Journal of Global Trends in Social Science

https://doi.org/10.70731/r97znf68

Machine Learning-Based Search Strategy for Water Object Retrieval in Cultural Tourism Safety Contexts

Fujiang Yuan [a], Chenxi Ye [b,*], Aichun Huang [c], Kai Xiao [b], Jiahao Qin [b], Xijun Chen [b]

[a] *School of Computer Science and Technology, Taiyuan Normal University, Jinzhong, 030619, Shanxi, China*
[b] *Faculty of Science and Technology,Hong Kong Baptist University ,Hong Kong, 999077 ,China*
[c] *Sino-German College of Intelligent Manufacturing, Shenzhen Technology University, ShenZhen,518118 ,China*

KEYWORDS

Cultural and Tourism Safety;
Machine Learning;
XGBoost;
Object Search

ABSTRACT

This research addresses the challenge of predicting deviations in the landing positions of objects dropped into water, with important implications for cultural tourism safety near lakes, rivers, and other natural attractions. An innovative optimization method for search strategies based on machine learning is proposed. A simulated dataset incorporating features such as drop height, water entry angle, drag coefficient, and object density enables detailed model comparisons. Five machine learning models—XGBoost, Random Forest, Decision Tree, Support Vector Machine (SVM), and Multi-Layer Perceptron (MLP)—are evaluated using Mean Squared Error (MSE), Mean Absolute Error (MAE), and the Coefficient of Determination. Experimental results show that XGBoost significantly outperforms the others, effectively capturing complex nonlinear relationships through its gradient boosting mechanism. In contrast, models like Decision Tree, SVM, and MLP exhibit lower predictive accuracy due to weaker generalization capabilities. This study provides a robust machine learning-based framework to enhance predictive accuracy and search efficiency in aquatic environments.

Introduction

With the rapid expansion of China's tourism industry, renowned tourist cities such as Hangzhou have witnessed a significant surge in visitor numbers. As a key component of the "Paradise on Earth," Hangzhou, with its unique tourism resources, welcomed over ten million visitors during the May Day Golden Week, with nearly 70% being interregional tourists. While this tourism boom has driven local economic growth, it has also given rise to various management and service challenges. In highly crowded tourist environments, accidental water drops of personal belongings frequently occur as visitors enjoy the scenery and leisure activities. In particular, the unintentional dropping of valuable items, such as smartphones, has become a notable social concern, highlighting the growing importance of cultural tourism safety. Ensuring the rapid and effective retrieval of such items is not only crucial for enhancing the tourist experience but also for maintaining the safety reputation of popular tourist destinations.[1].

* Corresponding author. E-mail address: u430233106@mail.uic.edu.cn

Received 10 April 2025; Received in revised form 25 April 2025; Accepted 28 April 2025; Available online 30 April 2025.

In response to the challenges associated with recovering lost items in aquatic environments, numerous scenic areas have begun to utilize specialized underwater retrieval devices that are designed to facilitate the prompt recovery of submerged objects [2]. It is important to recognize, however, that the path taken by these submerged items is subject to a variety of influences. Key factors that affect their trajectory include the physical characteristics of the objects themselves, such as their density, shape, and mass, as well as the environmental conditions of the water body in which they are located. For instance, elements like flow velocity, water drag, and prevailing weather conditions can all significantly impact the drift of these objects. As such, accurately predicting the potential drift range of lost items and creating effective strategies for searching them poses considerable technical difficulties in the realm of retrieval operations [3]. To address these issues and contribute to enhanced cultural tourism safety, this study aims to tackle these challenges by combining physical modeling techniques with data-driven methodologies to predict the drifting trajectories of submerged objects more accurately. Furthermore, it seeks to leverage cutting-edge intelligent algorithms, including those based on deep learning and reinforcement learning, to refine and enhance search strategies for locating these lost items. By engaging in empirical research focused on the movement behaviors of objects in the complex water conditions present in the scenic areas of Hangzhou, this research endeavors to create a comprehensive retrieval decision support model. This model is intended to provide valuable scientific guidance and practical operational insights that can be employed in real-world search and retrieval operations, ultimately improving the efficiency and effectiveness of recovery efforts in aquatic settings, and promoting a more secure and reassuring cultural tourism environment for both tourists and site managers alike [4].

Related Work

The task of locating objects in aquatic environments has long faced technical bottlenecks. Traditional retrieval operations mainly rely on manual observation and empirical analysis, which suffer from significant drawbacks such as high resource consumption and low positioning accuracy, often leading to inefficient searches and potential economic losses. To address this challenge, recent years have seen several innovative research advancements in the field of information retrieval. Anari et al. [5] integrated learning automata with swarm intelligence algorithms, optimizing search quality through ant colony behavior simulation. Wu et al. [6] developed an intelligent prediction model to solve bulk multi-item ordering problems, enhancing decision-making efficiency by combining machine learning with operational research methods.Furthermore, a series of pioneering studies have made breakthroughs in text information processing and recommendation algorithms. Notable works include the generalized nearest-neighbor retrieval framework proposed by Chen et al. [7], the personalized retrieval system based on graph contrastive learning by Li et al. [8], the adaptive k-nearest neighbor algorithm by Yadav et al. [9], and the intelligent clustering detection architecture designed by Shah et al. [10]. However, it is worth noting that most existing algorithmic frameworks are primarily designed for structured data and high-dimensional feature spaces, whereas aquatic environments exhibit significantly different dynamic characteristics. The complex interplay of water flow, drag effects, and sedimentation dynamics introduces strong nonlinearities in the movement trajectories of submerged objects. This unique setting makes it difficult for traditional data clustering methods and indexing optimization techniques to construct effective motion prediction models.

In order to meet the unique requirements of aquatic operations, this study introduces a groundbreaking solution that merges physical modeling with advanced intelligent algorithms. By integrating techniques such as XGBoost, deep neural networks, and ensemble learning, the research establishes a robust hydrodynamic feature learning model. This innovative model utilizes real-time environmental parameters to enhance the process of dynamic path planning. Unlike traditional manual search strategies that heavily depend on subjective experience, this data-driven approach excels in accurately capturing the complexities of fluid dynamics. Consequently, it is capable of generating precise predictions regarding optimal search areas, which in turn leads to a remarkable increase in the efficiency of object retrieval. Furthermore, this method not only streamlines the search process but also minimizes overall resource consumption, highlighting the advantages of employing a systematic, algorithm-based strategy in aquatic environments. By improving the accuracy and speed of retrieval operations, the proposed approach also contributes to a more responsive and intelligent cultural tourism safety management system, particularly in high-traffic scenic spots where accidental water drops are frequent..

Machine Learning-Based Search Strategy for Dropped Objects in Water Bodies

This research utilizes a modeling strategy based on data, incorporating machine learning techniques to forecast the movement of objects that have fallen into water settings. By assessing the predictive capabilities of various algorithms regarding object displacement, the research seeks to establish a solid foundation for effectively locating waterborne objects in real-life search and retrieval operations.

Simulation Dataset Construction

In this study, the dataset for dropped objects in water bodies is generated through randomized simulation of physical parameters, serving as training and evaluation data for machine learning models. Each data entry in the dataset represents a simulated object drop event and includes the following input features and output variables[11].

In this study, the motion trajectory of objects after falling into water is primarily influenced by gravity, fluid resistance, water entry angle, and object properties [12] (such as density and shape). Due to the complexity of water bodies, precise modeling typically involves numerical simulations of fluid dynamics, such as the Navier-Stokes equations. However, solving these high-order differential equations is computationally expensive and complex. Therefore, this study adopts a simplified physical modeling approach, assuming a static water environment to model the object's descent process, leading to the following trajectory calculation formulas:

Table 1 | Dataset Field Descriptions

Variable Name	Description	Unit
Height	Drop height	m
Angle		°
Water Resistance	Water resistance coefficient	-
Density	Object density	g/cm³
X_{drift}	Horizontal displacement	m
Y_{drift}	Settling depth	m

$$X_{drift} = \frac{H.sin(\theta).W}{Des}$$
$$Y_{drift} = \frac{H.cos(\theta).W}{Des} \quad (1)$$

H: Drop height, θ: Entry angle, W: Water resistance coefficient, Des: Object density, X_{drift}: Horizontal displacemen, Y_{drift}: Vertical depth. As the entry angle increases, horizontal drift increases (the object moves forward more), while vertical settling decreases (since larger angles result in more horizontal motion). When the water resistance coefficient increases, settling slows, and drift increases. Conversely, as object density increases, settling accelerates, and drift decreases[13].

By leveraging hydrodynamic theory under a still-water assumption, the derived equations for horizontal drift and vertical depth provide effective predictions of final object positions. Compared to traditional computational fluid dynamics (CFD) [14] simulations, this approach requires lower computational resources, making it suitable for training machine learning models to support waterborne object search operations.

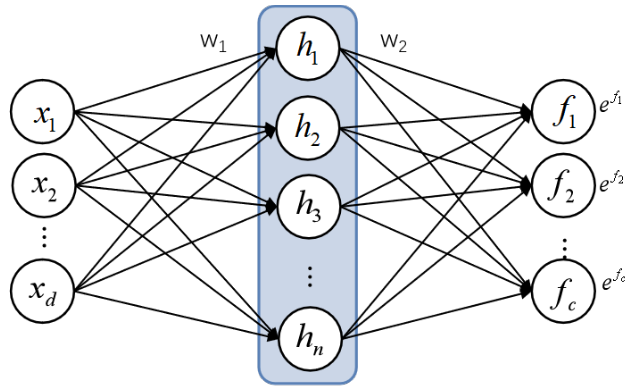

Figure 1 | MLP Model Architecture

Introduction to Deep Learning Model

MLP Models In recent years, the development of deep learning has made the Multilayer Perceptron (MLP) [15] a research hotspot. It has been widely applied in fields such as image processing, speech recognition, and natural language processing, achieving remarkable results in tasks like object detection, image classification, semantic segmentation, and machine translation.

MLP is a feedforward neural network architecture capable of mapping input vectors to output vectors. Its network structure typically consists of multiple fully connected neuron layers, where each neuron, except those in the input layer, employs a nonlinear activation function and is trained using the backpropagation algorithm. During model training, the network weights are first initialized. Then, the input data undergoes forward propagation to compute the weighted sum in hidden layers, which is transformed by the activation function to obtain the output.[16] Finally, the output layer generates the prediction results, and a loss function, such as Mean Squared Error (MSE) or Cross Entropy, is computed based on the ground truth labels. The backpropagation algorithm is then used to compute gradients and optimize network parameters. The MLP model architecture is illustrated in Figure 1.

This paper presents a four-layer Multi-Layer Perceptron (MLP) model, which consists of an input layer, three hidden layers, and an output layer. The design of this model facilitates the effective capture of intricate data patterns within the dataset, significantly boosting its overall learning capability.

a. The input layer plays a crucial role in the model by receiving the raw data inputs. In this layer, each neuron is designated to correspond to a specific feature of the input data, ensuring that all relevant attributes are adequately represented for subsequent processing.

b. The hidden layers, on the other hand, are integral to the model's function, as they perform the core operations of feature extraction and data mapping. The architecture of these hidden layers is fully connected, meaning that every neuron in a given layer is linked to all neurons in the previous layer. In this constructed model, three hidden layers have been implemented, and they utilize the ReLU (Rectified Linear Unit) activation function. This choice of activation function is particularly advantageous as it enhances the model's ability to represent nonlinear relationships within the data, further improving its performance and learning efficiency.

c. The output Layer: Responsible for generating the final prediction results. The number of neurons and the activation function in the output layer depend on the specific task requirements. For instance, binary classification tasks typically use the Sigmoid activation function, whereas multi-class classification tasks utilize Softmax

Decision Tree Model The decision tree model is recognized as a simple yet powerful tool in the field of data mining, commonly utilized in both classification and regression tasks. This model operates by creating a tree-like structure that transforms complicated decision-making processes into a series of straightforward judgments [17]. By doing so, it allows for more effective data segmentation and forecasting of outcomes. The architecture of a decision tree comprises several integral components: the root node, which symbolizes the entire dataset; internal decision nodes that signify the criteria for data splitting; and terminal nodes, or leaf nodes, which indicate the final decisions or classifications resulting from the analysis. During the development of a decision tree model, various evaluation metrics are employed to measure the effectiveness of the splits made within the data.

Among the most frequently used metrics are information entropy, information gain, and the Gini coefficient. These metrics are essential for assessing the changes in data purity that occur as a result of the division process. For example, information entropy can be mathematically expressed in a way that illustrates how it quantifies the level of uncertainty or disorder within a dataset before and after a split, thereby guiding the model in making more informed decisions:

$$Entorpy(S) = -\sum_{i=1}^{n} p_i log_2 p_i \tag{2}$$

Where S represents the current dataset, p_i represents the proportion of samples belonging to class i is denoted. Information gain reflects the reduction in uncertainty brought about by a particular feature in the dataset partitioning, and its formula is given by:

$$Gain(S, A) = Entorpy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entorpy(S_v) \tag{3}$$

Here, A is the candidate feature, S_v is the feature, and the subset corresponding to the value v of feature A is denoted.

Support Vector Machine Model Support Vector Machine (SVM) [18], as an efficient machine learning tool, performs excellently in handling regression problems. For the task of predicting the offset of the item drop location in a water body, the SVM regression model can build an accurate prediction system by minimizing the difference between the model's predicted offset and the actual observed value, thus enabling precise estimation of the item drop point's shift. The basic idea is to determine an optimal hyperplane that, within a certain error margin, positions most data points as close as possible to the hyperplane, ensuring good generalization capability when the model predicts unknown data.

In this application scenario, the shift in the item drop location is influenced by various factors such as water flow speed, direction, water temperature, and other environmental variables. SVM regression introduces a kernel function to map the input nonlinear features into a high-dimensional space, where the best-fitting hyperplane is sought to effectively capture the complex nonlinear relationships between variables. Furthermore, the model employs convex quadratic programming to ensure the stability of the global optimal solution and uses slack variables and an ϵ- insensitive loss function to balance model complexity and prediction accuracy, thereby enhancing robustness against outliers.

Random Forest Model The Random Forest model [19] is an ensemble learning method. Its basic idea is to construct a large number of randomly generated decision trees and combine the predictions from each tree to improve the overall model's stability and generalization ability. During the construction process, the model reduces the risk of overfitting commonly associated with individual decision trees by performing Bootstrap sampling on the original data and randomly selecting a subset of features at each node. This approach effectively captures the underlying complex relationships within the data[20].

To illustrate the prediction mechanism of Random Forest, the following formula is used. For regression problems, the final prediction result of the Random Forest is the average of the outputs from all the decision trees, and its mathematical expression is:

Where T represents the total number of decision trees, and $h_t(x)$ is the prediction output of the t-th tree

$$\hat{y} = \frac{1}{T}\sum_{t=1}^{T} h_t(x) \tag{4}$$

for the input x. This formula reflects the basic idea of reducing prediction variance through mean aggregation.

XGBoost Model XGBoost [21] is an efficient and scalable gradient boosting framework. Its core idea is to build decision trees incrementally using an additive model, minimizing prediction errors by optimizing the objective function, while also constraining model complexity to improve generalization ability and stability. In each iteration, XGBoost uses a second-order Taylor expansion to approximate the loss function, thereby capturing the variation in the objective function more accurately and accelerating the convergence rate. The objective function of XGBoost combines training error and a regularization term, and its expression is given by [22]:

$$Obj = \sum_{i=1}^{n} l(y_i, \hat{y}^{(t)}) + \sum_{k=1}^{t} \Omega(f_k) \tag{5}$$

$$\Omega(f_k) = \gamma T + \frac{1}{2}\lambda ||\omega||^2 \tag{6}$$

Where $l(y_i, \hat{y}^{(t)})$ represents the loss value for the i-th sample, $\hat{y}^{(t)}$ is the prediction result of the model after the t-th iteration. The regularization term $\Omega(f_k)$ is used to penalize model complexity, and γ and λ are the tuning parameters for the number of leaf nodes and the weights of the leaf nodes, respectively.

In each iteration, the model updates the overall output by adding the prediction contribution of the new tree, and its mathematical expression is:

$$\hat{y}^{(t)} = \hat{y}^{(t-1)} + f_t(x_i) \tag{7}$$

Where $f_t(x_i)$ represents the prediction contribution of the t-th tree for the sample x_i. This formula reflects the detailed process of how XGBoost approximates the model by progressively accumulating the outputs of decision trees.

Evaluation Metrics This paper uses three representative evaluation metrics to assess the prediction accuracy: Mean Squared Error (MSE), Mean Absolute Error (MAE), and the Coefficient of Determination. The mathematical expressions for these evaluation metrics are as follows:

$$MSE = \frac{1}{n} \times \sum_{i=1}^{n} (F_i - R_i)^2 \tag{8}$$

$$MAE = \sum_{i=1}^{n} \left| \frac{F_i - R_i}{R_i} \right| \tag{9}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n} (F_i - R_i)^2}{\sum_{i=1}^{n} (F_i - A_i)^2} \tag{10}$$

Where F_i is the predicted value for the i-th data point, R_i is the actual value for the i-th data point, n is the sequence length (number of samples), and A_i is the mean of all samples. Smaller values of MSE and MAE indicate smaller prediction errors and higher accuracy. R^2 takes values between 0 and 1, with a value closer to 1 indicating better fit of the neural network to the data, thus reflecting better model fitting ability.

Experimental Design and Results Analysis

The development tool selected for this paper is PyCharm, with the programming language Python 3.11.0. The Graphics Processing Unit (GPU) used is the NVIDIA GeForce GTX 4060, and the Central Processing Unit (CPU) is the i7-13600H, with 6GB of video memory. The experiment is based on a simulation-generated dataset for prediction, where the dataset is divided into 80% training data and 20% testing data for offset prediction. The models selected for prediction include MLP, Decision Tree, Support Vector Machine, Random Forest, and XGBoost. The parameters for each model are shown in the table 2-6.

This study explores the challenge of forecasting the displacement of objects as they enter water bodies, employing a comparative analysis of five distinct machine learning models: Decision Tree, Random Forest, Support Vector Machine (SVM), Multi-Layer Perceptron (MLP), and XGBoost. To rigorously assess the performance of these models, the study utilizes three quantitative evaluation metrics: Mean Absolute Error (MAE), Mean Squared Error (MSE), and the Coefficient of Determination (R^2). The findings of this analysis are comprehensively detailed in Table 7, which follows this discussion. Furthermore, the specific parameters utilized for each machine learning model are also outlined in the accompanying table, providing a clear understanding of the setup for the comparisons made in this study.

The experimental results indicate that XGBoost consistently surpasses all other models across all evaluated metrics. With a Mean Absolute Error (MAE) of 0.0112 and a Mean Squared Error (MSE) of 0.0002,

Table 2 | MLP Model Parameters

Parameter Name	Parameter Value
Learning Rate	0.001
Number of Iterations	200
Batch Size	32
Activation Function	Relu
Optimizer	Adam
Number of Hidden Layer Neurons	64
Training Set Ratio	80%
Test Set Ratio	20%

Table 3 | Decision Tree Model Parameters

Parameter Name	Parameter Value
Random Seed	42
Number of Target Variables	2
Number of Features	4
Training Algorithm	CART
Optimizer	Adam
Minimum Samples for Splitting	2
Minimum Samples per Leaf	1

Table 4 | Random Forest Model Parameters

Parameter Name	Parameter Value
Random Seed	42
Number of Target Variables	2
Number of Features	4
Number of Iterations	200
Bootstrap Sampling	True

Table 5 | SVM Model Parameters

Parameter Name	Parameter Value
Random Seed	42
Number of Target Variables	2
Number of Features	4
Number of Iterations	200
Kernel Coefficient	0.1
Penalty Parameter (C)	100

Table 6 | XGBoost Model Parameters

Parameter Name	Parameter Value
Random Seed	42
Number of Target Variables	2
Number of Weak Learners	300
Learning Rate	0.1
Maximum Tree Depth	6

Table 7 | Comparison of Evaluation Metrics for Different Models

Model	MAE	MSE	R^2
Decision Tree	0.0358	0.0026	0.9826
Random Forest	0.0185	0.0007	0.9953
Support Vector Machine	0.0529	0.0038	0.975
MLP	0.0522	0.0053	0.9609
XGBoost	0.0112	0.0002	0.9985

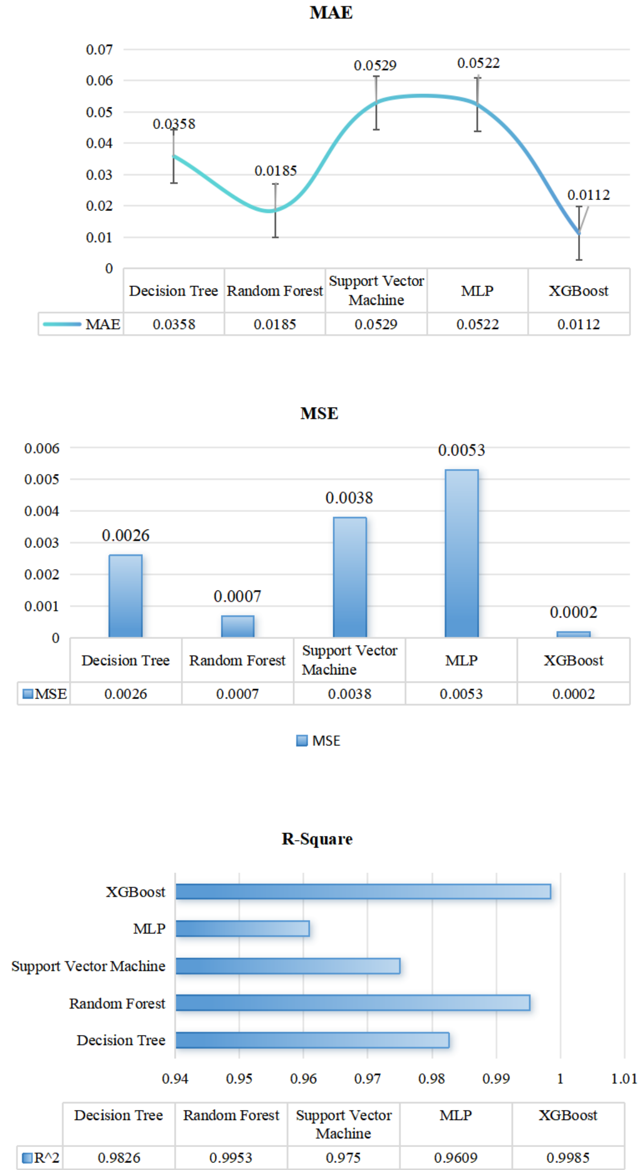

Figure 2 | Comparison Chart of Model Evaluation Metrics

XGBoost demonstrates markedly lower prediction errors compared to its competitors, signifying superior accuracy in estimating displacements. Furthermore, its R^2 value of 0.9985, which is very close to 1, suggests that the model accounts for 99.85% of the variance in the data, thereby illustrating an exceptional alignment between the predicted and actual values.

Ensemble learning models, particularly XGBoost and Random Forest, exhibit considerable advantages when it comes to predicting the positional displacements of objects in aquatic environments. Their impressive accuracy and robust generalization capabilities arise from their ability to collaboratively model the intricacies of complex environmental factors. In contrast, traditional modeling approaches such as decision trees, support vector machines (SVMs), and shallow neural networks, including multi-layer perceptrons (MLPs), tend to underperform due to their inherent limitations in representational capacity and often ineffective training methodologies.

Conclusion

This study systematically evaluates the performance of various machine learning models in predicting the positional displacement of objects falling into water bodies. Experimental results demonstrate that XGBoost, leveraging its gradient boosting mechanism and regularization strategies, significantly outperforms other models in both error control and data fitting, making it well-suited as the core algorithm for real-time search systems. Random Forest, due to its ensemble robustness, can serve as a complementary redundancy model. In contrast, traditional models (e.g., SVM, Decision Tree) and shallow MLPs are limited by their nonlinear representation capabilities, making them less adaptable to complex hydrodynamic scenarios. These findings provide a solid technical foundation for building intelligent retrieval systems that enhance cultural tourism safety by enabling faster and more accurate recovery of valuable items accidentally dropped into water at popular tourist destinations.

Acknowledgment The preferred spelling of the word “acknowledgment” in America is without an “e” after the “g”. Avoid the stilted expression “one of us (R. B. G.) thanks ...”. Instead, try “R. B. G. thanks...”. Put sponsor acknowledgments in the unnumbered footnote on the first page.

1. A B M K , B P G S .In search of water vapor on Jupiter: Laboratory measurements of the microwave properties of water vapor under simulated jovian conditions[J].Icarus, 2011, 212(1):210-223.
2. Yanming,Guo,Yu,et al.Deep learning for visual understanding: A review[J].Neurocomputing, 2016, 187(Apr.26):27-48.
3. Ning, H., Qian, S. and Zhou, T. (2023) ‘Applications of level set method in computational fluid dynamics: a review’, Int. J. Hydromechatronics, Vol. 6, No. 1, pp.1–33.
4. Chao, Q., Xu, Z., Shao, Y.Tao, J., Liu, C. and Ding, S. (2023) ‘Hybrid model-driven and data-driven approach for the health assessment of axial piston pumps’, Int. J.Hydromechatronics, Vol. 6, No. 1, pp.76–92.
5. B. Anari; J. A. Torkestani; A. Rahmani; "A Learning Automata-based Clustering Algorithm Using Ant Swarm Intelligence", EXPERT SYSTEMS, 2018.
6. Tao Wu; "Predictive Search for Capacitated Multi-Item Lot Sizing Problems", INFORMS JOURNAL ON COMPUTING, 2021.
7. Rihan Chen; Bin Liu; Han Zhu; Yaoxuan Wang; Qi Li; Buting Ma; Qingbo Hua; Jun Jiang; Yunlong Xu; Hongbo Deng; Bo Zheng; "Approximate Nearest Neighbor Search Under Neural Similarity Metric for Large-Scale Recommendation", CIKM, 2022.
8. Longbin Li; Chao Zhang; Sen Li; Yun Zhong; Qingwen Liu; Xiaoyi Zeng; "Graph Contrastive Learning with Multi-Objective for Personalized Product Retrieval in Taobao Search", ARXIV-CS.IR, 2023.
9. Nishant Yadav; Nicholas Monath; Manzil Zaheer; Rob Fergus; Andrew McCallum; "Adaptive Retrieval and Scalable Indexing for K-NN Search with Cross-Encoders", ARXIV-CS.IR, 2024.
10. Harsh Shah; Kashish Mittal; Ajit Rajwade; "Group Testing for Accurate and Efficient Range-Based Near Neighbor Search for Plagiarism Detection", ECCV, 2 024.
11. Peng Y, Yang X, Li D, et al. Predicting flow status of a flexible rectifier using cognitive computing[J]. Expert Systems with Applications, 2025, 264: 125878.
12. Yuan, F.; Huang, X.; Zheng, L.; Wang, L.; Wang, Y.; Yan, X.; Gu, S.; Peng, Y. The Evolution and Optimization Strategies of a PBFT Consensus Algorithm for Consortium Blockchains. Information 2025, 16, 268.
13. Tian Z, Zhao D, Lin Z, et al. Balanced reward-inspired reinforcement learning for autonomous vehicle racing[C]//6th Annual Learning for Dynamics & Control Conference. PMLR, 2024: 628-640.
14. Tian Z, Zhao D, Lin Z, et al. Efficient and balanced exploration-driven decision making for autonomous racing using local information[J]. IEEE Transactions on Intelligent Vehicles, 2024.
15. Lin Z, Zhang Q, Tian Z, et al. Slam2: Simultaneous localization and multimode mapping for indoor dynamic environments[J]. Pattern Recognition, 2025, 158: 111054.
16. Zhang C, Chen J, Li J, et al. Large language models for human–robot interaction: A review[J]. Biomimetic Intelligence and Robotics, 2023, 3(4): 100131.
17. Bai X, Peng Y, Li D, et al. Novel soft robotic finger model driven by electrohydrodynamic (EHD) pump[J]. Journal of Zhejiang University-SCIENCE A, 2024, 25(7): 596-604.
18. Allan P .Approximation theory of the MLP model in neural networks[J].Acta Numerica, 1999, 8:143-195.
19. Bui,DT,Pradhan,et al.Landslide Susceptibility Assessment in Vietnam Using Support Vector Machines, Decision Tree, and Naive Bayes Models[J].MATH PROBL ENG, 2012.
20. Tarabalka Y , Fauvel M , Chanussot J ,et al.SVM and MRF-Based Method for Accurate Classification of Hyperspectral Images[J].IEEE Geoscience & Remote Sensing Letters, 2010, 7(4):736-740.
21. Claudia,Lindner,Paul,et al.Robust and Accurate Shape Model Matching Using Random Forest Regression-Voting.[J].IEEE Transactions on Pattern Analysis & Machine Intelligence, 2015.
22. Rajkumar M N , Anbuchelvan R .A Novel Context-Aware Computing Framework with the Internet of Things and Prediction of Sensor Rank Using Random Neural XG-Boost Algorithm[J].Journal of Electrical Engineering & Technology, 2024(4):19.