Real-Time Data Acquisition in SCADA Systems: A JavaWeb and Swarm Intelligence-Based Optimization Framework

Lingyi Sun[1], Tieliang Sun[2], Ruojia Xin[3], Feng Yan[4], Yue Li[5], Hengyu Wang[6], Yecen Tian[7], Dongqing You[8], Yun Liu[9], Muhao Lv[10]
Pipe China Oil and Gas Control Center, Beijing, 10020, China[1, 2, 4, 5, 6, 7, 8, 9, 10]
Kunlun Digital Technology Co., Ltd. Beijing, 10020, China[3]

***Abstract*—This paper aims to improve the accuracy and efficiency of SCADA software design and testing for oil and gas pipelines. It proposes a JavaWeb-based SCADA software solution optimized by the Bird Foraging Search (BFS) algorithm combined with an Echo State Network (ESN) for enhanced testing and analysis. A multi-tiered distributed SCADA software architecture based on the Java EE framework was designed to provide real-time data acquisition, monitoring, control, and data analysis. The BFS algorithm was used to optimize the hyperparameters of the ESN model to improve testing accuracy and convergence speed. The BFS-ESN model was compared with other optimization algorithms such as PSO and DE. Experimental results show that the BFS-ESN model achieved a testing accuracy of 97.33% and faster convergence within 700 iterations. It outperformed other algorithms in both accuracy and convergence speed. The JavaWeb-based SCADA software design for oil and gas pipelines is feasible, and the BFS-ESN model significantly enhances the accuracy and efficiency of SCADA software testing. This approach demonstrates the potential for application in SCADA systems, with future research needed to simplify the model and extend its applicability for large-scale deployment.**

Keywords*—*JAVAWeb; oil and gas pipelines; SCADA software; design analysis; bird foraging search algorithm

I. Introduction

Supervisory Control and Data Acquisition (SCADA) is a kind of industrial control system for long-distance data acquisition and monitoring [1], which is widely used in long-distance oil and gas pipelines [2], and its ability to work properly directly affects the production and operation of oil and gas pipelines and even the national oil and gas industry [3]. In recent years, with the great development of pipeline construction, the normal operation of the pipeline has had a direct impact on the production and operation of oil and gas pipelines and even the national oil and gas industry [3]. In recent years, with the great development of pipeline construction, the pipeline scale is getting bigger and bigger, the demand for SCADA system software is getting higher and higher, the database management scale reaches millions of nodes, the supported devices are diversified, and the number of collection devices is required to reach hundreds or even thousands [4]. In order to improve the efficient, stable and safe operation of the system, combined with Java Web technology, the design of SCADA software, to build multi-level, distributed applications [5]. In the consideration of safe production, after the SCADA software for oil and gas pipelines passes the software test in the laboratory simulation environment, it is also necessary to select real scenarios at the operation site and use the SCADA software to carry out a controlled range of trial operation [6]. Therefore, the study of SCADA software analysis for oil and gas pipelines based on Java Web technology is of great significance for software design, which is not only conducive to the efficiency of software design, but also contributes to the efficient, stable and safe operation of SCADA systems [7].

The research on the design and analysis of SCADA software for oil and gas pipelines based on Java Web technology mainly includes the research on the demand analysis of SCADA system for oil and gas pipelines, Java Web development of SCADA software, and test analysis of SCADA software [8]. Oil and gas pipeline SCADA system requirements analysis is mainly to improve the performance of oil and gas pipeline SCADA system, study in [9] from the perspective of market research, analysed the existence of pipeline transmission characteristics, for this feature, designed a long-distance oil and gas pipeline SCADA system; study in [10] from the perspective of economic benefits, research on oil and gas pipeline SCADA project construction specific design ideas. SCADA software Java Web development is mainly to study the process of developing SCADA software based on Java Web technology, study in [11] used Java Web technology to develop SCADA software for oil and gas pipelines, and introduces the relevant functions and structure. SCADA software testing and analysis research is mainly the use of software testing process, according to the test indexes, analyse the effectiveness and functionality of SCADA software. The effectiveness and functionality of SCADA software is analysed according to the testing indexes. Currently, data-driven algorithms are used to test SCADA software, including neural networks [12], SVM [13], LSTM [14] and other methods. Although scholars at home and abroad have done a lot of rich research on the design and analysis of SCADA software for oil and gas pipelines based on Java Web technology, there are still some problems [15]: firstly, there are fewer research studies on the development of SCADA software for oil and gas pipelines based on Java Web technology; secondly, there are fewer methods for testing SCADA software for oil and gas pipelines; and lastly, the testing methods for SCADA software accuracy is not high enough.

In order to improve the accuracy of SCADA software testing and analysis methods for oil and gas pipelines based on Java Web technology, this paper provides relevant ideas for the design problems of SCADA software for oil and gas pipelines based on Java Web technology, focusing on the key problems of SCADA software design and testing for oil and gas pipelines, combining the Echo State Network [16] with the Bird Foraging Search Algorithm [17], and proposing a method based on the BFS-ESN network in Java Web framework of oil and gas pipeline SCADA software design testing method. The experimental results of relevant data analysis show that the SCADA software design method for oil and gas pipelines based on BFS-ESN network under the Java Web framework realises the SCADA software design and analysis problems of oil and gas pipelines, and improves the accuracy and efficiency of the SCADA software design test.

II. SCADA Software for Oil and Gas Pipelines

A. Java Web Technologies

Java Web is a web development framework based on Java technology [18] for building dynamic websites and enterprise applications. It includes a variety of technologies and server-side components, such as Servlet, JSP (JavaServer Pages), Spring, Hibernate, as well as web servers and application servers, such as Apache Tomcat, Jetty, GlassFish and WildFly. The Java Web technology system covers the full range of development needs from front-end presentation to back-end logic processing. Java Web technology system covers a full range of development needs from the front-end display to the back-end logic processing. Java Web schematic diagram shown in Fig. 1.

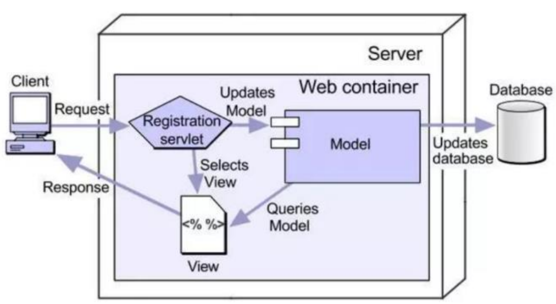


Fig. 1. Java web technology.

The technology system of Java Web includes front-end technologies (HTML, CSS, JavaScript) and back-end technologies (Servlet, JSP, MVC frameworks, database interaction technologies) [19]. Developers can use these technologies to create dynamic web pages, handle user requests, access databases, and implement business logic. The Java Web technology system and roles are shown in Fig. 2.

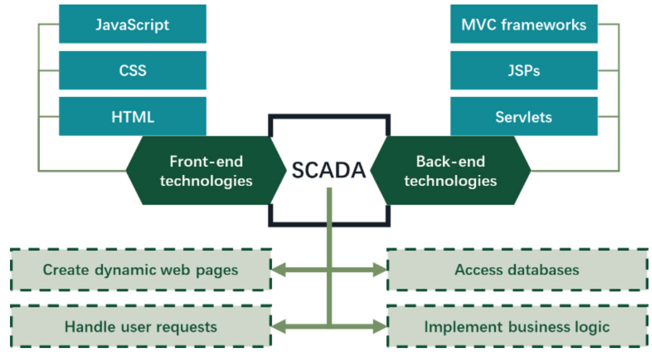


Fig. 2. Java Web technology system and role.

Java Web servers usually work based on a B/S (Browser/Server, browser and server) model (Fig. 3). The client sends an HTTP request to the server through the browser, and the server receives the request and processes it according to the type of request (static resource or dynamic resource). If it is a dynamic resource, the server forwards the request to the Web container (Tomcat), which loads the Servlet or JSP page, executes the corresponding business logic, and returns the generated dynamic content to the client browser.

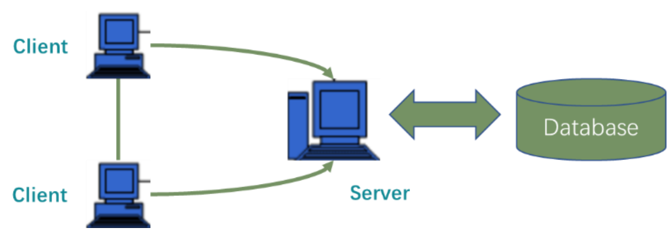


Fig. 3. Java web server fundamentals.

B. SCADA System for Oil and Gas Pipelines

Oil and gas pipeline SCADA (Supervisory Control And Data Acquisition) [20] systems are key industrial control systems used to monitor and control oil and gas transmission processes. Java Web-based SCADA software design typically involves the use of the Java EE (Jakarta EE) technology stack to build multi-tiered, distributed applications that provide real-time data acquisition, monitoring, control, and data analysis, as shown in Fig. 4.

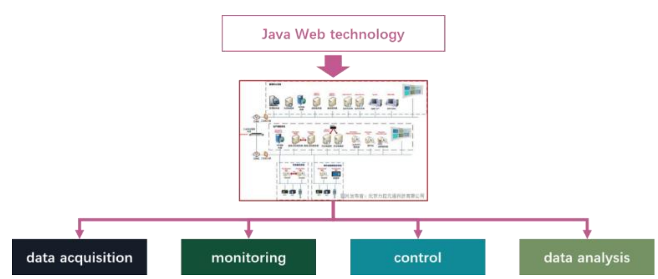


Fig. 4. Logical relationship between SCADA and Java Web.

In designing Java Web-based SCADA software for oil and gas pipelines, the multi-tier architecture provided by the J2EE platform can be used, including the client layer, Web layer, business logic layer, and enterprise information system layer [21]. Such an architecture helps to improve the modularity, maintainability, and scalability of the system.1) The client layer can be a Web-based interface for remote access; 2) the Web layer handles HTTP requests and responses and provides the user interface; 3) the business logic layer contains components that process data and execute control logic; and 4) the enterprise information system layer is responsible for integrating with external systems such as energy Management System (EMS). Fig. 5 shows Java Web-based SCADA software design architecture.

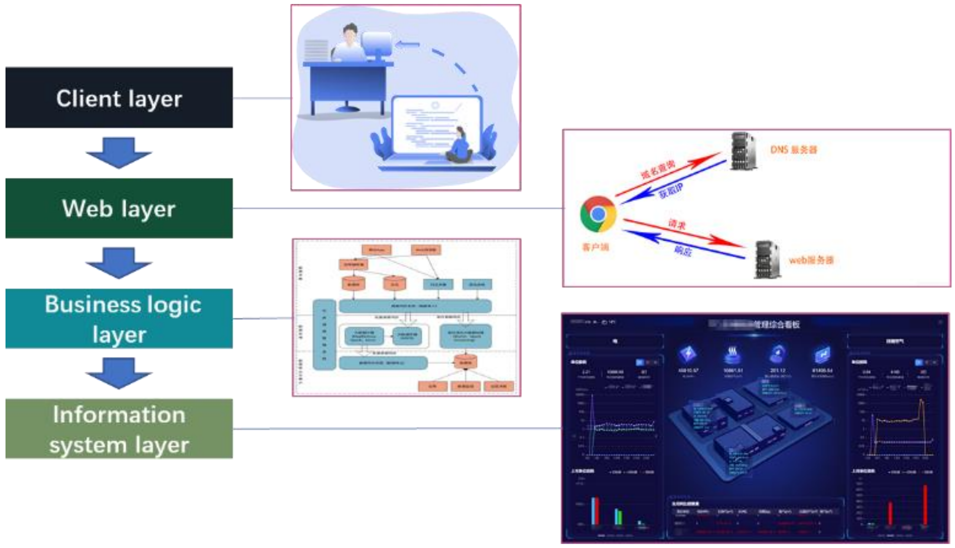


Fig. 5. Java Web-based SCADA software design architecture.

C. Design Thinking

According to the Java Web-based SCADA software design architecture for oil and gas pipelines, this paper uses various technologies and services provided by Java EE (Servlet, JavaServer Pages (JSP), Enterprise JavaBeans (EJB), Java Message Service (JMS), Java Transaction API (JTA)) to achieve high performance, reliability and security of the system; using a database management system (DBMS) to store historical data and real-time data; the use of Web services and RESTful APIs to achieve inter-system communication and data exchange, as shown in Fig. 6.

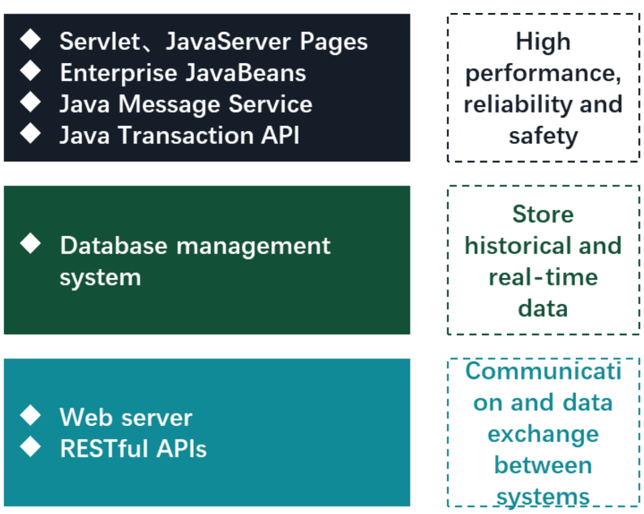


Fig. 6. Key technologies and tools.

With the help of these tools and technologies, security and reliability are fully considered to ensure that real-time data monitoring and control response can be provided, and interface interaction is designed to enable operators to easily monitor pipeline status and perform necessary control operations. The design flow of Java Web-based SCADA software is shown in Fig. 7.

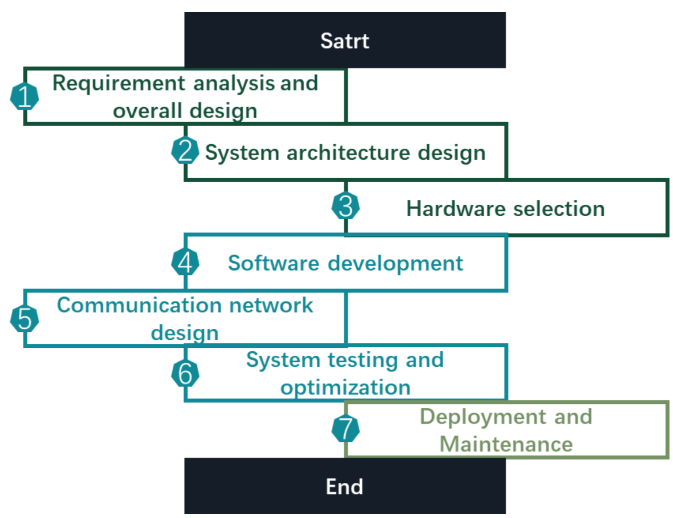


Fig. 7. Design flow.

III. Software Design Application and Test Analysis

A. Echo State Network

One type of recursive neural network is the echo state network (ESN) [22], which is typically used for time series prediction [19] and builds the network hidden layer using the "reserve pool" technique. The input layer, storage layer, and output layer comprise the vast majority of ESN. Fig. 8 illustrates its precise configuration. Following the initial randomization, the connection weights between the input layer and the storage pool $\boldsymbol{W}_{in}^{r\times n}$ are not trained and won't be altered. Similarly, the state feedback weight $\boldsymbol{W}^{r\times r}$ is arbitrarily initialized and does not require training, while the reserve pool input is derived from the output of the previous state of the input layer and the reserve pool, respectively. A reserve pool for the output layer weights $\boldsymbol{W}_{out}^{m\times r}$ must be trained, typically using the Ridge regression (Ridge regression) method for connection weights. This method is expressed as follows:

$$\boldsymbol{W}_{out} = \boldsymbol{Y}_{long}\boldsymbol{H}^T\left(\boldsymbol{H}\boldsymbol{H}^T + \lambda_r \boldsymbol{I}\right)^{-1} \tag{1}$$

The storage pool state and the regularization factor are represented by $\boldsymbol{H}$ and λ_r, respectively. The condition $\boldsymbol{H}$ of the pool is depicted below:

$$\boldsymbol{H}(t) = \tanh\left(\boldsymbol{W}_{in}\boldsymbol{X}_{long}(t) + \boldsymbol{W}\boldsymbol{H}(t-1)\right) \tag{2}$$

where the hyperbolic tangent activation function is indicated by tanh.

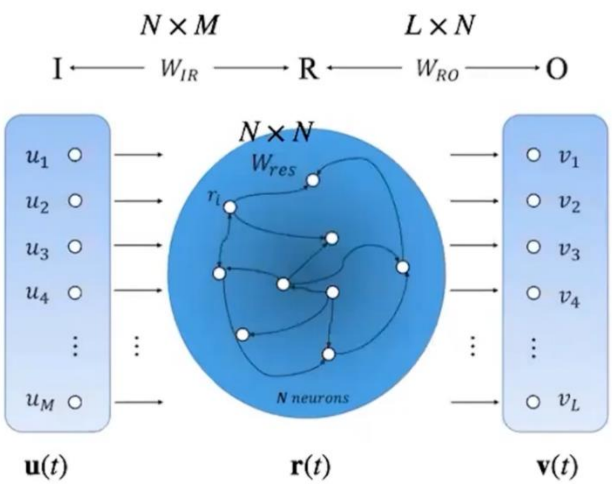


Fig. 8. Echo state neural network.

The Echo State Network (ESN) algorithm consists of two main phases: the initialization of weight parameters and the training process. During the initialization phase, the connection weights within the network are randomly generated. Specifically, the connection weights between neurons in the reservoir (or storage pool) are sparsely distributed, meaning that not all neurons are connected, which helps reduce computational complexity. In the training phase, only the weights connecting the reservoir to the output layer are adjusted, typically using methods like Ridge regression, while other weights remain fixed [23], where the ESN hyperparameters include the size of the storage pool N_r, the spectral radius SR, and the input scaling factor IS, storage pool sparsity SD.

A key characteristic of ESNs is the relatively large number of neurons present in the reservoir, which allows the network to capture complex dynamics. However, the performance of an ESN is highly dependent on its hyperparameters, making hyperparameter tuning a crucial step for optimal performance. The key hyperparameters include the size of the reservoir, which determines how much internal memory the network has; the spectral radius, which controls the stability and dynamic range of the network; the input scaling factor, which influences the sensitivity of the reservoir to input signals; and the sparsity of the reservoir, which affects the degree of connectivity between neurons. Proper adjustment of these hyperparameters is essential for improving the prediction accuracy of the ESN, especially in time series forecasting and other complex tasks.

B. Bird Foraging Search Algorithm

Drawing inspiration from the diverse behaviors exhibited by birds while foraging, this paper introduces a new swarm intelligence optimization algorithm known as the Birds Foraging Search (BFS) algorithm [17]. Each phase contributes uniquely to the optimization process.

In the flight search behavior phase, birds engage in exploratory movements, mimicking how birds scout for food sources over large areas. This phase primarily focuses on exploration, allowing the algorithm to cover a wide search space. The domain behavior phase, on the other hand, emphasizes exploitation, where birds refine their search within specific territories to locate better resources more precisely. The balance between these two phases—exploration and exploitation—is essential for optimizing the algorithm's performance. The cognitive behavior phase enhances the algorithm's search efficiency by allowing individual birds to learn from their past experiences. This phase simulates self-learning, enabling birds to adjust their strategies based on previously gathered information, which helps in avoiding redundant searches and accelerating convergence. The detailed search strategies for each phase are illustrated in Fig. 9.

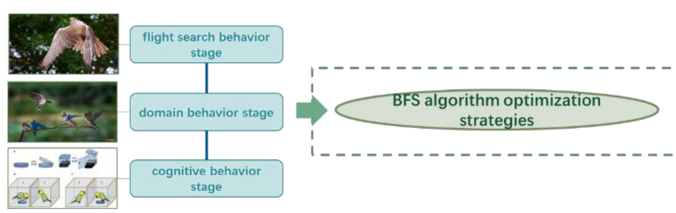


Fig. 9. Search strategy of the BFS algorithm.

The initial position of the bird in space is represented as follows:

$$X_i = UB - r_1 \cdot (UB - LB) \tag{3}$$

where X_i denotes the position of the ith bird, UB and LB denote the upper and lower bounds of the search space, respectively, and r_1 is a random value.

1) Flight search behaviour stage: Birds of prey, like falcons, typically choose areas with high potential for finding prey and hover over these regions while searching for food. Through extensive research, scientists have observed that the flight patterns of these raptors closely resemble a logarithmic spiral. This spiral movement allows the birds to efficiently cover a specific area, increasing their chances of locating prey. The spiral pattern is mathematically defined, and its unique structure enables the predator to maintain focus on a targeted zone while gradually expanding or contracting their search radius, optimizing their hunting strategy (as shown in Fig. 10):

$$X_i^{iter+1} = D_{i-p} \cdot e^{\theta} \cdot \cos(2\pi\theta) + PA^{iter} \quad (4)$$

where $i \in [1, 2, 3, \ldots, N]$, and N is the population size. θ is a random number between -1 and 1. $D_{i-p} = \left| X_i^{iter} - PA^{iter} \right|$ denotes the distance from the i[th] bird to the potential region, X_i^{iter} denotes the position of the i[th] bird at iter iteration, and PA^{iter} denotes the position of the potential region at iter iteration.

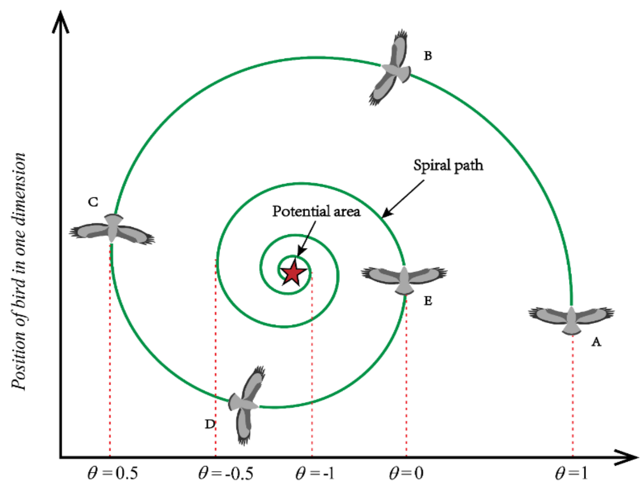


Fig. 10. Logarithmic spiral flight pattern.

2) Stages of domain behavior: In the Birds Foraging Search (BFS) algorithm, these two types of birds follow different movement behaviors. The territorial bird focuses on patrolling a small region around its current position, both to search for better food sources and to defend its territory from other competing birds. This localized exploration allows the territorial bird to refine its current solution while preventing others from entering its domain. Meanwhile, the invasive birds attempt to enter the territory in search of resources. The behavior of the territorial bird, balancing exploration and defense, can be mathematically modeled as follows:

$$X^{T,iter+1} = X^{T,iter} + r_d \cdot \lambda \quad (5)$$

Where, $X^{T,iter}$ is the position of the territorial bird at iterth iteration; r_d is a random value between -1 and 1, representing the search direction of the territorial bird; λ is a scale factor, which is generally set to $(X^{T,iter} - X^{S,iter})$, and $X^{S,iter}$ is the position of the suboptimal individual.

Once a territorial bird has claimed a region as its exclusive domain, all invading birds will begin moving toward that area in search of resources. At this stage, the territorial bird takes defensive action to protect its claimed space, using warning calls or chirping to deter the invaders. This defense mechanism can lead to two possible outcomes.

a) Scenario 1: In this case, the invasive bird's position is updated as it moves closer to the territory. The invasive bird's persistence reflects a focus on resource acquisition, and the algorithm models this movement as a rapid progression toward the territorial bird's domain, following a predefined mathematical update rule. This allows the invasive bird to keep adjusting its position despite the territorial bird's efforts to protect its resources.

$$X_j^{I,iter+1} = X_j^{I,iter} + r_2 \cdot (X^{T,iter} - IF \cdot X_j^{I,iter}) \quad (6)$$

Where, $X_j^{I,iter}$ is the position of the jth invasive bird at the iterth iteration, r_2 is a random number between 0 and 1; IF is the invasion factor, which determines how the position of the invasive bird changes, and the specific effects are shown in Fig. 11.

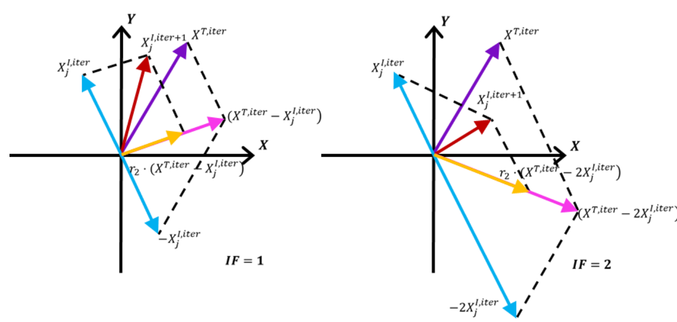


Fig. 11. Vector representation of invasive bird movements with different invasion factors.

b) Scenario 2: When the bird continues its search but fails to discover an improved solution, its current position remains unchanged from the previous one($X_i^{iter} = X_i^{iter-1}$). In this scenario, the bird's movement is modeled using a Gaussian distribution, allowing for random variations in its search path. This randomness helps the bird explore new areas more effectively.

$$X_{j,d}^{I,iter+1} = X_{j,d}^{I,iter} + r_3 \cdot (X_{k,d}^{I,iter} - X_{m,d}^{I,iter}) + r_4 \cdot (X_{l,d}^{I,iter} - X_{h,d}^{I,iter}) \quad (7)$$

The Scenarios 1 and 2 optimisation process is summarised as:

$$\begin{cases} X_j^{I,iter+1} = X_j^{I,iter} + r_2 \cdot (X^{T,iter} - IF \cdot X_j^{I,iter}) & if\ P^{iter} \le rand \\ X_{j,d}^{I,iter+1} = X_{j,d}^{I,iter} + r_3 \cdot (X_{k,d}^{I,iter} - X_{m,d}^{I,iter}) + r_4 \cdot (X_{l,d}^{I,iter} - X_{h,d}^{I,iter}) & otherwise \end{cases} \quad (8)$$

3) Cognitive behavioural stage: Cognitive behavior in birds is essentially a self-learning process that draws on accumulated experience. Birds use the information gathered from previous searches to refine their strategies, helping them avoid redundant or inefficient searching. This self-guided learning enhances their foraging efficiency, allowing them to focus on more promising areas. Cognitive behavior can be broken down into two distinct parts.

a) Scenario 1: In this case, the birds are continuously discovering better food sources, which means their current position differs from the previous one $X_i^{iter} \neq X_i^{iter-1}$. Here, the birds actively learn by leveraging the gradient information from their previous searches. By following this targeted search approach, they adjust their movements in a way that is guided by prior successes, which ultimately helps them refine their foraging strategy. This focused learning process not only allows them to zero in on better resources but also significantly speeds up the algorithm's convergence. The efficiency gained from this self-learning behavior means the birds are more likely to optimize their search pattern faster, avoiding unnecessary wandering and making the overall process more streamlined. The specific mathematical calculations for updating their positions during this phase are based on their learning from previous gradients.

$$X_i^{iter+1} = X_i^{iter} + r_5 \cdot (X_i^{iter} - X_i^{iter-1}) \tag{9}$$

Where X_i^{iter} and X_i^{iter-1} are the i-th bird at iter-th iteration and iter-1 iteration position respectively; r_5 is a random number between 0 and 1.

b) Scenario 2: The bird continues to search but fails to find a better result. That is, the current position is the same as the previous position ($X_i^{iter}=X_i^{iter-1}$). The process is implemented based on Gaussian distribution:

$$X_i^{iter+1}=Gaussian(X_{best}^{iter},\ \ \xi) \tag{10}$$

Where $Gaussian(X_{best}^{k},\ \ \xi)$ denotes a Gaussian distributed random number with mean X_{best}^{k} and standard deviation ξ ; X_{best}^{iter} is the population optimal solution for iter iteration number.

$$\xi = (\log(iter)/iter) \cdot abs(X_{best}^{iter} - r_6 \cdot X_i^{iter}) \tag{11}$$

where r_6 is a random value and $\log(iter)/iter$ denotes the size used to adjust the standard deviation.

To prevent such ineffective searches, BFS introduces a boundary control policy:

$$X_{i,d}^{iter} = UB - r_7 \cdot (UB - LB) \quad if \ \ X_{i,d}^{iter} < LB \ \ or \ \ X_{i,d}^{iter} > UB \tag{12}$$

where $X_{i,d}^{iter}$ denotes the position of the dth dimension of the ith individual at the iter iteration number and r_7 denotes the random value.

According to the optimisation strategy of the BFS algorithm, the pseudo code is shown in Table I.

TABLE I. BIRD FORAGING SEARCH ALGORITHM

Algorithm 1: Bird Foraging Search Algorithm
Set the parameter values UB, LB, N and Max_iter;
Random initialisation of bird populations in the solution space;
iter=1;
While iter<=Max_iter do
// Perform flight search phase
Individuals are generated by logarithmic spiral flight;
// Behavioural stages in the field of implementation
/* Territorial birds */
Generate a new location for the territorial bird;
/* Invasive bird */
Calculate the new location of the invasive bird;
// Implementation of the cognitive-behavioural stage
Consolidation of stocks;
Detecting transgressive constraints;
Evaluate the new location of the individual bird, update the location
End while
Output the optimal solution.

C. BFS-ESN Method

In order to improve the feasibility and accuracy of SCADA software design test for oil and gas pipelines based on Java Web technology, this paper uses to optimise the hyperparameters of ESN network using BFS algorithm to construct the SCADA software design test model, and the specific structure is shown in Fig. 12.

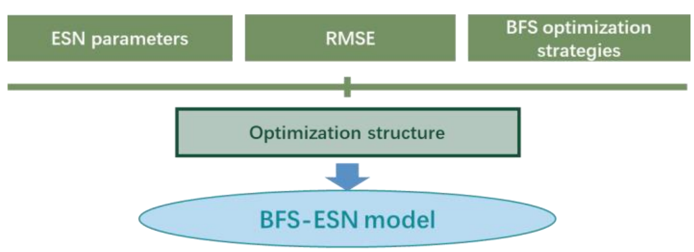


Fig. 12. BFS-ESN structure.

BFS-ESN uses ESN hyperparameters, i.e., storage pool size N_r , spectral radius SR , input scale factor IS , and storage pool sparsity SD as the optimisation variables, and the coding mode is real number coding mode, the specific coding structure is shown in Fig. 13.

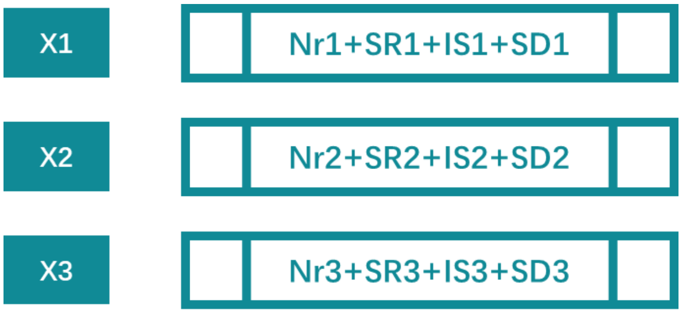


Fig. 13. BFS-ESN coding structure.

BFS-ESN uses RMSE as the optimisation fitness value and the on-the-fly search behavioural phase, domain behavioural phase, and cognitive behavioural phase of the BFS algorithm as the ESN optimisation strategy.

D. Test and Analysis Method for Designing SCADA Software

Taking the BFS-ESN model as the software testing model, in order to solve the oil and gas pipeline SCADA software design testing analysis, this paper proposes the oil and gas pipeline SCADA software design testing analysis method based on the BFS-ESN model, and the specific application flowchart is shown in Fig. 14. The application process of the BFS-ESN algorithm consists of the following processes: 1) According to the user's security and reliability of the system The application process of the BFS-ESN algorithm includes the following processes: 1) according to the user's demand for system safety and reliability, specify the actions and functions to be achieved by the control object; 2) design the topology of the SCADA system; 3) develop the front-end interface and the back-end logic of the SCADA software by using the JavaWeb technology; and 4) test the system by using the BFS-ESN model after the completion of the software development.

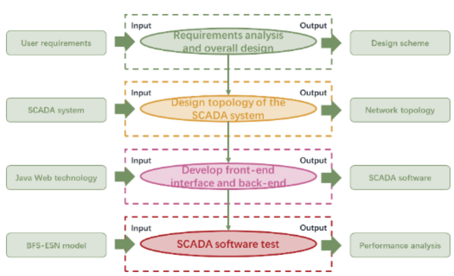


Fig. 14. Flow of BFS-ESN algorithm application.

IV. RESULTS AND DISCUSSION

A. Experimental Set-up

In order to verify the effectiveness and feasibility of the method proposed in this paper, this paper adopts the oil and gas pipeline SCADA system software and related data as the analysis data, and selects PSO, DE, ABC, CS, GSA, FA as the comparison algorithms of hyperparameter optimisation of the ESN network, and the parameter settings of each algorithm are shown in Table II. The population sizes of PSO, DE, ABC, CS, GSA, FA algorithms are chosen as 100 and the maximum number of iterations is set to 1000.

TABLE II. PARAMETER SETTINGS OF THE OPTIMISATION ALGORITHM FOR THE COMPARISON MODEL

Arithmetic	Parameterisation
PSO	w=0.6, c1=c2=2
DE	F=0.5, CR=0.9
ABC	limit=(N*D)/2, hired bees=scout bees=0.5N
CS	G0=100, α=20
GSA	β = 1.5, p = 0.25
FA	α = 0.2, β0 = 1 and γ = 1
BFS	Parameter-free optimisation algorithm

The experimental simulation system is Wins 10 and the programming language is Matlab2024a.

B. Analysis of Test Results

1) BFS optimisation performance analysis: In order to analyse the optimisation performance of BFS algorithms, in this section, the single peak benchmark functions (F01-F06) are used to analyse and compare the BFS, PSO, DE, ABC, CS, GSA and FA algorithms, and the basic information of the functions is shown in Table III.

The optimisation curves of the BFS, PSO, DE, ABC, CS, GSA and FA algorithms are given in Fig. 15. From Fig.15, it can be seen that in terms of convergence speed, BFS is significantly better than the remaining six algorithms; in terms of convergence accuracy, BFS is significantly better than the other optimisation algorithms.

TABLE III. TEST FUNCTION DESCRIPTION

Test function	name (of a thing)	dimension (math.)	realm	optimum value
F01	Sphere	30	[-100,100]	0
F02	Schwefel2.22	30	[-10,10]	0
F03	Schwefel 1.2	30	[-100,100]	0
F04	Schwefel2.21	30	[-100,100]	0
F05	Rosenbrock	30	[-30,30]	0
F06	Step	30	[-100,100]	0

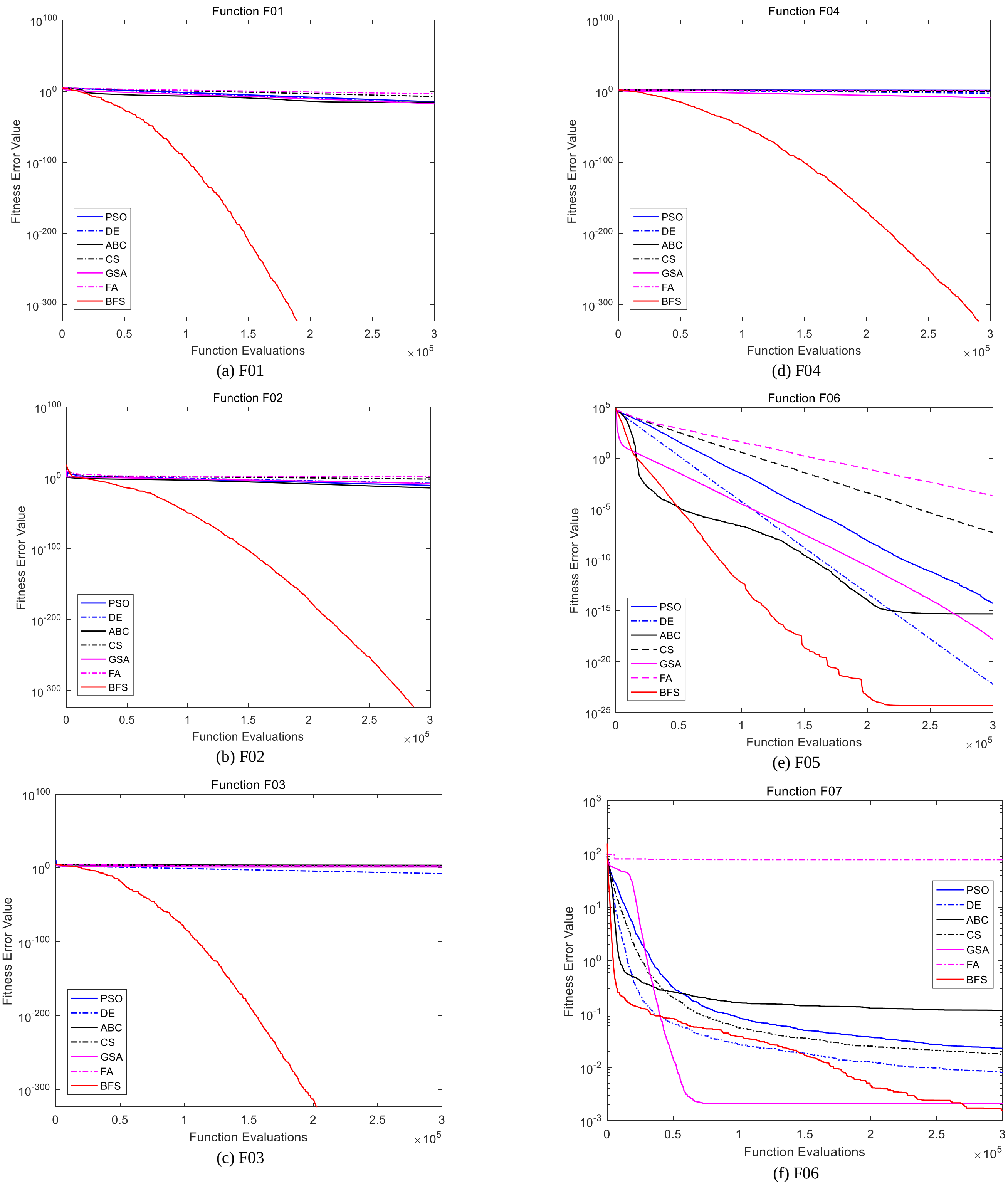


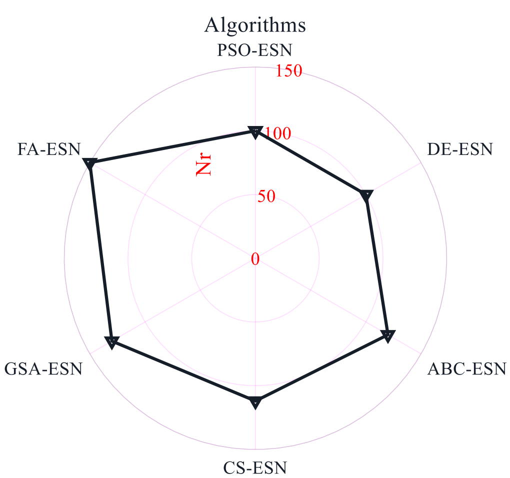
Fig. 15. Flow of BFS-ESN algorithm application.

2) Java web design software effectiveness analysis: Using Java Web technology, this paper designs SCADA system software for oil and gas pipelines, and the specific effect diagram is shown in Fig. 16. As can be seen from Fig. 16 the oil and gas pipeline SCADA system software achieves the functions of real-time data acquisition, monitoring, control and data analysis.

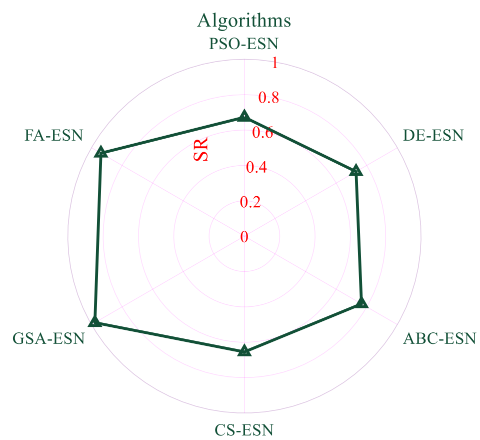
Fig. 16. Flow of BFS-ESN algorithm application.

3) SCADA software test and analysis: In order to verify the validity and feasibility of the SCADA software testing and analysis method based on the BFS-ESN model, this section uses PSO, DE, ABC, CS, GSA, FA algorithms to optimise the ESN model for comparative analysis with the BFS-ESN, and the results are shown in Fig. 16 and Fig. 17.

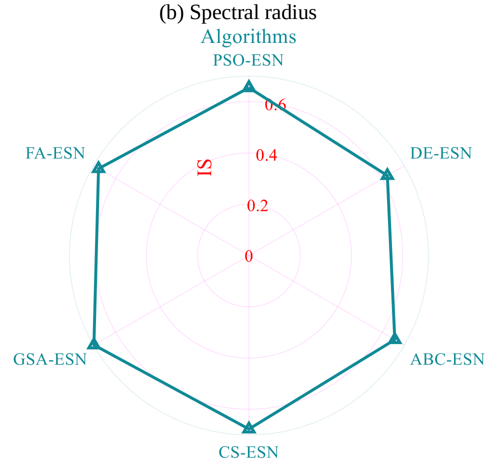
Fig. 17 gives the results of optimising the ESN hyperparameters for the comparison algorithms. As can be seen from Fig. 17, the results obtained by BFS for optimising the hyperparameters of the ESN model are as follows: storage pool size $N_r = 150$, spectral radius $SR = 0.9381$, input scale factor $IS = 0.6772$, storage pool sparsity $SD = 0.4951$.



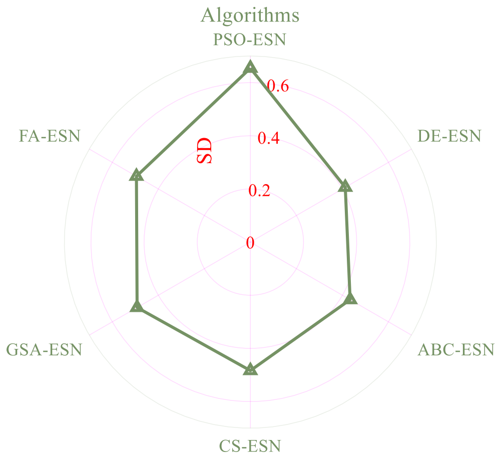
(a) Size of storage tanks



(b) Spectral radius



(c) Input scale factor



(d) Sparsity of storage tanks

Fig. 17. Contrasting algorithms to optimise ESN hyperparameters results.

The optimisation iteration curves of the compared algorithms are given in Fig. 18. From Fig. 18, it can be seen that the test accuracy of SCADA software design for oil and gas pipelines in Java Web framework based on BFS-ESN model is better than other algorithms, the optimisation iterations converge faster than other algorithms, and an accuracy of 97.33% is obtained at 700 iterations.

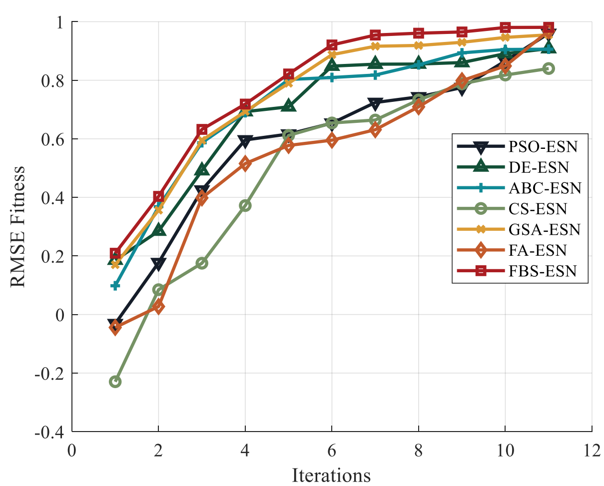


Fig. 18. Contrasting algorithm optimisation iteration curves.

V. Conclusion and Outlook

This paper presents the design and testing methods for oil and gas pipeline SCADA software based on JavaWeb technology, with the aim of improving accuracy, efficiency, and real-time data monitoring. The key innovation lies in the integration of the Bird Foraging Search (BFS) algorithm with the Echo State Network (ESN) to optimize the SCADA software design and testing process.

1) The paper outlines a multi-tiered software architecture using Java EE technologies to ensure the efficient operation of SCADA systems for oil and gas pipelines. The design includes real-time data acquisition, monitoring, control, and data analysis functionalities.

2), A novel combination of the Bird Foraging Search algorithm and Echo State Network is proposed to optimize the software testing process. This method enhances accuracy and convergence speed in SCADA system testing, outperforming other algorithms like PSO and DE.

3), Experimental results using actual SCADA data confirm the feasibility of the BFS-ESN model, showing significant improvements in testing accuracy, reaching 97.33% accuracy with faster convergence rates compared to traditional methods.

We also find some shortcomings: Firstly, the validation is based on specific SCADA data from oil and gas pipelines, which may limit the generalizability of the proposed method to other industrial applications. Secondly, while the model improves accuracy, its complexity might pose challenges in real-world implementation, particularly in terms of computational resource requirements. Finally, although the paper discusses system security briefly, the analysis of potential cybersecurity risks and scalability issues, particularly in larger pipeline networks, is not fully explored.

Future studies could extend the BFS-ESN approach to different SCADA systems in industries beyond oil and gas, to verify its broader applicability and performance. Reducing the computational complexity of the BFS-ESN model to make it more suitable for large-scale deployments in real-time environments. Exploring the integration of advanced security protocols and testing methods could strengthen the robustness of SCADA systems against potential cyber threats.

References

[1] Chen X. Zhou F., Hao X., Gong X., Lin X. Development Status, Challenges and Suggestions of SCADA System in China[J]. Industrial Technology Innovation, 2015, 2(1): 103-114.

[2] Moynihan B , Tronci E M , Hughes M C ,Moaveni B, Hines E.Virtual sensing via Gaussian Process for bending moment response prediction of an offshore wind turbine using SCADA data[J].Renewable Energy, 2024, 227.

[3] Deshpande S N , Jogdand R .A novel scheduling algorithm development and analysis for heterogeneous IoT protocol control system to achieve SCADA optimization: a next generation post covid solution[J].International Journal of Information Technology, 2023, 15:2123 - 2131.

[4] Gill S .SCADA/HMI software market to grow and drive digital transformation[J].Control Engineering Europe, 2023(05).

[5] Morrison R , Liu X , Lin Z .Anomaly detection in wind turbine SCADA data for power curve cleaning[J].Renewable Energy, 2022, 184.

[6] Rabie O B J , Balachandran P K , Khojah M , Selvarajan S. A Proficient ZESO-DRKFC Model for Smart Grid SCADA Security[J].Electronics, 2022, 11(24):4144 .

[7] Ma J , Yuan Y .Application of SCADA data in wind turbine fault detection - a review[J].Sensor Review, 2022.

[8] Mckinnon C , Tartt K , Carroll J , Mcdonald A. Comparison of novel SCADA Data Cleaning Technique for Wind Turbine Electric Pitch System[J]. Physics: Conference Series, 2022, 2151(1).

[9] Chretien A , Tahan A , Pelletier F .Wind Turbine Blade Damage Evaluation under Multiple Operating Conditions and Based on 10-Min SCADA Data[J]. energies, 2024, 17(5).

[10] Zhang G , Li Y , Zhao Y .A novel fault diagnosis method for wind turbine based on adaptive multivariate time-series convolutional network using SCADA data[J].Advanced Engineering Informatics, 2023.

[11] Dao P B , Barszcz T , Staszewski W J .Anomaly detection of wind turbines based on stationarity analysis of SCADA data[J].Renewable Energy, 2024, 232.

[12] Zhang L , Zhu H .A Simple Method of Non-Intrusive Load Monitoring Based on BP Neural Network[J].Journal of Physics: Conference Series, 2022, 2237(1) :012010.

[13] Laxmisagar H S , Hanumantharaju M C .FPGA implementation of breast cancer detection using SVM linear classifier[J].Multimedia Tools and Applications, 2023:1-24.

[14] Diqi M , Ordiyasa I W .Enhancing Stock Price Prediction in the Indonesian Market: a Concave LSTM Approach with Runrelu[J]. Mobile Robotics and Intelligent Systems, 2024, 18(3):69-77.

[15] Dbrowski T , Bednarek M ,A. Rosiński, Olchowik W.Engineering Application of a Product Quality Testing Method within the SCADA System Operator Education Quality Assessment Process[J].Applied Sciences, 2023.

[16] Ma Z F, Han J. J., Wang B , Zhang B, Sun K , Chen X. Ground resistance prediction of transmission line based on IPSO-NESN algorithm[J]. Intelligent Power,2024,52(06):116-122.

[17] Zhang Z, Huang C, Dong K, Huang H. Birds foraging search: a novel population-based algorithm for global optimisation[J]. Memetic Computing, 2019, 11(3):221-250.

[18] Wang F., Liu S. Java Web-based knowledge base management platform for optimised design of CNC machine tools[J]. Machine Tools and Hydraulics, 2022, 50(20):82-89.

[19] Wang J. , Gong J. X , Lin Z Q, Zhang X J. A multidimensional depth oriented fuzzy testing approach for Java Web[J]. Information Network Security, 2024(2):282-292.

[20] Huang H, Zhang W, Qi G. C. , Yan F , Chen P. Security Risks of Data Transmission in SCADA System of Oil and Gas Pipelines and Their Solutions[J]. Natural Gas Industry, 2013, 33(11): 115-120.

[21] Liu B , Zhang M. Design and application of SCADA system based on multi-control platform[J]. Electrotechnology,2023,(10):8-13+16.

[22] Bai Y. R., Lun S. X. Optimising time series prediction of echo state networks based on war strategy algorithm[J]. Journal of Bohai University(Natural Science Edition),2024,45(02):154-160.

[23] Guo Q. H., Lin H. Z., Li Y., Xie L. L., Liu T. Y. Short-term electricity price prediction based on NGO-VMD-SSA-ESN[J]. Electrotechnology,2024,(02):130-136.