Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

An external attention-based feature ranker for large-scale feature selection

Yu Xue [a], Chenyi Zhang [a], Ferrante Neri [b,a,*], Moncef Gabbouj [c], Yong Zhang [d]

[a] School of Software, Nanjing University of Information Science and Technology, Nanjing, 210000, China
[b] NICE Research Group, Department of Computer Science, University of Surrey, Guildford, GU2 7XS, UK
[c] Faculty of Information Technology and Communication Sciences, Tampere University, Tampere, 33900, Finland
[d] School of Information and Control Engineering, China University of Mining and Technology, Xuzhou, 221000, China

ARTICLE INFO

Keywords:
Feature selection
Attention mechanism
Classification
Neural network

ABSTRACT

An important problem in data science, feature selection (FS) consists of finding the optimal subset of features and eliminating irrelevant or redundant features. The FS task on high-dimensional data is challenging for the FS methods currently available in the literature. To overcome this limitation, we propose a novel feature selection method called External Attention-Based Feature Ranker for Large-Scale Feature Selection (EAR-FS) whose function is based on the logic of an attention mechanism and a hybrid metaheuristic. EAR-FS comprises three interdependent modules: (1) in the training module design, a multilayer perceptron network endowed with an attention module is trained to fit the dataset; (2) in feature ranking by attention, the trained attention module is used for attention updating and to rank features according to their importance; 3) in subset generation, a two-stage heuristic approach is applied to determine a small number of features that still guarantee high-accuracy performance. The experimental benchmark comprised 26 datasets of small, large and very large sizes, ranging from 15 to 12,533 features. Experiments performed against the state-of-the-art algorithms of FS show that our algorithm is efficient at selecting a small number of features from large datasets while guaranteeing excellent levels of classification accuracy. For instance, EAR-FS demonstrated its capability to reduce the features of the 11 Tumor dataset by 97% while maintaining a classifier accuracy of over 93%.

1. Introduction

Due to the rapid development of network information and big data, an increasing amount of information is flooding the entire network [1]. Redundant information poses a great challenge to the classification task. Large-scale datasets often contain plenty of redundant information, and when redundant features are fed into a classifier, this can lead to greatly reduced classification accuracy (CA) [2,3]. Therefore, a data preprocessing technique is required to reduce the sizes of datasets [4]. One of the most popular and effective preprocessing techniques is called feature selection (FS) [5,6]. FS involves the procedure of retaining crucial features while eliminating irrelevant and redundant ones, which can notably enhance classification accuracy and decrease computational time [7]. FS techniques have found widespread application in various domains such as financial analysis, safety prediction, cancer classification, and others.

The FS task can be regarded as a combinatorial optimisation problem whose goal is to find the minimal subset of features that enables the maximal accuracy of predictions [8–10]. Analogously, FS can be considered the variable reduction of a mathematical model, which aims to eliminate irrelevant or redundant features/variables without jeopardising the accuracy of the model. The cardinality of the search space associated with this combinatorial optimisation problem is $(2^n - 1)$ for n-dimensional feature sets. Therefore, for large-scale datasets, an exhaustive search would be unacceptable [11].

The earliest examples of FS methods make use of the heuristic criteria of information statistics to quickly eliminate invalid features. This approach is efficient and versatile for small problems but often ineffective for large-scale datasets. To overcome this limitation, evolutionary computing (EC) has attracted widespread attention in the field of FS. For example, Nguyen et al. [12] proposed a constrained competitive swarm optimiser with a support vector machine (SVM)-based surrogate model for FS. Jiao et al. [13] proposed a similarity-based repetitive subset processing method that effectively removes highly similar and poor solutions. Furthermore, [13] used a constraint approach to prioritise features that contain more strongly correlated information. Zhang et al. [14] used a binary differential evolution to perform the FS task.

Although broadly used, EC-based FS methods face some challenges. For example, due to the large size of the FS search space, EC methods

[*] Corresponding author at: NICE Research Group, Department of Computer Science, University of Surrey, Guildford, GU2 7XS, UK.
E-mail addresses: xueyu@nuist.edu.cn (Y. Xue), 202212490748@nuist.edu.cn (C. Zhang), f.neri@surrey.ac.uk (F. Neri), moncef.gabbouj@tuni.fi (M. Gabbouj), yongzh401@cumt.edu.cn (Y. Zhang).

https://doi.org/10.1016/j.knosys.2023.111084
Received 10 June 2023; Received in revised form 31 August 2023; Accepted 13 October 2023
Available online 18 October 2023

are likely to suffer from search stagnation and premature convergence [15]. In addition, since the evaluation of each candidate solution tends to be expensive, FS methods based on evolutionary algorithms often require long periods to provide satisfactory results [16,17].

Recently, FS methods based on the attention mechanism have been proposed [18]. Gui et al. [19] proposed an FS method based on the attention mechanism within the context of binary classification. Moreover, in the study presented in [19], a separable learning module was designed as the classifier.

It is worth noting that the action of the attention mechanism in deep learning, i.e. choosing the most relevant information from an image or text, is analogous to the FS task, i.e. the selection of the most relevant features. Until recently, there has been limited research on the integration of attention mechanisms with feature selection techniques. In previous studies, the computation involved in attention mechanisms often incurred significant computational costs, and their effectiveness did not demonstrate clear advantages compared to some newer algorithms. Based on this observation, we have proposed a novel feature selection method called External Attention-Based Feature Ranker for Large-Scale Feature Selection (EAR-FS), which successfully combines the neural network model and the FS method to achieve an excellent ability to identify effective features in high-dimensional data. The main contributions of this paper are summarised as follows:

1. We propose a novel feature ranking algorithm that uses attention units to score each feature, retain high-scoring features and remove low-scoring features in order to achieve FS.
2. We propose the use of neural networks with an embedded attention mechanism for feature training to learn feature weights. The integration of the attention mechanism enables the selected features to effectively represent the dataset and makes the FS results more interpretable and intuitively explainable.
3. We propose a novel generation strategy for candidate FS subsets. This strategy takes into account the number of features and its impact on classification accuracy. The proposed generation strategy greatly reduces the computational cost for large-scale datasets and, at the same time, comprehensively explores the search space and guides the search towards promising feature subsets.

The remainder of this paper is organised as follows: Section 2 reviews the related body of work about FS methods and attention mechanisms. Section 3 describes in detail our proposed methodology. Section 4 presents the experimental design and parameters. Section 5 reports the experimental results and analysis. Finally, Section 6 provides the conclusions of this work and suggests future directions for research.

2. Related work

To place our proposal in the proper context, in this section, we provide a brief literature review of FS methods and introduce some advances in the field of attention mechanisms.

2.1. Feature selection methods

Over the past few decades, several FS methods have been proposed. The FS methods in the literature can be broadly divided into filter, wrapper and embedded approaches [20].

The main difference between filter and wrapper approaches is that wrapper approaches include classification algorithms to evaluate the classification performance of the selected features. In filter approaches, FS does not require a classification algorithm. Since wrapper approaches focus more on the performance of the features in feature subsets, wrapper approaches tend to be more computationally expensive but better classified, while filter approaches are computationally cheaper but more general.

Embedded approaches integrate the FS process with the classifier training process, and the two are completed in the same optimisation process, i.e. FS is automatically carried out during the classifier training process [21]. In the embedded methods, convergence is often facilitated by adding the appropriate regularisation to make certain feature weights as small as possible.

In recent years, many FS methods based on deep learning and the attention mechanism have been proposed. Qiu et al. [22] introduced a batch-attention-based self-supervised mechanism into FS, which greatly reduces the dependence of FS tasks on labels. Aboozar et al. [23] proposed the DeepFS method, which is embedded into the DBM classifier. Mirzaei et al. [24] were the first to use teacher–student networks in FS tasks, and their method achieves good results in CA and clustering performance. Furthermore, a set of pioneering feature selection techniques grounded in evolutionary computation and demonstrating robust feature selection prowess have been introduced. In [25], an innovative binary rendition of the Rat Swarm Optimizer (RSO) has been proposed to address FS challenges. This version employs an S-shape transfer function and three distinct crossover techniques to ensure optimal feature selection capability. Awadallah et al. [26] have introduced a binary iteration of the Horse Herd Optimisation Algorithm (HOA) designed for FS tasks. In another work, Braik et al. [27] have presented an enhanced Chameleon Swarm Algorithm (CSA) comprising eight binary versions of CSA coupled with four transfer functions, yielding promising results in terms of feature selection effectiveness.

Nonetheless, despite demonstrating promising performance on small datasets, existing research exhibits certain limitations. Particularly when confronted with large-scale datasets, prevalent methods frequently grapple with challenges such as excessive computational demands and suboptimal feature selection outcomes.

2.2. Attention mechanism

The attention mechanism has been proven to be effective and can be applied to multiple fields of machine learning. It has achieved excellent results in computer vision, natural language processing (NLP) and other fields [28]. The earliest attention mechanisms were proposed in the field of computer vision, which attempted to mimic human attention by focusing the attention on parts of an image rather than the entire image. Mnih et al. [29] used the attention mechanism on the recurrent neural network model for image classification. Bahdanau et al. [30] used an attention-like mechanism to perform translation and alignment simultaneously on machine translation tasks, and their work was the first to apply attention to the NLP field. Hu et al. [31] proposed the adaptive recalibration of a channel by using attention weight. Vaswani et al. [32] proposed the transformer model and demonstrated that the attention mechanism is sufficient to build an advanced model. This has greatly promoted the development of attention mechanisms. Vision Transformer [33] was the first pure transformer structure to achieve excellent results in computer vision. After that, many better-performing transformer models were proposed, such as DERT [34], T2T [35], IPT [36], Swin-T [37] and NA [38].

3. Proposed method

In this section, we first introduce the overall architecture of EAR-FS which belongs to the class of embedded methods, and then present the details of each element composing the framework. The data flow of the entire model is shown in detail in Fig. 1. We divided the model into three blocks: training module design (TMD), feature ranking by attention (FRA) and subset generation (SG). The TMD block is composed of a multilayer perceptron (MLP) endowed with the attention mechanism. The TMD block processes the entire dataset with associated labels to train both the MLP and the attention module after normalising the data. Then, the FRA block processes the entire dataset (after normalisation) and trained attention module to rank the features based

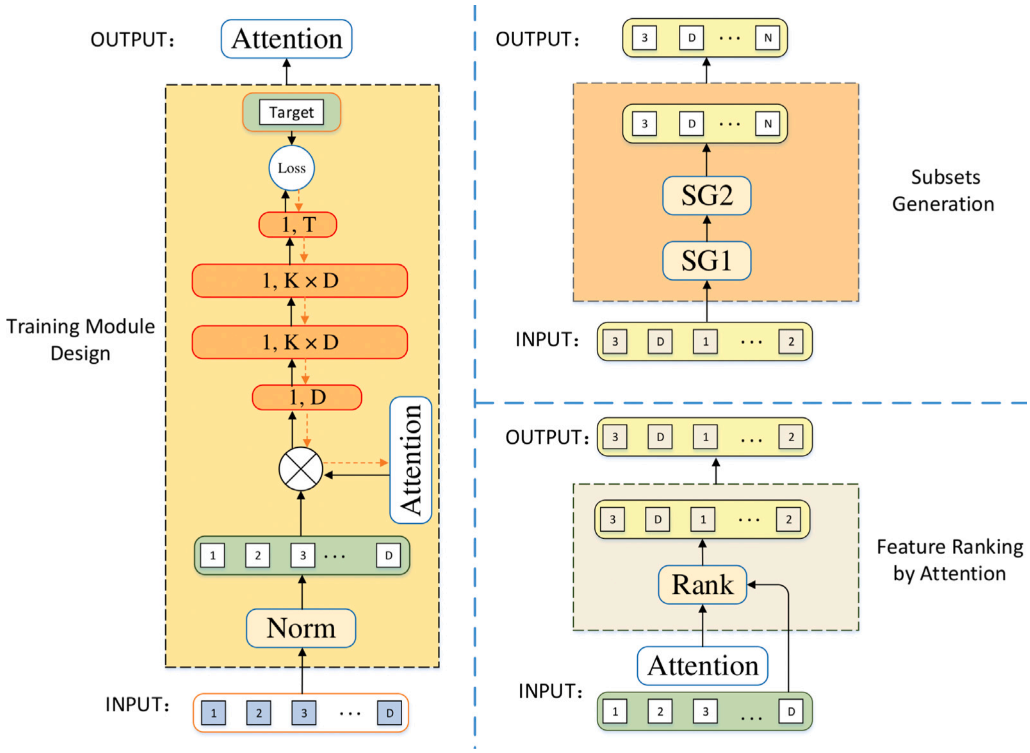


Fig. 1. Overall architecture of External Attention-Based Feature Ranker for Large-Scale Feature Selection. The original dataset (purple cells on white background) is inserted into the training module design to train the multilayer perceptron and attention module after data normalisation (white cells on green background). The trained attention module is then used by the feature ranking by attention block to rank the features (beige cells on yellow background). The ranked features and normalised data are then processed by the subset generation block to select the most appropriate N features out of the initial full list of D features.

on their importance with respect to their impact on accuracy (per the classification task performed by the MLP network). Finally, the SG block processes the list of features ranked by the attention module to select the suggested feature subset, i.e. to perform the FS task.

3.1. Training module design

Let $\mathbf{X}$ be the original dataset. Each instance $\mathbf{x}$ can be seen as a vector

$$\mathbf{x} = \left(x_1, x_2, \dots, x_D\right)$$

where each component represents a feature. The TMD block contains a normalisation module that preprocesses, for each instance, each feature of the dataset using the following formula:

$$x_i^* = \frac{x_i - min(\mathbf{x})}{max(\mathbf{x}) - min(\mathbf{x})} \tag{1}$$

thus generating the normalised vectors $\mathbf{x}^*$.

The normalised data feed the MLP network for its training. We used an MLP with two hidden layers. The exact size of the MLP network depends on the dataset. Let D be the original number of features in the dataset. We set the number of neurons in each hidden layer as $4 \times D$. This decision was made as a trade-off between the accuracy of the trained MLP network and the computational cost of its training.

The attention module is a positive definite vector of D whose elements signify the importance of each feature

$$\mathbf{a} = \left(a_1, a_2, \dots, a_D\right)$$

whose elements are processed by a sigmoid function to generate the vector $\mathbf{a}^*$

$$a_i^* = \frac{1}{\left(1 + e^{-a_i}\right)}.$$

At the beginning of the training, all the elements a_i^* are set equal to 1. Then, following the training, the values of a_i^* are varied. The attention vector $\mathbf{a}^*$ is used to emphasise the most important features. Specifically, the input for the MLP training are the vectors $\mathbf{x}^{\mathbf{a}}$ calculated as

$$\mathbf{x}^{\mathbf{a}} = \mathbf{x}^* \odot \mathbf{a}^*$$

where $\odot$ indicates the Hadamard product, i.e. the element-by-element product of the vectors.

The output $\mathbf{o}$ of the MLP network is a vector with as many elements as the number of classes and whose elements signify the probability that the instance $\mathbf{x}$ belongs to the corresponding class. The output vector $\mathbf{o}$ is calculated as

$$\mathbf{o} = \mathbf{W_3}\left(\mathbf{W_2}\left(\mathbf{W_1}\mathbf{x}^{\mathbf{a}} + \mathbf{b_1}\right) + \mathbf{b_2}\right) + \mathbf{b_3} \tag{2}$$

where $\mathbf{W_1}$, $\mathbf{W_2}$ and $\mathbf{W_3}$ are the weight matrices connecting the four MLP layers (input, two hidden and output), and $\mathbf{b_1}$, $\mathbf{b_2}$ and $\mathbf{b_3}$ are the corresponding bias vectors. In Eq. (2), the output vector $\mathbf{o}$ is calculated as the result of the products and sums of the matrices/vectors.

To determine the weights and biases of the MLP and the elements of the attention module, training by means of the dataset $\mathbf{X}$ is performed. Supervised learning is carried out by backpropagation. In this study, the training was conducted by stochastic gradient descent (SGD) with a cosine annealing strategy [39]. This choice was made based on the empirical consideration that this training algorithm guaranteed faster learning than other popular learning algorithms.

Let us indicate with $\mathbf{z}$ a vector containing all the MLP weights and biases as well as the elements of the attention vector

$$\mathbf{z} = \left(\mathbf{W_1}, \mathbf{W_2}, \mathbf{W_3}, \mathbf{b_1}, \mathbf{b_2}, \mathbf{b_3}, \mathbf{a}^*\right)$$

For each vector $\mathbf{x}$ of the training set $\mathbf{X}$ and for a set of parameters $\mathbf{z}$ univocally identifying an MLP network endowed with attention, an output vector $\mathbf{o}$ can be calculated using Eq. (2). Let $\mathbf{t}$ be a vector containing the ground truth associated with the input vector $\mathbf{x}$. Specifically, the element $t_j = 0$ if $\mathbf{x}$ does not belong to the jth class, and $t_j = 1$ if

$\mathbf{x}$ belongs to the jth class. The training minimises the loss function L given by

$$L(\mathbf{z}) = -\sum_j t_j \log_2\left(\frac{e^{o_j}}{\sum_j e^{o_j}}\right) + \frac{0.01}{\sum_i \left(\left(a_i^* - 0.5\right)^2\right)} \tag{3}$$

The first addend of the loss function in Eq. (3) is known as the cross-entropy loss function. The symbol o_j indicates the number of correct classifications. The relevant PyTorch documentation is available at CrossEntropyLoss. The second addend of the loss function in Eq. (3) is a correction term that promotes the selection of attention mechanisms that emphasise the importance of some features over others. It can be easily verified that the loss function $L(\mathbf{z})$ is positive definite.

The pseudocode of the TMD block is displayed in Algorithm 1.

Algorithm 1 Training Module Design

```
Input: Dataset vectors of type $\mathbf{x}$ and label vectors of type $\mathbf{t}$
Output: $\mathbf{a}$
Initialisation: $a_i \leftarrow 1$ for each element
for $k = 1$ : all the instances of the dataset do
    Generate $\mathbf{x}^*$ vectors by normalisation of the data per Eq. (1)
end for
sample a vector $\mathbf{z}$
repeat
    Apply SGD with cosine annealing strategy [39] on $L(\mathbf{z})$ as in Eq. (3)
        Ensure that the sigmoid function is applied to generate $\mathbf{a}^*$ and
        that the attention mechanism is applied to the input $\mathbf{x}^{\mathbf{a}} \leftarrow \mathbf{x}^* \odot \mathbf{a}^*$
        Calculate each output vector as in Eq. (2)
until MLP is trained to fit the data
return a
```

3.2. Feature ranking by attention

The calculated vector $\mathbf{a}$ encoding the attention mechanism and the normalised dataset are processed by the FRA block. The vector $\mathbf{a}$ is scanned and sorted in descending order, thus producing $\mathbf{a}_{\mathbf{Rank}}$. The elements of all the vectors $\mathbf{x}$ of the dataset $\mathbf{X}$ are permuted to reflect the order of the elements in $\mathbf{a}_{\mathbf{Rank}}$. The newly rearranged dataset indicated with $\mathbf{X}_{\mathbf{Rank}}$ is the output of FRA and is then used as an input of the following block.

3.3. Subset generation

To select the most appropriate features, the SG block processes the dataset $\mathbf{X}_{\mathbf{Rank}}$ in two stages, here indicated as SG1 and SG2. To explain the SG procedure, let us consider $\mathbf{X}_{\mathbf{Rank}}$ as a matrix whose rows are the instances and whose columns are the features. From the newly calculated dataset $\mathbf{X}_{\mathbf{Rank}}$, SG1 generates a number of sub-datasets. Each sub-dataset is obtained by retaining the first columns of $\mathbf{X}_{\mathbf{Rank}}$ and cancelling the others. By using this logic, SG1 uniformly samples M scenarios. The SG1 stage generates $M = 50$ sub-datasets, retaining 2%, 4%, 6% and up to 100% columns. Then, each of these sub-datasets is evaluated using a classifier, and its accuracy value is associated with it. Let us indicate $\mathbf{p}_k$ with $k = 1, 2, \ldots, 50$ as each of these potential feature sets, p_k its number of elements and $Acc\left(\mathbf{p}_k\right)$ its accuracy. The sub-dataset displaying the best performance is retained, and its associated subset of features $\mathbf{p}$ is recorded. It should be noted that in this study, we considered four classifiers (see Section 4 for details).

To ensure that slightly suboptimal yet promising scenarios characterised by a very small number of features are not neglected, a further check is carried out. The feature set $\mathbf{p}$ is compared with all the others, i.e. $\mathbf{p}_k$ $\forall k$, by calculating the score

$$u_k = \frac{Acc\left(\mathbf{p}_k\right) - Acc\left(\mathbf{p}\right)}{p_k - p} \tag{4}$$

Let $U = \left\{u_1, u_2, \ldots\right\}$ be the set containing all the u_k calculated. We want to identify the feature subset $\mathbf{p}_k$ corresponding to a positive u_k, which is as close as possible to 0, and use it as the output $\mathbf{p}$ of SG1. More formally,

$$\begin{aligned} &\mathbf{p} = arg\, min_{\mathbf{p_k}} U \\ &\text{subject to } 0 < u_k < K_p \end{aligned} \tag{5}$$

where K_p is a parameter to be set by the user. It is worthwhile commenting on Eqs. (4) and (5) to understand the logic of the algorithm. The numerator in Eq. (4) is always negative. Thus, if $p_k > p$, it follows that $u_k < 0$. This situation corresponds to $\mathbf{p}_k$ containing more features than $\mathbf{p}$ and displaying worse performance. Clearly, in this case, $\mathbf{p}_k$ is not of interest and can be discarded immediately. Conversely, $u_k > 0$ occurs when $p_k < p$. A large u_k value is likely to mean that $\mathbf{p}$ leads to a much more accurate prediction than $\mathbf{p}_k$ while containing only a marginally larger number of features. Hence, in this case, $\mathbf{p}_k$ can also be discarded. However, $u_k > 0$ and $u_k \approx 0$ are likely to correspond to $\mathbf{p}_k$ containing much fewer features than $\mathbf{p}$, associated with marginally worse accuracy. In this case, $\mathbf{p}_k$ is preferable to $\mathbf{p}$ and is used as the input for SG2. If there are no u_k scores belonging to the interval $\left]0, K_p\right[$, the original $\mathbf{p}$ corresponding to maximum accuracy is retained for SG2.

The SG2 stage processes the feature set $\mathbf{p}$ selected at SG1 to tune the search around this scenario. Let p be the length of $\mathbf{p}$. The SG2 stage explores the feature scenarios in the set $\{p - L, \ldots, p + L\}$, tests them using the classifier under consideration and elects the most promising scenario. In this study, we set $L = 25$. It must be considered that for datasets containing thousands of features, SG2 explores a modest portion of all the possible scenarios. Analogous to SG1, the u_k scores are calculated employing Eq. (4) for all the $\mathbf{p}_k$ sets considered by SG2, and then the procedure in Eq. (5) is also applied in this case.

To further explain the logic of SG, the selection of a feature subset is an optimisation problem, which, thanks to previously performed sorting, is reduced to the selection of only one parameter, i.e. p. To perform this selection, SG1 explores the search space and acts as a global search, while SG2 exploits the neighbourhood of a promising configuration and thus acts as a local search to 'end the game'. Thus, SG can be regarded as a hybrid metaheuristic. Algorithm 2 details the functioning principles of SG.

3.4. Fitness function

Feature selection presents itself as a multi-objective optimisation challenge, with classification accuracy and the count of features within the subset being two pivotal optimisation goals. These two indicators should be simultaneously considered within a unified objective function. Eq. (6) portrays a fitness function that utilises a weighting mechanism to combine the classification error rate and the number of selected features [27,40].

$$Fitness = \alpha \times ErrorRate + (1 - \alpha) \times \frac{\#Selected}{\#All} \tag{6}$$

In this equation, ErrorRate represents the classification error rate, #Selected signifies the size of the feature subset, and #All corresponds to the total number of features available in the dataset. The parameter α is responsible for weighing the classification error rate and the number of selected features. Since classification performance takes precedence, we have chosen $\alpha = 0.99$ for this study. To enhance the comprehensiveness of our study, we also present the results in terms of the fitness illustrated in Eq. (6).

Algorithm 2 Subset Generation

```
Input: Ranked dataset $\mathbf{X}_{\mathbf{Rand}}$, parameters $M$ and $K_p$, and a classifier
Output: Feature subset $\mathbf{p}$
SG1
Sample $M$ sub-datasets whose sizes grow at regular intervals by cancelling the columns of $\mathbf{X}_{\mathbf{Rand}}$ // Each sub-dataset is associated with a feature subset $\mathbf{p}_k$ with $k = 1, \dots, M$
for $k = 1 : M$ do
    Use the classifier to calculate the accuracy of each feature subset $Acc\left(\mathbf{p}_k\right)$
end for
Select the feature subset $\mathbf{p}$ with the higher accuracy
for $k = 2 : M$// We exclude the case $\mathbf{p}_k = \mathbf{p}$ // do
    Calculate $u_k$ by Eq. (4) and save $u_k$ in $U$
end for
Select from $U$ those $u_k \in \left]0, K_p\right[$, find the minimum among those and detect the corresponding $\mathbf{p}_k$
$\mathbf{p} \leftarrow \mathbf{p}_k$ // $\mathbf{p}$ has $p$ features
SG2
for $k = p - L : p + L$ do
    Use the classifier to calculate the accuracy of each feature subset $Acc\left(\mathbf{p}_k\right)$
end for
Select the feature subset $\mathbf{p}$ with the higher accuracy
for $k = p - L : p + L$// We exclude the case $\mathbf{p}_k = \mathbf{p}$ // do
    Calculate $u_k$ by Eq. (4) and save $u_k$ in $U$
end for
Select from $U$ those $u_k \in \left]0, K_p\right[$, find the minimum among those and detect the corresponding $\mathbf{p}_k$
$\mathbf{p} \leftarrow \mathbf{p}_k$
return $\mathbf{p}$
```

4. Experimental setup

The datasets for classification used in these experiments are shown in Table 1, where a link pointing to the corresponding dataset is displayed. We used the same large-scale datasets as in [40]. Notably, these datasets were from various application domains and were of different sizes. We broadly categorised datasets 01–09 as small-scale, 10–19 as large-scale and 20–25 as very large-scale. Most of these datasets were originally part of the UCI Machine Learning Repository https://archive.ics.uci.edu/ and the scikit-feature selection repository [4]. Each dataset was divided into two partitions: one was used as the training set, which was formed by randomly selecting 70% of the example from the original dataset. The other was used as the test set, and it consisted of the remaining examples.

To illustrate the versatility of our method, we selected four different classifiers for evaluation. These classifiers were SVM, k-nearest neighbours(KNN), decision tree (DT) and random forest(RF). All four classifiers served as evaluators, and KNN was used as the reference for comparison with the other algorithms.

We used the SGD optimiser to update the network parameters; the momentum was 0.9, and the weight decay was 0.00001. The cosine annealing strategy was used to gradually reduce the learning rate during training, and the minimum value of the learning rate was set to 0.001. With reference to Section 3.3, we set the three hyperparameters of the algorithm $M = 50$, $L = 25$ and $K_p = 0.1$.

Furthermore, we conducted additional experiments on a real-world dataset related to the coronavirus disease (SARS-CoV-2 or COVID-19) [41]. This dataset comprises 15 features and 233 samples. Additionally, we chose 8 state-of-the-art feature selection methods from [27] as comparison algorithms for this dataset, following the experimental conditions outlined in [27].

Table 1
Table of datasets.

No.	Dataset	Classes	Features	Samples
01	Zoo	7	16	101
02	Segmentation	7	19	210
03	WBCD	2	30	236
04	Ionosphere	2	34	351
05	Chess	2	36	3196
06	Lung1	2	56	32
07	Sonar	2	60	208
08	Movement	15	90	360
09	HillValley	2	100	606
10	MUSK1	2	166	476
11	Semeion	10	256	675
12	Madelon	2	500	2000
13	Isolet5	26	617	1559
14	MultipleFeatures	10	649	2000
15	CNAE-9	9	856	1080
16	PCMAC	2	3289	1943
17	Lung2	5	3312	203
18	RELATHE	2	4322	1427
19	BASEHOCK	2	4862	1993
20	TOX_171	4	5748	171
21	Prostate1	2	5966	102
22	Leukemia	2	7070	72
23	Amazon	50	10 000	1500
24	Prostate2	2	10 509	102
25	11Tumor	11	12 533	174

5. Results and analysis

The current section presents the experimental results of our study. Sections 5.1 and 5.2 present the results of two ablation studies. Section 5.3 showcases the results of the proposed EAR-FS compared to state-of-the-art feature selection methods. Additionally, Section 5.4 demonstrates the results against a high-performance algorithm using the COVID-19 dataset.

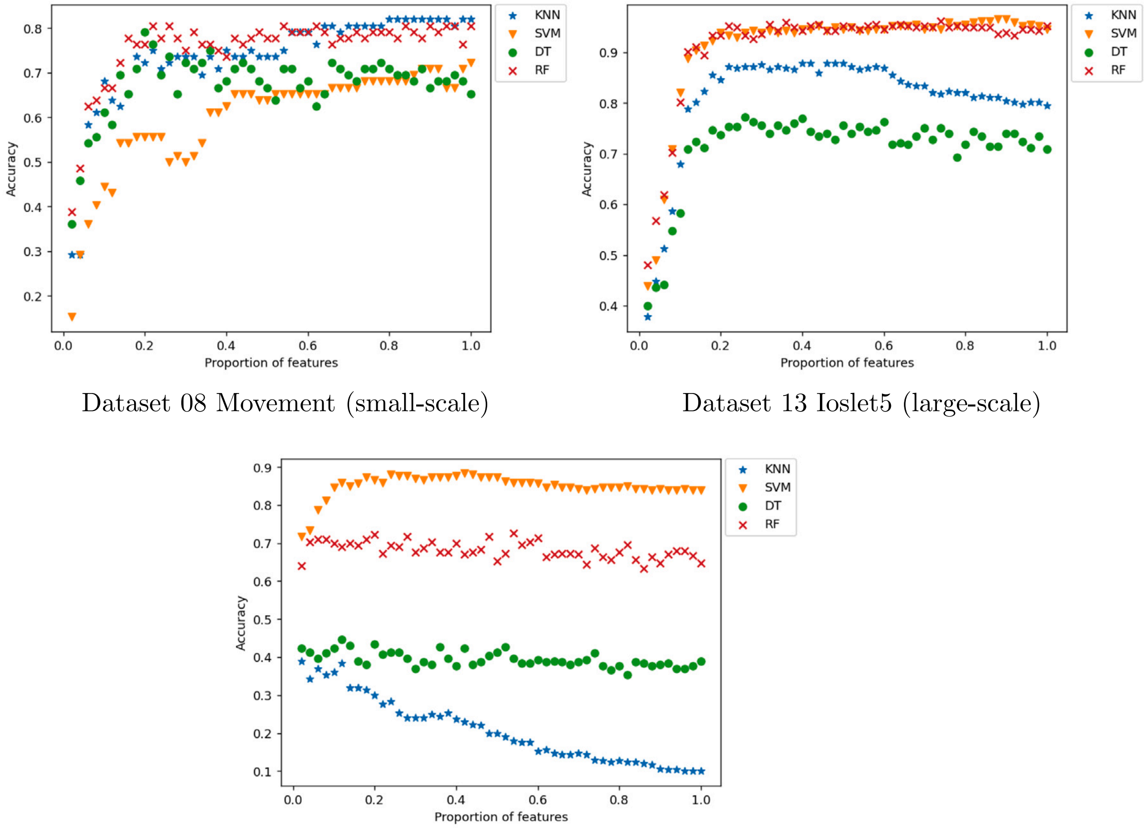
5.1. Ablation study 1: Accuracy of the four classifiers

We performed a preliminary experiment on the datasets listed in Table 1. The generated subsets were evaluated separately using the four classifiers mentioned above, i.e. SVM, KNN, DT and RF. To assess the performance of each classifier, we considered the CA of the classifier after FS was performed using EAR-FS and the size (SZ) of the corresponding feature subset. Due to the data, the results were stochastic; therefore, we ran EAR-FS for each dataset 30 times. Tables 2–4 display the results from each dataset as the mean ± standard deviation of the small-, large- and very large-scale datasets, respectively. The performance in terms of CA on the original datasets is shown in the column ‘Origin’. The original SZ is also reported in this column.

As seen in Tables 2–4, regardless of the classifier, our method could effectively reduce dimensionality and improve CA. However, when EAR-FS employed the KNN classifier, all the datasets appeared to consistently achieve good results. Furthermore, the SZ obtained using KNN was usually the smallest of the four classifiers.

The results from the small-scale datasets in Table 2 show that EAR-FS improved upon the accuracy of the original dataset with some improvements in terms of feature reduction. The results from the large-scale datasets appeared very promising, while the results from the very large-scale datasets were more impressive.

As shown in Table 3, for datasets 12, 17 and 18, EAR-FS with KNN could reduce more than 90% of the features. For datasets 15 and 16, the feature reduction rate was 80% to 90%; for datasets 13, 14 and 19, the reduced features comprised 70% to 80%; and for dataset 10, the feature reduction rate was about 60%. For dataset 11, the reduced features made up 43%. For the other three classifiers, all the datasets also obtained good CA. In particular, for SVM, the number of features



Dataset 08 Movement (small-scale)

Dataset 13 Ioslet5 (large-scale)

Dataset 23 Amazon (very large-scale)

Fig. 2. Accuracy achieved by the four classifiers – k-Nearest Neighbours (KNN), Support Vector Machine (SVM), Decision tree (DT) and random forest (RF) – at the end of SG1 for the M solutions of different sizes (the solution length is expressed as the ratio between the solution and the total number of features).

was reduced by more than 65% for most datasets (10, 12 and 14–19), and for DT, such datasets were 11–17 and 19. Furthermore, many datasets (10, 11, 14, 16 and 19) achieved good feature reduction with RF and the highest CA.

The results from the very large-scale datasets in Table 4 show that the performance of EAR-FS was excellent. For each dataset, EARFS with KNN could reduce the number of features by more than 90% and improve CA, especially for datasets 20, 21, 22, 24 and 25, wherein classification performance improvement was very significant. The SVM classifier performed very well on the large-scale datasets because of its characteristics, and EAR-FS with SVM also reduced the number of features and improved CA. For the DT and RF classifiers, the improvement was also significant.

Some examples of the performance of the subsets generated in SG1 are shown in Fig. 2 for one small-scale, one large-scale and one very large-scale dataset. The plots in Fig. 2 represent accuracy vs solution SZ expressed as a quota of the total number of features. We can see that the CA of the subsets first rose rapidly with the solution SZ and then flattened or decreased. This effectively shows that the valid features were densely concentrated in the front position of the feature set, while the later features were often redundant features or invalid features, thus confirming the efficacy of the attention module in ranking features by importance. Furthermore, these plots emphasised the differences among the classifiers and highlighted the concept that the number of features can be greatly reduced while still retaining or even improving the accuracy of the prediction model.

5.2. Ablation study 2: Architecture of the multilayer perceptron

In this section, we discuss the ablation study on the number of hidden layers in the MLP. Tables 5 and 6 show the performance of EAR-FS when different architectures were used in the MLP network for small-scale and large-scale datasets, respectively. The parameter K indicates the value of the number of hidden layer neurons divided by the number of output neurons.

As seen in Table 5, for most datasets (06, 07, 09, 10–12, 15 and 20), when the number of hidden layer neurons was four times the number of input features, CA was the highest. This may be because, for most datasets, the information contained is not fully learned by the network, so increasing the number of neurons can improve the network's ability to learn, improving CA. For datasets 04 and 21, the classification effect was best when $K = 3$. The reason for this may be that when $K = 3$, the neural network is deep enough to learn all the information of the dataset, and when K increases, the overfitting phenomenon will occur, resulting in a deviation from the expected feature ordering. For datasets 02 and 22, the best feature could be selected regardless of the value of K, and the number of features in the optimal subset was smallest when $K = 2$. For datasets 01, 08, 13, 14 and 16, the CA of the optimal subset reached the highest level when $K = 1$. Finally, the results from datasets 03, 05, 18 and 19 were best when $K = 0.5$.

Overall, this ablation study did not lead to fully conclusive results. We could not detect an MLP network architecture that was more advantageous than the others. The five architectures analysed in this section appeared to display similar performance. Since the $K = 4$

Table 2
Classification Accuracy (CA) and Size (SZ) achieved by external attention-based feature ranker for large-scale feature selection using k-Nearest Neighbours (KNN), Support Vector Machine (SVM), Decision Tree (DT) and Random Forest (RF) classifiers for small-scale datasets.

No.	Dataset		KNN		SVM		DT		RF	
			Mean ± Std	Origin	Mean ± Std	Origin	Mean ± Std	Origin	Mean ± Std	Origin
01	Zoo	CA	97.14 ± 2.33	79.31	97.62 ± 2.38	93.54	1.00 ± 0.00	93.54	99.05 ± 1.90	87.10
		SZ	6.8 ± 1.0	16	7.5 ± 1.2	16	7.8 ± 0.7	16	8.7 ± 1.2	16
02	Segmentation	CA	90.24 ± 0.71	77.59	83.33 ± 0.00	83.33	81.90 ± 1.17	76.19	89.52 ± 1.17	85.71
		SZ	3.1 ± 0.3	19	6.0 ± 0.0	19	4.5 ± 1.3	19	9.4 ± 3.3	19
03	WBCD	CA	97.81 ± 0.44	88.27	97.72 ± 0.43	95.61	94.91 ± 0.35	94.73	96.58 ± 0.26	96.49
		SZ	7.4 ± 2.4	30	10.7 ± 5.4	30	18.8 ± 11.5	30	9.8 ± 7.1	30
04	Ionosphere	CA	91.55 ± 1.41	83.67	89.44 ± 1.14	87.32	92.68 ± 1.76	88.73	94.51 ± 0.42	94.37
		SZ	6.5 ± 1.6	34	8.6 ± 3.7	34	8.1 ± 4.2	34	12.0 ± 2.5	34
05	Chess	CA	95.81 ± 0.23	87.85	95.58 ± 0.07	94.84	97.91 ± 0.24	97.19	98.06 ± 0.40	97.50
		SZ	24.0 ± 4.9	36	22.0 ± 1.8	36	30.2 ± 0.7	36	30.8 ± 1.7	36
06	Lung1	CA	99.56 ± 0.26	85.76	99.43 ± 0.16	71.74	98.57 ± 4.29	85.71	98.57 ± 4.29	57.14
		SZ	4.4 ± 1.2	56	2.0 ± 0.0	56	26.2 ± 18.9	56	10.9 ± 3.9	56
07	Sonar	CA	97.86 ± 1.67	78.61	91.67 ± 2.44	80.95	79.52 ± 3.23	71.43	90.00 ± 1.43	85.71
		SZ	24.4 ± 6.7	60	21.9 ± 5.1	60	14.2 ± 12.7	60	25.0 ± 11.1	60
08	Movement	CA	82.50 ± 0.68	71.58	72.22 ± 0.00	69.44	77.08 ± 2.86	66.67	83.61 ± 0.83	77.78
		SZ	69.6 ± 10.2	90	89.9 ± 0.3	90	24.3 ± 3.8	90	71.8 ± 12.4	90
09	HillValley	CA	52.38 ± 0.93	49.97	66.39 ± 0.00	98.92	58.52 ± 1.11	53.27	59.10 ± 0.86	54.10
		SZ	69.4 ± 7.6	100	65.8 ± 12.5	100	38.0 ± 12.7	100	45.1 ± 12.1	100

Table 3
Classification Accuracy (CA) and Size (SZ) achieved by external attention-based feature ranker using k-Nearest Neighbours (KNN), Support Vector Machine (SVM), Decision Tree (DT) and Random Forest (RF) classifiers for large-scale datasets.

No.	Dataset		KNN		SVM		DT		RF	
			Mean ± Std	Origin	Mean ± Std	Origin	Mean ± Std	Origin	Mean ± Std	Origin
10	MUSK1	CA	85.31 ± 0.73	80.43	84.38 ± 4.66	82.51	86.15 ± 1.68	73.43	86.15 ± 0.67	83.13
		SZ	69.7 ± 12.7	166	58.6 ± 22.2	166	107.1 ± 22.9	166	60.5 ± 16.5	166
11	Semeion	CA	89.63 ± 0.12	83.70	91.85 ± 0.81	85.19	74.22 ± 1.58	63.70	94.22 ± 0.30	91.11
		SZ	145.0 ± 10.3	256	143.4 ± 36.1	256	65.8 ± 26.4	256	211.2 ± 11.3	256
12	Madelon	CA	88.55 ± 0.24	71.25	62.55 ± 0.58	50.25	84.7 ± 0.46	74.75	88.05 ± 0.10	65.00
		SZ	11.0 ± 0.9	500	148.8 ± 8.2	500	19.4 ± 1.4	500	22.6 ± 7.4	500
13	Isolet5	CA	89.61 ± 0.16	79.49	96.41 ± 0.13	94.55	79.75 ± 0.37	72.12	96.34 ± 0.16	94.55
		SZ	145.7 ± 12.4	617	506.6 ± 17.2	617	153.2 ± 45.9	617	449.0 ± 135.9	617
14	MultipleFeatures	CA	98.65 ± 0.20	87.50	98.60 ± 0.12	97.75	96.50 ± 0.42	91.75	99.05 ± 0.10	99.00
		SZ	151.0 ± 24.6	649	73.0 ± 14.3	649	137.8 ± 85.8	649	286.8 ± 116.4	649
15	CNAE-9	CA	93.83 ± 0.22	83.80	97.69 ± 0.00	96.30	89.72 ± 0.35	85.86	94.44 ± 0.29	91.67
		SZ	133.3 ± 2.5	856	161.0 ± 8.5	856	178.8 ± 24.6	856	786.2±36.7	856
16	PCMAC	CA	85.30 ± 1.57	73.58	93.55 ± 0.37	88.16	90.44 ± 0.32	87.65	94.96 ± 0.24	91.77
		SZ	381.5 ± 105.5	3289	577.0 ± 77.0	3289	595.2 ± 260.3	3289	1671.4 ± 419.4	3289
17	Lung2	CA	1.00 ± 0.00	90.08	97.56 ± 0.00	95.08	97.56 ± 1.54	77.05	97.56 ± 0.00	86.89
		SZ	69.3 ± 27.5	3312	55.5 ± 25.9	3312	137.8 ± 163.1	3312	83.7 ± 37.9	3312
18	RELATHE	CA	89.72 ± 0.59	81.82	94.97 ± 0.39	88.34	91.01 ± 3.15	86.48	94.16 ± 0.68	89.04
		SZ	317.7 ± 66.2	4322	1155.5 ± 238.2	4322	1623.9 ± 520.7	4322	1197.8 ± 460.3	4322
19	BASEHOCK	CA	91.73 ± 0.57	81.13	97.24 ± 0.27	97.16	95.14 ± 0.38	92.64	97.39 ± 0.30	96.32
		SZ	1064.1 ± 203.7	4862	1524.2 ± 207.7	4862	1466.6 ± 434.5	4862	2771.0 ± 1144.3	4862

Table 4
Classification Accuracy (CA) and Size (SZ) achieved by external attention-based feature ranker using k-Nearest Neighbours (KNN), Support Vector Machine (SVM), Decision Tree (DT) and Random Forest (RF) classifiers for very large-scale datasets.

No.	Dataset		KNN		SVM		DT		RF	
			Mean ± Std	Origin	Mean ± Std	Origin	Mean ± Std	Origin	Mean ± Std	Origin
20	TOX_171	CA	90.29 ± 1.90	67.90	96.57 ± 1.14	96.15	70.57 ± 5.58	53.85	86.00 ± 2.37	80.77
		SZ	375.9 ± 183.6	5748	748.7 ± 352.8	5748	675.0 ± 418.8	5748	1431.6 ± 560.6	5748
21	Prostate1	CA	89.16 ± 0.00	72.11	95.24 ± 0.00	90.32	94.29 ± 1.90	87.10	94.29 ± 1.9	90.32
		SZ	94.0 ± 0.00	5966	262.80 ± 57.4	5966	2000.2 ± 1278.6	5966	96.8 ± 3.6	5966
22	Leukemia	CA	98.83 ± 0.16	87.31	100.00 ± 0.00	100.0 ± 0.00	97.36 ± 0.34	90.10	97.36 ± 0.34	81.82
		SZ	119.7 ± 7.7	7070	116.0 ± 0.0	7070	500.3 ± 149.6	7070	343.0 ± 162.3	7070
23	Amazon	CA	41.30 ± 0.53	36.27	88.27 ± 0.16	72.00	46.53 ± 0.83	41.55	73.83 ± 0.69	64.67
		SZ	779.7 ± 286.6	10 000	3009.7 ± 879.2	10 000	1357.7 ± 627.8	10 000	3755.9 ± 1514.1	10 000
24	Prostate2	CA	95.24 ± 0.00	71.73	100.0 ± 0.0	96.77	94.76 ± 1.43	90.32	98.57 ± 2.18	93.55
		SZ	227.0 ± 0.0	10 509	252.2 ± 75.6	10 509	1363.9 ± 1075.7	10 509	240.7 ± 14.1	10 509
25	11Tumor	CA	93.43 ± 2.87	76.04	100.00 ± 0.00	90.56	81.71 ± 2.29	71.70	93.14 ± 1.4	77.56
		SZ	383.2 ± 123.1	12 533	426.2 ± 218.4	12 533	3150.0 ± 885.6	12 533	1598.5 ± 481.7	12 533

Table 5
Classification Accuracy (CA) and Size (SZ) of the feature subset for various multilayer perceptron architectures for small-scale datasets (K is the ratio between the number of hidden layer neurons divided by the number of output neurons.).

No.	Dataset		K = 0.5	K = 1	K = 2	K = 3	K = 4
			Mean ± Std	Mean ± Std	Mean ± Std	Mean ± Std	Mean ± Std
01	Zoo	CA	98.16 ± 2.13	**98.36** ± 1.97	97.87 ± 2.25	98.06 ± 2.10	97.14 ± 2.33
		SZ	7.2 ± 1.2	7.2 ± 0.9	7.0 ± 1.1	6.8 ± 1.3	**6.8** ± 1.0
02	Segmentation	CA	90.22 ± 0.68	89.77 ± 0.67	**91.56** ± 0.55	90.38 ± 0.74	90.24 ± 0.71
		SZ	3.6 ± 0.3	3.9 ± 0.4	3.4 ± 0.4	**2.7** ± 0.2	3.1 ± 0.3
03	WBCD	CA	**98.44** ± 0.67	98.15 ± 0.30	96.93 ± 0.22	97.41 ± 0.60	97.81 ± 0.44
		SZ	8.1 ± 2.3	7.7 ± 2.3	7.7 ± 2.3	**6.8** ± 2.1	7.4 ± 2.4
04	Ionosphere	CA	90.62 ± 1.38	91.40 ± 1.28	91.20 ± 1.41	**91.59** ± 1.11	91.55 ± 1.41
		SZ	7.3 ± 1.9	**6.3** ± 1.4	6.4 ± 1.6	7.0 ± 1.7	6.5 ± 1.6
05	Chess	CA	**96.50** ± 0.17	95.53 ± 0.32	95.11 ± 0.48	95.34 ± 0.33	95.81 ± 0.23
		SZ	**20.1** ± 5.0	28.2 ± 6.7	24.4 ± 5.0	26.5 ± 4.1	24.0 ± 4.9
06	Lung1	CA	99.16 ± 0.13	99.47 ± 0.19	99.51 ± 0.24	99.26 ± 0.32	**99.56** ± 0.26
		SZ	5.1 ± 1.1	4.2 ± 1.1	4.0 ± 1.2	**3.9** ± 1.3	4.4 ± 1.2
07	Sonar	CA	97.22 ± 0.97	97.14 ± 1.77	96.97 ± 2.68	97.44 ± 1.86	**97.86** ± 1.67
		SZ	24.3 ± 6.6	22.5 ± 1.9	**21.6** ± 4.7	25.1 ± 8.8	24.4 ± 6.7
08	Movement	CA	81.97 ± 0.98	**82.73** ± 0.55	82.34 ± 1.12	81.73 ± 1.01	82.50 ± 0.68
		SZ	72.1 ± 11.2	71.4 ± 9.7	68.1 ± 8.6	**67.5** ± 11.1	69.6 ± 10.2
09	HillValley	CA	50.13 ± 0.91	49.95 ± 1.12	49.53 ± 0.48	52.19 ± 0.82	**52.38** ± 0.93
		SZ	60.5 ± 7.7	59.4 ± 5.6	**57.7** ± 7.6	72.8 ± 4.9	69.4 ± 7.6

Table 6
Classification Accuracy (CA) and Size (SZ) of the feature subset for various multilayer perceptron architectures for large-scale datasets (K is the ratio between the number of hidden layer neurons divided by the number of output neurons.).

No.	Dataset		K = 0.5	K = 1	K = 2	K = 3	K = 4
			Mean ± Std	Mean ± Std	Mean ± Std	Mean ± Std	Mean ± Std
10	MUSK1	CA	85.21 ± 1.12	84.84 ± 1.12	84.90 ± 1.16	84.97 ± 1.19	**85.31** ± 0.73
		SZ	**55.0** ± 16.0	76.5 ± 23.9	82.1 ± 24.5	87.3 ± 24.9	69.7 ± 12.7
11	Semeion	CA	85.57 ± 0.85	85.53 ± 1.24	84.61 ± 2.64	87.01 ± 0.84	**89.63** ± 0.12
		SZ	142.7 ± 11.5	123.8 ± 10.1	**99.4** ± 16.9	112.6 ± 28.9	145.0 ± 10.3
12	Madelon	CA	85.90 ± 1.79	86.88 ± 1.09	86.63 ± 0.93	87.38 ± 1.06	**88.55** ± 0.24
		SZ	11.2 ± 2.9	10.9 ± 2.1	**9.9** ± 1.5	11.5 ± 2.1	11.0 ± 0.9
13	Isolet5	CA	89.33 ± 0.43	**89.66** ± 0.57	89.23 ± 0.36	89.17 ± 0.35	89.61 ± 0.16
		SZ	202.5 ± 42.0	188.0 ± 16.2	152.5 ± 23.0	168.1 ± 17.4	**145.7** ± 12.4
14	MultipleFeatures	CA	98.50 ± 0.32	**98.70** ± 0.25	98.53 ± 0.26	98.68 ± 0.20	98.65 ± 0.20
		SZ	155.1 ± 38.1	165.5 ± 44.0	167.6 ± 30.8	**147.3** ± 41.4	151.0 ± 24.6
15	CNAE-9	CA	91.06 ± 0.55	91.48 ± 0.52	91.85 ± 0.59	92.22 ± 0.61	**93.83** ± 0.22
		SZ	145.6 ± 12.7	**124.7** ± 10.4	128.6 ± 4.9	128.3 ± 6.9	133.3 ± 2.5
16	PCMAC	CA	86.56 ± 1.71	**86.89** ± 1.73	86.39 ± 1.86	86.04 ± 1.93	85.30 ± 1.57
		SZ	238.2 ± 69.8	**191.3** ± 53.9	242.7 ± 90.6	255.3 ± 89.5	381.5 ± 105.5
17	Lung2	CA	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
		SZ	165.3 ± 33.9	127.1 ± 47.5	125.5 ± 45.1	112.3 ± 48.0	**69.3** ± 27.5
18	REALTHE	CA	**90.31** ± 0.72	90.16 ± 0.71	90.15 ± 0.71	90.12 ± 0.73	89.72 ± 0.59
		SZ	344.6 ± 62.9	341.5 ± 70.0	355.3 ± 90.5	359.3 ± 102.4	**317.7** ± 66.2
19	BASEHOCK	CA	**92.93** ± 0.52	92.42 ± 0.92	92.24 ± 0.90	92.01 ± 0.89	91.73 ± 0.57
		SZ	**706.2** ± 207.4	979.5 ± 207.4	942.0 ± 200.5	968.8 ± 240.6	1064.1 ± 203.7
20	TOX_171	CA	90.25 ± 1.94	88.46 ± 1.83	89.84 ± 2.23	90.18 ± 1.76	**90.29** ± 1.90
		SZ	480.4 ± 195.4	425.7 ± 135.8	530.2 ± 215.4	397.4 ± 148.9	**375.9** ± 183.6
21	Prostate1	CA	89.01 ± 0.32	88.97 ± 0.46	88.46 ± 1.67	**89.16** ± 0.00	**89.16** ± 0.00
		SZ	176.4 ± 47.8	103.0 ± 50.3	275.4 ± 93.4	**94.0** ± 0.0	**94.0** ± 0.0
22	Leukemia	CA	98.13 ± 0.32	98.42 ± 0.61	**98.95** ± 0.31	95.14 ± 0.17	98.83 ± 0.16
		SZ	**119.7** ± 5.8	134.9 ± 12.2	144.3 ± 8.8	135.4 ± 8.1	119.7 ± 7.7

architecture seemed to be the most promising and presented some relative advantages, it was chosen for the proposed EAR-FS. It can be noted that $K = 4$ corresponds to $4 \times D$ neurons in each hidden layer (see Section 3.1).

5.3. Comparison with state-of-the-art feature selection methods

In this section, we compare the performance of EAR-FS against that of seven FS methods, including the chaotic binary gaining sharing knowledge-based optimisation algorithm (CBi-GSK1) [42], sticky binary PSO with dynamic strategy (SBP-D) [43], quantum-based whale optimisation algorithm (QWOA) [44], bare-bones PSO with mutual information (MIBBPSO) [45], tree-based feature selection method (FWDT) [46], variable-size cooperative coevolutionary PSO (VS-CCPSO) [47] and feature selection approach based on the correlation-guided updating strategy and a surrogate technique (CUS-SPSO) [40]. To perform a fair comparison, we reproduced the experimental setup used in [40] and ran EAR-FS in these conditions.

Tables 7, 10 and 13 display the mean values and standard deviations of the CA of the eight FS algorithms under investigation across the 25 datasets. To strengthen the statistical significance of the results, we used a T-test with a 95% confidence level. In these tables, (+)/(−) indicates that the result from the EAR-FS algorithm was significantly better or worse than that of the comparison algorithm, while (=) indicates that there was no significant difference between the EAR-FS algorithm and the comparison algorithm. The best mean value of each dataset is highlighted in bold.

Table 7
Classification accuracy of the proposed external attention-based feature ranker for large-scale feature selection compared with seven state-of-the-art feature selection algorithms for nine small-scale datasets.

No.	Dataset	CBi-GSK1	SBP-D	QWOA	MIBBPSO	FWDT	VS-CCPSO	CUS-SPSO	EAR-FS
01	Zoo	86.37 ± 2.13(+)	87.27 ± 2.01(+)	87.21 ± 2.19(+)	87.63 ± 2.31(+)	85.43 ± 2.62(+)	88.24 ± 1.87(+)	87.63 ± 2.31(+)	**97.14** ± 2.33
02	Segmentation	84.24 ± 2.51(+)	84.66 ± 2.37(+)	83.65 ± 2.13(+)	84.79 ± 2.56(+)	81.69 ± 2.21(+)	84.33 ± 2.31(+)	85.93 ± 2.23(+)	**90.24** ± 0.71
03	WBCD	94.18 ± 0.94(+)	94.03 ± 0.78(+)	94.53 ± 1.13(+)	93.63 ± 1.32(+)	91.42 ± 0.83(+)	94.06 ± 1.02(+)	94.07 ± 0.64(+)	**97.81** ± 0.44
04	Ionosphere	88.43 ± 2.61(+)	85.19 ± 2.84(+)	88.81 ± 1.92(+)	88.69 ± 2.98(+)	84.27 ± 2.19(+)	88.58 ± 2.30(+)	89.78 ± 2.12(+)	**91.55** ± 1.41
05	Chess	93.70 ± 1.13(+)	94.45 ± 1.24(+)	93.58 ± 1.23(+)	94.21 ± 1.24(+)	90.46 ± 0.86(+)	93.86 ± 1.25(+)	94.93 ± 0.96(+)	**95.81** ± 0.23
06	Lung1	92.79 ± 8.62(+)	83.00 ± 11.49(+)	90.44 ± 5.23(+)	89.03 ± 8.24(+)	88.44 ± 6.19(+)	88.29 ± 7.26(+)	91.67 ± 8.34(+)	**99.56** ± 0.26
07	Sonar	81.85 ± 2.93(+)	80.89 ± 3.97(+)	81.56 ± 2.49(+)	81.18 ± 3.41(+)	74.56 ± 3.71(+)	81.98 ± 3.25(+)	82.29 ± 3.00(+)	**97.86** ± 1.67
08	Movement	72.67 ± 1.44(+)	74.86 ± 1.68(+)	74.70 ± 2.33(+)	74.11 ± 2.87(+)	72.68 ± 1.29(+)	73.86 ± 2.03(+)	75.06 ± 2.18(+)	**82.50** ± 0.68
09	HillValley	**56.38** ± 2.09(−)	54.86 ± 2.16(−)	55.14 ± 2.27(−)	55.87 ± 2.17(−)	53.37 ± 1.94(−)	55.79 ± 2.33(−)	56.07 ± 1.93(−)	52.38 ± 0.93

Table 8
Feature subset size of the proposed external attention-based feature ranker for large-scale feature selection compared with seven state-of-the-art feature selection algorithms for nine small-scale datasets.

No.	Dataset	Full size	CBi-GSK1	SBP-D	QWOA	MIBBPSO	FWDT	VS-CCPSO	CUS-SPSO	EAR-FS
01	Zoo	16	**4.1**(−)	5.6(−)	4.6(−)	5.2(−)	4.8(−)	4.9(−)	4.7(−)	6.8
02	Segmentation	19	6.3(+)	7.3(+)	6.6(+)	6.9(+)	6.9(+)	7.0(+)	6.7(+)	**3.1**
03	WBCD	30	4.2(−)	5.3(−)	4.1(−)	4.6(−)	5.0(−)	6.3(−)	**2.0**(−)	7.4
04	Ionosphere	34	7.1(+)	8.0(+)	8.2(+)	7.8(+)	6.9(+)	9.2(+)	**3.5**(−)	6.5
05	Chess	36	10.8(−)	14.7(−)	13.6(−)	13.0(−)	15.9(−)	16.2(−)	**11.1**(−)	24.0
06	Lung1	56	12.8(+)	23.3(+)	13.4(+)	19.4(+)	19.5(+)	21.1(+)	12.2(+)	**4.4**
07	Sonar	60	20.6(−)	25.0(+)	21.8(−)	25.5(+)	22.4(−)	26.1(+)	**20.5**(−)	24.4
08	Movement	90	32.9(−)	37.8(−)	31.4(−)	**30.0**(−)	33.2(−)	34.2(−)	32.9(−)	69.6
09	HillValley	100	39.4(−)	41.7(−)	44.5(−)	40.9(−)	42.1(−)	42.1(−)	**37.7**(−)	69.4

The numerical results in Table 7 clearly show the higher performance of EAR-FS over the other methods in terms of accuracy and for small datasets. It can be observed that EAR-FS significantly outperformed its seven competitors for datasets 01–08. However, for dataset 09 only, the proposed EAR-FS was outperformed.

Regarding large-scale datasets, it can be seen from Table 10 that the CA of EAR-FS was better than all of CBi-GSK1, SBP-D, QWOA, MIBBPSO, FWDT, VS-CCPSO and CUS-SPSO for nine datasets (11–19), with a statistically significant difference. For dataset 10, there was no significant difference between EAR-FS and SBP-D, QWOA, MIBBPSO and VS-CCPSO, while EAR-FS was significantly better than FWDT and significantly worse than CBi-GSK1 and CUS-SPSO.

Furthermore, Table 9 displays the fitness scores, see Eq. (6), of the proposed EAR-FS in comparison to seven other algorithms across the Fitness function. Notably, EAR-FS demonstrated exceptional proficiency on 8 out of the 9 small datasets.

Table 10 illustrates that although EAR-FS could achieve good results for relatively small-scale datasets, its superiority was not evident when compared with that of the other algorithms under consideration. However, when applied to the large-scale datasets, EAR-FS showed outstanding FS ability.

The results from the very large-scale datasets in Table 13 reveal that compared with the other algorithms, EAR-FS was superior for datasets 20, 21, 22, 24 and 25, and it was slightly better than the other algorithms for dataset 23. Moreover, as shown in Table 13, EAR-FS significantly outperformed all the other algorithms in all the very large-scale datasets.

Overall, the performance of EAR-FS was better than the other methods, as it was associated with the best accuracy for 23 of the 25 datasets considered in this study. These results allowed us to conclude that the proposed EAR-FS is capable of consistently producing FS subsets that are reliable in terms of associated accuracy.

The results of the solution SZ obtained by the EAR-FS algorithm and comparison method for the 25 datasets are shown in Tables 8, 11 and 14.

The results from the small datasets in Table 8 demonstrate that EAR-FS did not perform as well as CUS-SPSO in terms of the SZ of the feature subset. The best results were achieved by EAR-FS only for two datasets out of the nine considered in Table 8. This result was due to the logic of SG. The latter, while depending on constant hyperparameters (M, L and K_p), was designed to explore small portions of a large search space, returning a compact FS subset. This logic does not scale down well in the case of small datasets, as SG1 will consider a large number of configurations if the total number of features is of the same order of magnitude as M. Conversely, for datasets containing thousands of features, the solutions associated with $M = 50$ are a dramatic downsample of the original dataset. These solutions may contain promising scenarios that are then exploited by SG2.

This intuition was confirmed by the results from the large-scale datasets. The experimental results in Table 11 demonstrate that EAR-FS achieved the best results for seven (12, 13 and 15–19) of the 10 datasets under consideration. CBi-GSK1 achieved the smallest subsets for datasets 10 and 14. For dataset 11, FWDT generated the least number of features in the optimal subset. Moreover, as the number of features in a dataset increased, EAR-FS exhibited an increased ability to reduce the invalid or redundant features. Especially when the number of features reached more than 3000, the number of features contained in the optimal subset generated by EAR-FS was much smaller than that of other methods.

Moreover, when amalgamated with classification accuracy and the ratio of selected features, the proposed EAR-FS attains the most favourable fitness values for datasets 11–19 within the category of large datasets, as evidenced in Table 12. Notably, datasets 14 and 17 stand out, with fitness values reaching 0.01 and 0.00, respectively. This achievement can be attributed to the EAR-FS method's proficiency in both classification accuracy and the selection of an optimal number of features.

The results from the very large-scale datasets in Table 14 show that EAR-FS achieved outstanding results in terms of the SZ of the feature subsets for the very large-scale datasets. As shown in Table 14, the proposed EAR-FS detected the most compact FS subset for all the very large-scale datasets under consideration. It is worth noting that the EAR-FS solutions were one order of magnitude shorter than those produced by all the other FS algorithms.

The experimental results from all the datasets showed that EAR-FS achieved the best performance in terms of solution SZ in 15 cases out of 25. While its performance on small datasets was not competitive against that of the state-of-the-art FS algorithms, the performance of

Table 9
Fitness value of the proposed external attention-based feature ranker for large-scale feature selection compared with seven state-of-the-art feature selection algorithms for nine small-scale datasets.

No.	Dataset	CBi-GSK1	SBP-D	QWOA	MIBBPSO	FWDT	VS-CCPSO	CUS-SPSO	EAR-FS
01	Zoo	0.13 ± 0.01(+)	0.13 ± 0.00(+)	0.13 ± 0.00(+)	0.13 ± 0.00(+)	0.15 ± 0.01(+)	0.12 ± 0.00(+)	0.13 ± 0.00(+)	**0.03** ± 0.00
02	Segmentation	0.16 ± 0.00(+)	0.16 ± 0.01(+)	0.17 ± 0.01(+)	0.15 ± 0.00(+)	0.18 ± 0.01(+)	0.16 ± 0.00(+)	0.14 ± 0.00(+)	**0.10** ± 0.00
03	WBCD	0.06 ± 0.00(+)	0.06 ± 0.00(+)	0.05 ± 0.00(+)	0.06 ± 0.00(+)	0.09 ± 0.00(+)	0.06 ± 0.00(+)	0.06 ± 0.00(+)	**0.02** ± 0.00
04	Ionosphere	0.12 ± 0.00(+)	0.15 ± 0.01(+)	0.11 ± 0.00(+)	0.11 ± 0.00(+)	0.16 ± 0.02(+)	0.12 ± 0.00(+)	0.10 ± 0.00(+)	**0.08** ± 0.00
05	Chess	0.06 ± 0.00(+)	0.06 ± 0.00(+)	0.07 ± 0.00(+)	0.06 ± 0.00(+)	0.10 ± 0.00(+)	0.07 ± 0.00(+)	0.05 ± 0.00(+)	**0.04** ± 0.00
06	Lung1	0.07 ± 0.00(+)	0.17 ± 0.02(+)	0.10 ± 0.00(+)	0.11 ± 0.01(+)	0.12 ± 0.00(+)	0.12 ± 0.00(+)	0.08 ± 0.00(+)	**0.00** ± 0.00
07	Sonar	0.18 ± 0.01(+)	0.19 ± 0.01(+)	0.19 ± 0.02(+)	0.19 ± 0.00(+)	0.26 ± 0.02(+)	0.18 ± 0.00(+)	0.18 ± 0.01(+)	**0.03** ± 0.00
08	Movement	0.27 ± 0.00(+)	0.25 ± 0.02(+)	0.25 ± 0.01(+)	0.26 ± 0.02(+)	0.27 ± 0.03(+)	0.26 ± 0.01(+)	0.25 ± 0.01(+)	**0.18** ± 0.01
09	HillValley	**0.44** ± 0.06(−)	0.45 ± 0.04(−)	0.45 ± 0.04(−)	0.44 ± 0.06(−)	0.47 ± 0.05(=)	0.44 ± 0.06(−)	0.44 ± 0.03(−)	0.47 ± 0.03

Table 10
Classification accuracy of the proposed external attention-based feature ranker for large-scale feature selection compared with seven state-of-the-art feature selection algorithms for 10 large-scale datasets.

No.	Dataset	CBi-GSK1	SBP-D	QWOA	MIBBPSO	FWDT	VS-CCPSO	CUS-SPSO	EAR-FS
10	MUSK1	86.12 ± 1.66(−)	85.81 ± 1.49(=)	85.27 ± 1.79(=)	85.30 ± 2.31(=)	83.24 ± 1.46(+)	85.30 ± 2.58(=)	**86.23** ± 2.12(−)	85.31 ± 0.73
11	Semeion	87.84 ± 0.81(+)	88.18 ± 0.83(+)	87.90 ± 1.36(+)	88.02 ± 1.77(+)	84.62 ± 1.29(+)	87.57 ± 1.31(+)	88.16 ± 0.89(+)	**89.63** ± 0.12
12	Madelon	77.86 ± 1.20(+)	76.73 ± 1.42(+)	76.97 ± 1.72(+)	80.02 ± 1.83(+)	76.70 ± 1.58(+)	79.86 ± 1.96(+)	79.19 ± 1.28(+)	**88.55** ± 0.24
13	Isolet5	88.39 ± 1.89(+)	88.04 ± 1.00(+)	89.17 ± 1.66(=)	87.15 ± 1.79(+)	87.83 ± 1.78(+)	87.32 ± 1.03(+)	88.68 ± 0.91(+)	**89.61** ± 0.16
14	MultipleFeatures	95.81 ± 1.44(+)	95.97 ± 0.37(+)	94.18 ± 1.12(+)	95.09 ± 0.61(+)	91.83 ± 1.20(+)	94.92 ± 0.64(+)	96.04 ± 0.54(+)	**98.65** ± 0.20
15	CANE-9	85.29 ± 1.42(+)	82.34 ± 1.97(+)	87.16 ± 1.56(+)	86.27 ± 2.17(+)	84.26 ± 2.05(+)	84.79 ± 1.35(+)	87.42 ± 1.65(+)	**93.83** ± 0.22
16	PCMAC	77.84 ± 1.45(+)	76.97 ± 1.75(+)	77.61 ± 1.88(+)	77.64 ± 1.83(+)	77.43 ± 1.59(+)	78.89 ± 1.74(+)	78.01 ± 1.83(+)	**85.30** ± 1.57
17	Lung2	95.11 ± 2.10(+)	95.50 ± 1.77(+)	95.12 ± 0.83(+)	95.41 ± 1.50(+)	91.26 ± 1.74(+)	95.93 ± 1.41(+)	96.19 ± 1.29(+)	**1.00** ± 0.00
18	RELATHE	81.56 ± 1.56(+)	81.60 ± 2.36(+)	82.34 ± 1.15(+)	81.38 ± 2.15(+)	78.87 ± 1.92(+)	81.84 ± 1.97(+)	82.71 ± 1.65(+)	**89.72** ± 0.59
19	BASEHOCK	85.78 ± 1.86(+)	85.46 ± 2.04(+)	85.10 ± 1.78(+)	85.73 ± 1.93(+)	87.14 ± 1.24(+)	85.75 ± 1.44(+)	86.89 ± 1.89(+)	**91.73** ± 0.57

Table 11
Feature subset size of the proposed external attention-based feature ranker for large-scale feature selection compared with seven state-of-the-art feature selection algorithms for 10 large-scale datasets.

No.	Dataset	Full size	CBi-GSK1	SBP-D	QWOA	MIBBPSO	FWDT	VS-CCPSO	CUS-SPSO	EAR-FS
10	MUSK1	100	**50.2**(−)	76.6(+)	68.7(=)	75.3(+)	62.1(−)	69.0(=)	66.7(+)	69.7
11	Semeion	256	125.4(−)	123.8(−)	115.6(−)	120.3(−)	**110.1**(−)	129.5(−)	113.8(−)	145.0
12	Madelon	500	123.5(+)	237.1(+)	100.1(+)	226.3(+)	120.4(+)	206.4(+)	196.7(+)	**11.0**
13	Isolet5	617	176.1(+)	295.2(+)	200.6(+)	291.2(+)	184.2(+)	293.2(+)	270.5(+)	**145.7**
14	MultipleFeatures	649	**141.4**(−)	293.1(+)	155.1(=)	254.8(+)	155.7(=)	269.9(+)	244.7(+)	151.0
15	CNAE-9	856	368.3(+)	432.3(+)	379.6(+)	505.6(+)	451.2(+)	507.2(+)	432.3(+)	**133.3**
16	PCMAC	3289	770.6(+)	1638.6(+)	840.3(+)	1684.3(+)	923.6(+)	1583.2(+)	1529.6(+)	**381.5**
17	Lung2	3312	567.4(+)	1628.9(+)	580.6(+)	1592.1(+)	495.3(+)	1475.0(+)	1466.7(+)	**69.3**
18	RELATHE	4322	1110.7(+)	2147.1(+)	1889.4(+)	2099.7(+)	1580.1(+)	2102.5(+)	2059.5(+)	**317.4**
19	BASEHOCK	4862	1345.7(+)	2423.8(+)	1919.5(+)	2458.3(+)	1730.2(+)	2314.3(+)	2304.3(+)	**1064.1**

EAR-FS on datasets containing hundreds or thousands of features sets a new standard, since our proposed logic consistently achieved better accuracy than the other methods with solutions whose size was only 10% of the others. For example, for the largest dataset, i.e. 11Tumor, EAR-FS achieved an average accuracy of 93.43%, while CUS-SPSO obtained only 79.66%. This result was associated with an average solution length of 383.2 for EAR-FS compared with 5095.1 for CUS-SPSO.

Table 15 illustrates the fitness values derived from the proposed EAR-FS and seven comparative algorithms across very large datasets. It is evident that the EAR-FS algorithm consistently outperforms the competition in all very large-scale datasets. This superiority can be attributed to EAR-FS's enhanced capability in eliminating features, particularly in the context of extensive feature selection challenges. The integration of a neural network component within EAR-FS contributes to its heightened capacity for handling voluminous data and preserving pertinent information.

Furthermore, we have generated radar charts based on the data presented in the six aforementioned tables, depicting both classification accuracy and subset size. These charts are displayed in Fig. 3 for classification accuracy and Fig. 4 for subset size. These visual representations offer a more intuitive and illustrative view of the impact and effectiveness of EAR-FS.

5.4. Results on the dataset COVID-19

This section showcases the outcomes derived from the COVID-19 dataset. Table 16 provides insight into the classification accuracy, subset size, and fitness values of both the proposed algorithm and the eight comparison algorithms. Notably, the fitness values have been magnified by a factor of 100 for ease of interpretation, denoted by the use of percentages. Specifically, among the eight comparison algorithms, EAR-FS achieved the second-highest classification accuracy. Additionally, EAR-FS exhibited a smaller average subset size when compared to BCSA_S, which boasted the highest classification accuracy. Consequently, EAR-FS emerged as the algorithm with the best fitness value.

Moreover, in the classification of test set samples, EAR-FS demonstrated a 100% classification accuracy in 4 out of 20 independent experiments. In the remaining 16 experiments, the classification accuracy achieved a rate of 97.78%. The count of selected features ranged from 5 to 12, with a more concentrated distribution around 6 and 7. These findings, coupled with the insights provided in Table 16, sufficiently illustrate the competitiveness of the proposed EAR-FS algorithm outlined in this paper.

Table 12
Fitness value of the proposed external attention-based feature ranker for large-scale feature selection compared with seven state-of-the-art feature selection algorithms for 10 large-scale datasets.

No.	Dataset	CBi-GSK1	SBP-D	QWOA	MIBBPSO	FWDT	VS-CCPSO	CUS-SPSO	EAR-FS
10	MUSK1	0.14 ± 0.01(−)	0.15 ± 0.00(=)	0.14 ± 0.00(−)	0.15 ± 0.00(=)	0.17 ± 0.01(+)	0.15 ± 0.00(=)	**0.14** ± 0.01(−)	0.15 ± 0.00
11	Semeion	0.13 ± 0.00(+)	0.12 ± 0.00(+)	0.12 ± 0.00(+)	0.12 ± 0.00(+)	0.16 ± 0.01(+)	0.13 ± 0.00(+)	0.12 ± 0.00(+)	**0.10** ± 0.00
12	Madelon	0.22 ± 0.03(+)	0.23 ± 0.03(+)	0.23 ± 0.02(+)	0.20 ± 0.01(+)	0.24 ± 0.03(+)	0.20 ± 0.01(+)	0.21 ± 0.02(+)	**0.11** ± 0.00
13	Isolet5	0.12 ± 0.00(+)	0.13 ± 0.01(+)	0.11 ± 0.00(+)	0.13 ± 0.00(+)	0.12 ± .0.00(+)	0.13 ± 0.00(+)	0.12 ± 0.00(+)	**0.10** ± 0.00
14	MultipleFeatures	0.04 ± 0.00(+)	0.05 ± 0.00(+)	0.06 ± 0.00(+)	0.05 ± 0.00(+)	0.08 ± 0.00(+)	0.06 ± 0.00(+)	0.04 ± 0.00(+)	**0.01** ± 0.00
15	CANE-9	0.15 ± 0.01(+)	0.18 ± 0.01(+)	0.13 ± 0.00(+)	0.14 ± 0.00(+)	0.16 ± 0.01(+)	0.16 ± 0.00(+)	0.13 ± 0.00(+)	**0.06** ± 0.00
16	PCMAC	0.22 ± 0.03(+)	0.23 ± 0.02(+)	0.22 ± 0.02(+)	0.23 ± .0.02(+)	0.23 ± 0.02(+)	0.21 ± 0.01(+)	0.22 ± 0.01(+)	**0.14** ± 0.00
17	Lung2	0.05 ± 0.00(+)	0.05 ± 0.00(+)	0.05 ± 0.00(+)	0.05 ± 0.00(+)	0.09 ± 0.00(+)	0.04 ± 0.00(+)	0.04 ± 0.00(+)	**0.00** ± 0.00
18	RELATHE	0.19 ± 0.00(+)	0.19 ± 0.00(+)	0.18 ± 0.01(+)	0.19 ± 0.02(+)	0.21 ± 0.01(+)	0.18 ± 0.00(+)	0.17 ± 0.01(+)	**0.10** ± 0.00
19	BASEHOCK	0.14 ± 0.00(+)	0.15 ± 0.01(+)	0.15 ± 0.01(+)	0.14 ± 0.00(+)	0.13 ± 0.00(+)	0.14 ± 0.01(+)	0.13 ± 0.00(+)	**0.08** ± 0.00

Table 13
Classification accuracy of the proposed external attention-based feature ranker for large-scale feature selection compared with seven state-of-the-art feature selection algorithms for six very large-scale datasets.

No.	Dataset	CBi-GSK1	SBP-D	QWOA	MIBBPSO	FWDT	VS-CCPSO	CUS-SPSO	EAR-FS
20	TOX_171	71.26 ± 3.91(+)	71.38 ± 4.57(+)	70.65 ± 4.95(+)	70.96 ± 4.73(+)	68.31 ± 3.86(+)	71.02 ± 4.92(+)	71.95 ± 4.53(+)	**90.29** ± 1.90
21	Prostate1	75.31 ± 4.80(+)	75.38 ± 4.90(+)	75.26 ± 5.21(+)	75.39 ± 5.17(+)	73.11 ± 5.16(+)	75.81 ± 4.08(+)	76.45 ± 5.16(+)	**89.16** ± 0.00
22	Leukemia	90.15 ± 2.13(+)	90.72 ± 2.21(+)	90.13 ± 1.84(+)	91.26 ± 1.39(+)	86.46 ± 1.47(+)	89.10 ± 1.42(+)	91.16 ± 1.39(+)	**98.83** ± 0.16
23	Amazon	36.77 ± 1.22(+)	38.28 ± 1.38(+)	33.60 ± 1.13(+)	39.28 ± 1.30(+)	37.29 ± 1.36(+)	39.11 ± 1.54(+)	41.04 ± 1.28(+)	**41.30** ± 0.53
24	Prostate2	77.38 ± 4.56(+)	74.19 ± 5.01(+)	76.83 ± 4.17(+)	76.07 ± 4.11(+)	76.43 ± 4.51(+)	76.03 ± 4.36(+)	77.10 ± 3.32(+)	**95.24** ± 0.00
25	11Tumor	78.69 ± 2.43(+)	77.87 ± 2.87(+)	79.62 ± 3.71(+)	79.35 ± 3.79(+)	74.29 ± 2.92(+)	79.42 ± 3.46(+)	79.66 ± 3.43(+)	**93.43** ± 2.87

Table 14
Feature subset size of the proposed external attention-based feature ranker for large-scale feature selection compared with seven state-of-the-art feature selection algorithms for six very large-scale datasets.

No.	Dataset	Full size	CBi-GSK1	SBP-D	QWOA	MIBBPSO	FWDT	VS-CCPSO	CUS-SPSO	EAR-FS
20	TOX_171	5748	1382.4(+)	2859.4(+)	2775.9(+)	2811.3(+)	2811.2(+)	2741.3(+)	2665.1(+)	**375.9**
21	Prostate1	5966	2019.4(+)	2946.7(+)	2650.6(+)	2783.1(+)	2813.3(+)	2872.1(+)	2703.2(+)	**89.16**
22	Leukemia	7070	2843.5(+)	3509.7(+)	2883.3(+)	3115.2(+)	3256.4(+)	2992.8(+)	3202.6(+)	**119.7**
23	Amazon	10 000	3083.6(+)	4985.3(+)	4214.5(+)	4603.2(+)	4112.3(+)	4686.6(+)	4185.4(+)	**779.7**
24	Prostate2	10 509	4308.6(+)	5237.6(+)	4836.1(+)	4773.1(+)	4479.8(+)	4489.6(+)	4317.5(+)	**227.0**
25	11Tumor	12 533	6021.5(+)	6257.0(+)	5746.2(+)	5626.3(+)	5680.5(+)	5842.0(+)	5095.1(+)	**383.2**

Table 15
Fitness value of the proposed external attention-based feature ranker for large-scale feature selection compared with seven state-of-the-art feature selection algorithms for six very large-scale datasets.

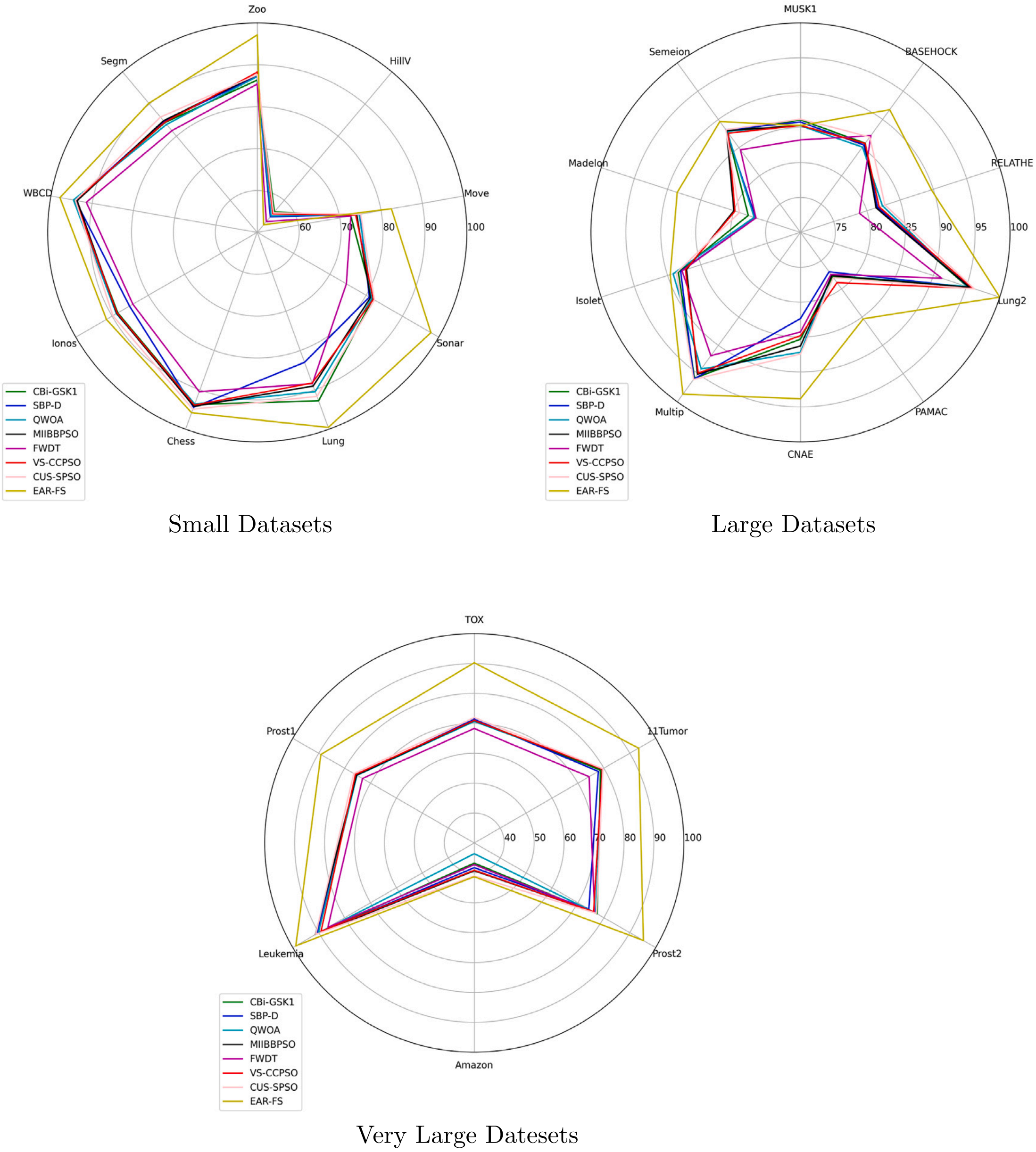
No.	Dataset	CBi-GSK1	SBP-D	QWOA	MIBBPSO	FWDT	VS-CCPSO	CUS-SPSO	EAR-FS
20	TOX_171	0.28 ± 0.01(+)	0.29 ± 0.02(+)	0.30 ± 0.01(+)	0.29 ± 0.02(+)	0.32 ± 0.02(+)	0.29 ± 0.01(+)	0.28 ± 0.01(+)	**0.09** ± 0.00
21	Prostate1	0.25 ± 0.01(+)	0.25 ± 0.02(+)	0.25 ± 0.02(+)	0.25 ± 0.01(+)	0.27 ± 0.02(+)	0.24 ± 0.01(+)	0.23 ± 0.01(+)	**0.10** ± 0.00
22	Leukemia	0.10 ± 0.00(+)	0.10 ± 0.00(+)	0.10 ± 0.01(+)	0.09 ± 0.01(+)	0.14 ± 0.00(+)	0.11 ± 0.00(+)	0.09 ± 0.00(+)	**0.01** ± 0.00
23	Amazon	0.62 ± 0.03(+)	0.62 ± 0.03(+)	0.66 ± 0.05(+)	0.60 ± 0.03(+)	0.62 ± 0.02(+)	0.60 ± 0.02(+)	0.59 ± 0.03(+)	**0.57** ± 0.01
24	Prostate2	0.23 ± 0.00(+)	0.26 ± 0.01(+)	0.23 ± 0.01(+)	0.24 ± 0.00(+)	0.23 ± 0.00(+)	0.24 ± 0.01(+)	0.23 ± 0.00(+)	**0.04** ± 0.00
25	11Tumor	0.22 ± 0.01(+)	0.23 ± 0.01(+)	0.20 ± 0.00(+)	0.21 ± 0.00(+)	0.26 ± 0.01(+)	0.20 ± 0.01(+)	0.21 ± 0.01(+)	**0.06** ± 0.00

Table 16
Result of the proposed external attention-based feature ranker for large-scale feature selection compared with eight state-of-the-art feature selection algorithms for COVID-19 dataset.

Metrics	BCSA_S	BHMICSA_S	BCSA_V	BHMICSA_V	BCSA_U	BHMICSA_U	BCSA_X	BHMICSA_X	EAR-FS
CA	93.47 ± 0.14(+)	**99.19** ± 0.93(−)	93.69 ± 0.18(+)	96.35 ± 0.78(+)	93.28 ± 0.51(+)	97.06 ± 1.02(+)	93.05 ± 0.26(+)	97.91 ± 0.48(+)	98.20 ± 0.01
SZ	4.85 ± 0.88(−)	9.15 ± 1.18(+)	**2.90** ± 0.31(−)	5.41 ± 2.61(−)	3.95 ± 0.76(−)	6.23 ± 1.52(=)	4.15 ± 0.59(−)	8.35 ± 1.66(=)	7.42 ± 3.01
FIT (%)	4.92 ± 0.06(+)	2.33 ± 0.00(+)	3.13 ± 0.15(+)	3.45 ± 0.00(+)	6.21 ± 0.25(+)	2.30 ± 0.00(+)	5.48 ± 0.15(+)	2.33 ± 0.00(+)	**2.27** ± 0.01

6. Conclusions and future work

This article proposed a novel method for FS called EAR-FS. The suggested method makes use of an attention mechanism trained jointly with an auxiliary neural network to rank the features of a dataset. Once the ranking has been performed, EAR-FS searches for a small-sized feature subset associated with high-accuracy performance through a hybrid heuristic search that combines global and local searches. This combination enabled a novel and efficient utilisation of the attention mechanism for FS. The proposed EAR-FS has been tested on small, large, and very large datasets, and its performance has been compared against that of modern FS algorithms representing the state of the art in the FS domain. The experimental results highlight that EAR-FS exhibits outstanding performance in terms of accuracy of the classifier and feature reduction for datasets containing more than 1000 features. Hence, the main advantage of the proposed method lies in its ability to effectively manage extensive feature sets. Conversely, the main limitation of EAR-FS is its performance sensitivity to three



Small Datasets

Large Datasets

Very Large Datesets

Fig. 3. Radar chart on the accuracy of the proposed external attention-based feature ranker for large-scale feature selection compared with eight state-of-the-art feature selection algorithms on the small, large, and very large datasets.

specific hyperparameters. Further investigation is needed to enhance the flexibility of the method so that the logic of sampling and searching in the feature space can be directly connected to the number of features in the original dataset.

CRediT authorship contribution statement

Yu Xue: Supervision, Project administration, Funding acquisition, Conceptualization. **Chenyi Zhang:** Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Conceptualization. **Ferrante Neri:** Writing – original draft, Supervision, Project administration, Methodology, Investigation, Conceptualization. **Moncef Gabbouj:** Supervision, Project administration. **Yong Zhang:** Supervision, Conceptualization.

Declaration of competing interest

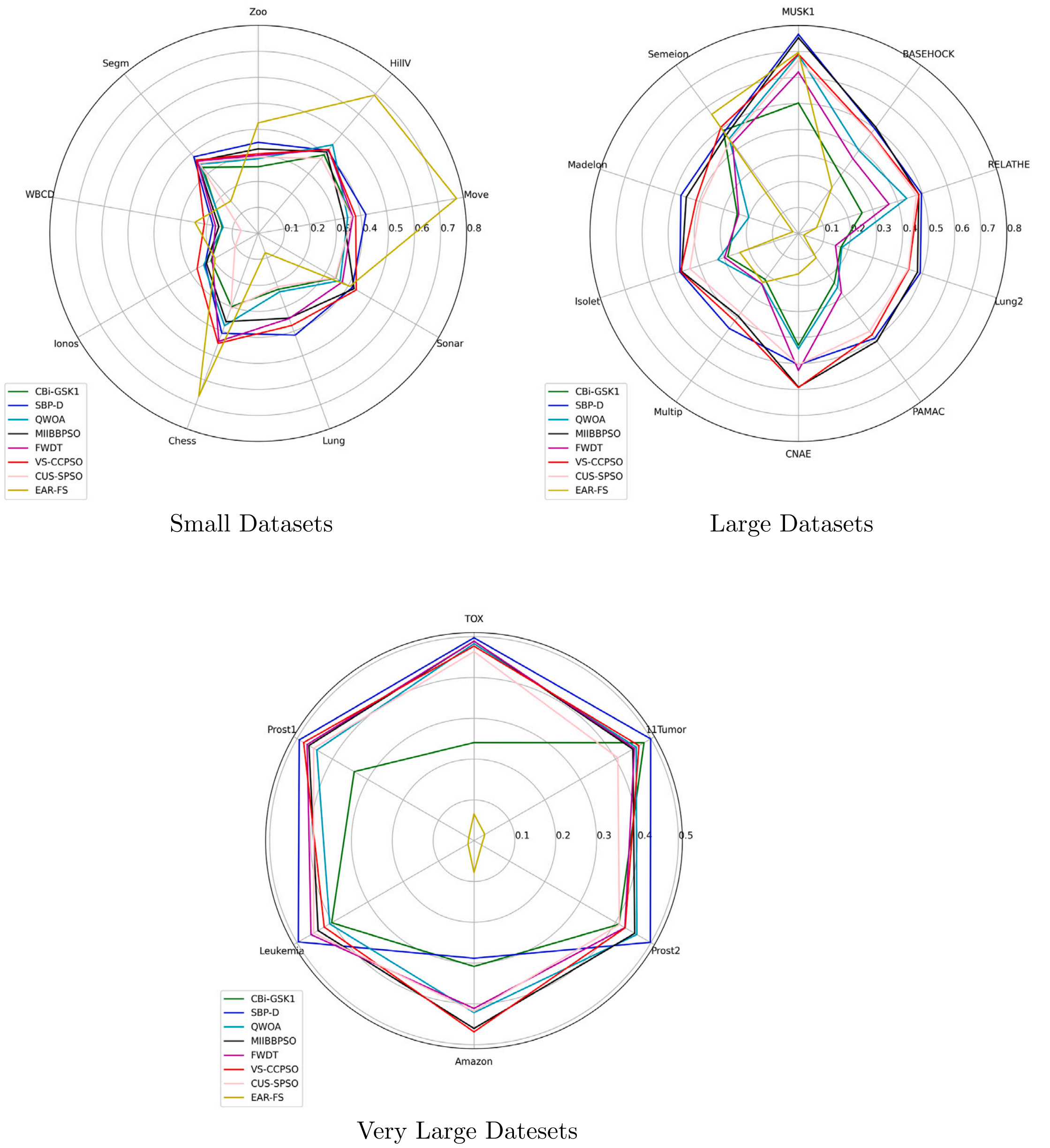
The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (62376127, 61876089, 61876185, 61902281) and the Natural Science Foundation of Jiangsu Province, China (BK20141005). This work was partially supported by the project Distinguished Professors of Jiangsu Province.



Small Datasets

Large Datasets

Very Large Datesets

Fig. 4. Radar chart on the proportion of the selected feature of the proposed external attention-based feature ranker for large-scale feature selection compared with eight state-of-the-art feature selection algorithms on the small, large, and very large datasets.

References

[1] M. Mafarja, S. Mirjalili, Whale optimization approaches for wrapper feature selection, Appl. Soft Comput. 62 (2018) 441–453.

[2] P. Jiang, Y. Xue, F. Neri, Convolutional neural network pruning based on multi-objective feature map selection for image classification, Appl. Soft Comput. 139 (2023) 110229.

[3] G. Zhang, X. Zhang, H. Rong, P. Paul, M. Zhu, F. Neri, Y. Ong, A layered spiking neural system for classification problems, Int. J. Neural Syst. 32 (8) (2022) 2250023:1–2250023:15.

[4] J. Li, K. Cheng, S. Wang, F. Morstatter, R.P. Trevino, J. Tang, H. Liu, Feature selection: A data perspective, ACM Comput. Surv. (CSUR) 50 (6) (2017) 1–45.

[5] J. Cai, J. Luo, S. Wang, S. Yang, Feature selection in machine learning: A new perspective, Neurocomputing 300 (2018) 70–79.

[6] T. Yin, H. Chen, Z. Yuan, T. Li, K. Liu, Noise-resistant multilabel fuzzy neighborhood rough sets for feature subset selection, Inform. Sci. 621 (2023) 200–226.

[7] X. Zhang, J. Li, Incremental feature selection approach to interval-valued fuzzy decision information systems based on λ-fuzzy similarity self-information, Inform. Sci. 625 (2023) 593–619.

[8] Y. Xue, B. Xue, M. Zhang, Self-adaptive particle swarm optimization for large-scale feature selection in classification, ACM Trans. Knowl. Discov. Data 13 (5) (2019) 1–27, Article No.: 50.

[9] R. Espinosa, F. Jiménez, J. Palma, Multi-surrogate assisted multi-objective evolutionary algorithms for feature selection in regression and classification problems with time series data, Inform. Sci. 622 (2023) 1064–1091.

[10] D. You, M. Sun, S. Liang, R. Li, Y. Wang, J. Xiao, F. Yuan, L. Shen, X. Wu, Online feature selection for multi-source streaming features, Inform. Sci. 590 (2022) 267–295.

[11] P. Wang, B. Xue, J. Liang, M. Zhang, Feature selection using diversity-based multi-objective binary differential evolution, Inform. Sci. 626 (2023) 586–606.

[12] B.H. Nguyen, B. Xue, M. Zhang, A constrained competitive swarm optimiser with an SVM-based surrogate model for feature selection, IEEE Trans. Evol. Comput. (2022) 1, http://dx.doi.org/10.1109/TEVC.2022.3197427.

[13] R. Jiao, B. Xue, M. Zhang, Solving multi-objective feature selection problems in classification via problem reformulation and duplication handling, IEEE Trans. Evol. Comput. (2022) 1, http://dx.doi.org/10.1109/TEVC.2022.3215745.

[14] Y. Zhang, D.-w. Gong, X.-z. Gao, T. Tian, X.-y. Sun, Binary differential evolution with self-learning for multi-objective feature selection, Inform. Sci. 507 (2020) 67–85.

[15] R. Jiao, B. Xue, M. Zhang, Benefiting from single-objective feature selection to multiobjective feature selection: A multiform approach, IEEE Trans. Cybern. (2022) http://dx.doi.org/10.1109/TCYB.2022.3218345.

[16] Y. Xue, X. Cai, F. Neri, A multi-objective evolutionary algorithm with interval based initialization and self-adaptive crossover operator for large-scale feature selection in classification, Appl. Soft Comput. 127 (2022) 109420.

[17] Y. Xue, H. Zhu, F. Neri, A feature selection approach based on NSGA-II with relieff, Appl. Soft Comput. 134 (2023) 109987.

[18] Y. Liao, R. Latty, B. Yang, Feature selection using batch-wise attenuation and feature mask normalization, in: 2021 International Joint Conference on Neural Networks (IJCNN), 2021, pp. 1–9, http://dx.doi.org/10.1109/IJCNN52387.2021.9533531.

[19] N. Gui, D. Ge, Z. Hu, AFS: An attention-based mechanism for supervised feature selection, Proc. AAAI Conf. Artif. Intell. 33 (01) (2019) 3705–3713.

[20] B. Xue, M. Zhang, W.N. Browne, X. Yao, A survey on evolutionary computation approaches to feature selection, IEEE Trans. Evol. Comput. 20 (4) (2016) 606–626.

[21] G. Kou, P. Yang, Y. Peng, F. Xiao, Y. Chen, F.E. Alsaadi, Evaluation of feature selection methods for text classification with small datasets using multiple criteria decision-making methods, Appl. Soft Comput. 86 (2020) 105836.

[22] Z. Qiu, W. Zeng, D. Liao, N. Gui, A-SFS: Semi-supervised feature selection based on multi-task self-supervision, Knowl.-Based Syst. 252 (2022) 109449.

[23] A. Taherkhani, G. Cosma, T.M. McGinnity, Deep-FS: A feature selection algorithm for deep Boltzmann machines, Neurocomputing 322 (2018) 22–37.

[24] A. Mirzaei, V. Pourahmadi, M. Soltani, H. Sheikhzadeh, Deep feature selection using a teacher-student network, Neurocomputing 383 (2020) 396–408.

[25] M.A. Awadallah, M.A. Al-Betar, M.S. Braik, A.I. Hammouri, I.A. Doush, R.A. Zitar, An enhanced binary rat swarm optimizer based on local-best concepts of PSO and collaborative crossover operators for feature selection, Comput. Biol. Med. 147 (2022) 105675, http://dx.doi.org/10.1016/j.compbiomed.2022.105675, URL https://www.sciencedirect.com/science/article/pii/S0010482522004632.

[26] M.A. Awadallah, A.I. Hammouri, M.A. Al-Betar, M.S. Braik, M.A. Elaziz, Binary horse herd optimization algorithm with crossover operators for feature selection, Comput. Biol. Med. 141 (2022) 105152, http://dx.doi.org/10.1016/j.compbiomed.2021.105152, URL https://www.sciencedirect.com/science/article/pii/S001048252100946X.

[27] M.S. Braik, A.I. Hammouri, M.A. Awadallah, M.A. Al-Betar, K. Khtatneh, An improved hybrid chameleon swarm algorithm for feature selection in medical diagnosis, Biomed. Signal Process. Control 85 (2023) 105073, http://dx.doi.org/10.1016/j.bspc.2023.105073, URL https://www.sciencedirect.com/science/article/pii/S1746809423005062.

[28] M.-H. Guo, T.-X. Xu, J.-J. Liu, Z.-N. Liu, P.-T. Jiang, T.-J. Mu, S.-H. Zhang, R.R. Martin, M.-M. Cheng, S.-M. Hu, Attention mechanisms in computer vision: A survey, Comput. Vis. Media 8 (3) (2022) 331–368.

[29] V. Mnih, N. Heess, A. Graves, et al., Recurrent models of visual attention, Adv. Neural Inf. Process. Syst. 27 (2014).

[30] D. Bahdanau, K. Cho, Y. Bengio, Neural machine translation by jointly learning to align and translate, 2014, arXiv preprint arXiv:1409.0473.

[31] J. Hu, L. Shen, G. Sun, Squeeze-and-excitation networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 7132–7141.

[32] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Adv. Neural Inf. Process. Syst. 30 (2017).

[33] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al., An image is worth 16x16 words: Transformers for image recognition at scale, 2020, arXiv preprint arXiv:2010.11929.

[34] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, S. Zagoruyko, End-to-end object detection with transformers, in: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16, Springer, 2020, pp. 213–229.

[35] L. Yuan, Y. Chen, T. Wang, W. Yu, Y. Shi, Z.-H. Jiang, F.E. Tay, J. Feng, S. Yan, Tokens-to-token vit: Training vision transformers from scratch on imagenet, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 558–567.

[36] H. Chen, Y. Wang, T. Guo, C. Xu, Y. Deng, Z. Liu, S. Ma, C. Xu, C. Xu, W. Gao, Pre-trained image processing transformer, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 12299–12310.

[37] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, B. Guo, Swin transformer: Hierarchical vision transformer using shifted windows, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 10012–10022.

[38] A. Hassani, S. Walton, J. Li, S. Li, H. Shi, Neighborhood attention transformer, 2022, arXiv preprint arXiv:2204.07143.

[39] G. Huang, Y. Li, G. Pleiss, Z. Liu, J.E. Hopcroft, K.Q. Weinberger, Snapshot ensembles: Train 1, get m for free, in: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings, OpenReview.net, 2017, URL https://openreview.net/forum?id=BJYwwY9ll.

[40] K. Chen, B. Xue, M. Zhang, F. Zhou, Correlation-guided updating strategy for feature selection in classification with surrogate-assisted particle swarm optimization, IEEE Trans. Evol. Comput. 26 (5) (2022) 1015–1029.

[41] C. Iwendi, A.K. Bashir, A. Peshkar, R. Sujatha, J.M. Chatterjee, S. Pasupuleti, R. Mishra, S. Pillai, O. Jo, COVID-19 patient health prediction using boosted random forest algorithm, Front. Public Health 8 (2020).

[42] P. Agrawal, T. Ganesh, A.W. Mohamed, Chaotic gaining sharing knowledge-based optimization algorithm: an improved metaheuristic algorithm for feature selection, Soft Comput. 25 (14) (2021) 9505–9528.

[43] B.H. Nguyen, B. Xue, P. Andreae, M. Zhang, A new binary particle swarm optimization approach: Momentum and dynamic balance between exploration and exploitation, IEEE Trans. Cybern. 51 (2) (2021) 589–603.

[44] R. Agrawal, B. Kaur, S. Sharma, Quantum based whale optimization algorithm for wrapper feature selection, Appl. Soft Comput. 89 (2020) 106092.

[45] X.-f. Song, Y. Zhang, D.-w. Gong, X.-y. Sun, Feature selection using bare-bones particle swarm optimization with mutual information, Pattern Recognit. 112 (2021) 107804.

[46] H. Zhou, J. Zhang, Y. Zhou, X. Guo, Y. Ma, A feature selection algorithm of decision tree based on feature weight, Expert Syst. Appl. 164 (2021) 113842.

[47] X.-F. Song, Y. Zhang, Y.-N. Guo, X.-Y. Sun, Y.-L. Wang, Variable-size cooperative coevolutionary particle swarm optimization for feature selection on high-dimensional data, IEEE Trans. Evol. Comput. 24 (5) (2020) 882–895.