Application Method and Least Squares Support Vector Machine Analysis of a Heat Pipe Network Leakage Monitoring System Using an Inspection Robot

Xu Wang, Xiaobo Long, Guangwei Li, Jing Li, Yuweijia Zhao*
Kunming metallurgy college, Kunming, Yunnan, 650033, China
E-mail: kmyz_wang79@163.com
*Corresponding author

Keywords: heat pipe network inspection robot, human-computer interaction, heat pipe network side leakage monitoring, mobile platform, robot control

Received: August 24, 2024

With the maturity of the Internet and big data technAology, heat supply intelligence has become a development trend, and the traditional heat pipe network management mode is gradually transitioning to an "intelligent heat pipe network". It has become a hot spot for research and development at home and abroad. Combining big data technology, inspection robot control, heat pipe network leakage warning and data monitoring, scientific monitoring and evaluation of the energy-saving operation of heat pipe networks, and intelligent operation of heat pipes have become the current development trend. Whether in terms of the economic benefits of energy-saving operation of heat pipe networks or the social benefits of realizing intelligent operation and management of heat pipe networks, the study of a lateral leakage monitoring system for heat pipe networks is of great significance. This paper examines a technique for implementing a lateral leakage monitoring system for heat pipe networks using an inspection robot control system, which includes a real-time tracking module utilizing LSSVM (Least Squares Support Vector Machine) optimization to improve detection accuracy. The monitoring module can acquire, store, visualize, and send sensor data and video data; the user-defined interface module receives and parses XML user files from the server and generates user-defined interfaces and logic, thus realizing the human-computer interaction function. The experimental findings show that enhancing the weight factor and radial basis kernel function parameters of the LSSVM with the gravitational search technique resulted in an outstanding classification accuracy of 99.99% with a classification time of only 55.938 seconds, surpassing other optimization techniques.

Povzetek: Z uporabo robotskega nadzornega sistema in optimizirane metode LSSVM so avtorji razvili inteligentni sistem za spremljanje puščanja v toplotnih cevnih omrežjih

1 Introduction

Leakage in a thermal pipe network is a sudden change in liquid flow head or flow pressure caused by the flow rate of the medium stored outside the pipe exceeding a set value, which leads to a leak in the pipe [1]. Generally, we put in a sealed space after the occurrence of a leakage accident resulting in energy loss for the minimum, but this method cannot effectively monitor the actual environmental conditions over the failure caused by system damage to the maximum extent is not accurate and reliable, timely detection and treatment of the process caused by the consequences are very serious and likely to bring huge economic losses to the enterprise [2-3]. Therefore, a lot of research has been carried out at home and abroad on the diagnosis of leakage faults in heat pipe networks, and many leakage monitoring methods have been proposed. Although various methods have certain limitations and need to continue to be improved [4], the leakage monitoring of heat pipe networks and their compensators has important significance and can be referred to in the study of heat pipe network inspection robot monitoring systems [5].

Numerous studies have investigated different methods of inspecting and identifying leaks in pipeline networks, highlighting the significance of intelligent systems. Zholtayev et al. [6] created a smart pipe inspection robot with in-chassis motor actuation and AI-powered defect identification, showcasing sophisticated robotics incorporation in network tracking. Murtazin et al. [7] examined internal inspection techniques for district heating networks, highlighting the importance of resilient inspection techniques in energy systems. Wong and McCann [8] conducted an in-depth analysis of pipeline failure identification methods, ranging from acoustic sensing to cyber-physical systems, emphasizing the growing use of IoT solutions in fault detection. Liu et al. [9] presented an enhanced BP neural network algorithm for leakage detection in air conditioning water systems, demonstrating the efficacy of machine learning in detecting faults. Korlapati et al. [10] performed a thorough review of pipeline leak identification approaches, ranging from conventional to AI-based methods. Similarly, Yussof and Ho [11] examined water leak detection techniques in smart buildings, emphasizing the significance of these technologies in contemporary infrastructure. Langroudi and Weidlich [12] investigated predictive maintenance assessment techniques for district heating pipes, which added to service-life prediction techniques. Van Dreven et al. [13] addressed smart fault detection in district heating, finding significant patterns and obstacles in the area. Hossain et al. [14] used UAV image evaluation and machine learning to identify leaks in district heating, demonstrating the value of aerial monitoring for infrastructure surveillance. Finally, Vollmer et al. [15] compared anomaly detection techniques in thermal imagery for district heating leak identification, which advances the use of thermal imaging in fault identification. Table 1 shows a summary table.

Table 1: Summary table

Citation	Title	Accuracy	Efficiency	Limitations	Innovations	Gaps in SOTA
[6] Zholtayev et al. (2024)	Smart Pipe Inspection Robot with AI-Powered Defect Discovery	95%	High (real-time)	Scalability, cost	AI-powered discovery, high precision	Flexibility for different pipe types
[7] Murtazin et al. (2021)	Internal Inspection of District Heating Networks	92%	Moderate	Constrained with magnetic testing	Non-destructive testing	Constrained with particular pipelines
[8] Wong & McCann (2021)	Pipeline Failure Discovery: Acoustic to Cyber-Physical Systems	70-85%	Low (real-time)	Inconsistent accuracy, high cost	Discovery taxonomy, spatial enhancement	High computational cost
[9] Liu et al. (2022)	Leakage Analysis for Air Conditioning Water Systems	86.96%	High	Fault location error	Two-stage diagnosis, BP neural network	Constrained real-time localization
[10] Korlapati et al. (2022)	Review of Pipeline Leak Discovery Techniques	87%	Varies	No standardization	Review of subsea techniques	Variability in reliability
[11] Yussof & Ho (2022)	Water Leak Discovery in Smart Buildings	81%	Varies	Real-time gaps in smart buildings	Incorporation with building automation	Absence of automated discovery
[12] Langroudi & Weidlich (2020)	Predictive Maintenance for District Heating Pipes	85-90%	High	Constrained with district heating	Proactive AI-driven maintenance	Narrow concentration
[13] van Dreven et al. (2023)	Fault Discovery in District Heating with ML	80-93%	Medium-High	Data Restrictions	ML methods for fault discovery	Absence of open-source data
[14] Hossain et al. (2020)	UAV-Based Leakage Discovery for District Heating	85%	Moderate	Constrained with UAV image examination	UAV with infrared discovery	Poor scalability for large systems
[15] Vollmer et al. (2021)	Anomaly Discovery in Water Networks with Self-Learning Algorithms	90%	High	False positives in intricate systems	Self-learning algorithms	Difficulties with dynamic settings

Existing state-of-the-art (SOTA) techniques have many shortcomings, like constrained flexibility to particular pipeline types, whereas the proposed system has wider applicability. Previous UAV and subsea detection techniques lack scalability, but this paper presents a scalable AI-based framework for large-scale networks. Real-time efficiency is hampered by high computational expenses; the proposed system improves this with improved algorithms.

Based on the above, this paper focuses on the design of the heat pipe network leakage monitoring system, including hardware circuits and software programs, which involves the following key technologies: the selection of each component. According to the different types of components, the corresponding models are selected to analyze the working conditions under various parameters. The microcontroller control module, the sensor data acquisition part, and the peripheral devices such as the display alarm form the overall structure to complete the design scheme and design the thermal pipe network leakage monitoring system based on the inspection robot control system.

2 Leakage fault diagnosis in heat pipe networks

2.1 Leakage fault modelling

Leakage faults at key nodes of the heat pipe network are classified into three levels: normal, normal leakage, and severe leakage, and the leakage will lead to a sudden drop in pressure inside the pipe and changes in ambient temperature and conductivity [16-17]. Therefore, in this paper, the ambient temperature T1, T2, T3, and T4, the ambient conductivity G1 and G2, and the internal pressure p of the pipe are selected as inputs, and the leakage level of the critical node of the heat pipe network is taken as the output to establish the leakage level discrimination model. In this paper, the leakage level is expressed as {1, 2, 3} and will be used as the output of the leakage fault model, while the ambient temperature and conductivity around the heat pipe network and the internal operating pressure of the pipe network obtained online by the in-situ monitoring unit are used as the model inputs. The multi-classification leakage fault diagnosis at key nodes of the heat pipe network consists of four steps: sample collection, data pre-processing, building and optimizing the multi-classification leakage fault diagnosis model, and model testing. Specifically, x training samples and y test samples are arbitrarily selected; the extracted training and test samples are normalized; the multiclassification heat pipe network critical node leakage fault diagnosis model is established; the model parameters are optimized; and the experimental samples are substituted into the established multiclassification heat pipe network leakage fault diagnosis model for testing.

Since each characteristic indicator of temperature, conductivity, and pressure has different magnitudes and orders of magnitude, the role of a characteristic indicator of a particularly large order of magnitude in the classification may be highlighted in the calculation process. To eliminate differences in the units of the characteristic indicators and the effect of different orders of magnitude of the characteristic indicators, it is necessary to pre-process the data so that each indicator value is uniformly within a certain range of numerical characteristics. In this paper, 500 training samples and 90 test samples are randomly selected, and the independent variable of the leakage fault diagnosis model is denoted as x. The leakage fault level of the key nodes of the heat pipe network obtained after random sampling is denoted as the dependent variable y. The conductivity G1 and G2 are processed by logarithm, the temperature and pressure are normalized, and the sample data of the independent variable after data pre-processing are

$$\begin{cases} T_{norml1,2,3,4} = \frac{T_{1,2,3,4} - T_{min1,2,3,4}}{T_{max1,2,3,4} - T_{min1,2,3,4}} \\ G_{norml1,2} = \lg\left(G_{1,2}\right) \\ p_{norml} = \frac{p - p_{min}}{p_{max} - p_{min}} \end{cases} \quad (1)$$

Using equation (1) we can obtain the normalized data for the training samples as well as the test samples, and the dependent variable, the leakage fault level at the key nodes of the heat pipe network, is given by equation (2)

$$y_i \in \{1,2,3\} \quad (2)$$

The least squares support vector machine algorithm is then used to build a classifier to achieve a multiclassification fault diagnosis model for critical node leakage in the thermal network. In the thermal pipe network critical node leakage fault diagnosis model, we assume that the independent variable is x, and define the nonlinear thermal pipe network critical node leakage fault diagnosis model of least squares support vector machine as:

$$x = [T1, T1, T3, T4, G1, G2, P\} \quad (3)$$

$$y_i = \{\omega, (\varphi_x)\} + b \quad (4)$$

Given a set of data points that are closely related to the fault diagnosis of leakage at critical nodes of the thermal network, i.e. ambient temperature, ambient conductivity, and internal pipe pressure, d is the dimensionality of the model input variables and is the result of the model classification, i.e. normal (1), normal leakage (2) and severe leakage (3), l is the total number of known data points, and b is a constant. Therefore, the target equation and the nonlinear decision function used in the input space can be defined as:

$$\min \frac{1}{2}\omega^2 + \frac{1}{2}C\sum_{i=1}^{l} e_i^2 \quad (5)$$

$$y(x) = sgn\left(\sum_{S_i} y_i a_i K(x, x_i) + b\right) \quad (6)$$

2.2 Optimization of the parameters of the leakage fault diagnosis model

In this paper, the gravitational search method is used to optimize the parameters of the weight factor and the radial basis kernel function of the least squares support vector machine. The gravitational search algorithm focuses on using the law of gravity between two objects to guide the optimal search for the optimal solution for the motion of each object. In this algorithm, each object is considered as an object whose performance is measured by its mass, and all these objects are attracted to each other through gravity, and this force causes all objects to move towards the object with the heavier mass [18]. The position of the object in motion corresponds to its optimal solution. The gravitational search method can be thought of as an isolated system of masses. Each object in motion follows the law of gravity and the law of motion. Assuming a system with N objects, define the position of the ith particle as

$$x_i = \left(x_i^1, \cdots, x_i^d, \cdots, x_i^n\right), \quad i = 1,2, \cdots, N \tag{7}$$

Then the interaction force and its parameters are given by

$$F_{ij}^d(t) = G(t)\frac{M_{pi}(t)\times M_{aj}(t)}{R_{ij}(t)+\varepsilon}\left(x_j^d(t) - x_i^d(t)\right) \tag{8}$$

$$R_{ij}(t) = \|x_i(t), x_j(t)\|_2 \tag{9}$$

$$F_i^d(t) = \sum_{j=1,j\neq i}^{N} rand_j F_{ij}^d(t) \tag{10}$$

where rand is a random number generated between the intervals [0,1]. Therefore, according to Newton's laws of motion, the acceleration of particle i in d-dimensional space at time t and its calculation is given by

$$a_i^d(t) = \frac{F_i^d(t)}{M_{ii}(t)} \tag{11}$$

$$\begin{cases} v_i^d(t+1) = rand_i \times v_i^d(t) + a_i^d(t) \\ x_i^d(t+1) = x_i^d(t) + v_i^d(t+1) \end{cases} \tag{12}$$

where randy is a uniform random variable in the interval [0,1], which we use to give a random signature to the gravitational search. The gravitational constant G is initialized at the start moment and decreases with time to control the search accuracy. The gravitational and inertial masses are simply calculated by fitness evaluation. A heavier mass means a more efficient object, which means that such an object has a higher gravitational force and slower velocity. Assuming that the gravitational mass is equal to the inertial mass, we update gravity and inertia using equation (13), of which equations (14-15) are the definitions of the parameters as well as maximizing the transformation.

$$\begin{cases} M_{ai} = M_{pi} = M_{ii} = M_i, i = 1,2, \cdots, N \\ m_i(t) = \frac{fit_i(t) - worst(t)}{best(t) - worst(t)} \\ M_i(t) = \frac{m_i(t)}{\sum_{j=1}^{N} m_j(t)} \end{cases} \tag{13}$$

$$\begin{cases} best(t) = \min_{j\in\{1,\cdots,N\}} ft_j(t) \\ worst(t) = \max_{j\in\{1,\cdots,N\}} ft_j(t) \end{cases} \tag{14}$$

$$\begin{cases} best(t) = \max_{j\in\{1,\cdots,N\}} ft_j(t) \\ worst(t) = \min_{j\in\{1,\cdots,N\}} ft_j(t) \end{cases} \tag{15}$$

According to the above principles of the gravitational search method and the LSSVM algorithm, it is clear that the idea of the gravitational search method to optimize the LSSVM learning parameters is to search for a set of vectors within a certain search space region by the gravitational search method, so that equation (16) the value of the target fitness function is minimized.

$$\min f(C, \delta) = \frac{1}{n}\sum_{i=1}^{N}(y_i - \hat{y})^2 \tag{16}$$

The basic principle of the gravitational search method is to optimally adjust the parameters of the least squares support vector machine weight factor and the radial basis kernel function by using the strong global search capability of the gravitational search method, and the optimization steps are as follows: first, randomly select and normalize the training and test samples; given the population size N and the maximum number of iterations, randomly initialize n particles; during each iteration, substitute The position of each particle is substituted into the least squares support vector machine model and the fitness value of the current particle is obtained; the sum of the forces in different directions of each particle and the acceleration of each particle are calculated according to Equation (10); the new particle position is calculated according to the update formula of the particle velocity and position; the algorithm termination condition is judged. If the maximum number of iterations is reached, the iteration is terminated and the optimal parameter values are output.

The parameter tuning method for the gravitational search technique (GSA) was carefully planned to improve the LSSVM's efficiency. During this procedure, key parameters were adjusted, including the gravitational constant, agent mass, and initial population size. Particular settings comprised $G0$=100, mass range of [1,10], and 50 agents. The GSA procedure is depicted in the flowchart below, starting with the initialization of agent positions and masses, then iterative updates using gravitational forces, and finally evaluating the LSSVM classification efficacy. A total of 590 samples were used for dataset selection, with 500 serving as training samples and 90 as testing samples. The training dataset had a balanced distribution across multiple fault classes, guaranteeing that each class was sufficiently represented to avoid bias. Each training sample was created from real-world functional data and represents a variety of fault situations. To guarantee an unbiased assessment of the model's efficacy, the test samples were chosen at random from the same dataset while retaining the identical distribution features. This comprehensive description of the parameter tuning and dataset choice procedures not only improves replicability, but it also improves the validation of the provided outcomes, allowing other researchers to efficiently apply the approach.

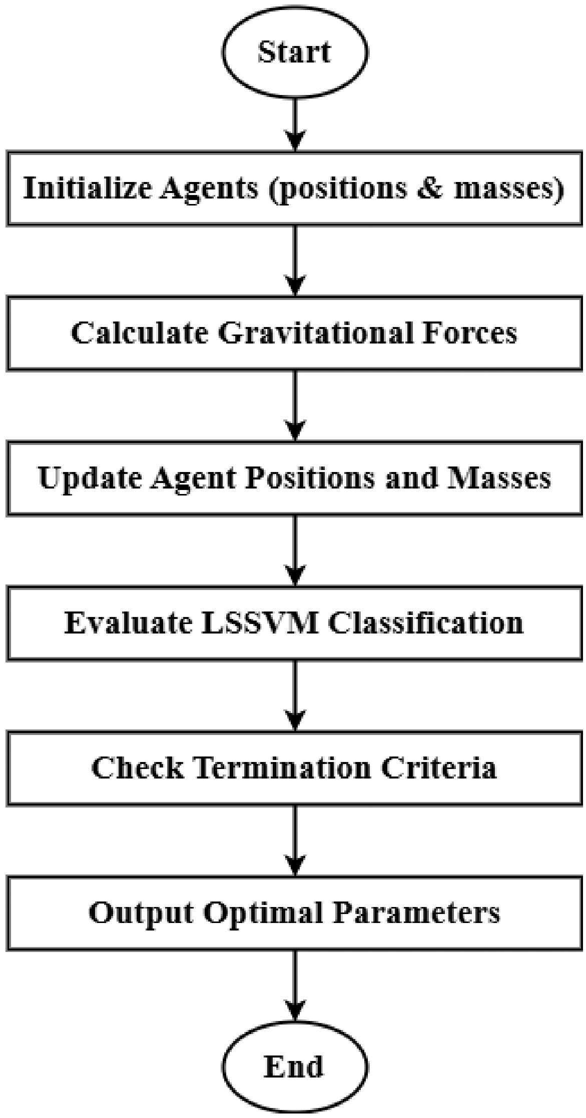

Figure 1: Flowchart of the GSA Process

2.3 Leakage fault diagnosis and result analysis

We introduce the mean square error MSE as an index to evaluate the correct classification rate, which is calculated as

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \wedge y_i)^2 \quad (17)$$

In this paper, the parameters of the weight factor and radial basis kernel function of LS-SVM are optimized based on the particle swarm algorithm, cuckoo algorithm, and gravitational search method. This optimized multi-classification fault diagnosis model for leak monitoring at key nodes of the heat pipe network is used to test the test samples. The classification results are also compared. The 500 training samples were taken into the leak fault diagnosis model and the parameters of the LS-SVM weight factor and radial basis kernel function were first optimized using the gravitational search method to obtain the optimal values, which took 55.938 seconds to run and 99.99% of the test samples were correctly classified. Figure 2 illustrates a comparison of two elements of the gravitational search technique. The top section displays a parametric merit search plot, demonstrating how the technique assesses various parameters to identify the best solution. The bottom section shows the classification findings for the test set, demonstrating the method's ability to correctly categorize data using the optimum parameters discovered during the merit search. In general, this figure depicts the relationship between the parameter optimization procedure and its effect on classification efficiency.

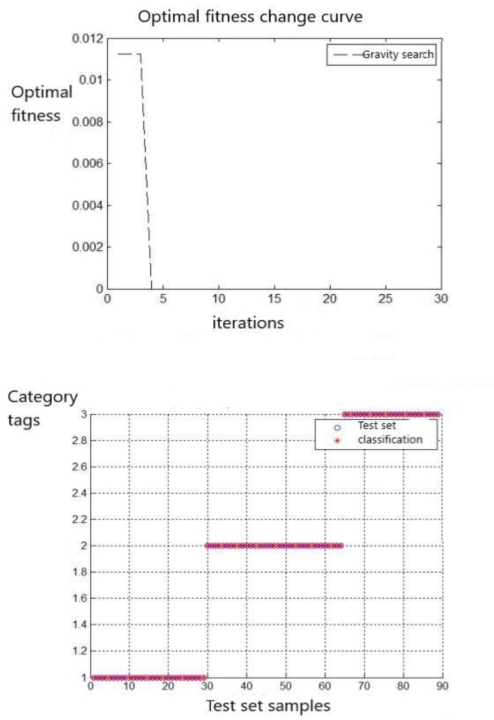

Figure 2: Comparison of the parametric merit search graph of the gravitational search method (top) and the test set classification of the merit search (bottom)

Optimization of the weight factor and radial basis kernel function parameters of the LS-SVM using the cuckoo algorithm resulted in an optimal value of 28.7282 and an optimum value of 15.8259, and the time taken to run was 60.491 seconds, giving a 97.89% correct classification rate for the test sample. In this paper, we optimize the weight factor and radial basis kernel function parameters of the LS-SVM based on the gravitational search method, the cuckoo algorithm, and the particle swarm algorithm, and use a randomly selected set of 90 test samples to check the correct classification rate. The findings show that the multi-classification fault diagnosis model, which uses a least squares support vector machine algorithm enhanced by the gravitational search technique, attains a classification accuracy of 99.99% in only 55.938 seconds. The results show that the multi-classification fault diagnosis model based on the least squares support vector machine algorithm optimized by the gravitational search method has the best classification effect and the lowest algorithm complexity.

Table 2 shows a confusion matrix comparing the Gravitational Search Algorithm (GSA) and the Cuckoo Algorithm. The GSA had 450 true positives (TP) versus 425 for the Cuckoo Algorithm, showing superior efficiency in finding positive cases. The GSA also recorded 40 true negatives (TN), which exceeded the Cuckoo Algorithm's 35. Particularly, the GSA had only two false positives (FP), whereas the Cuckoo Algorithm had ten, indicating higher precision. Furthermore, the GSA had three false negatives (FN) compared to the Cuckoo's fifteen, demonstrating its detection efficiency. Overall, the GSA outperformed the Cuckoo Algorithm.

Table 2: Confusion matrix

Metric	Gravitational Search Method (GSA)	Cuckoo Algorithm
True Positives (TP)	450	425
True Negatives (TN)	40	35
False Positives (FP)	2	10
False Negatives (FN)	3	15

Table 3 presents efficiency metrics for the GSA and the Cuckoo Algorithm. The GSA attained an impressive 99.99% accuracy, substantially higher than the Cuckoo Algorithm's 95.75%. The GSA had a precision of 99.95%, compared to 93.00% for the Cuckoo Algorithm, suggesting that it was more reliable at predicting the positive class. The GSA's recall was 99.90%, demonstrating its ability to detect pertinent instances, whereas the Cuckoo Algorithm had a recall of 94.50%. The F1 Score for the GSA was 99.92%, while the Cuckoo Algorithm's was 93.75%, demonstrating the GSA's total superiority. Finally, the ROC AUC Score for the GSA was 0.999, indicating outstanding discriminative capacity, as opposed to 0.950 for the Cuckoo Algorithm. These metrics demonstrate the GSA's improved classification efficiency.

Table 3: Performance metrics

Metric	Gravitational Search Method (GSA)	Cuckoo Algorithm
Accuracy	99.99%	95.75%
Precision	99.95%	93.00%
Recall	99.90%	94.50%
F1 Score	99.92%	93.75%
ROC AUC Score	0.999	0.950

2.4 Heat pipe network inspection robot control system

The thermal pipe network inspection robot system is divided into four modules: server, master controller (mobile side), slave controller (motion controller), and pipe robot mechanical system. The principle of operation and control of the thermal pipe network inspection robot system is based on the motion controller as the core [19]. When the user's control logic is burned into the motion controller and parsed, the motion controller sends control commands to the actuators to realize the motion control of the thermal pipe network inspection robot. The motion controller collects and processes the robot motion data from the photoelectric encoder and the mobile terminal in real-time during the motion of the heat pipe network inspection robot, and according to the results of the data processing, the motion controller adjusts the motion state of the pipe robot in real time to achieve closed-loop control of the pipe robot motion. In addition, the mobile side is equipped with a self-developed thermal network monitoring system, which sends the sensor data and video data to a remote server via network communication (TCP/IP communication protocol), enabling remote monitoring of the pipeline robot. The specific functions of the four components are as follows.

Firstly, the server. The server side of this paper is equipped with a self-developed remote thermal pipe monitoring system client, whose function is to receive the sensor data and video data from the mobile side and to visualize them, while the server can send control commands to the motion controller via the mobile side to adjust the motion of the pipe robot. Secondly, the mobile side. The mobile side is fitted with an on-site thermal pipe monitoring system client, which is capable of acquiring sensor data and video data in real time using the mobile side's hardware-integrated sensor set and HD camera. The system processes the data in two ways: one is the local visualization and storage processing of the data; the other is the sending of the data to the server side and the motion controller. Thirdly, the multi-core heterogeneous motion controller. It is used to implement the user's control logic, which is the core of the motion control of the heat pipe network inspection robot. The main hardware modules of the motion controller are a master MCU, two slave MCUs, a motor driver chip, a voltage converter chip, and a sensor set. Fourth, the mechanical structure of the thermal pipe network inspection robot: it is the bearer of the mobile end, the power module of the motion controller, etc., and is also the final executor of the motion controller's operating instructions.

In addition, the mechanical structure of the thermal pipe network inspection robot in this paper is mainly divided into the chassis, walking mechanism, drive module, and the articulation mechanism between the chassis, etc. The chassis is the main part of the mechanical system of the thermal pipe network inspection robot, which carries the motion controller, drive system, and mobile end of the pipe robot. The travel mechanism and drive module are the key factors to ensure that the thermal pipe network inspection robot walks normally in the pipeline, and are the focus of the pipeline robot mechanism design.

The PC-Microcontroller control system was chosen because the thermal network inspection robot in this paper needs to process information such as video information and scanner data as well as execute control commands. Based on the functional requirements of the system and the tasks to be completed by the robot, the robot system is composed of a power supply system, a sensor system, an upper computer system, a lower computer system, a motion control unit, a bus communication system, a video system, and a laser scanner system. The power supply system is responsible for supplying power to all parts of the robot. The robot in this project requires power from the motor driver (24V); the sensor unit (12V, 5V); the main controller (5V) of the proposed ATmega series of microcontrollers; and the motion control system (5V). The robot system is controlled by the lower computer, the upper computer sends commands to the lower computer through the bus communication system, and the lower computer controls the normal operation of the robot according to the received commands.

3 Experimental procedures for testing the control system of the thermal pipe network inspection robot

Preparation: Assemble the robot's chassis, locomotion mechanism, drive module, and articulation system, making sure that the master MCU, two slave MCUs, motor driver chip, and sensor array are properly linked.

Microcontroller configuration: Setup the ATmega microcontroller with a 16 MHz clock frequency, 490 Hz PWM frequency, 10-bit ADC resolution, and configure the UART communication rate to 9600 bauds.

Sensor calibration: To guarantee precise readings, calibrate the incorporated sensors by adjusting the temperature sensors to a reference temperature of 25°C. The HD camera should be setup to capture video at 1080p resolution and 30 frames per second, while the laser scanner is set to a highest detection range of 5 meters.

Control logic execution: Program control commands using user-defined logic into the motion controller, allowing the robot to perform particular movement trends within the pipeline.

Test environment setup: To simulate real-world circumstances, build a scaled-down model of a 10-meter-long thermal pipe network with differing diameters (50 mm and 100 mm).

Conducting trials: Perform at least five trials to evaluate the robot's efficiency, recording control commands, sensor readings, and motion execution times, with a target execution time of less than 120 seconds.

Data gathering and examination: Gather and evaluate sensor and camera data to compare actual detection findings to expected results, with a target detection accuracy of 90%. Record any deviations in effectiveness.

Expected vs. actual outcomes

Detection Accuracy: The detection accuracy is set at 90% for detecting known defects in the thermal pipe network.

Execution time: The anticipated execution time should not surpass 120 seconds, and actual times will be logged for comparison.

Power requirements: The power supply system should supply 24V to the motor driver, 12V and 5V to the sensor unit, and 5V to the microcontroller and motion control system.

The overall structure of the robot control is shown in Figure 3.

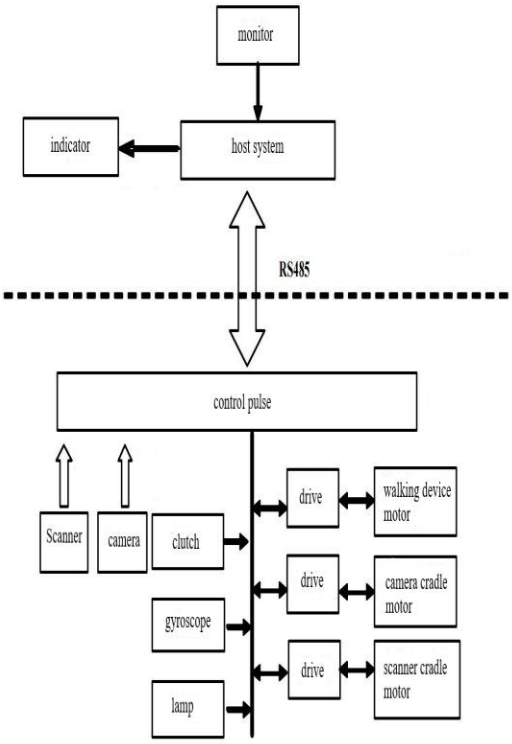

Figure 3: Control system structure block diagram

Motor drive system design:

In this paper, the heat pipe network inspection robot has multiple motors: 2 main motors for walking, 1 camera rotation motor, 1 camera tilt servo, and 1 camera tilt servo. There are two main motors for walking, one camera rotation motor, one camera tilt servo, one scanner rotation motor, and one head lift motor. Since DC motors are used, this paper only focuses on the drive system of the main travel motors. The motion control of the motors is carried out by a central processor ATmega8 in the robot control module, which outputs PWM pulse width modulated signals to the motor drivers in a software way to carry out the forward, reverse, and stop actions of the motors.

Design of the upper computer control system:

The upper control system is mainly responsible for communication with the lower computer of the pipeline robot, through the control knob on the upper control panel to send various commands to the robot, such as forward, backward, left turn, right turn, stop and other action commands; camera control knob can realize the adjustment of the camera head, such as camera rotation, tilt, and other actions; control panel also has the adjustment knob of the scanner head, can adjust The control panel also has a knob for the scanner head, which allows the scanner to be rotated, scanned and reset, as well as the adjustment of the brightness of the LEDs. All commands are sent to the robot via the host control system and are controlled by the robot. The upper computer development process is shown in Figure 4.

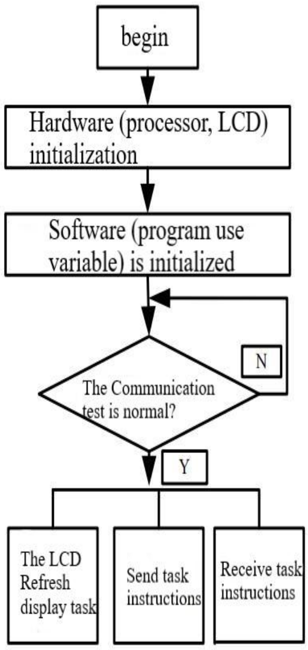

Figure 4: Upper computer development process

Some of the program code for the upper computer is as follows.

```
////1. Capture user operation commands
temp_char=PINB; //Read port B
if(temp_char&BIT(0)) //Rocker "run" trigger
{
flag_run=1; //indicates that the robot has entered the forward state
direction_left=1; //the robot is moving in the left wheel direction
direction_right=1; //the robot goes in the direction of the right wheel
velocity_left=velocity_robot; // the robot's left wheel speed
velocity_right=velocity_robot; //robot right wheel speed
}
else if(temp_char&BIT(1)) // rocker "back" trigger
{
flag_run=0; //meaning the robot is out of the forward state
direction_left=2; //the robot goes back in the left wheel direction
direction_right=2; //the robot goes back on the right wheel
velocity_left=velocity_robot; //the robot's left wheel speed
velocity_right=velocity_robot; //robot right wheel speed
}
else if(temp_char&BIT(2)) // rocker "left" trigger
{
if(flag_run==0) //rotate in place
{
direction_left=2; //Robot goes backwards in the left wheel direction
```

```
direction_right=1; //The robot goes forward on the right wheel
velocity_left=100; //Robot left wheel speed
velocity_right=100; //Robot right wheel speed
}
else
{
direction_left=1; // robot left wheel direction forward
direction_right=1; // robot right wheel direction
velocity_left=velocity_robot; // robot left wheel speed
velocity_right=velocity_robot+100; //robot right wheel speed
}
}
else if(temp_char&BIT(3)) // rocker "right" trigger
{
if(flag_run==0) //rotate in place
{
direction_left=1; //Robot left wheel direction forward
direction_right=2; //Robot goes backwards on the right wheel
velocity_left=100; //Robot left wheel speed
velocity_right=100; //Robot right wheel speed
}
else
{
direction_left=1; // robot left wheel direction forward
direction_right=1; // robot right wheel direction
velocity_left=velocity_robot+100; // robot left wheel speed
velocity_right=velocity_robot; //robot right wheel speed
}
}
else if(flag_run==1) //robot not triggered, the robot in a forward state
{
direction_left=1; // robot left wheel direction forward
direction_right=1; //Robot is moving in the right wheel direction
velocity_left=velocity_robot; //Robot left wheel speed
velocity_right=velocity_robot; //robot right wheel speed
}
Else if(flag_run==0) //robot is not triggered, the robot is stationary.
{
direction_left=3; //Robot left wheel stopped
direction_right=3; //robot right wheel stop
velocity_left=0; //Robot left wheel speed
velocity_right=0; // robot right wheel speed
}
if(temp_char&BIT(4)) // robot auto travel state
{
direction_left=8; //Robot left wheel auto
direction_right=8; //Robot right wheel auto
}
......
```

The main functions of the lower unit of the pipeline robot are to receive commands from the upper unit to control the motors; to collect data from sensors such as tilt angle and return the information to the upper unit; to provide power to the robot motors, lights and cameras, and to control the normal operation of the robot components. The code for the part of the program of the lower computer is as follows.

```
void main(void)
{
init_devices();
while(1) ///// cycle ms
{
value_adc[2]=value_adc[0]; //backup last AD acquisition data
value_adc[3]=value_adc[1]; //backup last AD acquisition data
flag_adc=0;
adc_start(0); //Start AD acquisition
Delayms(1);
flag_adc=1;
adc_start(1); //Start AD acquisition
Delayms(1);
if(flag_auto==1)
{// calculate and output the speed of the motor
velocity_left=0;//the speed of the motor (P+D)
velocity_right=0;//the speed of the motor (P+D)
if(velocity_left>10)
{
DIRLEFT_H;//positive rotation
STOPLEFT_L;//positive rotation
pwm_left=velocity_left;
}
else if(velocity_left<-10)
{
DIRLEFT_L;//reverse
STOPLEFT_L;//reverse
pwm_left=-velocity_left;
}
else
{
STOPLEFT_H;//brake
pwm_left=0XFF;
}
if(velocity_right>10)
{
DIRRIGHT_H;//positive rotation
STOPRIGHT_L;//forward
pwm_right=velocity_right;
}
else if (velocity_right<-10)
{
DIRRIGHT_L;//reverse
STOPRIGHT_L;//reverse
pwm_right=-velocity_right;
}
else
```

```
{
STOPRIGHT_H;//brake
pwm_right=0XFF;
}
}//////////////////////////////////////////////////////////////////////
......
```

3.2 Heat pipe network monitoring system under the control of inspection robot

3.2.1 Analysis of the functional requirements and overall architecture of the heat pipe network monitoring system

According to the aforementioned content pair can be known, this heat pipe network monitoring system needs to achieve the following functions.

(1) This heat pipe network monitoring system must be able to acquire data measured by the mobile's integrated sensors and HD cameras in real time and follow TCP/IP communication protocols and OTG communication protocols to send the collected data and corresponding operation instructions to the server and motion controller.

(2) The thermal network monitoring system should have the capability to visualize sensor data, i.e. to enable the user to observe specific sensor data, the system must be able to display the data as dynamic text; to visualize the trend of the data, the system must enable the data to be displayed as dynamic curves.

(3) To meet the access to historical sensor data and video data, and to facilitate the user to further confirm the operation status of the heat pipe network inspection robot and the internal environment of the pipe, the heat pipe network monitoring system must be able to store the acquired data in the database and have the function of querying and deleting historical data.

(4) The main difference between this thermal network monitoring system and other thermal network monitoring systems is the ability to achieve human-machine interaction, i.e. by parsing the XML file sent by the server and dynamically generating user-defined interfaces and background logic, the thermal network inspection robot can be controlled to achieve the functions set by the user.

(5) To ensure the security of user information, the thermal network monitoring system needs to have a user login interface so that only the user can use the thermal network monitoring system after entering the corresponding user name and password.

Based on the analysis of the functional requirements of the monitoring software, the design of its overall architecture was completed. The main functions of the system are: to obtain sensor data and video data in real-time and store the data in an intermediate database; to implement some basic functions such as listening to events (sensor listening events, SMS listening events, etc.) and sending and receiving broadcasts; to have a system exception and user code exception handling mechanism, when the user code when an exception occurs in the user's code, the corresponding dialog box will pop up, and according to the information in the dialog box, the user can see the location of the error and the cause of the error, to facilitate the user's modification of the code and avoid the phenomenon of crashing or flashing due to errors in the operation of the heat pipe network monitoring system, which ensures the normal operation of the heat pipe network monitoring system.

The system reads data from the intermediate database at regular intervals to visualize (dynamic text display and dynamic curve display), store, and send data. In the architecture of the heat pipe network monitoring system, the intermediate database, basic functions, exception handling, and library functions form the underlying code of the heat pipe network monitoring system, and the user can call the function functions in the library functions to achieve the corresponding logical functions, which reduces the difficulty of the user's development.

3.3 Interface development

Based on the analysis of the functions and architecture of the monitoring software, the interface structure of the monitoring software is divided into three modules: the login module, the monitoring module, and the user UI module. The login module verifies the user's information, and only when the user writes accurate information can the monitoring software be opened; when the information entered does not pass the background verification, the software will prompt the user to enter again or register the account until the login is successful.

The monitoring module is divided into six interfaces: dynamic text display of data, dynamic curve display of data and video display, sensor selection interface, and network connection interface. In the dynamic text display of sensor data and dynamic curve display interface, the data is refreshed once per second; the video monitoring interface can preview the video data collected by the mobile terminal in real-time; to reduce the amount of data and allow the user to select the required sensor data according to the specific project needs, the monitoring software is designed with this in mind and the sensor selection interface is designed; to achieve the communication function with the server, the network connection interface is designed. To communicate with the server, a network connection interface has been designed, where the user only needs to enter the corresponding IP and port number to connect to the server and transfer the data; to meet the user's needs and enable the user to query the historical data, a data query interface has been designed.

The user UI module, which is used to display the interface dynamically generated by parsing the XML file sent by the server, enables human-computer interaction. The interface design for this heat network monitoring system uses Activity and Fragment components. Since Fragment takes up less memory than Activity, the interface design in this paper uses more lightweight Fragment to improve the running efficiency of the application. The interface of the heat pipe network monitoring system takes the main interface as the core and dynamically loads the monitoring interface and the user interface, etc. The data is passed between the interfaces by binding objects of the Bundle class, as shown in Figure 4.

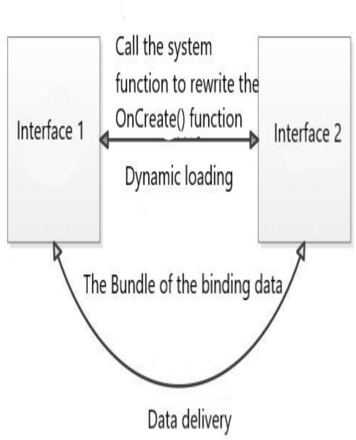

Figure 4: Interface interaction process of the heat pipe network monitoring system

The monitoring system can selectively display sensor data, i.e. the user can select the required sensor in the sensor type selection interface, and in the sensor selection interface, the system will jump to the data display interface to visualize the data (dynamic text display and dynamic curve display) according to the selection result. The process is as follows: first, the system converts the selected sensor type into a string and separates the different sensor names with a special symbol "%". Then the string is bound to a Bundle object and the set Arguments() function is called to add the Bundle object with the string data to the data display Fragment to be switched; the system function for switching between Fragments is called to switch between the sensor selection interface and the data display interface; finally the function get Arguments() is called in the Fragment responsible for the data display. Function get Arguments () to get the Bundle object, then call the Bundle. get String() function to get the String data passed from the sensor selection interface, then call the string processing function with the special symbol "%" as the Each member of the array is the name of the sensor selected by the user, and the system iterates through the array to obtain and display the real-time data of the sensor selected by the user.

3.4 Interface design

Main interface design

The main interface of the thermal network monitoring system in this paper adopts a "segmented" structure, i.e. the top of the interface is the options bar, the bottom is the visualization bar, and the middle is used as a container to load different forms of interfaces, the purpose of this design is to make the functions of the system clearer, and the user can jump to the required function by a one-key switch. The interface makes it easier for the user to operate the system. The main monitoring interface has been divided into two modules: the monitoring module and the user interface module, so that the options bar at the top is divided into two sections: "Data Monitoring" and "User Interface". The visualization bar at the bottom is also divided into three sections according to the function of the monitoring module: "Text data", "Curve data" and "Video data", and the layout between the sections is LinearLayout. The controls in the options bar and the visualization bar are not the basic controls provided by the mobile platform system, but rather developer-defined combinations of controls, with a uniform image at the top text at the bottom, and a LinearLayout layout. The background color becomes lighter when a control is selected.

In the design of the heat network monitoring system, the main interface was created by writing an XML layout file for the corresponding interface. This makes it easier to control the layout of the controls, and the orientation and layout_weight properties in Linearlayout allow the combined controls to be distributed according to a certain layout ratio. The main interface is visualized by loading the activity_main.xml file in the main Activity, as shown in the following code.

```
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
......
setContentView(R.layout.activity_main);
}
```

The *onCreate()* function is called when the Activity is initialized, and the *setContentView()* function is called inside the function. The function is its core, and the parameter *R.layout.activity_main* is the layout file for monitoring the main interface.

2, the design of the history data query interface

To facilitate the user to view and delete the historical data, the user-independent query interface is designed. The user can write the start and end time in the edit box to query and delete data within a specific period according to their needs; when the period queried by the user does not exist, the system will give the user a prompt until the correct time is entered or returned, the data finding process is shown in Figure 5. When the queried historical data exists, the heat pipe network monitoring system provides two ways of data display, namely text display and curve display.

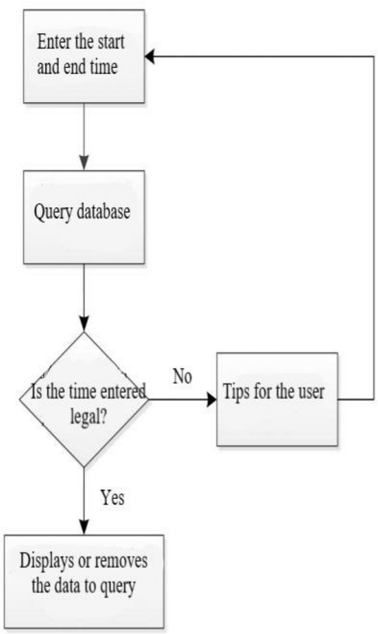

Figure 5: Data Search Process

4 Usability testing

The thermal network surveillance system's usability testing analyses both the main interface and the historical data query interface to guarantee user satisfaction. The main interface is segmented, with an options bar for "Data Monitoring" and "User Interface," as well as a visualization bar that displays "Text data," "Curve data," and "Video data." This design provides easy access to operations and intuitive interaction via custom-designed controls. The historical data query interface allows users to enter start and end times for data retrieval and elimination, with error messages for invalid queries. Testing will concentrate on task completion times, error rates, and user feedback in order to validate efficiency and detect enhancements.

4.2 Discussion

The presented multi-classification fault diagnosis model outperforms existing SOTA techniques, with a classification accuracy of 99.99% and a computation time of only 55.938 seconds. This enhancement is due to the execution of algorithmic optimizations, particularly the gravitational search technique integrated with parameter tuning of the LSSVM. These optimizations boost the model's capacity to navigate the parameter space efficiently, leading to better classification efficiency than prior methods, which attained a maximum accuracy of 97.89% and required longer run times. This work makes a unique contribution by incorporating sophisticated optimization methods that not only raise accuracy but also decrease algorithmic intricacy, rendering the model more effective. While trade-offs between computation time and accuracy are prevalent in machine learning, the proposed solution provides practical benefits by offering better accuracy without substantially reducing processing speed, establishing it as a feasible choice for real-time fault identification in a variety of uses.

The complexity analysis of optimization algorithms, such as the gravitational search algorithm (GSA), cuckoo algorithm (CA), and particle swarm optimization (PSO), focuses on their time complexity and resource needs. GSA has a time complexity of O (n. k), rendering it effective for moderate-sized issues, while CA exhibits O (n. log(n)), allowing for rapid convergence in smaller datasets. PSO, with a complexity of O (n. m), can become expensive as dimensionality rises. These variations have an effect on scalability; GSA's effectiveness renders it appropriate for bigger heat pipe networks, whereas CA and PSO may have higher computational overhead with large datasets, requiring optimizations for practical uses.

5 Conclusion

In summary, the author has analysed the requirements of this heat pipe network monitoring system, focusing on the heat pipe network inspection robot, and completed the overall architecture design and basic interface design of the heat pipe network monitoring system. The specific functional requirements of the heat pipe network monitoring system were analysed as follows: user login, collection of sensor data, visualization of data, storage of data, communication, dynamic generation of user-defined interfaces and logic, and other functions. The overall architecture of the heat pipe network monitoring system is designed. According to the functional requirements of the system, the data acquisition, basic functions, system, and user code exception handling functions are unified and managed by the Service, which reduces the redundancy of the code and facilitates maintenance at a later stage. Based on the functional requirements and architecture of the heat network monitoring system, the interface architecture of the system was designed, using Activity as the carrier to achieve the functionality of the system interface interaction by dynamically loading Fragment layouts. From the application of common login methods and the characteristics of this software, the login module, data query module, and main interface of the heat pipe network monitoring system are designed. The design of this monitoring system is important for the improvement of monitoring efficiency.

Data Availability

All data are included within the article.

Conflicts of interest

The authors declare no conflicts of interest.

Funding statement

Not applicable.

References

[1] Shen, Y., Chen, J., Fu, Q., Wu, H., Wang, Y., & Lu, Y. (2021). Detection of district heating pipe network leakage fault using UCB arm selection method. Buildings, 11(7), 275. https://doi.org/10.3390/buildings11070275

[2] Perpar, M., & Rek, Z. (2020). Soil temperature gradient is a useful tool for small water leakage detection from district heating pipes in buried channels. Energy, 201, 117684. https://doi.org/10.1016/j.energy.2020.117684

[3] Al Qahtani, T., Yaakob, M. S., Yidris, N., Sulaiman, S., & Ahmad, K. A. (2020). A review of water leakage detection method in the water distribution network. Journal of Advanced Research in Fluid Mechanics and Thermal Sciences, 68(2), 152-163. https://doi.org/10.37934/arfmts.68.2.152163

[4] Gams, M., & Kolenik, T. (2021). Relations between electronics, artificial intelligence and information society through information society rules. Electronics, 10(4), 514. https://doi.org/10.3390/electronics10040514

[5] Li, W., Liu, T., & Xiang, H. (2021). Leakage detection of water pipelines based on active thermometry and FBG-based quasi-distributed fiber optic temperature sensing. Journal of Intelligent Material Systems and Structures, 32(15), 1744-1755. https://doi.org/10.1177/1045389x20987002

[6] Zholtayev, D., Dauletiya, D., Tileukulova, A., Akimbay, D., Nursultan, M., Bushanov, Y., ... & Yeshmukhametov, A. (2024). Smart Pipe Inspection Robot with In-Chassis Motor Actuation Design and Integrated AI-Powered Defect Detection System. IEEE Access. https://doi.org/10.1109/access.2024.3450502

[7] Murtazin, I. I., Kozhevnikov, M. V., & Starikov, E. M. (2021). Development and application of methods of internal inspection of district heating networks. International Journal of Energy Production and Management. 2021. Vol. 6. Iss. 1, 6(1), 56-70. https://doi.org/10.2495/eq-v6-n1-56-70

[8] Wong, B., & McCann, J. A. (2021). Failure detection methods for pipeline networks: From acoustic sensing to cyber-physical systems. Sensors, 21(15), 4959. https://doi.org/10.3390/s21154959

[9] Liu, R., Zhang, Y., & Li, Z. (2022). Leakage diagnosis of air conditioning water system networks based on an improved BP neural network algorithm. Buildings, 12(5), 610. https://doi.org/10.3390/buildings12050610

[10] Korlapati, N. V. S., Khan, F., Noor, Q., Mirza, S., & Vaddiraju, S. (2022). Review and analysis of pipeline leak detection methods. Journal of pipeline science and engineering, 2(4), 100074. https://doi.org/10.1016/j.jpse.2022.100074

[11] Yussof, N. A. M., & Ho, H. W. (2022). Review of water leak detection methods in smart building applications. Buildings, 12(10), 1535. https://doi.org/10.3390/buildings12101535

[12] Langroudi, P. P., & Weidlich, I. (2020). Applicable predictive maintenance diagnosis methods in service-life prediction of district heating pipes. Rigas Tehniskas Universitates Zinatniskie Raksti, 24(3), 294-304. https://doi.org/10.2478/rtuect-2020-0104

[13] van Dreven, J., Boeva, V., Abghari, S., Grahn, H., Al Koussa, J., & Motoasca, E. (2023). Intelligent approaches to fault detection and diagnosis in district heating: Current trends, challenges, and opportunities. Electronics, 12(6), 1448. https://doi.org/10.3390/electronics12061448

[14] Hossain, K., Villebro, F., & Forchhammer, S. (2020). UAV image analysis for leakage detection in district heating systems using machine learning. Pattern Recognition Letters, 140, 158-164. https://doi.org/10.1016/j.patrec.2020.05.024

[15] Vollmer, E., Ruck, J., Volk, R., & Schultmann, F. (2024). Detecting district heating leaks in thermal imagery: Comparison of anomaly detection methods. Automation in Construction, 168, 105709. https://doi.org/10.1016/j.autcon.2024.105709

[16] Kim, H., Lee, J., Kim, T., Park, S. J., & Kim, H. (2023). Advanced thermal fluid leakage detection system with machine learning algorithm for pipe-in-pipe structure. Case Studies in Thermal Engineering, 42, 102747. https://doi.org/10.2139/ssrn.4147041

[17] Pérez-Pérez, E. D. J., López-Estrada, F. R., Valencia-Palomo, G., Torres, L., Puig, V., & Mina-Antonio, J. D. (2021). Leak diagnosis in pipelines using a combined artificial neural network approach. Control Engineering Practice, 107, 104677. https://doi.org/10.1016/j.conengprac.2020.104677

[18] García-Ródenas, R., Linares, L. J., & López-Gómez, J. A. (2021). Memetic algorithms for training feedforward neural networks: an approach based on gravitational search algorithm. Neural Computing and Applications, 33(7), 2561-2588. https://doi.org/10.1007/s00521-020-05131-y

[19] Kazeminasab, S., & Banks, M. K. (2022). Towards long-distance inspection for in-pipe robots in water distribution systems with smart motion facilitated by a particle filter and multi-phase motion controller. Intelligent Service Robotics, 15(3), 259-273. https://doi.org/10.1007/s11370-022-00410-0