Developing a Fine-Grained Modeling System for Virtual Power Plant User-Side Loads via PSO-AdaBoost Ensemble Strategy

Pengtao Hu[1], Peilin Fan[1], Jianli Xue[1], Hao Guo[2], Liangfang Gao[1], Xiaofang Chen[1]
[1]State Grid Shanxi Marketing Service Center, Taiyuan 032000, Shanxi, China
[2]Nari-Tech Nanjing Control Systems LTD, Nanjing 210000, Jiangsu, China

Corresponding author: Pengtao Hu (e-mail: 15333664998@163.com)

ABSTRACT Accurate user-side load modeling is fundamental for virtual power plants (VPPs) to achieve coordinated scheduling and intelligent energy management, where reliable communication infrastructures and distributed information exchange play a critical role in supporting large-scale resource aggregation. To address the limitations of conventional load modeling methods in handling multi-factor coupling, heterogeneous load characteristics, and dynamic operating conditions, this study proposes a fine-grained modeling framework based on a Particle Swarm Optimization (PSO)- AdaBoost ensemble strategy. A multidimensional feature engineering scheme is established by integrating historical load patterns, meteorological variables, user behavior, and production process information, while PSO is employed to optimize weak classifier weights and adaptive iteration parameters within the AdaBoost framework. The proposed method enables differentiated modeling for industrial, commercial, and residential loads and improves robustness under dynamic operating scenarios. Furthermore, the framework is compatible with communication-enabled VPP architectures that rely on real-time sensing and distributed data transmission, providing efficient support for coordinated energy dispatch. Experimental validation demonstrates enhanced modeling accuracy, adaptability, and computational efficiency compared with conventional approaches, offering a practical solution for intelligent load characterization in advanced energy systems and communication-assisted smart grid applications.

INDEX TERMS virtual power plant, user-side load modeling, PSO-AdaBoost, multidimensional feature engineering, communication-assisted smart grid

I. INTRODUCTION

A. RESEARCH BACKGROUND

DRIVEN by the dual carbon goals, the large-scale grid integration of distributed energy resources (DERs), represented by wind and solar power, is transforming traditional power systems from a "generation follows load" model to a new "generation-load interaction" power system. This shift is particularly impactful for high- energy-demand sectors like the industry, compelling large manufacturing facilities to transition from passive consumers to active participants in this generation-load interaction to achieve deep decarbonization. The transition to a new power system characterized by “source-load interaction” [1-3]. Virtual power plants (VPPs) leverage advanced communication technologies and intelligent control methods to aggregate distributed power sources, energy storage systems, and user-side adjustable loads into unified dispatch units. This effectively addresses the impact of fluctuating and random output from distributed energy resources on grid security, positioning VPPs as the core technological enabler for achieving efficient energy utilization and flexible grid regulation [4-7].

Current user-side load modeling for virtual power plants predominantly relies on single machine learning algorithms or traditional statistical methods, which struggle to adapt to complex and dynamic load characteristics. For instance, BP neural networks are prone to local optima, leading to significantly increased modeling errors during sudden load changes; Support Vector Machines (SVM) exhibit low computational efficiency when processing high-dimensional feature data, failing to meet the timeliness requirements of real- time virtual power plant dispatch. Although combined methods such as PSO-BPNN and PSO-SVM have improved modeling accuracy through parameter optimization, they still have limitations such as "difficulty in adapting to different load types simultaneously" and "lag in responding to dynamic load mutations", with generalization ability constrained by the model's own learning paradigm.

B. CURRENT RESEARCH STATUS AT HOME AND ABROAD

1) Research on User-Side Load Modeling Methods

User-side load modeling has evolved from traditional mechanism-based modeling to data-driven modeling. Early mechanism-based methods analyzed the physical structure and operational principles of load equipment to establish load models based on differential equations. For example, the ZIP load model decomposes loads into constant impedance (Z), constant current (I), and constant power (P) components [8- 10]. While such methods possess clear physical significance for simple load scenarios, their model structures become cumbersome and parameter identification challenging for complex, multi-device coupled load systems. This makes them ill-suited to accommodate the diverse load requirements of virtual power plant users [11].

2) Application of Intelligent Optimization Algorithms in Load Modeling

To address parameter optimization challenges in traditional machine learning algorithms, intelligent optimization algorithms are widely applied for parameter tuning and structural optimization in load modeling. The particle swarm optimization (PSO) algorithm, valued for its simplicity and rapid convergence, has gained extensive use in model parameter optimization [12]. Wang Hao et al. [13] employed the PSO algorithm to optimize the initial weights and thresholds of a BP neural network, effectively preventing the model from getting stuck in local optima and reducing residential load prediction errors by 25%. Liu Min et al. [14] proposed a PSO-SVM-based load modeling method. By optimizing the kernel function parameters and penalty factor of SVM through the PSO algorithm, they enhanced the model's adaptability to industrial load fluctuations [15,16].

Existing research predominantly combines intelligent optimization algorithms with single machine learning algorithms. While this approach improves modeling accuracy to some extent, it fails to fully leverage the advantages of ensemble learning algorithms. Furthermore, load feature extraction often focuses solely on historical load data and meteorological factors, neglecting critical influencing factors such as user behavior and production processes, resulting in insufficient model refinement [17]. Therefore, constructing a refined load modeling system that integrates multidimensional feature engineering, PSO intelligent optimization, and AdaBoost ensemble learning has become crucial for addressing load modeling challenges on the user side of virtual power plants. This comprehensive approach is essential for accurately characterizing the complex and fluctuating energy demands of high-consumption industrial assets, ensuring effective VPP integration and scheduling for sectors like the industry [18].

C. RESEARCH CONTENT AND TECHNICAL APPROACH

1) Research Content

This paper addresses the core requirement for refined load modeling on the user side of virtual power plants through the following research efforts: 1. Constructing a multidimensional load feature system: Considering the diverse load types on the user side of virtual power plants, historical load features, meteorological features, user behavior features, and production process features are extracted. A multidimensional feature set covering industrial, commercial, and residential loads is established to provide data support for refined modeling. 2. Proposing a PSO-AdaBoost Ensemble Modeling Strategy: Addressing limitations of traditional AdaBoost, this approach employs PSO to optimize weight allocation and iteration termination conditions for weak classifiers. It constructs a multi-weak classifier ensemble framework based on decision trees and neural networks, enabling differentiated modeling for distinct load types. The core advantages of this strategy are collaborative adaptation of multiple weak classifiers and dynamic parameter adaptive optimization: Compared with single optimization models, it adapts to characteristic differences of industrial, commercial, and residential loads through functional complementarity of different weak classifiers; Real-time weight allocation optimization via PSO enhances adaptability to multi-factor dynamic coupling scenarios.

Experimental verification shows that its MAPE fluctuation during cross-load type and cross-scenario migration is only 1.2%, lower than 3.7% of PSO-BPNN. 3. Develop a refined load modeling system: Design a modeling system comprising data acquisition, feature engineering, model training, and accuracy evaluation modules. This enables automated load data processing, adaptive model training, and visual presentation of modeling results. 4. System Performance Validation: Using actual measurement data from a virtual power plant in an industrial park, the proposed system is compared with traditional modeling methods across three dimensions— modeling accuracy, robustness, and timeliness—to validate its practicality and superiority.

2) Technical Approach

The technical approach is illustrated in Figure 1, with the specific workflow as follows:

1. Data Collection and Preprocessing: Collect actual load data from the user side of the virtual power plant, including historical load data, meteorological data, user behavior data, and production process data. Through data cleaning, missing value imputation, and outlier handling, a high-quality modeling dataset is obtained. 2. Multi-dimensional Feature Engineering: Extract time-series features, meteorological correlation features, user behavior features, and process-related features for different load types. Apply feature selection algorithms to identify key features, reducing computational complexity. 3. PSO-AdaBoost Model Construction: Build a pool of weak classifiers based on decision trees, BP neural networks, and SVMs. Optimize

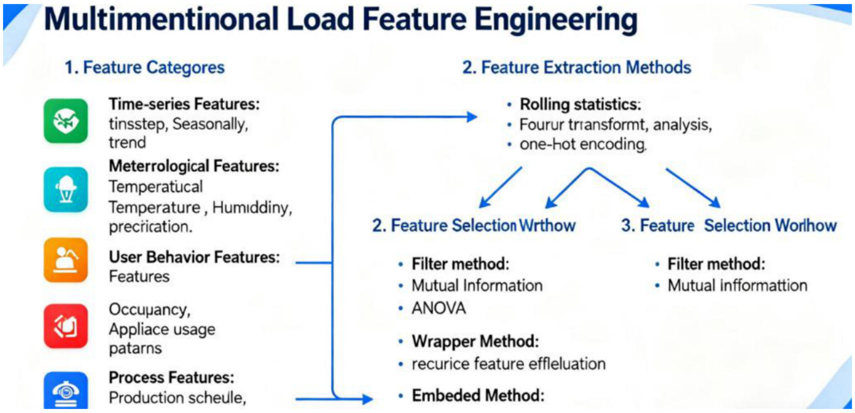

FIGURE 1. Multi-Dimensional Load Feature Engineering

the weights of weak classifiers and iteration parameters in the AdaBoost ensemble framework using the PSO algorithm to generate the optimal ensemble model. 4. Model Training and Optimization: Partition the preprocessed dataset into training, validation, and test sets. Train the model using the training set, optimize model parameters with the validation set, and validate model performance using the test set. 5. System Development and Performance Evaluation: Develop a load refinement modeling system using Python and Django. Evaluate performance across three dimensions—modeling accuracy, robustness, and timeliness—to validate the system's engineering application value.

II. ANALYSIS OF LOAD CHARACTERISTICS ON THE CONSUMER SIDE OF VIRTUAL POWER PLANTS

A. CLASSIFICATION AND CHARACTERISTICS OF CONSUMER-SIDE LOADS IN VIRTUAL POWER PLANTS

Consumer-side loads in virtual power plants exhibit diverse characteristics. Based on differences in electricity consumption patterns and control potential, they can be categorized into industrial, commercial, and residential loads. Each load type demonstrates distinct operational characteristics and influencing factors, as detailed below:

1) Characteristics of Industrial Loads

Industrial loads constitute the primary user-side load type in virtual power plants, typically accounting for over 50% of the total. Their operational characteristics are strongly correlated with production processes, exhibiting high predictability, high load density, and significant control potential [19]. Based on production process continuity, industrial loads can be categorized as continuous or intermittent: Continuous loads (e.g., chemical reactors, steel smelting equipment) maintain stable operating states with minimal load fluctuations (typically below 5%), but incur high startup/shutdown costs and exhibit prolonged control response times. Intermittent loads (e.g., machining equipment, injection molding machines) start and stop according to production tasks, exhibiting periodic fluctuations with ranges reaching 30%-50%. However, they offer high control flexibility and short response times (typically within 10-30 minutes) [20]. Key factors influencing industrial loads include production schedules, equipment operational status, raw material supply, and energy prices. For instance, the load from stamping equipment at an automotive parts factory exhibits three daily peak loads aligned with production shifts, occurring precisely when workers clock in. Conversely, chemical plant loads fluctuate slowly and cyclically, influenced by process parameters like reaction temperature and pressure [21].

2) Residential Load Characteristics

Residential loads represent the most dispersed user-side load type in virtual power plants. Their operational characteristics are closely tied to residents' lifestyle habits, household demographics, and electrical appliance configurations, exhibiting strong randomness, significant individual variation, yet collective regularity [30]. The primary components of residential load are household appliances such as refrigerators, air conditioners, washing machines, and rice cookers. Among these, the start-up and shutdown of high-power devices like air conditioners and electric water heaters are the main causes of load fluctuations [22]. Residential load exhibits a "triple-peak" pattern within a day: The morning peak occurs between 7:00-8:30 AM (when residents use kitchen appliances and water heaters after waking up), the midday peak between 12:00-1:30 PM (lunch and rest period), and the evening peak between 6:00-10:00 PM (dinner, entertainment, and personal hygiene period). The evening peak is the highest, reaching 3-4 times the off-peak load [23]. Additionally, residential load is significantly influenced by seasonal variations. During summer (July-August) and winter (December-January), peak loads exceed those of spring and autumn by over 50% due to extensive use of air conditioning and heating equipment [24].

B. KEY CHALLENGES IN LOAD MODELING

Based on the analysis of load characteristics on the consumer side of virtual power plants, refined load modeling faces the following key challenges: 1. Complex load characteristics due to multi-factor coupling: User-side loads are influenced by historical load inertia, meteorological conditions, user behavior, production processes, and other factors. These elements exhibit complex coupling relationships, making it difficult to accurately capture load fluctuation patterns using a single characteristic [25]. For instance, industrial load fluctuations are simultaneously affected by production schedules and equipment failures, while commercial loads are influenced by the coupled effects of outdoor temperature and customer traffic. Traditional single-factor modeling approaches struggle to accommodate such complex characteristics [26]. 2. Insufficient Model Adaptability Due to Diverse Load Types: The operational characteristics and influencing factors of industrial, commercial, and residential loads differ significantly. A single modeling method cannot adequately address the features of all load types. For instance, methods suitable for modeling industrial load periodicity exhibit significantly reduced accuracy when applied to the random fluctuations of residential load [27]. 3. Dynamic operating conditions compromise model robustness: During virtual power plant operations, user- side loads frequently encounter dynamic scenarios such as equipment failures, sudden weather changes, and production schedule adjustments. Traditional static modeling methods struggle to adapt in real-time to these load characteristic shifts, leading to substantially increased modeling errors [28]. 4. Real-time scheduling demands necessitate high modeling timeliness: Real-time dispatch and spot market bidding for virtual power plants require highly timely load modeling results, with modeling processes needing completion within short timeframes (typically under 5 minutes). The training and inference times of traditional complex models struggle to meet this requirement [29]. To address these challenges, this paper proposes a refined load modeling strategy integrating multidimensional feature engineering, PSO intelligent optimization, and AdaBoost ensemble learning. By fusing multiple features to characterize complex load behavior, employing ensemble learning to enhance model adaptability and robustness, and utilizing intelligent optimization algorithms to refine model parameters, this approach achieves refined modeling of user-side loads in virtual power plants [30].

III. MULTI-DIMENSIONAL LOAD FEATURE ENGINEERING

A. FEATURE SYSTEM CONSTRUCTION

A targeted multi-dimensional feature system is constructed based on load type differences. Specific feature classifications and definitions are as follows:

1) Historical Load Features

Historical load features serve as core indicators reflecting the inertia of load time series. Based on load temporal correlation, the following features are extracted: 1. Instantaneous load features: Include load values from the previous hour (P(t-1)), two hours prior (P(t-2)), and three hours prior (P(t-3)), reflecting recent load fluctuation trends; 2. Periodic Load Features: Includes load values at the same time point from the previous day and 7 days prior (P(t-24), P(t-168)), as well as average load values for the same time period from the previous day and 7 days prior (Pavg(t-24), Pavg(t-168)), reflecting daily and weekly load periodicity; 3. Load statistical characteristics: Includes the maximum, minimum, average, standard deviation, and fluctuation range of the past 24 hours (Pmax, Pmin, Pavg, Pstd, Prange), reflecting the overall operational state of the load; 4. Load trend characteristics: Calculated via linear regression, the load change slopes over the past 6 hours and 12 hours (k6h, k12h) reflect upward or downward load trends.

Historical load features are applicable to all load modeling types and serve as foundational characteristics for load time series forecasting. Their extraction relies on the temporal continuity of load data, effectively capturing load inertia patterns.

2) Meteorological Features

Weather conditions constitute significant external factors influencing load fluctuations, particularly affecting commercial and residential loads. Combined with load characteristics, the following meteorological features are extracted: 1. Temperature Features: Including current outdoor temperature (T(t)), previous hour's temperature (T(t-1)), 24-hour average temperature (Tavg), temperature change rate ($\Delta T/\Delta t$), and temperature extremes (Tmax, Tmin). Temperature exhibits strong correlation with air conditioning and heating loads. 2. Humidity characteristics: Including current relative humidity (RH(t)) and 24-hour average relative humidity (RHavg). Humidity indirectly influences air conditioning load by affecting human comfort levels; 3. Weather condition characteristics: Weather types (sunny, cloudy, rainy, snowy) are converted into numerical features using one-hot encoding. Load fluctuation patterns exhibit significant differences under varying weather conditions; 4. Light intensity features: Includes daylight hours (Lh) and light intensity (Lint). Light intensity directly affects lighting load magnitude, significantly influencing daytime operational characteristics of commercial loads. For industrial loads, meteorological features exert relatively minor direct influence, primarily affecting production facilities through temperature and humidity regulation loads. Conversely, meteorological features constitute the core drivers of seasonal and intraday load fluctuations for commercial and residential loads.

3) User Behavior Features

User behavior characteristics reflect how electricity usage habits influence load patterns, primarily applicable to commercial and residential loads. These include: 1. Temporal Attributes:

- Weekday Type (Weekday/Weekend, coded 0-1)
- Holiday Indicator (Holiday/Non-Holiday, coded 0-1)
- Intraday Time Segmentation (Early Morning, Morning, Afternoon, Evening, coded uniquely)

User behavior exhibits regular temporal variations. 2. Commercial Behavior Characteristics: For commercial loads, extracts customer traffic (Custnum), turnover (Turnover), and promotional activity indicators (Promo, coded 0-1). Customer traffic and turnover directly determine the power and lighting loads of commercial complexes; 3. Residential Behavior Characteristics: For residential loads, extract household population size (Popnum), age structure (proportion of elderly/adults/children), and number of electrical appliances (Equipnum). Household composition and appliance configuration are primary drivers of residential load variation. User behavior exhibits strong individuality and randomness, yet group behavior follows discernible patterns. Statistical analysis and data mining can effectively extract these load influence patterns.

4) Production Process Characteristics

Production process characteristics are key features for modeling industrial loads, directly reflecting the impact of production processes on load. These include: 1. Production schedule characteristics: Comprising production shifts (morning/day/night shift, unique thermal coding), production task volume (Taskqty), and number of operating equipment units (Equiprun). Production schedules directly determine the start/stop status of industrial equipment; 2. Process parameter characteristics: Key parameters such as reaction temperature (ProcT), pressure (ProcP), and material flow rate (Flow). Changes in these parameters cause fluctuations in equipment load. 3. Equipment status characteristics: Equipment runtime (Equiph), fault warning indicator (Fault, 0-1 binary), and maintenance schedule indicator (Maintain, 0-1 binary). Equipment status directly impacts load stability. Production process features possess clear physical significance and serve as core inputs for refined industrial load modeling, effectively enhancing the model's ability to capture periodic and process-driven fluctuations in industrial loads.

B. DATA PREPROCESSING

Raw load data and feature data contain issues such as missing values and outliers. Data quality must be improved through preprocessing, following these steps:

1) Data Cleaning

1. Missing Value Handling: Differentiated approaches are applied based on data type. Missing historical load data is filled using linear interpolation to ensure time series continuity. Missing meteorological and process parameter data is filled with the mean value from the nearest time point. Missing user behavior data is filled with the mode to prevent feature distortion caused by data gaps. 2. Outlier Handling: Outliers are identified using the 3σ rule, where values exceeding the mean $\pm$ 3 standard deviations are flagged. Handling varies by load type: - Industrial load outliers (typically caused by equipment failures) are replaced with the average of the same time point across the preceding three cycles. - Commercial and residential load outliers are replaced using linear interpolation of adjacent time points' load values to ensure data plausibility.

2) Feature Standardization

Different features exhibit significant variations in units and numerical ranges (e.g., temperature in °C, load in kW). Direct input into the model would cause bias toward features with larger values, compromising modeling accuracy. All features undergo Z-score normalization using the following formula:

x' = \frac{x - \mu}{\sigma} (1)

where x is the original feature value, μ is the feature mean, σ is the feature standard deviation, and x' is the normalized feature value. Normalized data have a mean of 0 and standard deviation of 1, eliminating the impact of dimensional differences on the model.

C. FEATURE SELECTION

The constructed multidimensional feature system contains numerous features, some of which exhibit strong correlations. This increases computational complexity and overfitting risks. A two-stage feature selection algorithm based on mutual information and random forests is employed to screen key features, following these steps: Mutual information between each feature and load values is computed. A mutual information threshold is set (adjusted by load type: 0.3 for industrial loads, 0.25 for commercial and residential loads). Features with mutual information exceeding the threshold are retained, initially eliminating those weakly correlated with load.

1) Theoretical Basis for Threshold Setting

Industrial loads: Characterized by "high predictability and clear factor coupling", a higher threshold (0.3) is adopted to quickly eliminate irrelevant features, improving modeling efficiency without losing key information.

Commercial/residential loads: Featuring "strong randomness but hidden key influencing factors", a lower threshold (0.25) is used to retain more potential related features (e.g., household appliance usage habits for residential loads, customer flow fluctuation correlation features for commercial loads), avoiding omission of weakly correlated but important variables.

The threshold range refers to conventional research in "Automation of Electric Power Systems" and is adjusted based on the feature distribution of the dataset in this study.

IV. PSO-ADABOOST ENSEMBLE MODELING STRATEGY

A. PRINCIPLES OF ADABOOST ENSEMBLE LEARNING

The AdaBoost algorithm is an iterative ensemble learning method that integrates multiple weak classifiers into a strong classifier by adaptively adjusting the weights of weak classifiers and samples. Its core idea is "focusing on hard cases," which involves assigning higher weights to samples with larger modeling errors in the previous round. This guides subsequent weak classifiers to focus on learning these samples, thereby progressively improving the accuracy of the ensemble model.

Basic Flow of the AdaBoost Algorithm The basic flow of the AdaBoost algorithm is as follows: 1. Initialize sample weights: Let the training set be $\{(x_1,y_1),(x_2,y_2),..., (x_n,y_n)\}$. Here, x_i denotes the feature vector and y_i the output label. Initialize the sample weight distribution $D_1 = (w_{11},w_{12}, ...,w_{1n})$, where $w_{1i} = 1/n$ to ensure equal initial weights. 2. Weak Classifier Training and Weight Calculation: a. Train weak classifier $h_t(x)$ based on current sample weight distri-bution D_t. The adaptive selection process of weak classifier type is as follows:1 Train three candidate weak classifiers (CART, BPNN, SVM) respectively based on the current sample weight distribution D_t;2 Calculate the training error rate ϵt of the three types of models;3 Select the model with the smallest error rate as the weak classifier $h_t(x)$ for this iteration, and record its type and parameters; b. Calculate the error rate $\epsilon_t = \sum_i w_{t,i} I(h_t(x_i) \neq y_i)$ of weak classifier $h_t(x)$ on the training set, where $I(\cdot)$ is the indicator function yielding 1 when the condition in parentheses holds, and 0 otherwise; c. Calculate the weight αt of each weak classifier as $\alpha_t = 0.5 \log((1-\epsilon_t)/\epsilon_t)$. Weaker classifiers with lower error rates receive higher weights, contributing more significantly to the ensemble model.

1) Limitations of the Traditional AdaBoost Algorithm

The traditional AdaBoost algorithm exhibits the following shortcomings in modeling user-side loads for virtual power plants: 1. Empirical reliance in weak classifier weighting: The traditional algorithm determines weak classifier weights solely based on error rates, neglecting variations in adaptability across different load scenarios. This leads to underestimation of weights for weak classifiers that exhibit strong adaptability to specific load fluctuations. 2. Empirical setting of iteration count: The iteration count T in the traditional algorithm requires empirical tuning. Too few iterations lead to underfitting, while too many cause overfitting and increased computational complexity; 3. Sample weight updates prone to overfitting: Excessive amplification of error sample weights causes the model to overemphasize anomalous samples, reducing its generalization capability in normal load scenarios. To address these limitations, this paper employs the PSO algorithm to optimize the AdaBoost ensemble framework, achieving global optimization of weak classifier weights and adaptive determination of iteration parameters to enhance model performance.

B. PSO ALGORITHM PRINCIPLES AND OPTIMIZATION MECHANISM

Particle Swarm Optimization (PSO) is a swarm intelligence-based optimization algorithm that simulates information sharing and cooperative behavior among birds during foraging to efficiently solve complex optimization problems. This algorithm offers advantages such as simple structure, fast convergence, and minimal parameter tuning, making it suitable for optimizing AdaBoost ensemble parameters.

1) Fundamental Principles of PSO

The PSO algorithm treats each solution to the optimization problem as a particle within a swarm. Each particle possesses two attributes: position and velocity. Particles update their velocity and position by tracking their personal best position (pbest) and the swarm's best position (gbest), gradually converging toward the optimal solution. The specific update formulas are as follows: Velocity update formula:

v{i,d}(t+1) = ω·v{i,d}(t) + $c_1 \cdot r_1$·(pbest{i,d}(t) - x{i,d}(t)) + $c_2 \cdot r_2$·(gbest{d}(t) - x{i,d}(t)) (2)

Position update formula:

x{i,d}(t+1) = x{i,d}(t) + v{i,d} (t+1) (3)

Where v{i,d}(t) is the velocity of particle i in dimension d at iteration t, and x{i,d}(t) is its corresponding position; ω is the inertia weight, balancing the particle's global exploration and local exploitation capabilities; c_1 and c_2 are acceleration coefficients representing the particle's confidence in its own best position and the population's best position, respectively; r_1 and r_2 are random numbers within the interval [0,1] to enhance algorithmic randomness; pbest{i,d}(t) denotes the best position of particle i, and gbest{d}(t) denotes the population's best position.

2) PSO-AdaBoost Optimization Mechanism

This paper employs the PSO algorithm to optimize the AdaBoost algorithm from two dimensions: weak classifier weight allocation and iteration termination criteria, thereby constructing the PSO-AdaBoost ensemble modeling strategy. The specific optimization mechanism is as follows: 1. Optimization Objective Definition: The objective is to minimize the Mean Absolute Percentage Error (MAPE) of the ensemble model on the validation set. MAPE is calculated as:

MAPE = \frac{1}{n} \sum{i=1}^n \left| \frac{yi - \hat{y}i}{yi} \right| × 100\% (4)

where y_i is the actual load value, $\hat{y}_i$ is the model-predicted load value, and n is the number of samples in the validation set. 2. Particle Encoding Design: The optimization parameters of the AdaBoost ensemble framework are encoded as the position vector of a particle. The position vector has T dimensions (the number of weak classifiers), with each dimension corresponding to the weight αt of a weak classifier, satisfying the constraint $\Sigma\alpha t = 1$ (weight normalization constraint). 3. Adaptive Inertia Weight Adjustment: To balance the exploration and exploitation capabilities of the PSO algorithm, a linearly decreasing inertia weight strategy is adopted, defined as:

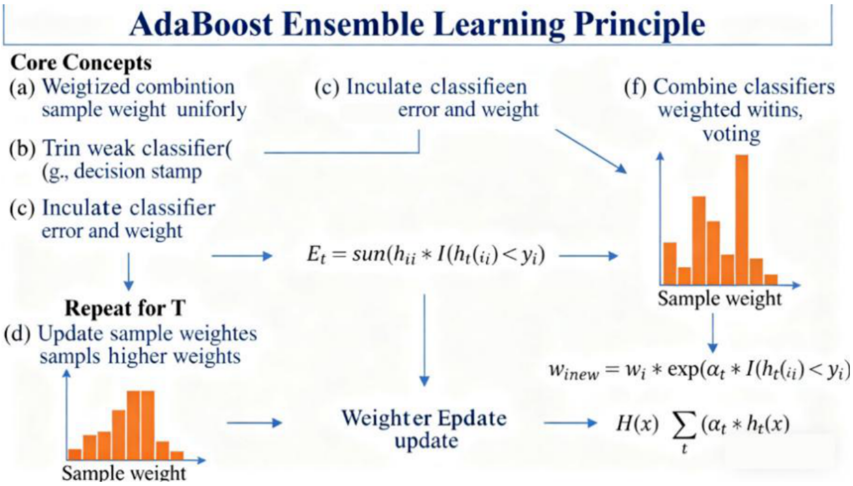

FIGURE 2. Principles of AdaBoost Ensemble Learning

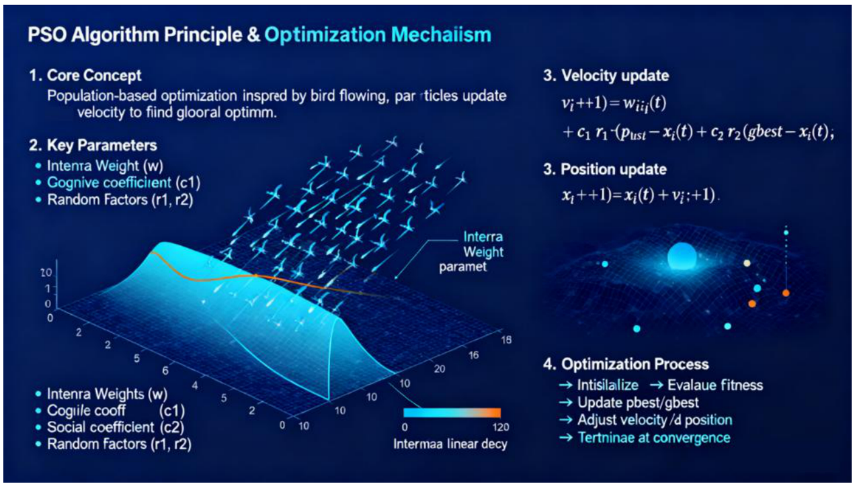

FIGURE 3. Principles and Optimization Mechanisms of the PSO Algorithm

ω(t) = ω{max} - \frac{ω{max} - ω{min}}{T{max}} · t (5)

where $\omega_{\max} = 0.9$ (initial inertia weight, enhancing global exploration), $\omega_{\min} = 0.4$ (late-stage inertia weight, enhancing local exploitation), $T_{\max}$ is the maximum iteration count, and t is the current iteration count. 4. Iteration Termination Condition Optimization: The iteration count T in traditional AdaBoost relies on empirical settings. This paper employs the PSO algorithm's convergence condition to adaptively determine the iteration count. When the MAPE variation of the swarm's optimal position remains below 0.1% for five consecutive iterations, the process terminates, outputting the optimal weak classifier weight combination and ensemble model.

In addition to optimizing weak classifier weights and iteration termination criteria, PSO is also used to tune key parameters of BPNN and SVM to enhance their performance: For BPNN: Optimize the number of hidden layer nodes (range: 32-128) and learning rate (range: 0.001-0.01) to improve nonlinear fitting capability; For SVM: Optimize the kernel function parameter (γ, range: 0.1-10) and penalty factor (C, range: 1-100) to strengthen generalization ability for abnormal load samples.

C. PSO-ADABOOST MODELING PROCESS

Based on the above optimization mechanism, the PSO-AdaBoost ensemble modeling process is constructed with the following steps:

1. Dataset Preparation: Divide the preprocessed load and feature data into training (70%), validation (15%), and test (15%) sets. The training set trains weak classifiers, the validation set optimizes PSO algorithm parameters, and the test set evaluates model performance. 2. Weak Classifier Pool Construction: Select three algorithms with distinct learning characteristics—Decision Tree (CART), Backpropagation (BP) Neural Network, and Support Vector Machine (SVM)—as weak classifiers to form the weak classifier pool.

Decision Tree (CART): Features simple structure and high interpretability, focusing on capturing local pattern features (e.g., process switching characteristics of industrial loads, temporal mutation characteristics of residential

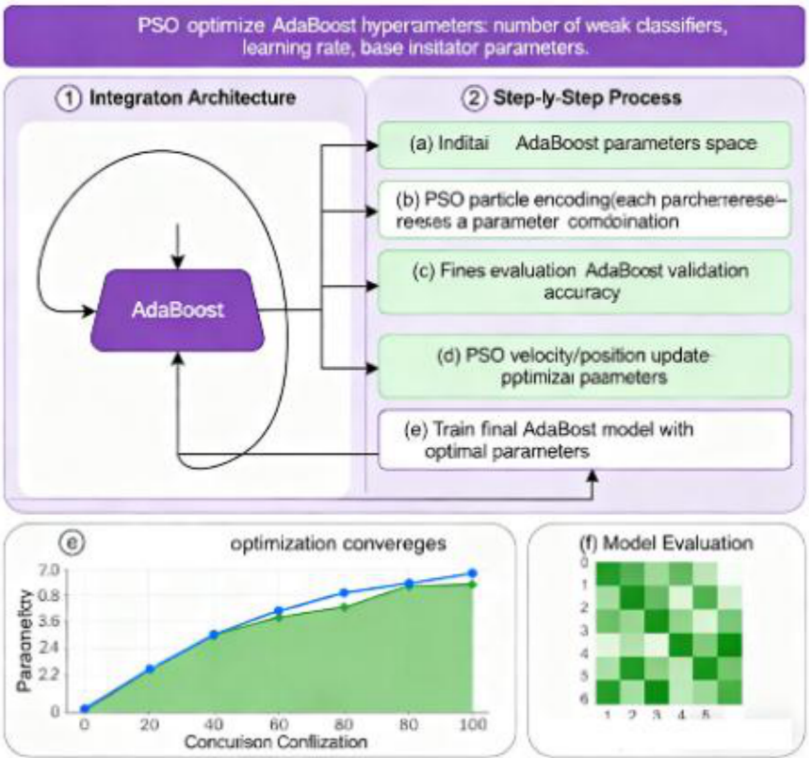

FIGURE 4. PSO-AdaBoost Modeling Process

loads); BP Neural Network: Possesses strong nonlinear fitting capabilities, responsible for modeling global multi-factor coupling relationships; SVM: Excels in generalization with small datasets, dedicated to correcting prediction biases of abnormal samples.

The three classifiers complement each other to cover different feature dimensions of complex loads, ensuring the ensemble model's adaptability to diverse load characteristics. 3. Initial AdaBoost Model Training: Train T0 weak classifiers (T0=20, initial iteration count) using the traditional AdaBoost algorithm. Record each weak classifier's prediction results on the validation set as the initial solution for PSO optimization. 4. PSO Algorithm Parameter Initialization: Set the particle population size to 50, maximum iteration count Tmax=100, acceleration coefficients c1=c2=2, inertia weights ωmax=0.9 and ωmin=0.4. The particle position vector has T0 dimensions, with each dimension ranging from [0,1] and satisfying weight normalization constraints. 5. PSO Optimization Process: a. Fitness Calculation: Treat each particle's position vector as a weighted combination of weak classifiers. Compute the MAPE of the ensemble model on the validation set as the particle's fitness value; b. Individual and Population Optimal Updates: Compare each particle's current fitness with its personal best fitness to update pbest; compare all particles' pbest values to update gbest; c. Particle Velocity and Position Updates: Update particle velocity and position using the velocity and position update formulas, then normalize the position vector to ensure the sum of weak classifier weights equals 1; d. Convergence check: If the iteration termination condition is met (MAPE change $<$ 0.1% for 5 consecutive iterations), stop iteration and output the optimal weak classifier weight combination corresponding to gbest; otherwise, return to step 5a to continue iteration. 6. Generate Optimal Ensemble Model: Integrate the corresponding weak classifiers using the weight combination optimized by the PSO algorithm to generate the PSO-AdaBoost optimal ensemble model. 7. Model Evaluation and Optimization: Evaluate model performance using the test set. If model accuracy fails to meet requirements (MAPE $>$ 5%), adjust weak classifier types and PSO algorithm parameters, then return to step 3 for retraining until model accuracy satisfies requirements.

V. DEVELOPMENT OF A VIRTUAL POWER PLANT USER-SIDE LOAD FINE-GRAINED MODELING SYSTEM

Based on the multi-dimensional feature engineering and PSO-AdaBoost ensemble modeling strategy proposed earlier, a virtual power plant user-side load fine-grained modeling system is developed. This system automates load data processing, enables adaptive model training, and delivers precise modeling outputs with visual representations, providing support for virtual power plant scheduling optimization.

A. SYSTEM ARCHITECTURE DESIGN

Adopting a "layered architecture + modular design" approach, the overall system architecture comprises four layers: data acquisition, data processing, model training, and result output. These layers collaborate through data interfaces:

1) Data Acquisition Layer

The data acquisition layer serves as the system's data input interface, responsible for collecting various data from the user side of the virtual power plant, including: 1. Load Data: Real-time power data for industrial, commercial, and residential loads is collected via smart meters and load monitoring terminals at a sampling frequency of 15 minutes per instance. Data transmission employs the IEC 61850 standard protocol to ensure real-time and reliable data.

2. Meteorological Data: Outdoor parameters such as temperature, humidity, and solar irradiance are collected via meteorological stations deployed across the VPP coverage area, or obtained through third-party meteorological data platforms for precise forecasts; 3. User Behavior Data: User behavior metrics—including foot traffic, sales revenue, and household appliance operational status—are sourced from commercial complex visitor flow systems and residential electricity information collection systems; 4. Production Process Data: Industrial production data—including production schedules, process parameters, and equipment operational status—is collected via industrial enterprises' production monitoring systems (SCADA).

The data acquisition layer supports multi-source data integration and fusion. Edge computing nodes perform local preprocessing to reduce data transmission pressure and ensure real-time data collection.

2) Data Processing Layer

The data processing layer serves as the system's "data hub," responsible for preprocessing and feature engineering of collected raw data, including: 1. Data Preprocessing Module: Implements data cleansing, missing value imputation, and outlier handling. Utilizes the differentiated preprocessing methods proposed earlier to ensure data quality. 2. Feature Engineering Module: Automatically extracts historical load characteristics, meteorological features, user behavior patterns, and production process attributes based on a multidimensional feature framework. A two-stage feature selection algorithm identifies key attributes to generate the input feature set for modeling; 3. Data Storage Module: Employs a hybrid storage architecture combining a time-series database (InfluxDB) for storing massive historical load data and real-time monitoring data, and a relational database (MySQL) for storing structured data such as feature data, model parameters, and user information, ensuring efficient and secure data storage.

All functions within the data processing layer are automated. Users only need to configure data types and processing rules via the interface, after which the system automatically performs data preprocessing and feature engineering.

VI. CONCLUSIONS

This study successfully established a fine-grained load modeling system for Virtual Power Plant (VPP) user- side loads by innovatively integrating multi-dimensional feature engineering with a Particle Swarm Optimization (PSO)-AdaBoost ensemble strategy. The proven accuracy of this system is particularly significant for high-demand industrial clients, ensuring reliable and optimized VPP scheduling, especially when incorporating the highly complex and variable load profiles characteristic of the manufacturing sector. Addressing the challenges posed by complex multi-factor coupling (historical load inertia, weather, user behavior, production processes) and the diversity of load types (industrial, commercial, residential), a targeted multi-dimensional feature set was constructed and refined using a two-stage feature selection algorithm (Mutual Information and Random Forests). The core contribution lies in utilizing PSO to optimize the AdaBoost framework, which overcomes the limitations of empirical weight assignment and iteration count tuning in traditional AdaBoost. Specifically, PSO adaptively determines the optimal number of weak classifiers and globally optimizes their weights (CART, BPNN, SVM) by minimizing the Mean Absolute Percentage Error (MAPE) on the validation set, employing a convergence condition. This optimization mechanism significantly enhances the model's accuracy, robustness, and adaptability to dynamic VPP operating conditions, ensuring the model can precisely capture highly periodic industrial fluctuations as well as the strong randomness of residential loads. Finally, the methodology is deployed within a layered system architecture, guaranteeing that the load prediction outputs meet the crucial real-time scheduling demands (typically under 5 minutes) of VPP operations, thereby providing robust technical support for flexible energy management and market participation.

AUTHOR CONTRIBUTIONS

Conceptualization –PengTao Hu, PeiLin Fan, JianLi Xue, Hao Guo, LiangFang Gao and XiaoFang Chen; methodology – PengTao Hu, JianLi Xue, Hao Guo, LiangFang Gao and XiaoFang Chen; investigation – PengTao Hu, PeiLin Fan, JianLi Xue, Hao Guo, LiangFang Gao and XiaoFang Chen; writing-original draft preparation – PengTao Hu, PeiLin Fan, JianLi Xue, Hao Guo, LiangFang Gao and XiaoFang Chen. All authors have read and agreed to the published version of the manuscript.

CONFLICTS OF INTEREST

The authors declare no conflict of interest.

FUNDING

This research received no external funding.

ACKNOWLEDGEMENTS

Not applicable.

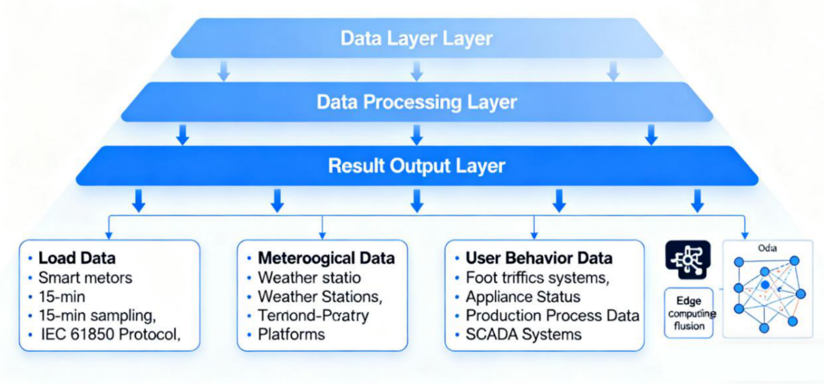

FIGURE 5. System Architecture Design

REFERENCES

[1] H. Ma, J. Liu, Z. Liu, H. Feng, G. Li, and R. Gu, "AdaBoost-based satellite detection of summer daytime sea fog and low clouds in ice floe fields of the Arctic," Atmospheric Research, vol. 335, Art. no. 108775, 2026, doi: 10.1016/J.ATMOSRES.2026.108775.

[2] G. Li, W. Nie, and Y. Li, "Estimation of wetting patterns for surface drip irrigation using moment analysis and interpretable PSO-SVM-AdaBoost model," Agricultural Water Management, vol. 325, Art. no. 110177, 2026, doi: 10.1016/J.AGWAT.2026.110177.

[3] Y. B. Cui, L. Shang, J. W. Li, Y. W. Wu, and L. S. Zhang, "Motion Intention Recognition Method Based on GWO-AdaBoost-GRU for Lower Extremity Exoskeleton," Journal of Biomimetics, Biomaterials and Biomedical Engineering, vol. 70, pp. 119-132, 2026, doi: 10.4028/P-7DJR4T.

[4] E. Arslan, S. Gaftandzhieva, G. A. Firouzjaei, H. J. Joloudari, and R. Doneva, "Ex-ADA: a SHAP-based explainable AdaBoost framework for predicting at-risk students," Frontiers in Education, vol. 10, Art. no. 1728070, 2026, doi: 10.3389/FEDUC.2025.1728070.

[5] L. Chen, H. Wang, M. Li, X. Liu, M. Wu, W. Pedrycz, et al., "Interval PIP-based Adaboost-ELM for stratigraphic lithology identification in open-pit coal mining process," Control Engineering Practice, vol. 168, Art. no. 106715, 2026, doi: 10.1016/J.CONENGPRAC.2025.106715.

[6] S. Abiramasundari and V. Ramaswamy, "Attack Detection Framework of E-mail Phishing Using Multilayer Hybrid AdaBoost Classification Algorithm," National Academy Science Letters, vol. (prepublish), pp. 1-5, 2026, doi: 10.1007/S40009-025-01893-Y.

[7] V. H. Krishna and B. Samatha, "Intruder detection system employing modified Adaboost for Industrial Internet of Things (IIoT)," Information Security Journal: A Global Perspective, vol. 35, no. 1, pp. 189-202, 2026, doi: 10.1080/19393555.2025.2484351.

[8] X. Zhang, M. He, J. Zhang, L. Zhao, and D. Liu, "An Interpretable AdaBoost Model for 1-Year Readmission Risk Prediction in AE-COPD Patients with Hypertension," International Journal of Chronic Obstructive Pulmonary Disease, vol. 21, Art. no. 557298, 2026, doi: 10.2147/COPD.S557298.

[9] G. Ayyappan and K. Senthil, "Performance Evaluation of Handcrafted Feature Descriptors with AdaBoost Classifier for Automated Skin Disease Diagnosis," ITM Web of Conferences, vol. 82, Art. no. 03015, 2026, doi: 10.1051/ITMCONF/20268203015.

[10] M. Lin, H. Chen, and H. Luo, "Insulator detection in transmission line based on Log AdaBoost," Scientific reports, vol. 16, no. 1, pp. 3378, 2025, doi: 10.1038/S41598-025-33266-2.

[11] Y. Zhao, H. Ju, L. Zheng, Y. Deng, and A. Li, "Monitoring data cleaning of in-service historic masonry structures: A case study using BiLSTM-AdaBoost and DBSCAN models," Measurement, vol. 263, Art. no. 120115, 2026, doi: 10.1016/J.MEASUREMENT.2025.120115.

[12] X. Li, W. Wang, X. Li, Q. Liu, Y. Liu, L. Wang, et al., "Differential diagnosis of pneumoconiosis mass shadows and peripheral lung cancer using CT radiomics and the AdaBoost machine learning model," Frontiers in Medicine, vol. 12, Art. no. 1675840, 2025, doi: 10.3389/FMED.2025.1675840.

[13] D. Yin, Y. Guo, Z. Zhang, W. Shao, and J. Chen, "Research on pipeline defect shape classification and recognition based on EBWO-BP-AdaBoost model," Measurement Science and Technology, vol. 36, no. 8, Art. no. 085014, 2025, doi: 10.1088/1361-6501/ADF90F.

[14] K. Shang, Y. Yang, J. Wang, M. Shuai, Z. Dang, and Q. Liu, "Dynamic Risk Assessment of Wind Farms Under Extreme Gust Disturbances Using BiLSTM, Adaboost, and Adaptive Kernel Density Estimation for Enhanced Prediction," Electronics, vol. 14, no. 23, pp. 4584, 2025, doi: 10.3390/ELECTRONICS14234584.

[15] S. Karuppiah, A. J. Rose, C. Subasini, and F. V. F. Sangeetha, "Enhanced medical image analysis using hybrid Henry gas Solubility optimization algorithm with optimized AdaBoost stacked neural networks," Expert Systems With Applications, vol. 296, no. Part C, Art. no. 128882, 2026, doi: 10.1016/J.ESWA.2025.128882.

[16] Z. Li, Y. Xiang, L. Pei, L. Wan, and C. Wang, "Oversampling AdaBoost Cycle Slip Detection Method for Small Sample Data in GNSS Signal Processing," Journal of Shanghai Jiaotong University (Science), vol. (prepublish), pp. 1-11, 2025, doi: 10.1007/S12204-025-2866-9.

[17] K. S. Sahoo, A. Samantra, C. Mohapatra K, P. B. Swain, Z. Ali, and G. Muhammad, "LABDT (Lexi AdaBoost Decision Tree): An Approach for Legal Outcome Prediction Fusing Lexical, Semantic, and Similarity-based features," International Journal of Computational Intelligence Systems, vol. 18, no. 1, pp. 279, 2025, doi: 10.1007/S44196-025-01003-2.

[18] X. Fang, B. Xin, Z. Zhan, A. A. M. Newton, F. Yan, and Z. Jin, "Classification of wool and cashmere fiber based on LBP and DWT features: performance comparison of random forest, AdaBoost, and KNN classifiers," The Journal of The Textile Institute, vol. 116, no. 11, pp. 2663-2675, 2025, doi: 10.1080/00405000.2025.2450851.

[19] H. Zhong, R. Yuan, S. Yu, J. Yi, M. Zhao, and H. Wu, "SCGAN-TrAdaBoost: a skip-connection generative adversarial network with transfer AdaBoost for bearing fault diagnosis with limited data," Measurement Science and Technology, vol. 36, no. 10, Art. no. 106127, 2025, doi: 10.1088/1361-6501/AE0CEA.

[20] R. Sayeri, B. Barzegar, B. Y. Rad, N. Mikaeilvand, and T. H. M. Najaran, "Evolutionary AdaBoost ensemble: A machine learning framework for depression detection," Machine Learning with Applications, vol. 22, Art. no. 100748, 2025, doi: 10.1016/J.MLWA.2025.100748.

[21] Z. Meng, Z. Wang, X. Liu, S. Liu, and Y. Zhang, "Real-time individual subway destination prediction: An AdaBoost graph neural network," Engineering Applications of Artificial Intelligence, vol. 160, no. Part A, Art. no. 111830, 2025, doi: 10.1016/J.ENGAPPAI.2025.111830.

[22] K. Zhu, S. Hu, Y. Kong, J. Zhou, J. Teng, W. Luo, et al., "Enhancing Sustainable Disaster Risk Management: Landslide Susceptibility Evaluation Using AdaBoost-CB Ensemble and Multi-Dimensional Vegetation Metrics in Yuanling County, China," Sustainability, vol. 17, no. 21, pp. 9358, 2025, doi: 10.3390/SU17219358.

[23] S. Chidambaram, S. Ramesh, S. Geetha, and D. P. C. Cyril, "Early Detection of Cardiovascular Disease Using AdaBoost Convolutional Random Arithmetic Trigonometric Algorithm," Cognitive Computation, vol. 17, no. 6, pp. 153, 2025, doi: 10.1007/S12559-025-10511-4.

[24] Y. Han, A. An, W. Li, and H. Dong, "A New Incipient Faults Diagnosis Method Combining SAE and AdaBoost Algorithm for Vehicle Power Supply with Imbalanced Datasets," Processes, vol. 13, no. 10, pp. 3343, 2025, doi: 10.3390/PR13103343.

[25] H. Yao and C. Wan, "Multi-factor portfolio optimization: A combined random Forest–AdaBoost model with cost-sensitive learning," Pacific-Basin Finance Journal, vol. 94, Art. no. 102946, 2025, doi: 10.1016/J.PACFIN.2025.102946.

[26] W. Ding, C. Zhang, X. Li, L. Zhang, and N. Chen, "Enhanced forecasting of coastal chlorophyll-a through AdaBoostoptimized LSTM models," Acta Oceanologica Sinica, vol. 44, no. 7, pp. 147-160, 2025, doi: 10.1007/S13131-025-2492-2.

[27] C. Li, B. Chen, and S. Li, "Graph discriminative dynamic sampling AdaBoost for imbalanced cardiovascular disease diagnosis," Expert Systems With Applications, vol. 297, no. Part C, Art. no. 129546, 2026, doi: 10.1016/J.ESWA.2025.129546.

[28] L. Qi, T. Kawaguchi, and S. Hashimoto, "Detection and Classification of Low-Voltage Series Arc Faults Based on RF-Adaboost-SHAP," Electronics, vol. 14, no. 19, pp. 3761, 2025, doi: 10.3390/ELECTRONICS14193761.

[29] A. Wiangkham, A. Ariyarit, P. Aengchuan, W. Homjabok, N. Wattikornsirikul, A. Koptonglang, et al., "Artificial intelligence-based optimization of pack carburizing parameters for low carbon steel with different wood charcoals using AdaBoost-GRNN," The International Journal of Advanced Manufacturing Technology, vol. 140, pp. 5399-5418, 2025, doi: 10.1007/S00170-025-16508-5.

[30] E. Guo, G. Wang, J. Hu, W. Yan, P. Zhao, and A. Zhang, "Stacked BiLSTM–Adaboost Collaborative Model: Construction of a Precision Analysis Model for GABA and Vitamin B9 in the Foxtail Millet," Agronomy, vol. 15, no. 9, pp. 2077, 2025, doi: 10.3390/AGRONOMY15092077.