

Support Vector Machine-based Optimization of Human-Machine Friendly Interactive Interfaces for Products

Jian Sun

Department of Information and Media
Tongling Polytechnic
Tongling 244000, P. R. China
sunjian1980110@163.com

Qing-Bing Sang*

School of Artificial Intelligence and Computer Science
Jiangnan University
Wuxi 214122, P. R. China
qingbings@jiangnan.edu.cn

Zhi-Xuan Chang

Baekseok University
Seoul 06695, South Korea
cn0080@163.com

*Corresponding author: Qing-Bing Sang

Received August 8, 2024, revised February 7, 2025, accepted July 25, 2025.

ABSTRACT. *Traditional Human-Computer Interaction (HCI) interface optimisation methods often rely on designers' experience or simple heuristic algorithms, which are difficult to effectively deal with the trade-offs between multiple design goals in complex systems and also neglect the visual perception characteristics of the human eye. With the increasing complexity of HCI systems, there is a growing demand for methods that can automatically optimise, adapt to different scenarios and take into account the visual properties of the human eye. In order to solve these problems, this paper proposes a method for optimising the human-computer friendly interaction interface of products based on visual perception and support vector machine. Firstly, by introducing the Visual-perceptual Efficiency Index (VEI), the method is able to quantitatively take into account the visual characteristics of the human eye, which ensures that the optimisation results are more in line with the actual user experience. Secondly, by combining the hybrid optimisation method of Support Vector Machine (SVM) and Genetic Algorithm (GA), the efficient optimisation of large-scale interface elements is achieved, which improves the optimisation efficiency and the quality of the results. The experimental results show that in the optimisation of the operation interface of a mechanical product, the new method improves the visual perception efficiency index (VEI) from 4.718 to 5.246, which is an improvement of 11.19%. Meanwhile, in the user evaluation, the operation efficiency was improved by 16.34%, the error rate was reduced by 40%, and the user satisfaction was improved by 22.22%. These results demonstrate the effectiveness and potential of the method in this paper in practical applications.*

Keywords: human-computer interaction interface optimisation; visual perception; support vector machine; genetic algorithm; visual perception efficiency index

1. Introduction. As a bridge for information exchange between humans and machines, the design quality of Human-Computer Interaction (HCI) interfaces directly affects the usability, operational efficiency and user experience of the system [1, 2]. With the rapid development of modern industry and information technology, the functions of various types of equipment and systems are becoming more and more complex, and the requirements for human-computer interaction interfaces are also increasing. Efficient, intuitive, easy-to-use human-computer interaction interface can not only enhance the efficiency of the operator, but also effectively reduce operational errors, which is of great significance for improving production safety and economic efficiency. Especially in some high-risk and high-pressure working environments, such as industrial control systems, aerospace, medical equipment and other fields, optimally designed human-computer interaction interfaces can significantly reduce human errors and lower the risk of accidents [3]. In addition, as people pay more and more attention to user experience, good HCI design has become one of the key factors for product competitiveness [4]. Therefore, in-depth study of human-computer interaction interface optimisation methods not only has important theoretical value, but also has a wide range of practical application prospects. Traditional HCI interface design mainly relies on designers' experience and intuition, which is difficult to cope with increasingly complex system requirements and diverse user groups. Machine learning methods, especially techniques such as Support Vector Machines (SVMs), can learn and extract valuable patterns from a large amount of user behaviour data, providing data-driven decision support for interface optimisation. These methods can not only automate the analysis of user preferences and behavioural characteristics, but also dynamically adjust the interface layout and interaction modes according to the needs of different scenarios and user groups [5]. For example, by analysing a user's sequence of operations and line-of-sight trajectory, machine learning algorithms can predict the functions that a user is most likely to need and place them in the most accessible locations. In addition, machine learning techniques can also achieve personalised interface recommendations, automatically adjusting the arrangement and display of interface elements according to the usage habits and preferences of different users, so as to provide a more user-friendly interaction experience. Based on the above background, this paper aims to explore a human-computer interaction interface optimisation method that combines visual perception theory and machine learning technology. Specifically, this study combines the distribution characteristics of optic cone cells with SVM classification algorithms, proposes a visual perception-based human-computer interaction interface optimisation design method, and carries out global optimisation through genetic algorithms in order to achieve a more intelligent and efficient interface layout design.

1.1. Related work. Human-computer interaction interface optimisation has been an important research direction in the field of human-computer interaction. Traditional optimisation methods mainly include experience-based design, heuristic algorithms and user studies. Blair-Early and Zender [6] proposed ten usability principles for user interface design, which provide important guidance for interface design. These principles, which include system state visibility, system-real world fit, and user control and freedom, are still widely used today. However, this empirically based approach often relies on the subjective judgement of the designer and it is difficult to quantify the effect. Lamberti [7] proposed an interface layout optimisation method based on a simulated annealing algorithm. This method can automatically generate interface layouts that meet specific criteria, but the computational complexity is high, which makes it difficult to be applied to large-scale real-time optimisation. User research methods are also widely used for interface optimisation. Joseph and Murugesh [8] proposed a series of user experience metrics,

including task completion time, error rate and user satisfaction. These methods provide quantitative metrics for interface evaluation, but often require a large investment of time and resources. Overall, traditional HCI optimisation methods provide an important theoretical foundation and practical guidance for interface design. However, these methods still present challenges in dealing with complex systems, enabling personalisation and balancing multiple design guidelines. In recent years, with the development of artificial intelligence and big data technology, the research on human-computer interaction interface optimisation presents new trends and directions. Cheng and Liu [9] developed an interface optimisation system based on eye tracking. The system optimises the interface layout by analysing the user's visual trajectory to improve the efficiency of information acquisition. Although this approach takes into account the user's visual perception characteristics, it fails to make full use of machine learning techniques to improve the optimisation effect. In terms of personalised interfaces, Du et al. [10] proposed an interface recommendation algorithm based on collaborative filtering. The algorithm can recommend suitable interface layouts based on user preferences and usage history. However, this approach may lead to "over-personalisation", affecting the consistency and learnability of the interface. Soui et al. [11] developed a multi-objective optimisation framework that considers aesthetic, functional and usability objectives. The framework uses a genetic algorithm to search for Pareto optimal solutions. However, the efficiency and convergence of this method in dealing with high-dimensional optimisation problems still need to be further investigated. Lv et al. [12] developed a deep neural network-based interface layout optimisation method. The method is able to capture the complex relationships between interface elements and generate a more reasonable layout. However, the adaptability of the method in dealing with dynamically changing interface requirements needs to be improved. In summary, the current research on human-computer interaction interface optimisation mainly focuses on the directions of intelligent, personalised and multi-objective optimisation [13, 14, 15]. However, there are still some problems in the existing research, such as failing to fully consider the visual perception characteristics of the human eye, optimisation efficiency needs to be improved, and insufficient interpretability. These problems can be alleviated by introducing visual perception theory and machine learning methods such as SVM. Visual perception theory can provide guidance for interface optimisation that is more in line with the physiological characteristics of the human eye, while SVM can improve the optimisation efficiency and enhance the interpretability of the model.

1.2. Motivation and contribution. Existing HCI optimisation methods usually rely on designers' experience or simple heuristic algorithms, which are difficult to effectively deal with the trade-offs of multiple design goals in complex systems. The significant differences in user requirements and interface element characteristics in different application scenarios often make generic optimisation methods unsatisfactory in specific domains. In addition, traditional optimisation methods are computationally inefficient when dealing with large-scale interface elements, making it difficult to meet the demand for real-time optimisation. At the same time, these methods often ignore the visual perception characteristics of the human eye, resulting in a gap between the optimisation results and the actual user experience. In order to solve the above problems, this paper proposes a method for optimising the human-computer friendly interaction interface of a product based on visual perception and support vector machine, which improves the rationality of the interface layout and user experience. The main innovations and contributions of this work include: (1) Aiming at the challenge of multi-objective trade-offs in human-computer interaction interface optimisation, the visual perception-based optimisation model proposed in this paper is able to better balance multiple factors, such as the importance of

interface elements, the frequency of use and the intensity of visual perception. By introducing the Visual-perceptual Efficiency Index (VEI), the quantitative consideration of the visual characteristics of the human eye is achieved, which improves the practicality and user-friendliness of the optimisation results. This improvement is especially significant in the interface optimisation of complex systems. (2) To address the problem of inefficient optimisation of large-scale interface elements, this paper proposes a hybrid optimisation method combining SVM and Genetic Algorithm (GA). SVM is used to quickly generate the initial layout scheme, while GA is used to carry out global optimisation, which significantly improves the optimisation efficiency. This method is not only capable of handling large-scale interface elements, but also obtains high-quality optimisation results in a reasonable time, providing the possibility of real-time interface optimisation. (3) In order to improve the interpretability and reliability of the optimisation results, this paper proposes an interface division method based on visual perception intensity levels. This method divides the human-computer interaction interface into different visual perception intensity regions, which provides clear guidance for the layout of interface elements. Meanwhile, intelligent classification and layout of different types of interface elements is achieved by introducing SVM classifiers, which enhances the versatility and adaptability of the optimisation method. Ultimately, this paper validates the effectiveness of the proposed method through a real case study. In the optimisation of the operator interface of a mechanical product, the new method significantly improves the VEI, and also achieves significant improvements in user evaluation, including improved operational efficiency, reduced error rate and increased user satisfaction. These results demonstrate the potential and value of the method in this paper in practical applications.

2. Relevant technologies.

2.1. Visual characteristics of cones. The human eye is the main sensory organ for acquiring external visual information, and about 80% of external information is obtained through the visual system. The cone cells in the human retina show a non-uniform distribution in their physiological structure, and this distribution has an important influence on the visual perception of the human eye [16, 17]. Cone cells in the human retina are mainly responsible for colour vision and high-resolution vision. Humans usually have three types of optic cone cells, which are sensitive to red, green and blue light respectively, and this forms the basis of human colour vision. The responsive properties of the optic cone cells allow the human eye to distinguish between different colours. This perceptual mechanism is important in designing the colour scheme of a user interface to ensure that users can easily identify and distinguish between different elements. Cone cells are densely distributed in the central concave region of the retina, which is the most sensitive part of the human eye. The density of cones in the central concave region determines our ability to see detail, which is critical for designing high-resolution displays and icon sizes.

As shown in Figure 1, the central concave area is very densely populated with cone cells, which is why this area is the most acute part of our vision and is responsible for high-resolution vision and colour perception. This figure shows how the density of cone cells is highest in the central concave region of the retina and gradually decreases in the limbic region [18]. As the position moves towards the edge of the retina, the number of optic cone cells decreases significantly. This non-uniform distribution helps to explain why the centre of our vision captures more detail and colour information, whereas peripheral vision is more sensitive to motion and objects in low light conditions. Cone cells are most densely distributed in the central recess region of the retina, and the density of cone cells shows a clear tendency to decrease with increasing centrifugal distance. This non-uniform

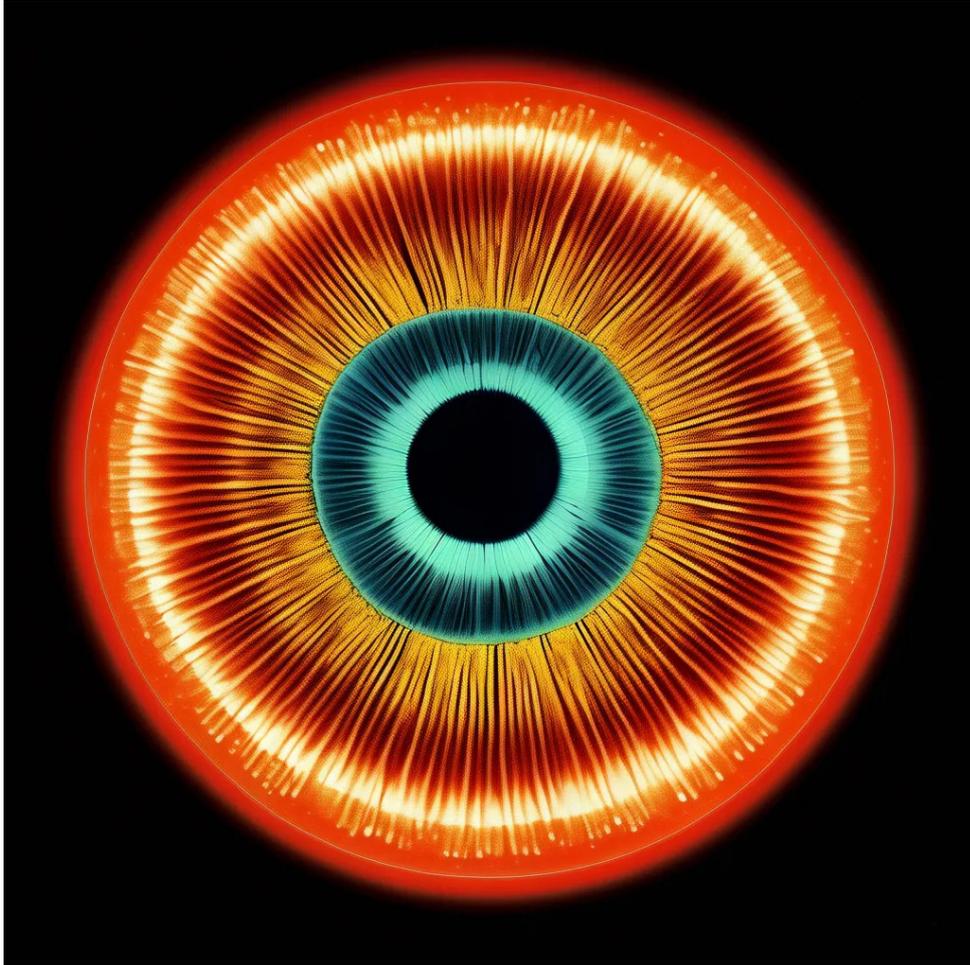


Figure 1. Schematic illustration of the non-uniform distribution of cone cells in the human eye.

distribution can be described by the following mathematical model:

$$D(e) = D_{\max} \exp(-k e^2). \quad (1)$$

where $D(e)$ denotes the density of the optic cone cells when the centrifugal degree is e , D_{\max} is the maximum density at the central concavity, and k is the attenuation coefficient.

This distributional property leads to the “centrifugal effect” of the human eye’s visual perception [19], i.e., as the distance between the target and the gaze point increases, the difficulty of the human eye’s localisation of the target also increases. This distribution of cone cells determines the highest visual acuity and spatial resolution in the centre of the human visual field. In the normal direction of the human eye, the central concave region occupies about 2° of the visual angle, and the visual perception ability is the strongest in this region. As the viewing angle increases, the visual perception ability gradually decreases [20]. It is generally believed that the effective gaze angle of the human eye for visual focusing ranges from 24° to 26° . The relationship between visual acuity and centrifugality can be expressed by the following equation:

$$VA(e) = \frac{VA_0}{1 + e/e_2}. \quad (2)$$

where $VA(e)$ is the visual acuity at centrifugal degree e , VA_0 is the maximum visual acuity at the central concavity, and e_2 is the centrifugal degree when acuity is reduced to half.

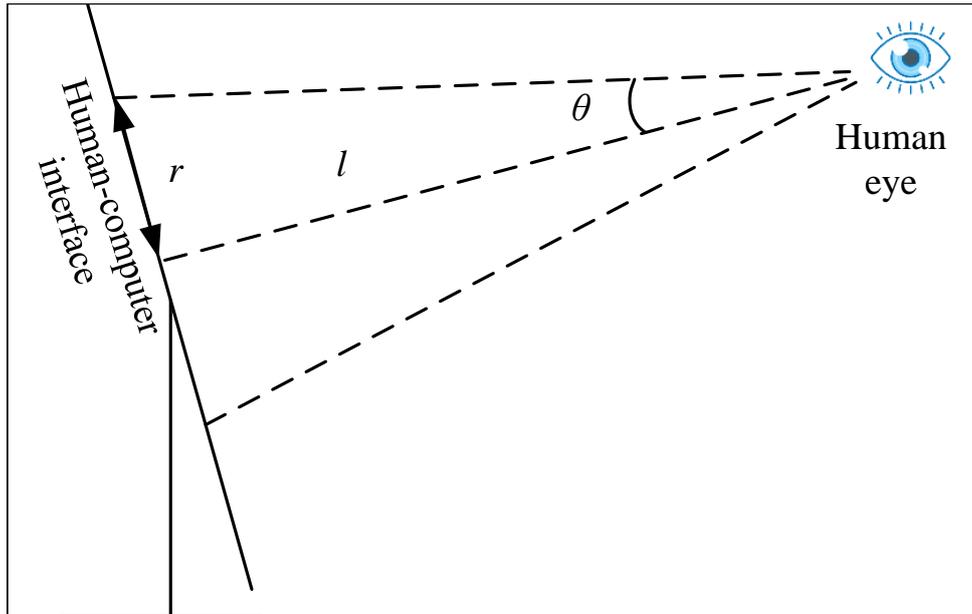


Figure 2. Visual perception domain of the human eye.

In human-computer interaction interface design, it is important to make full use of this visual feature of the visual cone cells. Specifically, placing important visual elements in the centre of the field of view can improve users' visual perception efficiency and operation accuracy. On the contrary, placing secondary information in the peripheral area of the visual field can avoid interference with the main task. In order to quantify this effect, we can introduce a visual perceptual efficiency index (VEI):

$$\text{VEI} = \sum_{i=1}^n w_i \text{VA}(e_i), \quad (3)$$

where w_i is the importance weight of the i th visual element and $\text{VA}(e_i)$ is the visual acuity of the location of the element. By maximising the VEI, the optimal design of the human-computer interaction interface can be achieved.

Therefore, optimising the design of human-computer interaction interfaces based on the distribution characteristics of visual cone cells can significantly improve the usability and user experience of the interfaces. Such optimisation not only improves the user's operating efficiency, but also reduces visual fatigue, which is particularly important for systems that are used for long periods of time. This section outlines the basic visual characteristics of cone cells and introduces the relevant mathematical models, which lays a theoretical foundation for the subsequent establishment of visual perception models and optimisation of human-computer interaction interfaces.

2.2. Visual perception intensity level. Based on the cone cell distribution characteristics discussed above, we need to further establish the visual perception intensity level model to provide more specific guidance for the optimal design of human-computer interaction interfaces. When the human eye's vision spreads from the centre to the periphery of the visual field, the visual perception intensity shows a decreasing characteristic consistent with the distribution of the optic cone cells [21, 22]. In order to quantify this characteristic, we can classify HCI into different visual perception intensity levels.

Let the distance between the human eye and the human-computer interaction interface be l , the angle between the line of sight from the human eye to the target point and

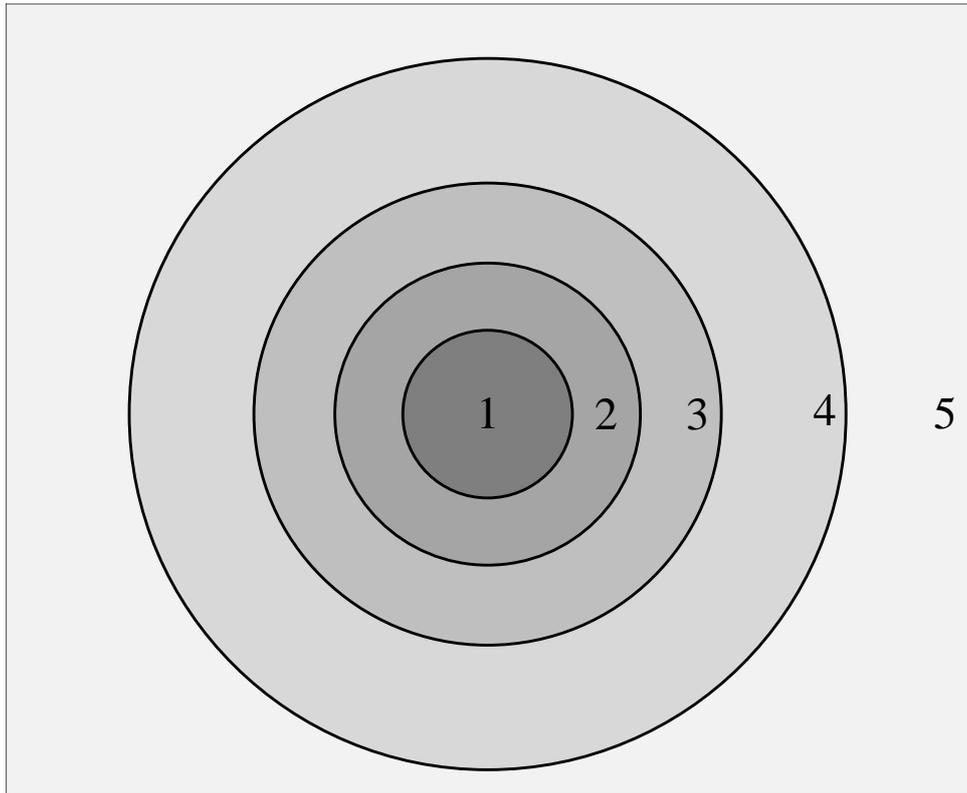


Figure 3. Visual Perception Intensity Modelling.

the centre line be θ , and the radius of the visual perception domain be r . The visual perceptual domain of the human eye [23] is shown in Figure 2. According to the geometric relationship, we can get:

$$r = l \tan \theta. \quad (4)$$

Taking the visual perception centre of the HCI interface as a circle, we can divide the interface into areas with different visual perception levels. The visual perception level decreases from inside to outside. Considering that most HCI interfaces are rectangular, we can divide the rectangular interface into visual domains with different visual perception levels, as shown in Figure 3.

For ease of computation and implementation, we can discretise the visual perceptual domain. Let the side length of the smallest unit square be a , and its area be equal to the area of the first level of the visual perception domain, i.e.:

$$a^2 = \pi r^2. \quad (5)$$

From this, the side length of the square $a = r$. Based on this minimal unit, we can divide the whole HCI interface into different levels of visual perception domains.

We can define the visual perception intensity level as x_j , where $j = 1, 2, \dots, m$, and m is the maximum number of levels. Usually, we can divide the interface into 5–7 levels according to the actual needs. The visual perception intensity corresponding to each grade can be expressed by the visual acuity function $VA(e)$ in Section 2.1:

$$x_j = VA(e_j) = \frac{VA_0}{1 + e_j/e_2}. \quad (6)$$

where e_j is the average centrifugal degree of the visual perceptual domain at level j .

This visual perception intensity classification method provides a quantitative basis for the design of human-computer interaction interface. In practical application, we can

reasonably arrange the position of interface elements according to the characteristics of different visual perception level areas. For example, important control elements or key information are placed in the high-level (centre) area, while secondary information or infrequently used functions are placed in the low-level (peripheral) area. In this way, we can make full use of the visual perception characteristics of the human eye to improve the usability and user experience of the interface.

2.3. Principle of SVM. SVM is a powerful machine learning algorithm widely used in pattern recognition and data classification [24, 25]. In this study, we introduce SVM into the HCI optimisation design process to improve the intelligence and efficiency of optimisation.

The core idea of SVM is to find an optimal hyperplane in the feature space such that samples of different classes can be maximally separated. For the linearly separable case, the goal of SVM is to maximise the classification interval. For the non-linearly separable case, SVM maps the samples to a high-dimensional feature space by means of a kernel trick, where a linear classification boundary is found.

Consider a binary classification problem given a set of training samples $\{(x_i, y_i)\}_{i=1}^N$, where $x_i \in \mathbb{R}^d$ is the d -dimensional feature vector and $y_i \in \{-1, +1\}$ is the category labels. The optimisation objective of the SVM can be expressed as:

$$\min_{w, b, \{\xi_i\}} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^N \xi_i \quad (7)$$

subject to

$$y_i(w^\top x_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad i = 1, \dots, N. \quad (8)$$

where w is the weight vector, b is the bias term, ξ_i is the slack variable, and C is the penalty parameter to balance interval maximisation and classification error minimisation.

By introducing Lagrange multipliers and applying dyadic theory, we can obtain the dyadic problem for SVMs:

$$\max_{\{\alpha_i\}} \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j x_i^\top x_j \quad (9)$$

subject to

$$\sum_{i=1}^N \alpha_i y_i = 0, \quad 0 \leq \alpha_i \leq C, \quad i = 1, \dots, N. \quad (10)$$

where α_i is the Lagrange multiplier.

Solving this pairwise problem yields the optimal α^* , which in turn yields w^* and b^* . For the nonlinearly differentiable case, we introduce the kernel function $K(x_i, x_j)$, which implicitly maps the samples to the high-dimensional feature space. Commonly used kernel functions include [26]: (1) Polynomial kernels: $K(x_i, x_j) = (\gamma x_i^\top x_j + r)^d$. (2) Radial basis function (RBF) kernel: $K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2)$. (3) Sigmoid nucleus: $K(x_i, x_j) = \tanh(\gamma x_i^\top x_j + r)$.

In the human-computer interaction interface optimisation problem, we can take the features of interface elements (e.g., importance, frequency of use, intensity of visual perception, etc.) as input features, and the optimal positions of the elements as output labels. By training the SVM model, we can learn a function that maps the interface element features to the optimal location.

Specifically, we can construct the feature vector $x = [d_i, f_i, x_j, \dots]^\top$, where d_i is the importance of the element (as defined in Section 2.1), f_i is the frequency of use, and x_j is the visual perception intensity level (as defined in Section 2.2), as well as other relevant

features. The output label y_i can be either a discrete positional category or a continuous coordinate value, depending on how we define the optimisation problem.

In this way, SVM can help us find an intelligent layout strategy for interface elements that takes into account both the visual perception properties of the human eye and the attributes and usage characteristics of the elements themselves. In the next chapter, we will discuss in detail how to combine SVM with the visual perception model introduced earlier to construct a complete HCI interface optimisation method.

3. Methods for optimising the human-machine interaction interface of products.

3.1. HCI optimisation model. Based on the visual perception characteristics and SVM principles introduced in the previous sections, we will now construct a comprehensive HCI interface optimisation model. This model aims to organically combine visual perception strength, interface element characteristics and machine learning techniques to achieve more intelligent and efficient interface layout optimisation.

First, we define the required symbols, as shown in Table 1.

Table 1. Symbols of the HCI optimisation model

No.	Symbol	Define
1	$Y = \{y_1, y_2, \dots, y_n\}$	HCI interface visual perceptual elements collection
2	$D = \{d_1, d_2, \dots, d_n\}$	Element importance set, $d_i \in [0, 1]$
3	$F = \{f_1, f_2, \dots, f_n\}$	Element usage frequency set, $f_i \in [0, 1]$
4	$S = \{s_1, s_2, \dots, s_n\}$	Elemental area set
5	$X = \{x_1, x_2, \dots, x_m\}$	A collection of visual perception intensity levels, as defined in Section 2.2
6	$Q = \{q_1, q_2, \dots, q_m\}$	Aggregate area of perceived area for each intensity level
7	$P = \{p_1, p_2, \dots, p_n\}$	The set of element positions, $p_i = (u_i, v_i)$ denotes the coordinates of element y_i

Our goal is to find an optimal element layout scheme that maximises the overall visual perception efficiency. For this purpose, we define the VEI as follows:

$$V_{EI} = \sum_{i=1}^n \sum_{j=1}^m w_i \cdot x_j \cdot \frac{q_{ij}}{s_i} \quad (11)$$

where $w_i = \alpha d_i + \beta f_i$ is the combined weight of element y_i ; α and β are the weight coefficients that satisfy $\alpha + \beta = 1$; q_{ij} denotes the area occupied by the element y_i in the j -th region of the visual perception intensity.

The optimisation problem can be formulated as:

$$\max_P V_{EI} \quad (12)$$

subject to

$$\begin{cases} \sum_{i=1}^n q_{ij} = q_j, & j = 1, \dots, m, \\ \sum_{j=1}^m q_{ij} = s_i, & i = 1, \dots, n, \\ \sum_{j=1}^m q_j = \sum_{i=1}^n s_i, \\ p_i \in \Omega, & i = 1, \dots, n, \end{cases} \quad (13)$$

where Ω denotes the feasible layout region.

To introduce SVM into the optimisation process, we construct the following feature vectors:

$$\mathbf{x}_i = [d_i, f_i, s_i, x_{c(i)}]^\top \quad (14)$$

where $x_{c(i)}$ denotes the visual perception intensity level of the current location of the element y_i .

We transform the optimisation problem into a classification problem, i.e., divide the interface region into a number of discrete positional categories and train the SVM classifier to predict which positional category each element should belong to. The training data can be obtained from expert experience or user feedback. The former is provided by HCI design experts with excellent interface layout solutions. The latter is collected through A/B testing to collect feedback and ratings of real users on different layouts.

The trained SVM model can quickly predict the optimal position of a given element, which greatly improves the optimisation efficiency. At the same time, in order to ensure the global optimality, we can take the prediction result of SVM as the initial solution, and then fine-tune it by local search or other optimisation algorithms. This comprehensive model combines visual perception theory and machine learning techniques, which not only considers the physiological characteristics of the human eye, but also makes full use of historical data and experience.

3.2. SVM-based optimisation model solving. Based on the HCI optimisation model developed in Section 3.1, we will use a combination of SVM classification and GA to solve the optimisation problem. This approach takes advantage of both the fast classification capability of SVM and the global search by GA to obtain a better solution.

3.2.1. SVM classification model. First, we divide the interface region into K discrete location categories, each corresponding to a specific region. Then, we construct SVM classifiers to predict which location category each interface element should belong to. The feature vectors are defined as shown in Equation (14). The label is defined as $y_i \in \{1, 2, \dots, K\}$, which denotes the optimal location category to which the element y_i belongs.

We use the RBF kernel function to train the SVM classifier:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2) \quad (15)$$

where γ is the kernel function parameter, which needs to be determined by cross-validation.

3.2.2. GA-based fine-tuning. After obtaining the initial layout scheme using SVM classifier, we use GA for further optimisation. The main steps of GA are as follows:

Step 1. Coding: encode the location of each interface element as a chromosome. For example, an integer encoding can be used, with each gene indicating the category of location to which the element belongs.

Algorithm 1 Pseudo-code for solving SVM based optimisation model

Input: set of interface elements Y , set of feature vectors X , SVM classifier M
Output: Optimal Layout Programme P^*

- 1: Classify all elements using SVM classifier M to get initial layout P_0
- 2: Initialise GA population, add P_0 to the population
- 3: **while** termination condition not met **do**
- 4: Calculate the fitness of each individual in the population (VEI)
- 5: Selection operation
- 6: Perform crossover operations
- 7: Perform mutation operations
- 8: Apply elite retention strategies
- 9: **end while**
- 10: **return** P^*

Step 2. Initialise the population: use the prediction result of SVM as one individual in the initial population, and the rest of the individuals are randomly generated.

Step 3. Adaptation function: The VEI defined in section 3.1 is used as the adaptation function, as shown in Equation (11).

Step 4. Selection: The roulette wheel selection method is used to select the best individuals.

Step 5. Crossover: use a single-point crossover operation with a crossover probability of p_c .

Step 6. Variation: use uniform variation with a variation probability of p_m .

Step 7. Elite retention: retain the best individuals in each generation.

Step 8. Termination condition: the maximum number of iterations is reached or there is no improvement in the optimal solution for consecutive generations.

The pseudo-code for solving the SVM-based optimisation model is shown in Algorithm 1.

In order to obtain the best performance, the parameters of SVM and GA need to be tuned. The main parameters include:

- (1) SVM parameters: kernel function parameter γ , and penalty coefficient C .
- (2) GA parameters: population size N , crossover probability p_c , mutation probability p_m , and maximum number of iterations T . Grid search and cross-validation methods can be used to determine the optimal values of these parameters.

3.2.3. Algorithm complexity analysis. Assume that the number of interface elements is n , the number of location categories is K , the GA population size is N and the maximum number of iterations is T . Then the time complexity of the algorithm includes SVM classification $\mathcal{O}(n)$, GA initialisation $\mathcal{O}(N)$ and GA iteration $\mathcal{O}(TNK)$. Therefore, the overall time complexity is $\mathcal{O}(n + N + TNK)$.

This hybrid optimisation approach based on SVM and GA combines the fast prediction capability of machine learning and the global search capability of meta-heuristic optimisation algorithms to obtain high-quality HCI layout solutions in a reasonable time.

4. Experimental results and analysis. In order to verify the effectiveness of our proposed method of optimising the HCI interface of a product based on visual perception and SVM, we choose the operation interface of a mechanical product as an example for analysis. This case is representative because the design of the product operation interface is directly related to the accuracy and timeliness of equipment control, which has an important impact on operation safety and efficiency.

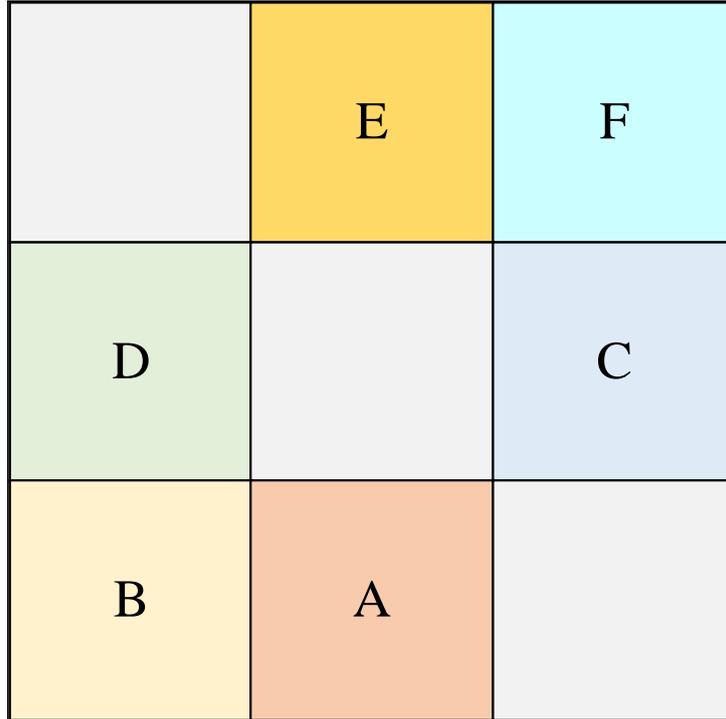


Figure 4. Product interface before optimisation

Table 2. Parameters of the hydraulic interface elements

Module	Importance d_i	Frequency f_i	Area s_i
A	0.8	0.9	18
B	0.7	0.8	22
C	0.6	0.7	79
D	0.9	0.2	198
E	0.5	0.6	51
F	0.3	0.4	18

4.1. **Experimental setup.** Based on the statistical data from several experiments, we set the distance between the human eye and the operator console interface to L cm. Based on the visual perception intensity level model in Section 2.2, we classified the interface into 6 visual perception levels ($K = 6$). The product operator interface has six main modules: pressure display (A), flow control (B), temperature monitoring (C), emergency stop (D), system status (E), and auxiliary functions (F), as shown in Figure 4.

4.2. **Data preparation.** We collected the following data as model inputs:

- (1) Elemental importance: $D = d_A, d_B, d_C, d_D, d_E, d_F$
- (2) Frequency of use: $F = f_A, f_B, f_C, f_D, f_E, f_F$ (16)
- (3) Element area: $S = s_A, s_B, s_C, s_D, s_E, s_F$

Through expert evaluation and historical data analysis, we obtained initial values for these parameters, as shown in Table 2:

4.3. **SVM model training.** We use historical data and excellent layout schemes designed by experts as training data to construct feature vectors and train SVM classifiers. We divide the interface region into 9 location categories ($K = 9$) corresponding to a 3×3

Table 3. Parameterisation of the SVM solver model

Parameters	Numerical value
RBF kernel function parameters	$\gamma = 0.1$
SVM penalty factor	$C = 10$
Population size	$N = 100$
Crossover probability	$p_c = 0.8$
Probability of mutation	$p_m = 0.1$
Maximum number of iterations	$T = 1000$
Weighting factor	$\alpha = 0.6, \beta = 0.4$

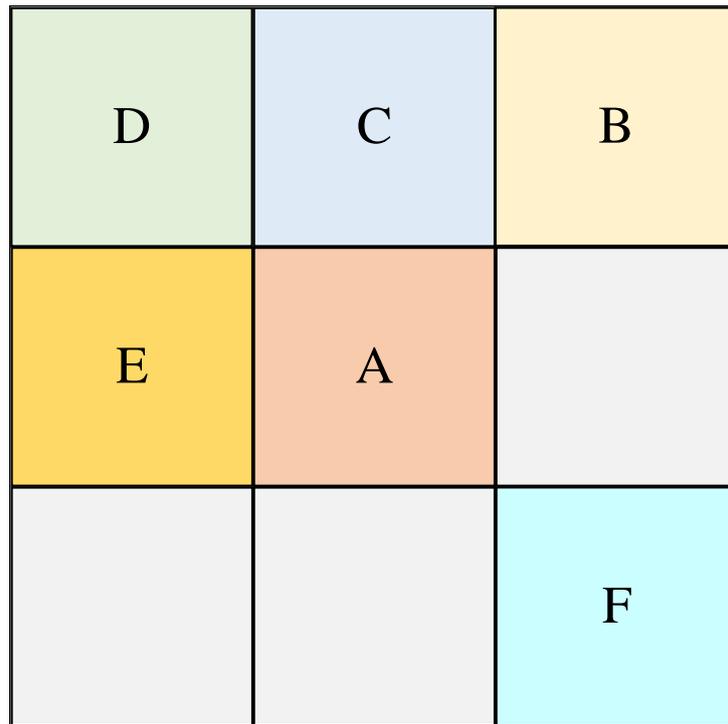


Figure 5. Product interface before optimisation

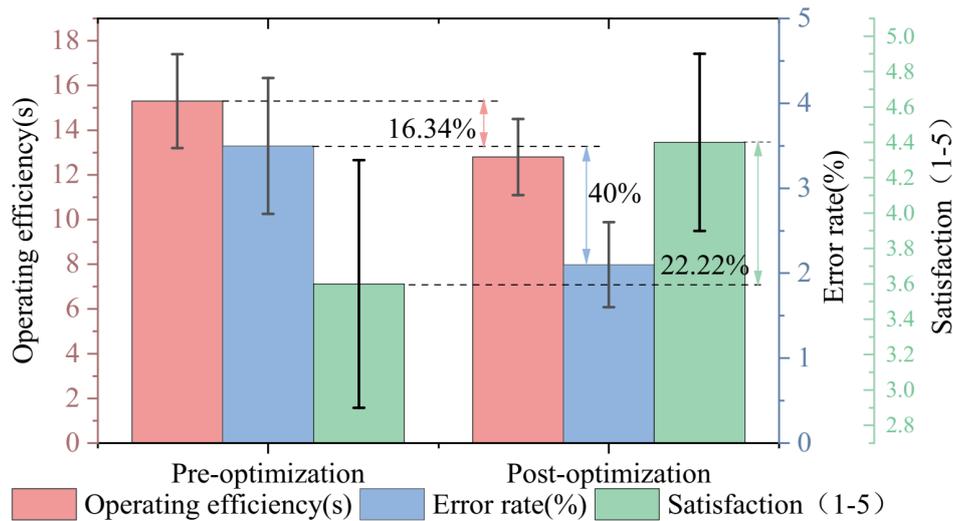
grid layout. Using grid search and 5-fold cross-validation, we obtained the optimal SVM parameters. Then, based on the initial classification results of the SVM, we use a genetic algorithm for further optimisation. In addition, we use the VEI defined in Section 3.1 as the fitness function, and the weight coefficients are set to $\alpha = 0.6$ and $\beta = 0.4$. Ultimately, the parameter settings of the SVM solution model proposed in this paper are shown in Table 3.

4.4. Result analysis. After optimisation, we obtained the new product operator interface layout, as shown in Figure 5. Table 4 demonstrates the location categories and corresponding visual perception intensity levels of each module before and after optimisation.

The main changes to the product interface before and after optimisation include: (1) The Emergency Stop (D) module was moved to the upper left corner, which is one of the most visually perceptible areas. (2) The pressure display (A) and flow control (B) modules have been moved closer to the centre. (3) The temperature monitoring (C) module has been moved to the upper centre position to improve its visibility. (4) The system status (E) module has been moved to the centre position for easy monitoring. (5) The auxiliary

Table 4. Layout comparison before and after optimisation

Module	Position category before optimisation	Visual intensity before optimization	Position category after optimisation	Visual intensity after optimization
A	9	x_6	5	x_3
B	8	x_5	6	x_4
C	7	x_5	3	x_2
D	4	x_3	1	x_1
E	6	x_4	4	x_3
F	3	x_2	8	x_5

Figure 6. User assessment results (mean \pm standard deviation)

function (F) module is kept in the lower right corner, in keeping with its low-priority nature.

Calculating the VEI before and after the optimisation, we get $V_{EI}^{\text{before}} = 4.718$ before optimisation and $V_{EI}^{\text{after}} = 5.246$ after optimisation. The results show that the VEI after optimisation has increased by 11.19%, which indicates that the new layout is more in line with the visual perception characteristics of the human eye and the operational needs. The emergency stop module (D) is placed in the area with the strongest visual perception, which is in line with its high importance characteristics. The pressure display (A) and flow control (B) modules have been moved closer to the centre, taking into account their high frequency of use. The auxiliary function module (F) has been moved to the peripheral area, which is consistent with its low importance and low frequency of use.

4.5. User evaluation. In order to further verify the optimisation effect, 20 operators were invited to evaluate the interface before and after the optimisation. The evaluation indexes include: operation efficiency, error rate and subjective satisfaction. The results are shown in Figure 6.

The results of the user evaluation showed that the optimised interface showed a significant improvement in all aspects, especially the significant reduction of the error rate by 40%, which is essential for the security of the hydraulic operating system. At the same time, user satisfaction increased by 22.22%. In summary, this example analysis proves the effectiveness of our proposed human-computer interaction interface optimisation method based on visual perception and support vector machines. The method not only improves

the visual perception efficiency of the interface, but also significantly improves the operation performance and user experience.

5. Conclusion. In this paper, a product human-computer friendly interaction interface optimisation method based on visual perception and support vector machine is proposed, which effectively solves the limitations of traditional optimisation methods in dealing with multi-objective trade-offs of complex systems and considering the visual characteristics of the human eye. By introducing VEI, the model is able to more accurately quantify the influence of the visual characteristics of the human eye on the interface layout, which significantly improves the practicality of the optimisation results. In addition, the hybrid optimisation approach combining SVM and GA further enhances the ability to handle large-scale interface elements, ensuring the efficiency of the optimisation process and the quality of the results. The following conclusions can be drawn from the experiments conducted on the operation interface of a mechanical product:

(1) The optimisation model based on visual perception can effectively balance multiple factors such as the importance of interface elements, frequency of use and intensity of visual perception, which improves the reasonableness of the optimisation results.

(2) The combination of SVM and GA not only improves the optimisation efficiency, but also enhances the versatility and adaptability of the method, enabling it to cope with different types of interface optimisation problems.

(3) The interface segmentation method of visual perception intensity levels improves the interpretability of the optimisation results and provides clear guidance for interface design.

(4) The proposed method performs well in real-world applications and not only improves the visual perception efficiency, but also significantly improves the user experience in terms of operation efficiency, error rate and satisfaction. The experiments in this paper are mainly based on a single mechanical product operator interface, which may limit the generalisation ability of the method. Future work should consider validating the method on more different types of human-computer interaction interfaces, such as mobile applications, web interfaces, etc., in order to further validate the effectiveness and applicability of the method. In addition, the introduction of deep learning techniques into the optimisation process can be explored to improve the model's ability to understand and process the relationships of complex interface elements.

REFERENCES

- [1] H. R. Hartson, and D. Hix, "Human-computer interface development: concepts and systems for its management," *ACM Computing Surveys*, vol. 21, no. 1, pp. 5–92, 1989.
- [2] R. Sharma, V. I. Pavlovic, and T. S. Huang, "Toward multimodal human-computer interface," *Proceedings of the IEEE*, vol. 86, no. 5, pp. 853–869, 1998.
- [3] T.-Y. Wu, H. Li, S. Kumari, and C.-M. Chen, "A Spectral Convolutional Neural Network Model Based on Adaptive Fick's Law for Hyperspectral Image Classification," *Computers, Materials & Continua*, vol. 79, no. 1, pp. 19–46, 2024.
- [4] T.-Y. Wu, A. Shao, and J.-S. Pan, "CTOA: Toward a Chaotic-Based Tumbleweed Optimization Algorithm," *Mathematics*, vol. 11, no. 10, 2339, 2023.
- [5] T.-Y. Wu, H. Li, and S.-C. Chu, "CPPE: An Improved Phasmatodea Population Evolution Algorithm with Chaotic Maps," *Mathematics*, vol. 11, no. 9, 1977, 2023.
- [6] A. Blair-Early, and M. Zender, "User interface design principles for interaction design," *Design Issues*, vol. 24, no. 3, pp. 85–107, 2008.
- [7] L. Lamberti, "An efficient simulated annealing algorithm for design optimization of truss structures," *Computers & Structures*, vol. 86, no. 19–20, pp. 1936–1953, 2008.

- [8] A. W. Joseph, and R. Muruges, "Potential eye tracking metrics and indicators to measure cognitive load in human-computer interaction research," *Journal of Scientific Research*, vol. 64, no. 1, pp. 168–175, 2020.
- [9] S. Cheng, and Y. Liu, "Eye-tracking based adaptive user interface: implicit human-computer interaction for preference indication," *Journal on Multimodal User Interfaces*, vol. 5, pp. 77–84, 2012.
- [10] F. Du, C. Plaisant, N. Spring, and B. Shneiderman, "Visual interfaces for recommendation systems: Finding similar and dissimilar peers," *ACM Transactions on Intelligent Systems and Technology*, vol. 10, no. 1, pp. 1–23, 2018.
- [11] M. Soui, M. Chouchane, M. W. Mkaouer, M. Kessentini, and K. Ghedira, "Assessing the quality of mobile graphical user interfaces using multi-objective optimization," *Soft Computing*, vol. 24, pp. 7685–7714, 2020.
- [12] Z. Lv, F. Poiesi, Q. Dong, J. Lloret, and H. Song, "Deep learning for intelligent human-computer interaction," *Applied Sciences*, vol. 12, no. 22, 11457, 2022.
- [13] H. Sun, X. Zhang, Y. Zhao, Y. Zhang, X. Zhong, and Z. Fan, "A novel feature optimization for wearable human-computer interfaces using surface electromyography sensors," *Sensors*, vol. 18, no. 3, 869, 2018.
- [14] Y. K. Meena, H. Cecotti, K. Wong-Lin, A. Dutta, and G. Prasad, "Toward optimization of gaze-controlled human-computer interaction: Application to hindi virtual keyboard for stroke patients," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 26, no. 4, pp. 911–922, 2018.
- [15] Y. Sun, C. Xu, G. Li, W. Xu, J. Kong, D. Jiang, B. Tao, and D. Chen, "Intelligent human computer interaction based on non redundant EMG signal," *Alexandria Engineering Journal*, vol. 59, no. 3, pp. 1149–1157, 2020.
- [16] J. A. Diego-Mas, D. Garzon-Leal, R. Poveda-Bautista, and J. Alcaide-Marzal, "User-interfaces layout optimization using eye-tracking, mouse movements and genetic algorithms," *Applied Ergonomics*, vol. 78, pp. 197–209, 2019.
- [17] B. P. Schmidt, A. E. Boehm, W. S. Tuten, and A. Roorda, "Spatial summation of individual cones in human color vision," *PLoS One*, vol. 14, no. 7, e0211397, 2019.
- [18] B. P. Schmidt, R. Sabesan, W. S. Tuten, J. Neitz, and A. Roorda, "Sensations from a single M-cone depend on the activity of surrounding S-cones," *Scientific Reports*, vol. 8, no. 1, 8561, 2018.
- [19] W. N. Grimes, A. Songco-Aguas, and F. R. Rieke, "Parallel processing of rod and cone signals: retinal function and human perception," *Annual Review of Vision Science*, vol. 4, no. 1, pp. 123–141, 2018.
- [20] R. F. Cooper, D. H. Brainard, and J. I. Morgan, "Optoretinography of individual human cone photoreceptors," *Optics Express*, vol. 28, no. 26, pp. 39326–39339, 2020.
- [21] H.-K. Shen, P.-H. Chen, and L.-M. Chang, "Human-visual-perception-like intensity recognition for color rust images based on artificial neural network," *Automation in Construction*, vol. 90, pp. 178–187, 2018.
- [22] M. Manassi, Y. Murai, and D. Whitney, "Serial dependence in visual perception: A meta-analysis and review," *Journal of Vision*, vol. 23, no. 8, pp. 18–18, 2023.
- [23] F. Abd-Alhamid, M. Kent, C. Bennett, J. Calautit, and Y. Wu, "Developing an innovative method for visual perception evaluation in a physical-based virtual environment," *Building and Environment*, vol. 162, 106278, 2019.
- [24] V. K. Chauhan, K. Dahiya, and A. Sharma, "Problem formulations and solvers in linear SVM: a review," *Artificial Intelligence Review*, vol. 52, pp. 803–855, 2019.
- [25] H. Wang, G. Li, and Z. Wang, "Fast SVM classifier for large-scale classification problems," *Information Sciences*, vol. 642, 119136, 2023.
- [26] M. Goudjil, M. Koudil, M. Bedda, and N. Ghoggali, "A novel active learning method using SVM for text classification," *International Journal of Automation and Computing*, vol. 15, pp. 290–298, 2018.