zůstává výzvou – zejména pokud je důležité zachovat jejich interpretovatelnost. Tento článek shrnuje existující optimalizační techniky pro FIS, včetně funkce tunefis v MATLABu a adaptivních neuro-fuzzy inferenčních systémů (ANFIS). Následně představuje tři nové metody pro datově řízené, avšak interpretovatelné ladění FIS: (1) adaptivní úpravy vstupních funkcí příslušnosti pomocí inverzní optimalizace na základě nových měření, (2) ladění konsekventů pravidel pomocí gradientní metody a (3) kalibrace parametrů využívající Bayesovskou inferenci a simulaci Monte Carlo pro učení s ohledem na nejistotu. Každá metoda je navržena tak, aby zlepšila výkonnost systému a zároveň zachovala logickou a sémantickou strukturu fuzzy modelů. Navržené metody jsou následně aplikovány na praktický příklad a porovnány.

KLÍČOVÁ SLOVA

Fuzzy logika • Optimalizace • Ladění parametrů • Bayesovská inference • Funkce příslušnosti

ABSTRACT

Fuzzy inference expert systems (FIS) provide a powerful framework for modeling complex, nonlinear, and uncertain systems by emulating human-like reasoning through fuzzy rules and membership functions. While widely applied across engineering and decision-making domains, tuning these systems to accurately reflect measured data remains a challenge—particularly when preserving interpretability is essential. This paper reviews existing optimization techniques for FIS, including MATLAB's tunefis function and adaptive neuro-fuzzy inference systems (ANFIS). It then proposes three novel methods for data-driven yet interpretable tuning of FIS: (1) adaptive updates of input membership functions through inverse optimization based on new measurements, (2) gradient-based tuning of rule consequents, and (3) a Bayesian parameter calibration framework utilizing Markov Chain Monte Carlo sampling for uncertainty-aware learning. Each method is designed to enhance system performance while upholding the logical and semantic structure of fuzzy models. The proposed methods are then applied to a practical example and compared.

KEYWORDS

Fuzzy logic • Optimization • Parameter tuning • Bayesian inference • Membership function

[*] Supervisor: prof. Ing. Petr Štemberk, Ph.D., D.Eng.

1. INTRODUCTION

Fuzzy inference expert systems (FIS) are widely used for modeling and decision-making in environments characterized by imprecision, uncertainty, and where only limited data is available. Unlike classical logic, fuzzy logic allows for gradual transitions between truth values, making it well-suited for applications involving vague or subjective information. This flexibility has led to its successful deployment in domains such as control systems, diagnostics, and artificial intelligence (Sheena et al. (2017)).

The performance of fuzzy systems can be increased by manual tuning, particularly of membership functions and rule parameters. To support engineering judgment with data-driven evidence, various optimization techniques have been developed. These range from gradient-based methods to metaheuristic algorithms and neuro-fuzzy approaches, enabling more efficient and systematic model calibration (Moreno-Velo et al. (2003)).

This paper reviews existing methods for FIS optimization and introduces several new approaches designed to enhance both the adaptability and interpretability of fuzzy models. The proposed methods include adaptive membership function updates, rule consequent tuning, and a Bayesian framework for parameter calibration. Together, these methods provide a flexible toolkit for refining fuzzy inference systems using data.

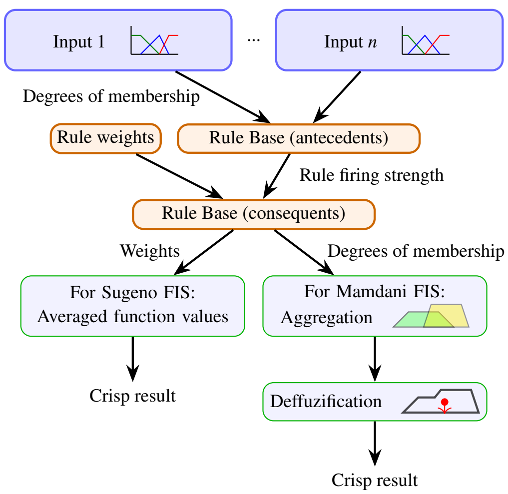
2. FUZZY INFERENCE SYSTEMS AND THEIR OPTIMIZATION

Fuzzy logic inference systems are used for modeling various phenomena and decision making by mimicking the approximate "fuzzy"reasoning used by humans. Unlike classical Boolean logic, where statements are true or false, fuzzy logic allows intermediate degrees of truth, enabling reasoning in conditions where data are incomplete or vague (Zadeh (1965)). This is possible in fuzzy inference systems through the use of fuzzy sets in decision rules. Fuzzy sets are objects without crisp, clearly defined boundaries (MathWorks, Inc. (1994–2025*c*)) - membership in those sets is a matter of degree - element can be partially a member of set A and $\neg A$ simultaneously.

The growth of fuzzy logic applications in recent decades has been substantial. Its capabilities have been successfully employed in various domains, including consumer electronics (e.g. cameras, washing machines, and air conditioners), industrial control systems, decision support systems, medical diagnostics, financial modeling, and artificial intelligence. These systems benefit from the ability of fuzzy logic to handle imprecision, model nonlinear systems, and incorporate expert knowledge without requiring an exact mathematical model (MathWorks, Inc. (1994–2025*c*)).

A fuzzy inference system (FIS) typically consists of a set of linguistic rules in the form of "IF-THEN" statements, fuzzy membership functions (MF) that determine degrees of membership of system's inputs to defined linguistic fuzzy sets, and an inference mechanism that combines these rules to produce a fuzzy output. This output is then defuzzified into a crisp value suitable for decision making or control (Mendel (1995)).

Among the most widely used fuzzy inference systems are the Mamdani and Sugeno types, each suited to different types of applications. The Mamdani FIS, introduced by Ebrahim Mamdani in 1975 (Blej & Azizi (2016)), is characterized by using fuzzy sets, not only in rules antecedents but also in consequents. This approach closely resembles human reasoning and is thus widely used in applications where interpretability is a priority. In contrast, the Sugeno FIS, developed by Takagi, Sugeno, and Kang in 1985 (Blej & Azizi (2016)), uses fuzzy sets in the rule antecedents but employs mathematical functions as rule consequents. Therefore, Sugeno fuzzy systems are more computationally efficient and easier to integrate with optimization and learning algorithms (MathWorks, Inc. (1994–2025*a*)). The basic structure of fuzzy inference systems is depicted in figure 1.



Obrázek 1: *Flowchart of Mamdani and Sugeno fuzzy inference systems.*

2.1. Existing Methods for FIS Optimization

Optimizing FIS with data is essential for enhancing their performance and accuracy in various applications. Several methods have been developed to fine-tune FIS parameters, including membership functions and rule bases.

MATLAB's tunefis function

The Fuzzy Logic Toolbox in MATLAB (version R2024b) offers the `tunefis` function, which facilitates the optimization of fuzzy inference systems by minimizing a specified cost function. This optimization is achieved through the adjustment of selected parameters, including input and output membership functions and the rule base. The tuning process can employ various optimization algorithms, such as genetic algorithms, particle swarm optimization, and pattern search methods (MathWorks, Inc. (1994–2025*b*)).

This functionality is particularly beneficial for initial system tuning based on empirical data or for approximating complex systems, such as artificial neural networks (MathWorks, Inc. (2024)).

However, the `tunefis` function presents certain limitations that may affect the interpretability of the resulting FIS. One notable issue is the necessity to impose constraints on tunable parameters to preserve the logical consistency of the system. For instance, consider an input variable "height" with fuzzy sets labeled "short," "average," and "tall," each defined by specific membership functions. Without appropriate constraints, the tuning algorithm might adjust the membership function for "tall" to overlap with or fall below that of "average," thereby violating the intended semantic structure of the fuzzy sets. In `tunefis`, constraints can only be specified as fixed numerical bounds; however, logical consistency relies on relative relationships between parameters.

Another concern is that the `tunefis` function does not inherently normalize the membership functions. Consequently, for a given input, the sum of membership degrees across all fuzzy sets can be less than, or greater than one. This may complicate the interpretability of the system. For example, an input value of 1.71 meters could result in membership degrees of 0.5 for "short," 0.7 for "average," and 0.3 for "tall," totaling 1.5. While such non-normalized approaches can be advantageous in certain contexts, they often introduce ambiguity in the interpretation of membership degrees.

Adaptive Neuro-Fuzzy Inference System (ANFIS)

ANFIS integrates the learning capabilities of artificial neural networks with fuzzy logic. This hybrid approach enables the modeling of complex, nonlinear systems by learning from data while maintaining interpretability through fuzzy rules.

ANFIS is an artificial neural network with a multi-layer architecture copying structure of Sugeno type FIS. The system utilizes a hybrid learning algorithm that combines gradient descent and least squares estimation to optimize the input parameters of the membership functions and the parameters of the rules consequent functions that are usually in a form of first order polynomial (Karaboga & Kaya (2019)).

To enhance the performance and generalization capabilities of ANFIS, various optimization techniques have been proposed. Metaheuristic algorithms such as Genetic Algorithms or Particle Swarm Optimization have been integrated with ANFIS to optimize its parameters effectively (Moayedi et al. (2020)).

3. PROPOSED METHODS FOR FIS OPTIMIZATION

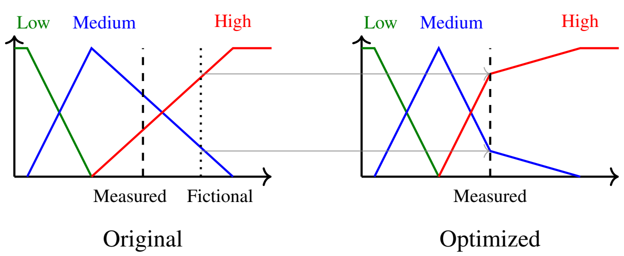
The following methods were developed to fine-tune fuzzy inference systems with data preserving their interpretability. All of the optimization methods presented are applicable to the Sugeno and Mamdani systems with an arbitrary number of input parameters and one output.

3.1. Adaptive Update of Membership Functions Based on New Measurements

To enhance the adaptability of a fuzzy inference system (FIS) to evolving datasets, the `set_MF_trends` function provides a mechanism for updating input membership functions based on newly observed input-output pairs. This function modifies the shapes of selected membership functions to better reflect new measurement data, while preserving interpretability and consistency within the fuzzy model. The manipulated membership functions must support incremental definition via data points (e.g., polygons or two connected monotonic splines); such membership functions are referred to as *adaptive*.

The method is grounded in constrained numerical optimization. Given a new input measurement and a corresponding desired output value, the function searches for a set of hypothetical input values (referred to as "fictional inputs") that would yield the same output under the current FIS configuration. This inverse mapping is achieved using the Sequential Least Squares Programming (SLSQP) algorithm by SciPy Community (2008–2025). The optimization problem is subject to equality constraints ensuring that the FIS output matches the desired output. The objective function minimizes the deviation of the fictional inputs from the actual measurements, normalized over the domain of each input. This normalization can be configured in a flexible manner, allowing certain inputs to be penalized more or less than others if desired.

Once the optimal fictional inputs are identified, the function compares the active fuzzy sets, i.e., those with nonzero membership values, corresponding to both the actual and fictional inputs. If the same fuzzy sets are active, the associated adaptive membership functions are updated by incorporating new defining data points so they have for actual inputs the same degrees of membership as for the fictional inputs (see figure 2). However, if different fuzzy sets are activated, implying that peaks or boundaries of the membership functions would need to shift, the function halts execution. This safeguard prevents unintended logical inconsistencies in the FIS structure.



Obrázek 2: *Comparison of input's membership functions before and after `set_MF_trends` optimization.*

The `set_MF_trends` function thus offers a controlled and interpretable approach to tuning adaptive fuzzy models in response to new data. However, its application to large datasets or datasets with significant uncertainty is limited. One potential solution involves randomly selecting a subset of data points and, if the corresponding membership function updates are feasible (i.e., the optimization does not halt), evaluating the model cost on the remaining data using a chosen cost function. This process can be repeated multiple times, and the model configuration with the lowest cost can be selected, or weighted average of MFs shapes can be taken (with their inverse cost as weights). Alternatively, clustering techniques may be applied to large datasets, followed by selecting a few representative (average) measurements from each cluster for adaptation.

3.2. Rules Consequent Tuning Algorithm

The proposed method (referred to as `train_consequent`) leverages the interpretability of fuzzy rules whose consequent parts are defined in a fuzzy manner. For example, a rule may take the form: *"IF (antecedent) THEN result is Low with 20% and Medium with 80%"*. These percentages are referred to as consequent weights. This formulation reduces the number of output membership functions required, thereby helping them retain their semantic meaning. The optimization algorithm iteratively adjusts the consequent weights within a fuzzy inference system (FIS) in order to minimize a selected cost function (e.g., mean squared error, Huber loss, etc. - see Tonyloi (2024)) between the predicted and target outputs.

The input to `train_consequent` consists of an arbitrary number of input-output pairs. Optionally, each sample may be assigned a weight to reflect its importance or reliability during training. At the core of the tuning procedure lies a custom gradient-based optimization algorithm. For each training sample, the algorithm computes the predicted output and evaluates the associated cost using the selected loss function.

The gradient of the cost function with respect to the rule consequent weights is then estimated. This gradient reflects the sensitivity of the prediction error to changes in the consequent weights. For each training sample, its contribution to the gradient is computed based on the local discrepancy between the target and predicted output, the influence of each rule via its firing strength, and an estimate of how variations in individual consequent weights would affect the prediction. The resulting gradients from individual samples are then aggregated across the entire training set, with each sample's influence modulated by its optional weight.

The rule weights are updated using a normalized gradient descent approach. The gradient is first scaled to a fixed step size, and a tentative update is applied. If this update reduces the cost function value, it is accepted; otherwise, the step size is halved and the update is retried. This adaptive mechanism continues until either a successful update is found or the step size falls below a predefined threshold. This strategy avoids instability due to overly large updates while maintaining steady convergence toward a lower error.

To ensure numerical stability and preserve interpretability, the consequent weights are normalized after each update so that they are non-negative and sum to one within each rule. This normalization guarantees that the rules remain interpretable as weighted combinations of output terms or functions. The optimization proceeds iteratively until a maximum number of iterations is reached or the step size becomes too small to allow further improvement.

The outcome of the procedure is a tuned fuzzy inference system whose rule base more effectively captures the underlying patterns present in the training data, thereby enhancing predictive accuracy. As the algorithm relies on gradient descent, it is inherently constrained to finding local minima of the cost function. This characteristic helps preserve the interpretability of the system, as the original rule structure remains largely intact and only the consequent weights are adjusted. However, it also implies that the quality of the final model strongly depends on the initial design of the FIS. A poorly initialized rule base may lead to suboptimal performance, as the algorithm is not capable of performing global structural modifications.

3.3. Bayesian Calibration of Fuzzy Inference Systems

The function `Bayesian_learn` implements a Bayesian framework for parameter calibration of fuzzy inference systems (FIS), utilizing the Metropolis-Hastings algorithm, a Markov Chain Monte Carlo (MCMC) sampling technique (Kruschke (2015)). This method facilitates robust estimation of parameters in the presence of noise and uncertainty, while allowing the incorporation of prior knowledge in a principled manner through prior probability distributions.

At the core of this approach is a probabilistic model in which selected FIS parameters are treated as random variables. The user provides prior distributions for each tunable parameter (e.g., normal or uniform distributions), which encode initial beliefs about their plausible values. Based on the provided input-output training data, the method evaluates how well each sampled parameter configuration explains the observed outputs using a likelihood function. This function quantifies the probability of the data given a parameter configuration, and is based on the residuals (differences between predicted and observed values) using a user-defined noise model. This design offers flexibility in modeling measurement uncertainty.

The sampling itself is performed via the Metropolis-Hastings algorithm, which iteratively proposes new parameter values by perturbing the current state with normally distributed noise scaled by a user-defined proposal step size. Proposed samples are accepted or rejected based on their posterior probability, which combines prior knowledge and data likelihood. The method supports additional user-defined parameter constraints, enabling the enforcement of physical or logical relationships between parameters (e.g., orderings or bounds).

The MCMC process begins from user-specified or automatically-initialized values. After a designated burn-in period (during which early samples are discarded to mitigate initialization effects), the remaining samples are used for inference. The final parameter values used to update the FIS can be set either as the posterior mean or as the maximum a posteriori (MAP) estimate.

The function also reports the acceptance ratio of the MCMC process, which serves as a diagnostic for tuning the proposal step size. An acceptance ratio that is too low indicates that the proposal distribution is overly narrow, while a ratio that is too high suggests it is too broad—both of which can impede convergence.

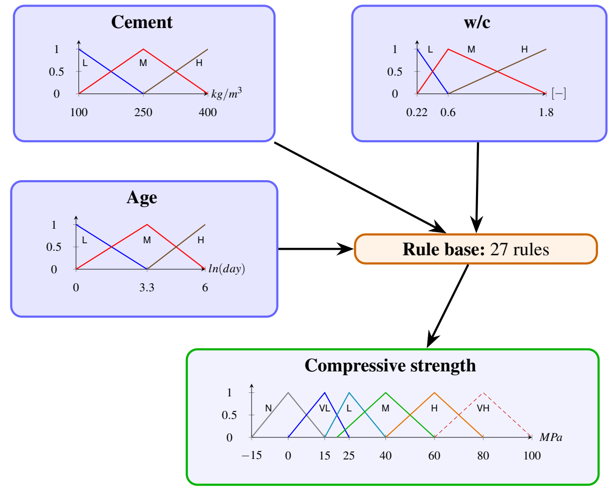
This optimization algorithm is particularly advantageous in applications where interpretability, prior expert knowledge, and uncertainty quantification are important. However, the method is computationally intensive: each proposed parameter set requires reconstructing the FIS and processing all data points. Therefore, although the algorithm's structure is theoretically applicable to Mamdani-type systems, its practical utility is limited to Sugeno-type FIS, owing to their significantly faster inference times.

4. APPLICATION

Proposed optimization methods where used to create a system that predicts concrete compressive strength. Data for FIS tuning and system validation was taken from Yeh (2007). This dataset contains 1030 instances and 9 attributes (8 input features and 1 output). The features represent quantitative measurements of materials used in concrete (e.g., cement, water, aggregates) in kg/m³, along with the age of the sample in days. There are no missing values, and the dataset originates from the study by Yeh (1998) on modeling concrete strength using neural networks .

Since expert systems (like FISs) are usually used in situations where only limited number of experiment were done (Basheer & Hajmeer (2000)), only 10 specimens were chosen at random from the dataset to represent the learning dataset and rest was used after systems optimization to validate the tuning method. For the FIS only 3 most significant inputs (cement content, water–cement ratio and age) were considered, to keep the problem easy to design and understand. The FIS used for optimization is schematically showed in figure 3.

All of the three proposed optimization methods were applied separately and later compared by various metrics.



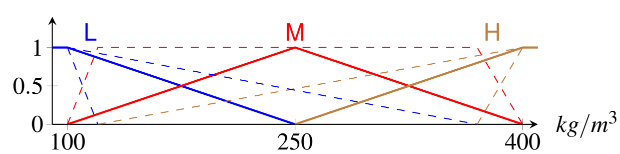
Obrázek 3: *Fuzzy inference system architecture for concrete compressive strength estimation.*

4.1. Optimized Parameters of the FIS

The `set_MF_trends` optimization was applied as follows. The training dataset was randomly divided into five subsets, each containing two samples. Each subset was used to optimize the original FIS (with use of monotone splines as MF), while the remaining subsets were used to compute the R^2 score. These scores (if positive) were used as weights, and the final shape of the membership functions was determined as a weighted average of the MF shapes from the optimized FISs.

The `train_consequent` algorithm was applied to the training dataset and terminated after 20 iterations (withou reaching minimum allowed step size). This limit was set to prevent overfitting. In the final step, the change in average squared error was less than 2 ‰.

In the case of `Bayesian_learn`, the positions of the peaks of the M (medium) membership functions were optimized. Uniform prior distributions were used. A visualization of one of the input membership functions, along with its prior probability distribution, is shown in Figure 4, where the dashed lines indicate the limiting positions of the MFs. The output membership functions were replaced by their centroid values, effectively approximating the original Mamdani-type FIS with a Sugeno-type system for faster computation. A normal distribution was assumed for the measurement noise.



Obrázek 4: *Prior distributions of the cement content MF parameters used in the Bayesian FIS optimization.*

4.2. Validation

To evaluate the performance of the FISs optimized by the three proposed methods, a comparison with validation data was conducted. The prediction accuracy was assessed using the coefficient of determination (R^2), the root mean squared error (RMSE), and the mean absolute error (MAE). These metrics capture different aspects of predictive quality: R^2 reflects the proportion of variance explained by the model, RMSE emphasizes larger errors, and MAE provides a robust measure of average error magnitude.

The table 1 summarizes the validation results for each optimization method.

Tabulka 1: *Validation metrics for FISs optimized using different methods*

Optimization Method	R^2	**RMSE**	**MAE**
Original FIS	*0.37*	*12.9*	*10.4*
`set_MF_trends`	*0.45*	*12.1*	*9.7*
`train_consequent`	*0.61*	*10.2*	*8.1*
`Bayesian_learn`	*0.57*	*10.7*	*8.7*

5. CONCLUSIONS

This paper addressed the challenge of tuning fuzzy inference systems in a data-driven yet interpretable way. While conventional tools such as ANFIS and MATLAB's `tunefis` offer robust solutions for many applications, they often compromise interpretability. To bridge this gap, we introduced three complementary methods: adaptive inverse optimization of membership functions, gradient-based tuning of rule consequents, and Bayesian calibration through MCMC sampling. Each approach enhances the model's fit to observed data while respecting the logical structure and interpretability of fuzzy models. Applied to a practical example, these methods demonstrated their ability to improve performance without significant loss of transparency. The algorithm `train_consequent` performed best compared to the other proposed algorithms; however, this may vary depending on the specific problem and the quality of non-optimised FIS.

ACKNOWLEDGEMENTS

The presented work has been realized within Institutional Support by Ministry of Industry and Trade of the Czech Republic and the CTU in Prague project SGS25/039/OHK1/1T/11.

Reference

Basheer, I. A. & Hajmeer, M. (2000), 'Artificial neural networks: fundamentals, computing, design, and application', *Journal of Microbiological Methods* **43**, 3–31.
URL: *https://www.elsevier.com/locate/jmicmeth*

Blej, M. & Azizi, M. (2016), 'Comparison of mamdani-type and sugeno-type fuzzy inference systems for fuzzy real time scheduling', *International Journal of Applied Engineering Research* **11**(22), 11071–11075. CRMEFO, MATSI Lab, Mohamed 1st University Oujda, Morocco.
URL: *https://www.researchgate.net/publication/311795100_Comparison_of_Mamdani-type_and_Sugeno-type_fuzzy_inference_systems_for_fuzzy_real_time_scheduling*

Karaboga, D. & Kaya, E. (2019), 'Adaptive network based fuzzy inference system (anfis) training approaches: a comprehensive survey', *Artificial Intelligence Review* **52**, 2263–2293.
URL: *https://doi.org/10.1007/s10462-017-9610-2*

Kruschke, J. K. (2015), *Doing Bayesian Data Analysis: A Tutorial with R, JAGS, and Stan*, 2 edn, Academic Press. Available online via ScienceDirect.
URL: *https://nyu-cdsc.github.io/learningr/assets/kruschke_bayesian_in_R.pdf*

MathWorks, Inc. (1994–2025*a*), *Mamdani and Sugeno Fuzzy Inference Systems*, MathWorks. Accessed: April 12, 2025.
URL: *https://www.mathworks.com/help/fuzzy/types-of-fuzzy-inference-systems.html*

MathWorks, Inc. (1994–2025*b*), *tunefis*, MathWorks. Accessed: April 12, 2025.
URL: *https://www.mathworks.com/help/fuzzy/tunefis.html*

MathWorks, Inc. (1994–2025*c*), 'What is fuzzy logic?'. Accessed: 2025-04-10.
URL: *https://www.mathworks.com/help/fuzzy/what-is-fuzzy-logic.html*

MathWorks, Inc. (2024), 'Fuzzy logic controller tuning | fuzzy logic, part 4 [video]', `https://www.youtube.com/watch?v=MbUaxn-7bJI`. Accessed: April 12, 2025.

Mendel, J. M. (1995), 'Fuzzy logic systems for engineering: a tutorial', *Proceedings of the IEEE* **83**(3), 345–377.

Moayedi, H., Raftari, M., Sharifi, A., Wan Jusoh, W. A. & Rashid, A. S. A. (2020), 'Optimization of anfis with ga and pso estimating ratio in driven piles', *Engineering with Computers* **36**, 227–238.
URL: *https://link.springer.com/article/10.1007/s00366-018-00694-w*

Moreno-Velo, F., Baturone, I., Senhadji, R. & Sanchez-Solano, S. (2003), Tuning complex fuzzy systems by supervised learning algorithms, *in* 'The 12th IEEE International Conference on Fuzzy Systems, 2003. FUZZ '03.', Vol. 1, pp. 226–231 vol.1.

SciPy Community (2008–2025), *scipy.optimize.minimize*. Accessed: 2025-04-13.
URL: *https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.minimize.html*

Sheena, A. D., Ramalingam, M. & Anuradha, B. (2017), 'A comprehensive study on fuzzy inference system and its application in the field of engineering', *International Journal of Engineering Trends and Technology* **54**, 36–40.

Tonyloi, I. (2024), 'Understanding cost functions and their role in measuring the accuracy of machine learning models'. Medium article.
URL: *https://datascienceafrica.medium.com/understanding-cost-functions-and-their-role-in-measuring-the-accuracy-of-machine-learning-models-63ae4f7d03c5*

Yeh, I.-C. (1998), 'Modeling of strength of high-performance concrete using artificial neural networks', *Cement and Concrete Research* **28**(12), 1797–1808.
URL: *https://www.sciencedirect.com/science/article/pii/S0008884698001653*

Yeh, I.-C. (2007), 'Concrete compressive strength data set'. Accessed: 2025-04-15.
URL: *https://archive.ics.uci.edu/ml/datasets/Concrete+Compressive+Strength*

Zadeh, L. (1965), 'Fuzzy sets', *Information and Control* **8**(3), 338–353.
URL: *https://www.sciencedirect.com/science/article/pii/S001999586590241X*