Journal of Electrical and Computer Engineering Innovations (JECEI)
Journal homepage: http://www.jecei.sru.ac.ir

Research paper

Sonar Target Classification Using a Decision Fusion Method Based on a Fuzzy Learning Automata

Sajjad Mahmoodi Khah, Seyed Hamid Zahiri*, Iman Behravan

Department of Electronic, Faculty of Electrical Engineering and Computer, University of Birjand, Birjand, Iran.

Article Info

Article History:
Received 19 June 2025
Reviewed 29 July 2025
Revised 26 September 2025
Accepted 07 October 2025

Keywords:
Sonar
Classification
Fuzzy learning automata
Performance parameters
Data fusion

*Corresponding Author's Email Address: *hzahiri@birjand.ac.ir*

Abstract

Background and Objectives: Sonar data processing helps in identifying and tracking targets with unstable echoes, which conventional tracking methods often misidentify. Recently, RLA has significantly improved the accuracy of undersea target detection compared to traditional sonar object recognition techniques that tend to lack robustness and precision.
Methods: This research utilizes a combination of classifiers to improve the accuracy of Sonar data classification for complex tasks like identifying marine targets. Each classifier creates its own data pattern and maintains a model. Ultimately, a weighted voting process is carried out by the fuzzy learning automata algorithm among these classifiers, with the one receiving the highest votes being the most impactful on performance improvement.
Results: We compared the performance of SVM, RF, DT, XGBoost, ensemble methods, R-EFMD, T-EFMD, R-LFMD, T-LFMD, ANN, CNN, TIFR-DCNN+SA, and joint models against the proposed model. Given the differences in objectives and databases, we focused on benchmarking the average detection rate. This comparison examined key parameters including, Precision, Recall, F1_Score, and Accuracy, to highlight the superior performance of the proposed method compared to the others.
Conclusion: The results obtained with the analytical parameters Precision, Recall, F1_Score, and Accuracy have been examined and compared with the latest similar research and the values of 88.6%, 90.2%, 89.02% and 88.6% have been obtained for each of these parameters in the proposed method, respectively. Also, in this research, the impressive performance of the new method compared to the Sonar data fusion by the conventional learning automata method is evident.

This work is distributed under the CC BY license (http://creativecommons.org/licenses/by/4.0/)

How to cite this paper:
S. Mahmoodi Khah, S. H. Zahiri, I. Behravan, "Sonar target classification using a decision fusion method based on a fuzzy learning automata," J. Electr. Comput. Eng. Innovations, 14(1): 211-226, 2026.
DOI: 10.22061/jecei.2025.11966.846
URL: https://jecei.sru.ac.ir/article_2426.html

2345- 3044/© 2026 The Authors. Published by Shahid Rajaee Teacher Training University.

Introduction

The classification and recognition of underwater objects is a crucial area of research in modern underwater acoustics. As advancements in noise reduction technology continue, underwater targets have become increasingly quiet, posing challenges for passive Sonar detection and recognition [1]. The classification of sonar data constitutes a significant challenge owing to the complexities inherent in the underwater environment. Contributing factors such as sound scattering, ambient noise, and frequency-dependent absorption further exacerbate the difficulties encountered in this domain [2]. The identification of surface and underwater sources through machine learning (ML) involves the classification of input features. Unlike a human sonar operator, who is limited to observing a single direction at any given moment, an automated ML system can concurrently process data from multiple directions. Building upon the demonstrated success of ML techniques in enhancing performance through the utilization of engineered features, this study introduces an innovative methodology based on learning automata (LA) [3], [4].

Given the limitations outlined above and the inimitable benefits of artificial intelligence (AI) methods for addressing recognition challenges in complex and blurred ambiances, various researchers have begun employing learning-based techniques to classify underwater sound. Moreover, the effectiveness of AI-based methods in similar domains, like acoustic scene classification [5], [6], acoustic event detection [7], and speaker identification [8], [9]. Considering the latest trends in underwater Sonar classification, we limit the scope of our study to learning-based procedures. Classical ML models perform well with small datasets but may not achieve satisfactory accuracy with large datasets that have diversified feature spaces. Given the success of reinforcement learning (RL) procedures in various fields, multiple studies have adopted and proposed RL-based approaches for classifying underwater sounds [3], [10]. However, many researchers have enhanced and proposed various reinforcement learning algorithms (RLAs) for Sonar objective classification and cognition [11]. In supervised ML, a dataset is given to the learning algorithm along with labels that indicate the correct output for the given data. Algorithms such as k-nearest neighbor (KNN), support vector machine (SVM), random forests (RF), and artificial neural networks (ANN) are examples of this learning.

In this work, the data fusion (DF) problem in Sonar data classification is considered due to its importance in various applications such as navigation and marine surveillance. However, we must mention that the mechanism of fuzzy learning automata algorithm (FLAA) has not been used in this field yet. We intended to check whether using mechanisms related to FLAA can be effective and efficient in DF at the decision level. However, data integration at the level of data, decision, and feature has been used in the problem of Sonar data classification. But until now, the use of an ML method such as FLAA to increase the ability to classify targets has been neglected. In this article, we measured the remarkable performance of the proposed method for 6 different objectives with Precision, Recall, F1_Score, AUC, and Accuracy indicators. Noise and acoustic interferences make the act of identification difficult in the vast and diverse oceanic and marine environments. In most marine devices, target detection is done by human operators, and with the development of this method in detecting various targets, the speed and accuracy of identification can be increased, and human errors can be reduced in these cases.

Due to the numerous parameters in the learning automata algorithm (LAA), all of which impact performance, finding the optimal set of values is a challenging task. We spent a significant amount of time refining these parameters to achieve the best results. Our previous research required extensive adjustments to various LAA parameters to achieve the desired performance. This experience led us to consider isolating the more effective parameters and identifying which ones have the greatest impact on LAA. We also explored using a smart tool to automatically adjust these parameters when necessary, allowing us to leverage LAA's full capabilities and obtain answers more quickly, accurately, and efficiently.

Therefore, we considered controlling these key parameters with a fuzzy controller and providing a soft computing tool that combines the capabilities of fuzzy systems and LA systems. In fact, the algorithm we introduced in this paper benefits from both systems' features, whereas previous research only used traditional LAA. This creates a pathway for LAA to be applied as a soft computing approach in many areas.

In the following article, we will examine, in order of related research, a number of machine learning algorithms used, the learning automata algorithm, fuzzy evaluation, the method, data, and device used, and the results obtained.

Related Work

In many research studies, Environment-based performance is emphasized to obtain the most expected benefits in reinforcement learning (RL), which is one of the main branches in the field of ML [12]. Valdenegro-Toro et al. used a convolutional neural network (CNN) to detect the object of an undersea Sonar image, and after training the network, the average detection rate in test sets reached 90% [13]. A Sonar objective cognition procedure based on a shallow CNN has fault cognition and insufficient model strength. Ferguson et al. proposed the use of a deep CNN to detect the sound of an undersea ship in a shallow water ambiance. In this article, a data augmentation technique is introduced, and the criterion for comparing data integration performance at the feature level is the precision parameter [14]. Huo et al. proposed a classification method for Sonar target detection based on semi-synthetic data training and transfer learning for small sample Sonar datasets. Experiments indicate that transfer training and semi-synthetic training can help increase model cognition accuracy [15].

In reference [16], a hybrid dragonfly algorithm is proposed to train a multi-layer perceptron (MLP) neural network to design a classifier in solving complex issues and to distinguish true targets from fake objectives in Sonar applications. In this paper, by combining DA and ChoA algorithms, the researchers were able to achieve a suitable classification rate and execution time compared to the separate performance of each algorithm. Using six ML algorithms such as KNN, RF, SVM, decision tree (DT), extreme gradient boosting (XGboost), and ensemble methods, Reddy et al. conducted research with the help of Sonar data to find sea mines. The Ensemble method is the combination of RF, XGboost, and Voting Classifier. Comparative results including Accuracy, Precision, Recall, and F1-score for all these algorithms are presented in this paper [17]. In this research [18], the first used 4 classification models separately to classify Sonar data. Then, by combining those classifiers with LA algorithm to achieve the best solution and by determining the optimal coefficients for each classifier, they were able to achieve significant results compared to similar works. The outcomes corresponding to the analytical metrics of Precision, Recall, F1 Score, and Accuracy have been reported, with respective values of 88.6%, 90.2%, 89.02%, and 88.6% achieved by the proposed method. Also, Wang et al presented a method of identifying active Sonar targets based on multi-domain transformations and precision-based fusion networks. The results of the experiments show that by using multi-domain transformations, active Sonar echoes can be accurately detected. Improved by 10.5% compared to single domain methods. In addition, the identification performance of different fusion models such as the early fusion model with resnet (R-EFMD) as the backbone of multi-domain attention-based feature extractor (MAFE), early fusion model with swin transformer (T-EFMD) as the backbone of MAFE, late fusion model with resnet (R-LFMD) as the backbone of single domain feature extractor (SFE) no attention-based feature extractor (AFE) module, and late fusion model with swin transformer (T-LFMD) as the backbone of SFE no AFE module has been compared [19]. Tian et al. designed a collaborative learning model for underwater Sonar recognition. In this study, a lightweight multiscale residual deep neural network (MSRDN) is developed utilizing efficient network design strategies. This approach results in a reduction of 64.18% in parameters and 79.45% in floating-point operations (FLOPs) compared to the original MSRDN, while maintaining accuracy. It decreases a little. Then, a combined model of wave representation and time-frequency-based models was presented. The results of deterministic experiments prove that the performance improvement of the proposed techniques from mutual ML has advantages such as favorable recognition accuracy [20].

Yang et al. implemented a spatial attention deep convolutional neural network (DCNN) for marine mammal call detection. This method tends to use spatial attention (SA) to help the DCNN achieve better detection performance. Time-frequency image recognition-DCNN (TFIR-DCNN) is designed at the beginning of this method. Then, SA is added to the TFIR-DCNN to help the TFIR_DCNN focus on the location of call features in the time and frequency domains. Favorable marine mammal contact detection test results have been reported [21]. Ahmed et al. investigated an underwater audio signal classification model with a deep learning method. A regular neural network is also implemented to classify audio as input features. Comparing the performance of this classifier and the general results of the presented models is promising [22].

Algorithms

In this section, we use classifier fusion to increase the accuracy of the classification of complex problems. In fact, each of these classifiers builds its own model on the data and stores this model. In the final classification step, a vote occurs among the classifiers, and the class receiving the highest number of votes is deemed to have had the greatest impact on the classification [18].

K-Nearest Neighbor

KNN is a classification algorithm and there are mainly two phases in classification. The first phase is learning, in which a classification is made using the training data, and in the second phase, the evaluation of the classifier is done [23], [24]. As presented in Fig. 1, the new data computes the distance of each of its neighbors according to the K value. Then, it specifies the class that contains the maximum number of nearest neighbors to it [25].

After collecting KNN, we simply select most of them to predict the training sample class. The agents that affect the operation of this algorithm are K value, Euclidean distance, and parameter normalization. For a precise understanding of the algorithm's performance and according to the set of training data shown in (1), the steps are as follows. First, the training set is stored, and then the Euclidean distance for each new unlabeled data among two points x and y in all training data points is calculated using (2).

$$\{(x(1), y(1)), (x(2), y(2)), \dots, (x(m), y(m))\} \tag{1}$$

$$d = \sqrt{\sum_{k=1}^{N} (x_k - y_k)^2} \tag{2}$$

KNNs are determined, and the maximum number of nearest neighbors is assigned to a class. After saving the training, all the parameters should be set to normal so that the calculations become easier. The value of K affects the algorithm because it can be used to create the boundaries of each class. The best solution is selected first by checking the data. Larger solutions of K are more accurate because they decrease the net noise, but this is not guaranteed.

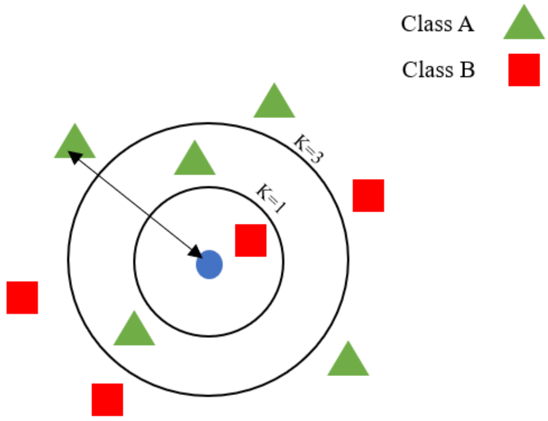


Fig. 1: KNN classifier method.

Naïve Bayesian

Bayes' theorem is one of the most widely used theorems for inferential calculations and many advanced ML models. Naïve Bayesian (NB) argument plays an impressive role in science [26]. This analysis permits us to respond problems for which frequent technical approaches were not constructed. In other words, the frequentist paradigm [27].

We assume that the dataset contains n samples x_i, $i = 1 \dots n$, which include p features, i.e., $x_i = (x_{i1}, x_{i2}, \dots, x_{ip})$. Every sample is assumed to belong to only one class $y \in \{y_1, y_2, \dots y_m\}$. Most predictive models in ML produce numeric predictions for each sample x_i. This distinction indicates the class membership of that item in class y_j. If the dataset includes only negative and positive samples, $y \in \{0, 1\}$, then the predictive model can be handled as a ranker or a classifier. By mounting a threshold t on the ranking score, $s(x)$, like that $\{s(x) \geq t\} = 1$, the ranker becomes a crisp classifier [28].

This learning method involves creating a NB probabilistic model that assigns a posterior class eventuality to each sample: $P(Y = y_j \mid X = x_i)$. The straightforward classifier utilizes these eventualities to assign a sample to a class. By applying the theorem introduced in (3) and simplifying the notation, we derive (4).

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{3}$$

$$P(y_j|x_i) = \frac{P(x_i|y_j)P(y_j)}{P(x_i)} \tag{4}$$

Ward that the numerator in (4) is the joint probability of x_i and y_j. Consequently, the numerator can be expressed in this way. Let us consider that the individuals x_i are independent of each other. This hypothesis implies that $P(x_1 \mid x_2, x_3, \dots x_p, y_j) = P(x_1 \mid y_j)$, which we can plug into (4), and we obtain (5).

$$P(y_j|x) = \frac{\prod_{k=1}^{p} P(x_k|y_j)P(y_j)}{P(x)} \tag{5}$$

By applying the explicit classification principle, we can compute the counter value for each class and select the one with the highest value. This approach is known as the maximum posterior rule [29]. The computed class posterior probabilities serve as natural rating scores. Reapplying the probability theorem allows us to rewrite (5) and derive the general (6).

$$P(y_j|x) = \frac{\prod_{k=1}^{p} P(x_k|y_j)P(y_j)}{\prod_{k=1}^{p} P(x_k|y_j)P(y_j) + \prod_{k=1}^{p} P(x_k|y_j^c)P(y_j^c)} \tag{6}$$

Decision Tree

Data mining is used to extract useful information from large datasets and to display it in easy-to-interpret visualizations. DTs are one of the most effective methods for data mining; they have been widely used in several disciplines [30]. They are user-friendly, unambiguous, and resilient even when faced with missing data. Both discrete and continuous variables can serve as either target variables or independent variables [31].

Inductive inference is the process of moving from concrete examples to common models. In one way, the object is to learn how to classify targets or situations by analyzing a set of instances whose classes are known. Samples are typically represented as feature-value vectors that give the numerical or nominal values of a fixed collection of properties. Learning input contains a set of such vectors, each belonging to a known class, and the output contains a mapping from feature values to classes. This mapping should accurately classify both the given samples and other unseen samples [32].

A DT is a tree-based technique where any path starting from the root is defined by a data-separating sequence until a Boolean outcome is reached at the leaf node [33]-[36]. It serves as a hierarchical representation of knowledge relationships that includes nodes and connections. When relationships are used for classification, nodes represent objectives [37]-[39]. Classification algorithms are capable of handling a vast volume of information. It can be used to make assumptions regarding categorical class names, to classify knowledge on the basis of training sets and class labels, and to classify newly obtainable data. Fig. 2 illustrates the general structure of DT.

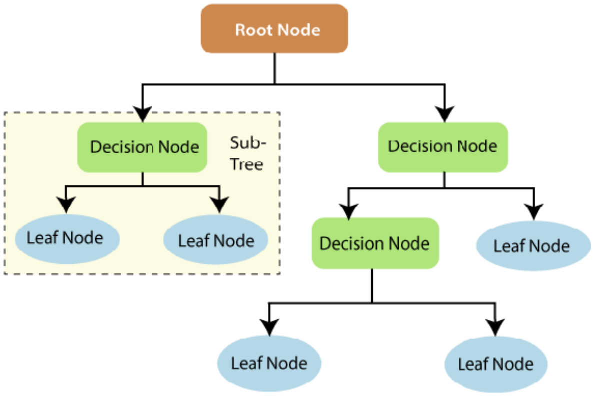


Fig. 2: General structure of Decision Tree [40].

Multi-Layer Perceptron

ANNs are structures inspired by brain performance. These networks can compute model performance estimation and manage non-linear and linear functions by learning from data generalizing and their relationships to unsighted situations. One of the most main ANNs is MLP. It is a potent modeling tool that exerts a supervised learning method using data samples with certain outputs. This method creates a non-linear function model that makes it possible to predict the output data from the given input data [41]. To comprehend MLP, a short description of single-layer perceptron (SLP) and single neuron perceptron has been prepared. The first type is the simplest ANN and has only one output to which all inputs are linked, and the values of x_i, w_i and y are inputs, weighting of the neuron and predictive binary class respectively, which are described in Fig. 3 of the steps of weighting, summation and transfer function.

Fig. 3: Perceptron steps: weighting, sum, and transfer steps.

Also, Fig. 4 shows its simplified model and the transfer function is calculated in (7).

$$y = f(z) \; and \; z = \sum_{i=0}^{n} w_i x_i \tag{7}$$

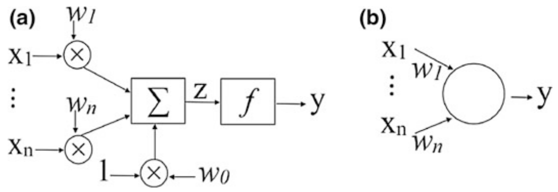


Fig. 4: Perceptron models: a) steps. b) Simplified.

$x_0=1$, y is the output and w_0 is the bias or threshold value. The transfer function has different forms such as unit step, linear, and sigmoid. Fig. 5 shows an example of the linear and nonlinear functions, which detaches the data into two classes. A Function can be represented by the dot product among the input and the weight vectors in (8).

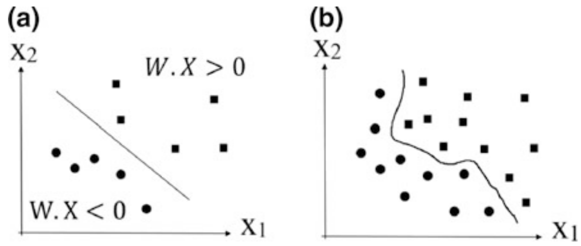


Fig. 5: Input patterns: a) linear. b) nonlinear.

$$\sum_{i=0}^{n} w_i x_i = 0 \tag{8}$$

Learning Automata Algorithm

Automatic learning is an easy model for adaptive decision-making in an anonymous stochastic ambiance. It is purported that its performance can be considered analogous to the learning processes exhibited by living organisms in similar environments. General instances of such positions are cases where an inexperienced person learns to perform the right motions or an individual who finds the best track from home to the office. The structure efforts various operations and chooses new operations based on the response of the environment to the past acts. The structure of such adaptive selection of operations and decisions is indicated by LA. The learning problem the appropriate operation is complicated by the verity that ambiance responses are not entirely reliable because they are stochastic and the corresponding probability distribution is anonymous [18], [42].

The learning method in the field of LA is as follows. Every time it cooperates with the environment, it automatically and stochastically selects an action based on a probability distribution. After the ambiance responds to a chosen action, it automatically updates its operation probability distribution. Then, a new operation is chosen according to the renovated eventuality distribution, and the solution of the environment is extracted for this act, and this method is rerun. The updated algorithm for the operation probability distribution is called the RLA [42]. LA is more practical and effective in discovering the exact best solutions for complicated optimization issues. The dimensions (DIMs) of the points are equal to the number of automata used in the LAA. In other words, for the N-DIM problem, this algorithm contains N automata [43].

Every automata is accountable for exploring one DIM and operates separately in the ambiance. The *i-th* LA can be defined as the model ⟨x_i, A_i, r, P_i, U⟩ where $x_i = \{x_i\}$ represents the set of possible positions in the *i-th* DIM. As well as, x_i is the next state in the DIM i ($x_i \in [x_{min,i}, x_{max,i}]$), the maximum and minimum amounts in DIM i are $x_{max,i}$ and $x_{min,i}$, respectively. In automatic learning, $A_i = \{a_{l,\eta}\}$ is the set of feasible operations that the LA can perform in the DIM i, $a_{l,\eta}$ demonstrates that an operation is right ($l=2$) or left ($l=1$) moves and η is the length of step. Note that r is a scalar value and represents a RL signal that is generated through the ambiance to demonstrate the quality of the movement x_i during the step in the selected route. As well as, Pi includes two possibilities p_1 and p_2. p_1 and p_2 respectively demonstrate the probability of choosing the right route and the left route in the *i-th* DIM. Suppose the right route is chosen, and the probability of selecting one cell among k cells located on the route determines the probability p_2. As well as, U is a procedure for calculating the eventualities of operations, P.

In the proposed method, each DIM is divided into D cells. This means that x_i is segmented into D subsets, with each subset containing all the dimensional states that are found within the respective cell. Thus, $D \times N$ cells are generated for the N-dimensional space of exploration where $\omega_{c,i}$ is a cell width in the DIM i and is computed using (9).

$$\omega_{c,i} = \frac{x_{max,i} - x_{min,i}}{D} \tag{9}$$

At the beginning of the operation exploration, it must be able to select one of two possible directions to appraise the selection of the best solution in the route. Therefore, the value of $L_2(x_i)$ is determined by the amounts of the k adjacent cells in the right route, where k is a predefined integer amount and $c_{i,j}$ is cell j in DIM i. As well as, j is computed by (10) and the amount of a route can be evaluated by (11).

$$j = floor\left(\frac{x_i - x_{min,i}}{\omega_{c,i}}\right) \tag{10}$$

$$L_l(x_i) = (1-\lambda_1)\sum_{m=1}^{k-1} \lambda_1^{m-1} v_{l,m}^* + \lambda_1^{k-1} v_{l,k}^* \qquad l = 1,2 \tag{11}$$

where $v_{l,m}^*$ represent the variable of the vector m that is placed in the direction of l. Also, λ_1 is computed with the conditions $0 \le \lambda_1 \le 1$ and $(1-\lambda_1)\sum_{m=1}^{k-1} \lambda_1^{m-1} + \lambda_1^{k-1} = 1$, provided that the relation $(1-\lambda_1)\lambda_1^{k-2} \ge \lambda_1^{k-1}$ is established. The two probabilities p_1 and p_2 are obtained from (12) and (13).

$$p_1(L_l(x_i)) = \frac{e^{\frac{L_l(x_i)}{\tau}}}{\sum_{s=1}^{2} e^{\frac{L_s(x_i)}{\tau}}} \qquad l = 1,2 \tag{12}$$

$$p_2(c_{i,j+s}) = \frac{e^{\frac{(V(x_i)|_{x_i \epsilon c_{i,j+s}})}{2\tau}}}{\sum_{z=1}^{k} e^{\frac{(V(x_i)|_{x_i \epsilon c_{i,j+z}})}{2\tau}}} ; l = 1,2 \; ; \; s = 1, \dots, k \tag{13}$$

where $V(x_i)$ is the cell value. The τ parameter makes a balance among search and utilization. With selecting a cell, the operation proceeds to the new cell with a step length that can be expressed in the act of η in (14). Thus, when L_1 is chosen, the current dimensional state of x_i changes to $x_i = x_i - \eta$ and when L_2 is selected, x_i moves to $x_i = x_i + \eta$.

$$\eta = \omega_{c,i}(\xi + \zeta) \; ; \; \zeta \in (0,1] \tag{14}$$

where the distance among the former cell and the chosen cell ζ and ξ is a stochastic number. Next, an amplification signal is applied to investigate the next state x_i. Just after the dimensional state x_i is transferred to x_i', the *i-th* variable of the current state $X(x_i)$ is changed by $X(x_i')$. According to (15), the amplification signal is allocated to cell $c_{i,j}$. The amplification signal is used to update the cell value $c_{i,j}$ and is obtained according to (16).

$$r\big(X(x_i')\big) = \begin{cases} 1, & if\ F(X(x_i')) \le F(X_{best}) \\ 0, & otherwise \end{cases} \tag{15}$$

$$\begin{aligned} V(x_i)|_{x_i \epsilon c_{i,j}} \leftarrow r\big(X(x_i)\big) + \alpha_1 V(x_i)|_{x_i \epsilon c_{i,j}} + \\ +(1-\alpha_1)\big((1-\lambda_2)L_{max}(x_i) + \lambda_2 L_{min}(x_i)\big) \end{aligned} \tag{16}$$

The solution is desirable when $r=1$ and $r=0$ indicates an unfavorable answer. Also, $L_{max}(x_i) = max\ \{L_1(x_i), L_2(x_i)\}$ and $L_{min}(x_i) = min\ \{L_1(x_i), L_2(x_i)\}$ are two estimated path values at x_i. $L_{max}(x_i)$ has a greater impression on the cell value than $L_{min}(x_i)$. Thus, the parameter λ_2 must be given in such a way that this relation $(1-\lambda_2) > \lambda_2$ is true. The weights α_1 and $(1-\alpha_1)$ show the impression of past evaluations and route values on the new evaluation, respectively. In equation (17), the relationship among X_{best} and X and is shown [44].

$$X_{best} \leftarrow \begin{cases} \begin{matrix} X(x_i'), X(x_i') = \\ = [x_i, \dots, x_{i-1}, x_i', x_{i+1}, \dots, x_N] \end{matrix} & if\ r = 1 \\ X_{best} & otherwise \end{cases} \tag{17}$$

Fuzzy Evaluation

The model proposed in this research employs a fuzzy inference system (FIS), which is an optimization method that receives distinct inputs and relates those inputs to output with some rules. The final output is obtained from the aggregated optimized result of the exclusive rule [45]. FIS is the process of formulating the mapping from a given input to an output using fuzzy logic. The mapping then prepares a basis from which decisions can be made, or patterns discerned. The FIS process in the Fuzzy Logic Toolbox includes the sections of fuzzing input variables, using a fuzzy operator (AND or OR) in the antecedent, the concept from antecedent to result, aggregating the consequences into rules, and fuzzification, which is also shown in Fig. 6. These sometimes cryptic and odd names have very specific meaning that is defined in the following steps.

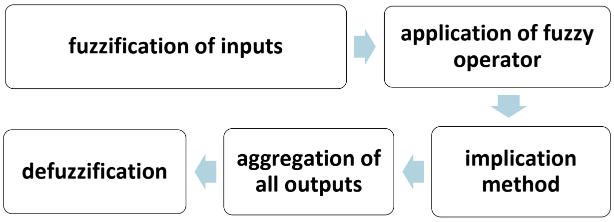


Fig. 6: Fuzzy evaluation steps.

Methodology

To increase the classification accuracy of complex problems, it is possible to use a combination of classifications that use the same learning algorithm but with different complexities and parameters. Hybrid classifiers use the fusion of several classifiers. In fact, these classifiers each build their own pattern on the data and save this model. Eventually, for the final classification, a vote is held between these classifications, and the class that gets the most votes will be the class that has had the greatest impact on the classification. In this work, we defined coefficients to weight the classifiers, and in order to achieve the best accuracy, we implemented voting and found the optimal coefficients by FLAA. We proceeded with this process in five steps. Fig. 7 shows the overall process.

In the first step, we created and stored Sonar data in six classes with specific DIMs and samples.

In the second step, we loaded those data into the introduced classification training algorithm and after running the algorithm, we saved the precision, recall, F1_score, AUC, and accuracy results of each of the classification models related to the Sonar data. Four classifiers (KNN, NB, DT, and MLP) were used in this research.

In the third step, the stored models and data were loaded into the FLAA.

In the fourth step, we created and integrated functions for weighting the categories.

In the fifth step, a fuzzy system was included in the algorithm to find optimal control parameters.

In the last step, to find the best accuracy answer with the majority vote, we ran the FLAA to find the optimal coefficients of the classifiers and saved the results.

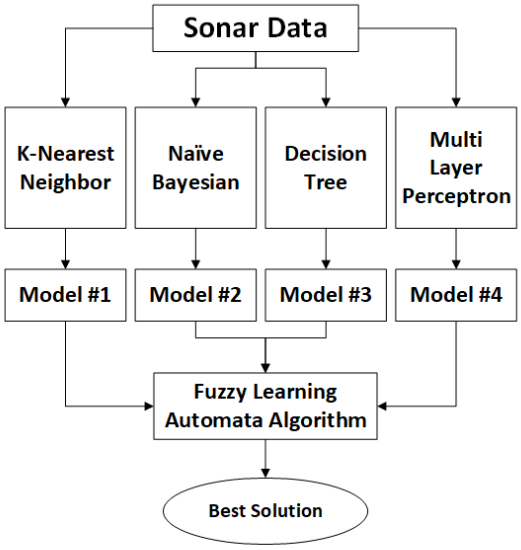


Fig. 7: The general process of the proposed method.

Data and Device

In this work, a dataset of Sonar targets with six different classes and DIM of 123x129 was used. The sonar datasets employed in this study are derived from micro-Doppler signatures and data collected through practical experimentation within a cavitation tunnel. These datasets are systematically organized utilizing a mathematical model that characterizes the return signal of the target propellers.

Also, these targets in different subclasses include different viewing angles and signal-to-noise ratios. The Specifications of the targets are demonstrated in Table 1.

Table 1: Specification of objects

Class Number	Name	Type of Application
1	logistic	Military
2	aircraft carrier	Military
3	Destroyer	Military
4	Landing ship	Military
5	Submarine	Military
6	Chinese oceanic	Tug boat

This program is implemented on a system with Intel® Core™ i7-6500U CPU (2.50-2.59) GHz processor specifications, 8 GB RAM, and MATLAB R2020b software.

Results and Discussion

In this study, we intend to investigate how selecting appropriate control parameters through a fuzzy system can improve the performance of classification combinations using LA. Also, to better examine the models' efficiency, the Accuracy, Precision, Recall, F1_Score, and AUC methods are reported in Tables 3 to 5.

The graphs of each model are also shown in Fig. 8 to 15.

These metrics are derived from the Confusion Matrix is given in Table 2.

Table 2: Confusion matrix for calculating evaluation criteria

		Predicted Class	
		Positive	Negative
Actual Class	Positive	True Positives (TP)	False Negatives (FN)
	Negative	False Positives (FP)	True Negatives (TN)

True Positive gives the count of predictions that belong to positive classes and are correctly identified. True negative gives the count of predictions that belong to the negative class and correctly classified as negative. False Positive gives the counts which are predicted are true but actually not true and vice versa for False Negative.

In (18), Precision represents the probability that the predicted category is consistent with the actual category. Recall represents the probability that the actual category is consistent with the predicted category in (19). Also, (20) represents the probability that Accuracy can predict the actual category. And the criterion F1-Score represents the harmonic results between Precision and Recall in (21).

$$Precision = \frac{TP}{TP + FP} \tag{18}$$

$$Recall = \frac{TP}{TP + FN} \tag{19}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{20}$$

$$F1 - Score = \frac{2 * Precision * Recall}{Precision + Recall} \tag{21}$$

Initial Classification

In the first model, the data was trained by a KNN classifier with a nearest neighbor rate of 5. The results obtained from this run are listed in Table 3. Fig. 8 and Fig. 9 show the performance of Model 1 on Sonar data with confusion matrix and ROC plots for 6 different classes.

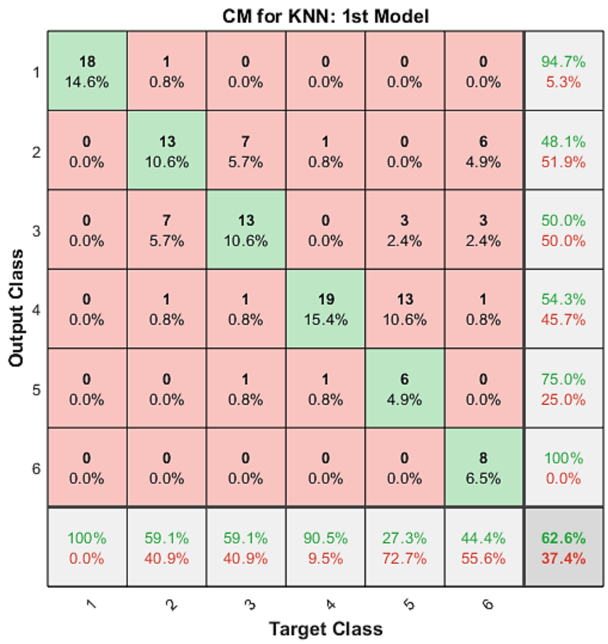


Fig. 8: Confusion matrix chart for KNN - 1st Model.

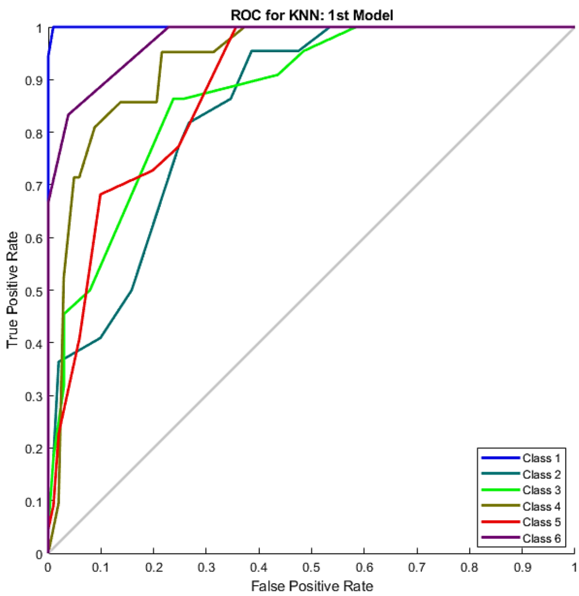


Fig. 9: ROC chart for KNN - 1st Model.

In the second model, the data was trained by an NB classifier. The results obtained from this run are listed in Table 4. Fig. 10 and Fig. 11 show the performance of Model 2 on Sonar data with a confusion matrix and ROC plots for 6 different classes.

Table 3: KNN model performance results

Classes	Precision (%)	Recall (%)	F1_Score (%)	AUC	Accuracy (%)
1	94.73	100	97.29	0.99	62.60
2	48.14	59.09	53.06	0.84	
3	50	59.09	54.16	0.87	
4	54.28	90.47	67.85	0.93	
5	75	27.27	40	0.88	
6	100	44.44	61.53	0.97	

Table 4: NB model performance results

Classes	Precision (%)	Recall (%)	F1_Score (%)	AUC	Accuracy (%)
1	78.26	100	87.80	0.98	66.66
2	54.54	27.27	36.36	0.83	
3	45.71	72.72	56.14	0.88	
4	64.28	85.71	73.46	0.95	
5	91.66	50	64.70	0.89	
6	92.85	72.22	81.25	0.98	

Table 5: DT model performance results

Classes	Precision (%)	Recall (%)	F1_Score (%)	AUC	Accuracy (%)
1	100	100	100	1	88.61
2	76.92	90.90	83.33	0.98	
3	82.60	86.36	84.44	0.98	
4	94.73	85.70	90	0.99	
5	86.95	90.90	88.88	0.98	
6	100	77.77	87.5	0.99	

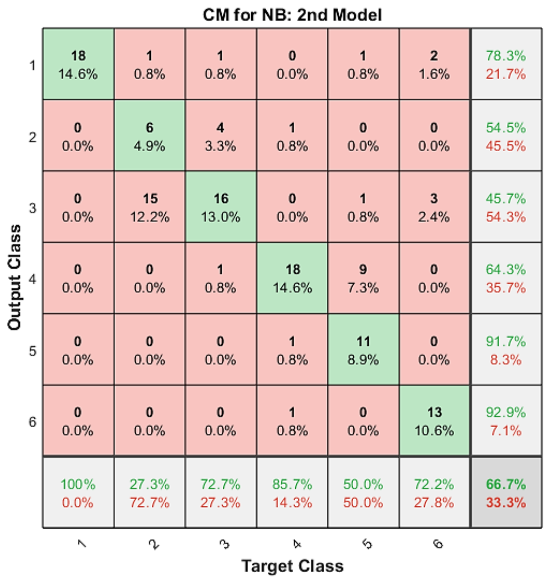


Fig. 10: Confusion matrix chart for NB - 2nd Model.

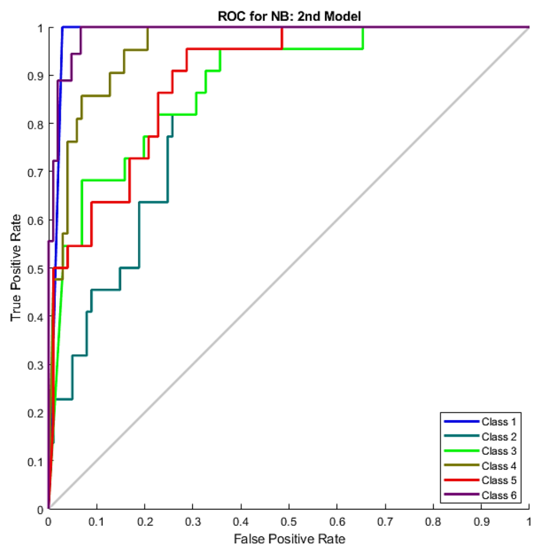


Fig. 11: ROC chart for NB - 2nd Model.

In the third model, the data was trained by a DT classifier. The results obtained from this run are listed in Table 5. Fig. 12 and Fig. 13 show the performance of Model 3 on Sonar data with confusion matrix and ROC plots for 6 different classes.

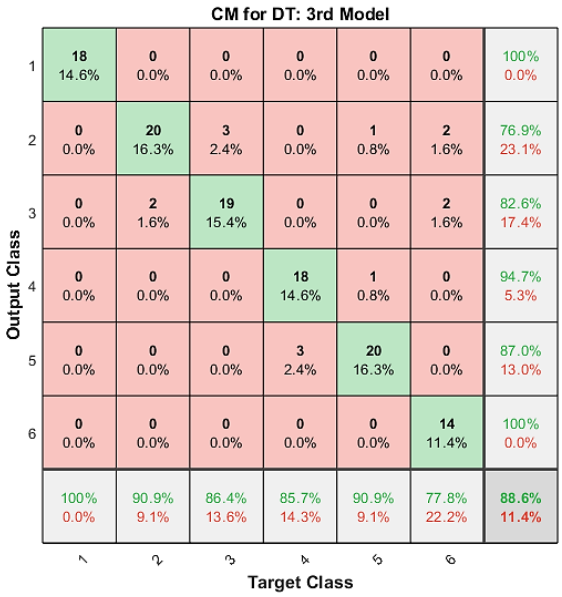


Fig. 12: Confusion matrix chart for DT - 3rd Model.

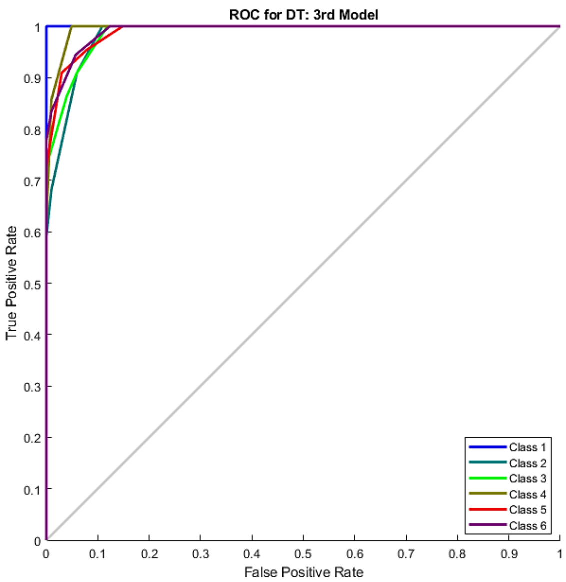


Fig. 13: ROC chart for DT - 3rd Model.

In the fourth model, the data was trained by an MLP classifier.

The results obtained from this run are listed in Table 6. Fig. 14 and Fig. 15 show the performance of Model 4 on Sonar data with confusion matrix and ROC plots for 6 different classes.

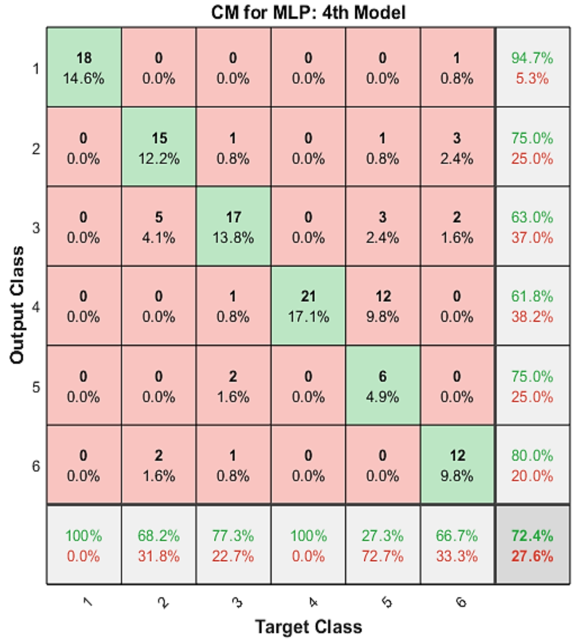


Fig. 14: Confusion matrix chart for MLP - 4th Model.

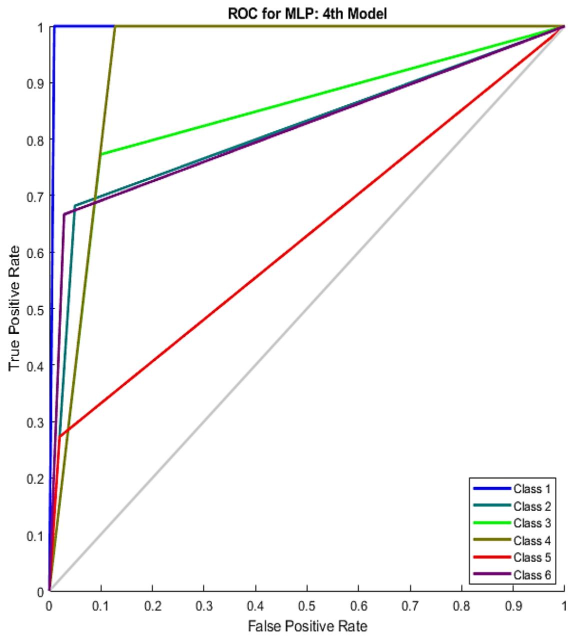


Fig. 15: ROC chart for MLP - 4th Model.

Fuzzy System Specifications

According to the explanations provided in the previous section, Fig. 16 presents a general diagram of the fuzzy system used in the LA algorithm, where the parameters (Nfe(max) and BestNfe) constitute the inputs and the parameters K and D constitute the outputs of this system.

Also, Table 7 shows the rules used in this system and the specifications of the parameters.

Table 6: MLP model performance results

Classes	Precision (%)	Recall (%)	F1_Score (%)	AUC	Accuracy (%)
1	94.73	100	97.29	0.99	
2	75	68.18	71.42	0.81	
3	62.96	77.27	69.38	0.83	72.35
4	61.76	100	76.36	0.93	
5	75	27.27	40	0.62	
6	80	66.66	72.72	0.81	

Table 7: FLAA rules specifications

Number		BestNfe		Nfe(max)		K		D
1		Good		Primary		Low		Low
2		Good		Middle		Low		Low
3	If	Good	And	Final	Then	Low	And	Low
4		Bad		Primary		Low		High
5		Bad		Middle		High		Low
6		Bad		Final		High		High

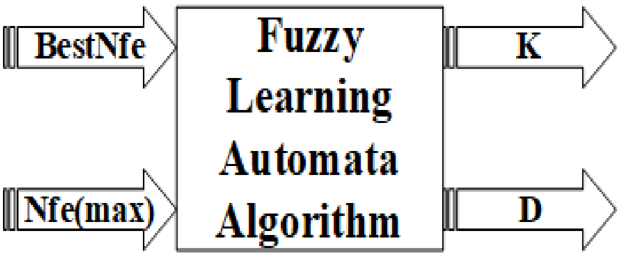


Fig. 16: general diagram of the fuzzy system.

The three-dimensional phase diagram of K, Nfe(max) and BestNfe is shown in Fig. 17 and the three-dimensional phase diagram of D, Nfe(max) and BestNfe is shown in Fig. 18.

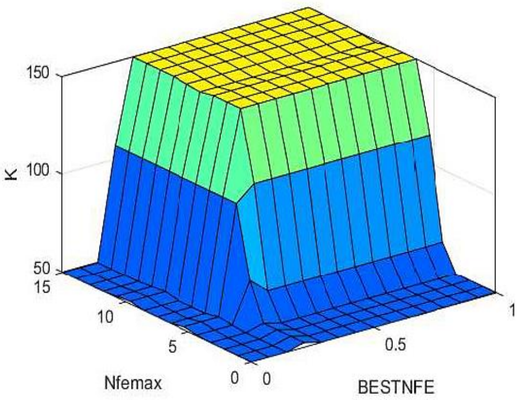


Fig. 17: The 3D phase diagram of K, Nfe(max) and BestNfe.

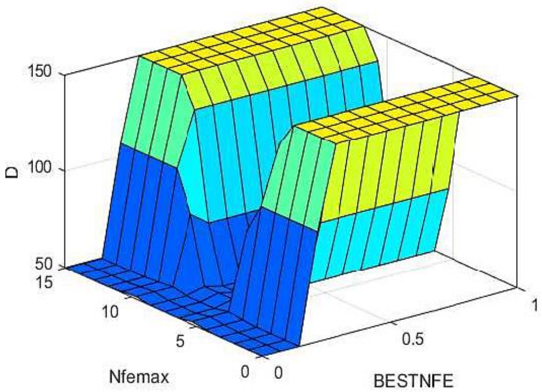


Fig. 18: The 3D phase diagram of D, Nfe(max) and BestNfe.

Fusion Operation

As described in the work process in the previous sections. The stored models of each class are weighted using the FLAA and weighted summation functions in the defined range. To achieve the best accuracy and decision by obtaining the best solutions for the classifications and fusion it by the FLAA. Due to the fact that in this process the effective parameters in the FLAA are very effective. The results of Accuracy, Precision, Recall, F1_Score, and AUC are reported for the K, D, and Nfe(max) fuzzy parameters.

Also, for better comparison, the fusion operation has been performed with a conventional LA algorithm, the performance results of which are reported in Table 8. The complexity matrix and ROC plots of this performance for the values of K = 200, D = 200, and Nfe(max) = 15 are shown in Fig. 19 and Fig. 20.

Table 8: LA performance results for selected weights [w1:2, w2:2, w3:4, w4:3]

Classes	Precision (%)	Recall (%)	F1_Score (%)	AUC	Accuracy (%)
1	100	94.73	97.29	0.97	
2	90.90	83.33	86.95	0.90	
3	95.45	80.76	87.50	0.89	85.36
4	95.23	71.42	81.63	0.85	
5	59.09	100	74.28	0.95	
6	72.22	100	83.87	0.97	

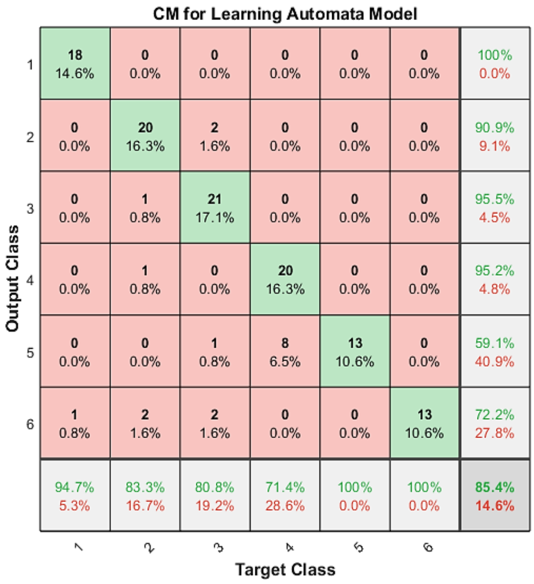


Fig. 19: Confusion matrix chart in LA.

In Table 9, the results are reported by changing the values of weighted of classifiers parameters. Also, the performance of Sonar DF by the FLAA is shown in Fig. 21 and Fig. 22.

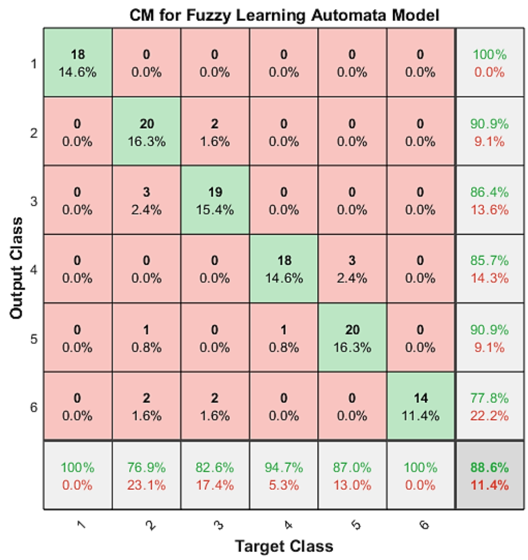


Fig. 21: Confusion matrix chart in FLAA.

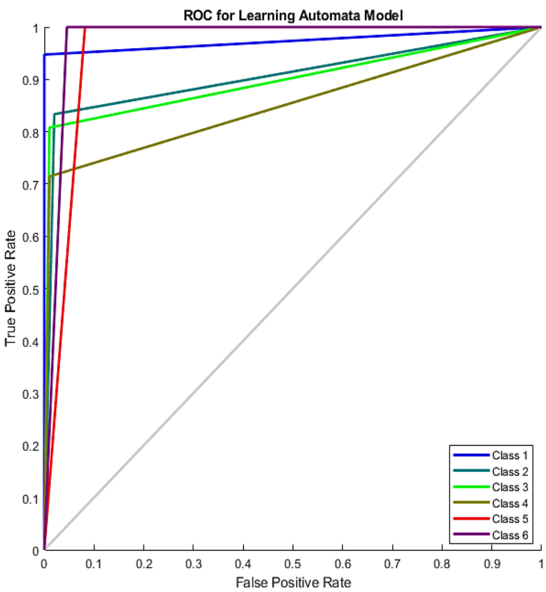


Fig. 20: ROC chart in LA.

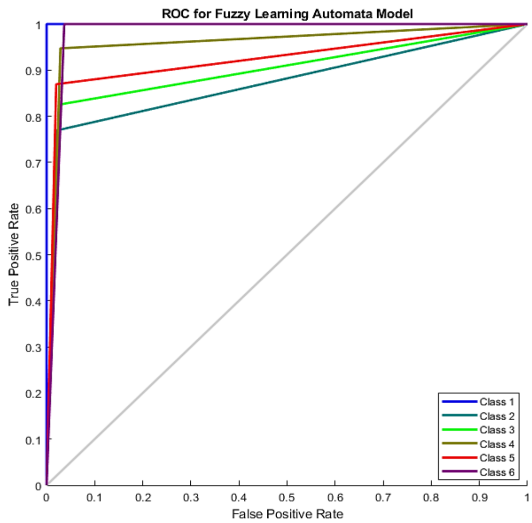


Fig. 22: ROC chart in FLAA.

Table 9: FLAA performance results for selected weights [w1:1, w2:3, w3:5, w4:5]

Classes	Precision (%)	Recall (%)	F1_Score (%)	AUC	Accuracy (%)
1	100	100	100	1	
2	90.90	76.92	83.33	0.87	
3	86.36	82.60	84.44	0.89	88.61
4	85.71	94.73	90	0.95	
5	90.90	86.95	88.88	0.92	
6	77.77	100	87.50	0.98	

Table 10: Performance comparison of conventional and fused classification models

No.	Model	Precision (%)	Recall (%)	F1_Score (%)	Accuracy (%)
1	SVM	71.4	70	70	83.9
2	RF	70	77.78	73.68	76.19
3	DT	90	75	81.81	80.95
4	XGboost	80	80	80	80.95
5	Ensemble Method	60	75	66.67	71.45
6	R-EFMD	79.27	76.5	77.86	78.25
7	T-EFMD	79.51	81.5	80.49	80.25
8	R-LFMD	78.82	80	79.4	79.25
9	T-LFMD	83.17	86.5	84.8	84.5
10	ANN	63.71	64.58	64.14	65.57
11	CNN	78.47	79.39	78.92	65.57
12	TFIR-DCNN+SA	73.55	66.14	69.65	66.14
13	Joint	79.5	80.12	79.49	79.8
14	**FLAA (proposed)**	**88.6**	**90.2**	**89.02**	**88.6**

Table 11: Comparison of the performance of the combined models with LAA and FLAA.

No.	Model	Precision (%)	Recall (%)	F1_Score (%)	Accuracy (%)
1	LAA	85.48	88.37	85.25	85.36
2	**FLAA (proposed)**	**88.6**	**90.2**	**89.02**	**88.6**

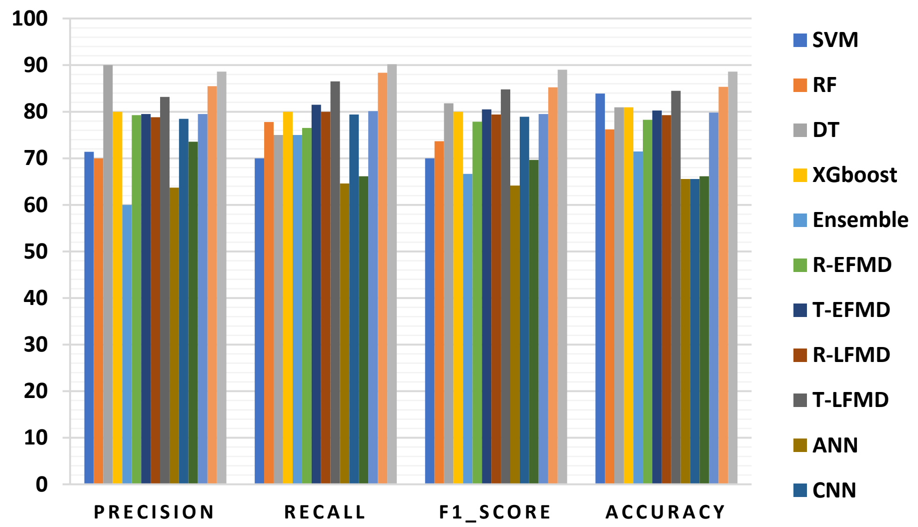


Fig. 23: Functional comparison of Precision, Recall, F1_Score, and Accuracy parameters.

Table 10 shows the important parameters and computational requirements of introduced models such as SVM, RF, DT, XGboost, ensemble method, R-EFMD, T-EFMD, R-LFMD, T-LFMD, ANN, CNN, TIFR-DCNN+SA, and joint. The results of these models have been compared with the proposed model. Considering that the objectives and databases are different, we benchmarked the average detection rate. In this comparison, Precision, Recall, F1_Score, and Accuracy parameters have been considered and investigated to show the superior performance of the proposed method compared to other procedures. Table 11 highlights that the most significant finding is the superior performance of the FLAA method relative to the prior approach that did not incorporate fuzzy control. This enhancement in performance provides strong evidence for the efficacy of employing an intelligent system. Notably, even though the complexity of the problem was increased by expanding the number of classes, the proposed method consistently outperformed all alternative methods. Also, in Fig. 23, the graph of this comparison is illustrated to show the results of each of the models side by side, and the optimal performance of the DF method with the other algorithms is quite evident.

Conclusion

This study addresses the challenge of integrating classification results derived from FLAA and Sonar datasets. The Sonar dataset, comprising six distinct target categories characterized by varying capabilities and specifications, was analyzed utilizing the FLAA approach. The detection of targets in marine environments is complicated by sound wave interference and ambient noise, which pose significant obstacles. Traditionally, classification of such data has been performed manually, a process prone to a high likelihood of target misidentification. The application and combination of machine learning techniques offer the potential to enhance target detection accuracy. In this research, four classification models were initially applied independently to the Sonar data. Subsequently, these classifiers were integrated using the LAA to optimize performance by determining the optimal weighting coefficients for each classifier. This combined approach yielded results that surpass those reported in comparable studies. Performance metrics including Precision, Recall, F1_Score, and Accuracy were evaluated and compared against recent literature, with the proposed method achieving values of 88.6%, 88.53%, 90.2%, and 88.6%, respectively. Furthermore, the study demonstrates the superior efficacy of the proposed method relative to the conventional LA-based Sonar DF approach.

The main advantage of this work was to provide a soft computing method for dealing with sonar data. In this method, unlike LAA, the parameters are not controlled manually but are controlled intelligently using a fuzzy controller. This innovation prevents LAA from getting stuck in local solutions in the search space. In this way, we were able to improve the performance in terms of both speed and accuracy. Furthermore, numerous parameters within the LAA exhibit stochastic variability and lack adaptive intelligence. To address this, we enhanced these parameters by integrating fuzzy systems, thereby facilitating more rapid and improved convergence.

Some limitations can be mentioned. The proper setting of FLAA rules is the proper selection of basic classifiers and the existence of appropriate databases for training basic classifiers. In the future, it is possible to perform tasks such as optimizing LA control parameters using meta-heuristic methods for better convergence, using intelligent methods for optimal parameter selection, and using the proposed method in dealing with incomplete and missing databases. This also adds some complexity to our system.

Previously, it was just an LAA, but now a fuzzy system has been added to it, and we may have to pay more for implementation.

We recommend that, despite the implementation of a fuzzy system in this study, certain components of the system—specifically the membership functions and the rule base—may not have been optimized. Future work should focus on refining these elements to enhance system performance.

Author Contributions

Sajjad Mahmoudi Khah simulated the proposed method in MATLAB. Seyed Hamid Zahiri and Iman Behravan supervised and consulted in this research's design, implementation, and results. All authors discussed important sections and contributed to the final text.

Acknowledgment

We sincerely thank the respected referees for their accurate review of this paper. Also, we sincerely thank the ICT Research Institute and Connectivity and Communication Technologies Development Headquarters and Dr. Kharrat.

Funding

This research received no external funding.

Conflict of Interest

The authors declare no potential conflict of interest regarding the publication of this work. In addition, the ethical issues including plagiarism, informed consent, misconduct, data fabrication and, or falsification, double publication and, or submission, and redundancy have been completely witnessed by the authors.

Abbreviations

SONAR	Sound and Range Navigation
DF	Data Fusion
RLA	Reinforcement Learning Algorithm
CNN	Convolutional Neural Network
AI	Artificial Intelligence
ML	Machine Learning
SVM	Support Vector Machine
KNN	K-Nearest Neighbor
RF	Random Forest
DT	Decision Tree
XGboost	Extreme Gradient Boosting
R-EFMD	Early Fusion Model with Resnet
T-EFMD	Early Fusion Model with Swin Transformer
R-LFMD	Late Fusion Model with Resnet
T-LFMD	Late Fusion Model with Swin Transformer
SA	Spatial Attention
TFIR	Time Frequency Image Recognition
DCNN	Deep CNN
SLP	Single-Layer Perceptron
MLP	Multi-Layer Perceptron
NB	Naïve Bayesian
LA	Learning Automata
DIM	Dimension
LAA	Learning Automata Algorithm
FLAA	Fuzzy Learning Automata Algorithm
FIS	Fuzzy Inference System
AUC	Area Under the ROC Curve
CM	Confusion Matrix
ROC	Receiver Operating Characteristic

References

[1] W. Tang, "Highlight model of echoes from sonar targets," Acta Acustica, 19(2): 92-100, 1994.

[2] C. Erbe, S. A. Marley, R. P. Schoeman, J. N. Smith, L. E. Trigg, C. B. Embling, "The effects of ship noise on marine mammals—a review," Front. Mar. Sci., 6: 606, 2019.

[3] X. Luo, Y. Feng, "An underwater acoustic target recognition method based on restricted Boltzmann machine," Sensors, 20(18): 5399, 2020.

[4] L. Zhang, D. Wu, X. Han, Z. Zhu, "Feature extraction of underwater target signal using mel frequency cepstrum coefficients based on acoustic vector sensor," J. Sensors, 2016(1): 7864213, 2016.

[5] J. Abeßer, "A review of deep learning based methods for acoustic scene classification," Appl. Sci. 10(6), 2020.

[6] I. J. Ananya, S. Suad, S. H. Choudhury, M. A. Khan, "A comparative study on approaches to acoustic scene classification using CNNs," in Proc. Mexican International Conference on Artificial Intelligence: 81-91, 2021.

[7] M. Yu et al., "Spatiotemporal event detection: A review," Int. J. Digital Earth, 13(12): 1339-1365, 2020.

[8] Z. Bai, X.-L. Zhang, "Speaker recognition based on deep learning: An overview," Neural Networks, 140: 65-99, 2021.

[9] R. M. Hanifa, K. Isa, S. Mohamad, "A review on speaker recognition: Technology and challenges," Comput. Electr. Eng., 90: 107005, 2021.

[10] S. Shen, H. Yang, X. Yao, J. Li, G. Xu, M. Sheng, "Ship type classification by convolutional neural networks with auditory-like mechanisms," Sensors, 20(1): 253, 2020.

[11] M. Hassaballah, A. I. Awad, Deep learning in computer vision: principles and applications. crc Press, 2020.

[12] W. Xiong, X. Gu, Y. Cui, "Tracking and data association based on reinforcement learning," Electron., 12(11): 2388, 2023.

[13] M. Valdenegro-Toro, "Best practices in convolutional networks for forward-looking sonar image recognition," in Proc. OCEANS 2017-aberdeen: 1-9, 2017.

[14] E. L. Ferguson, R. Ramakrishnan, S. B. Williams, C. T. Jin, "Convolutional neural networks for passive monitoring of a shallow water environment using a single sensor," in Proc. 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP): 2657-2661, 2017.

[15] G. Huo, Z. Wu, J. Li, "Underwater object classification in sidescan sonar images using deep transfer learning and semisynthetic training data," IEEE Access, 8: 47407-47418, 2020.

[16] F. Mousavipour, M. R. Mosavi, "Sonar data classification using neural network trained by hybrid dragonfly and chimp optimization algorithms," Wireless Pers. Commun., 129(1): 191-208, 2023.

[17] B. N. K. Reddy, C. A. Vaithilingam, S. Kamalakkannan, "SONAR based under water mine detection using machine learning algorithms," in Proc. 2024 4th International Conference on Innovative Practices in Technology and Management (ICIPTM): 1-4, 2024.

[18] S. Mahmoudikhah, S. Zahiri, I. Behravan, "Fusion of classifiers using learning automata algorithm," J. Electr. Comput. Eng. Innovations, 13(1): 65-80, 2025.

[19] Q. Wang, S. Du, W. Zhang, F. Wang, "Active sonar target recognition method based on multi-domain transformations and attention-based fusion network," IET Radar Sonar Navig., 18(10): 1814-1828, 2024.

[20] S. Z. Tian, D. B. Chen, Y. Fu, J. L. Zhou, "Joint learning model for underwater acoustic target recognition," Knowledge-Based Syst., 260: 110119, 2023.

[21] H. Yang, Y. Huang, Y. Liu, "Spatial attention deep convolution neural network for call recognition of marine mammal," in Proc. International Conference on Autonomous Unmanned Systems: 2725-2733, 2022.

[22] F. Ahmad, M. Z. Ansari, R. Anwar, B. Shahzad, A. Ikram, "Deep learning based classification of underwater acoustic signals," Procedia Comput. Sci., 235: 1115-1124, 2024.

[23] A. K. Bathla, S. Bansal, M. Kumar, "OKC classifier: an efficient approach for classification of imbalanced dataset using hybrid methodology," Soft Comput., 26(21): 11497-11503, 2022.

[24] K. Taunk, S. De, S. Verma, A. Swetapadma, "A brief review of nearest neighbor algorithm for learning and classification," in Proc. 2019 International Conference on Intelligent Computing and Control Systems (ICCS): 1255-1260, 2019.

[25] D. Wettschereck, T. Dietterich, "Locally adaptive nearest neighbor algorithms," Adv. Neural Inf. Process. Syst., 6, 1993.

[26] D. A. Berry, D. A. Berry, Statistics: a Bayesian perspective. Duxbury Press Belmont, CA, 1996.

[27] D. Berrar, "Bayes' theorem and naive Bayes classifier," 2019.

[28] D. Berrar, "An empirical evaluation of ranking measures with respect to robustness to noise," J. Artif. Intell. Res., 49: 241-267, 2014.

[29] D. Berrar, P. Flach, "Caveats and pitfalls of ROC analysis in clinical microarray research (and how to avoid them)," Briefings Bioinf., 13(1): 83-97, 2012.

[30] T. Hastie, R. Tibshirani, J. H. Friedman, J. H. Friedman, The elements of statistical learning: data mining, inference, and prediction. Springer, 2009.

[31] Y. Y. Song, L. Ying, "Decision tree methods: Applications for classification and prediction," Shanghai Arch. Psychiatry, 27(2): 130, 2015.

[32] J. R. Quinlan, "Learning decision tree classifiers," ACM Comput. Surv. (CSUR), 28(1): 71-72, 1996.

[33] F. J. Yang, "An extended idea about decision trees," in Proc. 2019 International Conference on Computational Science and Computational Intelligence (CSCI): 349-354, 2019.

[34] J. Liang, Z. Qin, S. Xiao, L. Ou, X. Lin, "Efficient and secure decision tree classification for cloud-assisted online diagnosis services," IEEE Trans. Dependable Secure Comput., 18(4): 1632-1644, 2019.

[35] A. Brifcani, A. Issa, "Intrusion detection and attack classifier based on three techniques: A comparative study," Eng. Tech. J., 29(2): 368-412, 2011.

[36] A. S. Eesa, Z. Orman, A. M. A. Brifcani, "A novel feature-selection approach based on the cuttlefish optimization algorithm for intrusion detection systems," Expert Syst. Appl., 42(5): 2670-2679, 2015.

[37] A. Suresh, R. Udendhran, M. Balamurgan, "Hybridized neural network and decision tree based classifier for prognostic decision making in breast cancers," Soft Comput., 24(11): 7947-7953, 2020.

[38] Priyanka, D. Kumar, "Decision tree classifier: A detailed survey," Int. J. Inf. Decis. Sci., 12(3): 246-269, 2020.

[39] A. S. Eesa, A. M. Abdulazeez, Z. Orman, "A dids based on the combination of cuttlefish algorithm and Decision Tree," Sci. J. Univ. Zakho, 5(4): 313-318, 2017.

[40] C. Z. Janikow, "Fuzzy decision trees: Issues and methods," IEEE Trans. Syst., Man Cybern. Part B Cybern., 28(1): 1-14, 1998.

[41] C. M. Bishop, Neural networks for pattern recognition. Oxford university press, 1995.

[42] M. A. Thathachar, P. S. Sastry, Networks of learning automata: Techniques for online stochastic optimization. Springer Science & Business Media, 2011.

[43] Q. Wu, H. Liao, "Function optimisation by learning automata," Inf. Sci., 220: 379-398, 2013.

[44] N. S. Shahraki, S. H. Zahiri, "DRLA: Dimensionality ranking in learning automata and its application on designing analog active filters," Knowledge-Based Syst., 219: 106886, 2021.

[45] I. Ahmed, I. Sultana, S. K. Paul, A. Azeem, "Employee performance evaluation: A fuzzy approach," Int. J. prod. Perform. Manag., 62(7): 718-734, 2013.

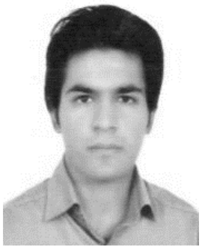
Biographies

Sajjad Mahmoudi Khah received the B.Sc. in Electrical Engineering, and Electronic from University of Sajjad in 2015 and M.Sc. in Electrical Engineering, and electronic integrated circuits from University of Birjand in 2018. He is currently a doctoral student in electrical engineering majoring in electronics at Birjand University, Birjand, Iran. His research interest includes soft computing, machine learning, circuit optimization, and swarm intelligence.

- Email: sajjadmahmoudikhah@birjand.ac.ir
- ORCID: 0009-0001-2184-2544
- Web of Science Researcher ID: NA
- Scopus Author ID: NA
- Homepage: NA

Seyed Hamid Zahiri received the B.Sc., M.Sc., and Ph.D. degrees in Electronics Engineering respectively from Sharif University of Technology in 1993, Tarbiat Modarres University in 1995, and Ferdowsi University of Mashhad in 2005. Currently, he is Professor with the Department of Electronics Engineering, University of Birjand, Birjand, Iran. His research interests include pattern recognition, evolutionary algorithms, swarm intelligence algorithms, and soft computing.

- Email: hzahiri@birjand.ac.ir
- ORCID: 0000-0002-1280-8133
- Web of Science Researcher ID: NA
- Scopus Author ID: NA
- Homepage: https://cv.birjand.ac.ir/zahiri/fa



Iman Behravan received the B.Sc. in Electronics Engineering from Shahid Bahonar University of Kerman, Kerman, Iran. Also, he received his M.Sc., Ph.D., and post-doctoral degrees from the University of Birjand, Birjand, Iran. His research interests include big data analytics, pattern recognition, machine learning, swarm intelligence, and soft computing.

- Email: i.behravan@birjand.ac.ir
- ORCID: 0000-0003-0319-1765
- Web of Science Researcher ID: NA
- Scopus Author ID: 5326-2017
- Homepage: https://scholar.google.com/citations?user=w9GKiVcAAAAJ&hl=en