Kolmogorov-Arnold Network Based Feature Extraction of Support Vector Machines

Selami Beyhan[1] and Meric Cetin[2*]

[1]Department of Artificial Intelligence and Data Engineering, Yildiz Technical University, Davutpasa Campus, Esenler, Istanbul, selami.beyhan@yildiz.edu.tr

[2*]Department of Computer Engineering, Pamukkale University, Kinikli, Denizli, mcetin@pau.edu.tr , *Correspondence author

Abstract: Support Vector Machines (SVMs) have been applied to the small datasets using pre-processing methods since their learning-based feature extraction capabilities are limited compared to deep neural networks. In this study, a Kolmogorov-Arnold Network (KAN) is used for feature extraction to achieve a high approximation performance of SVMs. The proposed pre-processing layer involves optimization of functions and scaling parameters using the Adolescent Identity Search Algorithm (AISA). First, trigonometric basis functions are optimized with constant parameters. Then, the parameters of these functions are optimized to construct an optimal Kolmogorov-Arnold network using a smaller number of basis functions. In the proposed model, the internal layer functions of the KAN become trigonometric functions, while the external layer functions are Gaussian distribution functions in the kernel-induced feature space of the SVM model. At the same time, the hyperparameters of the SVM model, V C dimension, ε, and σ of the kernel functions are optimized for further performance improvement. The signal properties of the input and feature layers such as energy, central frequency, entropy, correlation, principal components, and clusters are analyzed to discuss the information content of the layers. The variation in feature signal properties provides important evidence in the KAN regression layer literature regarding how the classification and regression performance is improved using the KAN layer. The proposed KAN-SVM model has been applied for the regression and classification of benchmark datasets. The resulting KAN layer and hyperparameters for the identification of a nonlinear dynamic system are discussed by considering the mathematical model of the system. In addition, input-output data recorded from a newly_developed real-time experimental setup has been accurately identified using the proposed KAN-SVM model.

Keywords: Support Vector Machine, Kolmogorov-Arnold Representation, Feature Extraction, AISA.

1 Introduction

Support vector machines are one of the best-known models in machine learning for solving classification and regression problems. In addition to its strong theoretical background, its easy applicability allows it to be used in embedded systems [1, 2]. Due to its structure, the problem is easily solved linearly in the nonlinear kernel space, where it is created by the kernel function with the distance between the input data. However, it does not have the enhanced feature extraction property to reveal the characteristics of input data as in deep learning models. The difficulty of designing big datasets is trying to be overcome with online SVM models. Besides, SVMs have limitations in feature extraction, especially when dealing with complex data structures. Use of pre-determined features in data pre-processing increases model accuracy [3]. This approach also has advantages such as data compatibility, dimension reduction, complexity reduction, good generalization, and noise resistance [4]. Feature extraction provides better results, especially in complex data structures, as it better represents the data distribution [5, 6]. Dimensionality reduction allows the model to be trained faster and generalize better. This enables the use of SVM with less computational cost and faster training times. Applying feature extraction to reduce the model's complexity and training time helps the model produce solutions faster. This process must be managed according to the characteristics and requirements of the dataset; otherwise, different results may be obtained depending on the nature of the dataset. In some cases, feature extraction may discard necessary features, resulting in the loss of important information [7]. In addition, eliminating noise or unreasonable features through feature extraction can increase its robustness. In [8], a comprehensive analysis and application results have been demonstrated for the least squares structural twin bounded support vector machine considering the class scatter distribution. Recently, autoencoder based feature extraction is proposed to construct multi-view support vector machines for multi-class classification [9].

Kolmogorov-Arnold Networks, a new neural network architecture [10], have been increasingly used as feature extractors in signal processing applications where understanding complex data structures is important. Conventional neural networks perform a weighted sum of the inputs at each node with parameters and feed the result into a statically defined nonlinearity. In contrast, KANs perform nonlinear computations represented by B-splines on the edges leading to each node, then simply sum the inputs at the node. Instead of learning weights, they do this by learning the system spline parameters. This allows complex real-valued functions to be learned more efficiently and accurately. KANs also encourage a more modular approach to feature extraction and transformation by distributing activation functions across edges. Modularity increases the interpretability and flexibility of the network and allows for more detailed representations of the input data [11]. Conventional SVM optimization strategies focus on using hyperparameter optimization or static dimensionality reduction techniques. These approaches may not be successful in capturing high-order nonlinear relationships in complex

datasets. Standard kernels project data onto a fixed feature space, which may not be compatible with the underlying distribution of the input signal. In contrast, KANs offer a difference by replacing fixed activation functions with learnable univariate functions at the edges.

Due to the aforementioned characteristics, many studies have been conducted in recent years using the KAN structure to perform feature extraction. In one of these, KAN has been used as a feature extractor in [12] for inertial measurement units based human activity recognition tasks. In a different study [13], researchers have proposed Conv-KAN, replacing the convolutional kernel with a set of univariate nonlinear functions, KAN, to improve the feature extraction capability. The orthogonality properties in the Chebyshev-KAN architecture [14] allow for more efficient feature extraction with fewer parameters, which results in superior accuracy with minimal processing time. In addition to feature extraction based applications, many applications using KAN architecture for classification [15], clustering [16], function approximation [17], decision making [18], system identification [19], genetic programming [20], control [21, 22] and artificial intelligence applications [23] have been implemented recently. Using KAN in combination with shallow models that require less computation time and memory space not only reduces the computational burden, but also improves the learning accuracy by revealing semantic features from the data. Performing the feature extraction process using KAN provides the model with advantages such as analytical expression, generalization ability, and interpretability. In this way, KAN extracts features from the data and can successfully model complex relationships. When there are fewer expressions representing important features in the dataset, the model extracts more meaningful features. In addition, the rapid applicability of KAN makes it easier to work in real-time applications and with big datasets, thus making it an effective tool in the field of signal processing and artificial intelligence. Therefore, KAN has been used for pattern recognition itself and feature extraction of other machine learning models.

This paper focuses on improving the feature extraction capability of the SVM model through optimized KAN functions. In this process, the adolescent identity search algorithm was chosen due to its unique multi-stage search strategy. While conventional meta-heuristic algorithms may experience early convergence problems in high-dimensional spaces, AISA, which simulates the identity formation process, is strong in escaping local optima. This is important in the proposed model because optimizing the trigonometric basis functions in the KAN layer is quite difficult. The main contributions of this study are as follows: i) The pre-processing layer based on KAN improved the feature extraction capability of the SVM model and contributed to a more effective model due to its practical applicability. ii) Optimizing KAN functions using the recently proposed adolescent identity search optimization algorithm has improved the performance of the SVM model without

increasing its complexity, enhancing generalization performance and stability. iii) The signal properties of the input and feature layers are analyzed in detail. Depending on the type of input signals, the variation in certain properties has been observed to contribute to improved accuracy. By integrating KAN as a dynamic pre-processing layer, the proposed model not only transforms the data but also actively shapes the feature space to make it more separable for SVM. The proposed model offers greater interpretability than traditional deep learning methods. Combining the KAN layer with SVM minimizes computational burden. This creates a suitable structure for real-time signal processing applications where speed and generalization success are prioritized.

2 Support Vector Machines

Although SVM is generally designed for classification problems, it can also be applied to regression problems with the Vapnik loss function [24]. The purpose of the Support Vector Regression (SVR) method is to create an error-insensitive hyperplane by separating data classes and contracting this area according to the extreme value of the data. In ε −SVR, instead of grouping data points around a hyperplane, a margin is tried to be found between data points. This margin is controlled by ε, which determines the error tolerance of the regression model. Thus, the regression model is forced to stay within a certain error limit. Consider that p data pairs $\left[(x_1, y_1), \dots, (x_p, y_p)\right]$ where $x_p \in \Re^d$ and $y_p \in \Re$ exist. x is the input vector, y is the output vector and d denotes the number of dimensions of input data. The aim of ε −SVR is to find the $f(x)$ linear function with the largest deviation ε for the given input-output pairs. The linear function is defined as $f(x) = w^T \rho(x + b)$ where $w \subset \Re^n$ is the weight vector and n indicates the number of dimensions of weight vector. $b \subset \Re$ is a deviation and $\rho(.)$ is the nonlinear transformation from $\Re^n$ to high-dimensional space. For ε −SVR, the convex optimization problem is written as $min \frac{1}{2}\|w\|^2$ subject to $\forall: |y_i - (x_i^T w + b)| \leq \varepsilon, i = 1, \dots, p$ where w indicates the margin. When it is not always possible to find the perfect function for all points, some errors are allowed so that the hard margin becomes the soft margin. To deal with this problem, slack variables (ξ_i, ξ_i^*) are used for convex optimization problems based on the equality constraint as in Eq. (1) and Eq. (2)

$$min\left(\frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l}\xi_i + \xi_i^*\right) \quad \text{subject to} \tag{1}$$

$$\begin{aligned} &y_i - (x_i^T w + b) \leq \varepsilon + \xi_i, \\ &(x_i^T w + b) - y_i \leq \varepsilon + \xi_i^*, \\ &\xi_i \geq 0,\ \xi_i^* \geq 0. \end{aligned} \tag{2}$$

The hyperparameter C determines the balance between the flatness of the straight function and the amount at which larger deviations from ε are tolerated. Then, $|\xi|_\varepsilon$ is defined by Eq. (3)

$$|\xi|_\varepsilon = \begin{cases} 0, & if\ |\xi| \le \varepsilon \\ |\xi| - \varepsilon, & otherwise. \end{cases} \tag{3}$$

This problem is easily solved in the Lagrangian dual form. If the problem is convex and the constraints satisfy the sufficiency condition, the solution of the dual problem is considered the optimal solution. In terms of Lagrangian multipliers α_i, α_i^*, which act as forces pushing the estimates towards the target value y_i, the weight vector can be written as

$$w = \sum_{i=1}^{p} (\alpha_i - \alpha_i^*)\, \rho(x_i). \tag{4}$$

Let's rearrange the $f(x)$ function by substituting Eq.(4) into the $f(x)$ function as

$$f(x) = \sum_{i=1}^{p} (\alpha_i - \alpha_i^*)\, \rho(x_i)\rho(x) + b = \sum_{i=1}^{p} (\alpha_i - \alpha_i^*)\, K(x_i, x) + b \tag{5}$$

where $K(x_i, x)$ is introduced as the kernel function. Not every dataset can be classified in two-dimensional space. In such cases, different kernels are used instead, as increasing the size will increase the processing load. The most commonly used among them are polynomial kernel, Gaussian radial-basis function kernel, sigmoid kernel, etc.

3 Kolmogorov-Arnold Networks

Multi Layer Perceptrons (MLPs) and fixed activation functions are not very successful in feature extraction due to information loss. To enhance the nonlinearity, an activation function is usually included between two convolution layers. In particular, the number of parameters in MLP networks does not scale linearly with the number of layers, leading to a lack of interpretability [25]. In addition, some activation functions used limit the capacity and potentially prevent nodes from learning complex features [26]. On the other hand, KANs replace linear weights along the edges of the network with spline-based univariate functions [10]. These functions are specifically structured as learnable activation functions. This can be particularly useful in machine learning applications. The function approximation capability of KANs allows them to perform effective feature extraction in high-dimensional and complex problem spaces. As a function that directly maps inputs to a high-dimensional space, it creates separable features when it does this. In addition, the KAN architecture can be used to move lower-dimensional inputs to a more separable space. In particular, KAN-based transformation features obtained from image or time series data can be used to train

a machine learning model. Using this method with hybrid models can accelerate the learning process and increase the generalization ability of the model. In addition, a solution is provided to the efficiency and interpretability limitations encountered in MLPs.

In order to overcome the challenges of limited generalizability, interpretability and reproducibility in predictive models or to address the limitations of black-box machine learning modeling in real-world applications, Kolmogorov-Arnold Networks can be preferred. KANs are a new neural network architecture [10] based on the Kolmogorov-Arnold representation theorem [27]. This theorem suggests that any continuous multivariable function can be expressed as a sum of continuous univariable functions. KANs show potential advantages in terms of accuracy and interpretability compared to MLP models. Thanks to these features, it can model and interpret complex systems with high performance. KANs have fully connected structures like MLPs. However, unlike MLPs that use fixed activation functions at their nodes, KANs use adaptive and dynamically adjustable activation functions on the edges. Therefore, KANs do not have linear weight matrices; each weight parameter is replaced by a learnable 1D function represented as a B-spline. Despite this, KANs usually lead to much smaller computational graphs compared to MLPs [28]. According to the Kolmogorov-Arnold representation theorem, any continuous multivariate function on a bounded domain can be expressed as the sum of a finite number of continuous univariate functions. For any smooth function $f: [0,1]^{n_{in}} \rightarrow \Re^{n_{out}}$, the theorem asserts that there exist continuous univariate functions Ψ_q and $\phi_{p,q}$ such that:

$$f(x) = f(x_1, \dots, x_{d_{in}}) = \sum_{q=1}^{2n_{in}+1} \Psi_q \left(\sum_{p=1}^{n_{in}} \phi_{p,q}(x_p) \right). \quad (6)$$

A representative example layer for a KAN with n_{in} −dimensional inputs and n_{out} −dimensional outputs is illustrated in Figure 1.

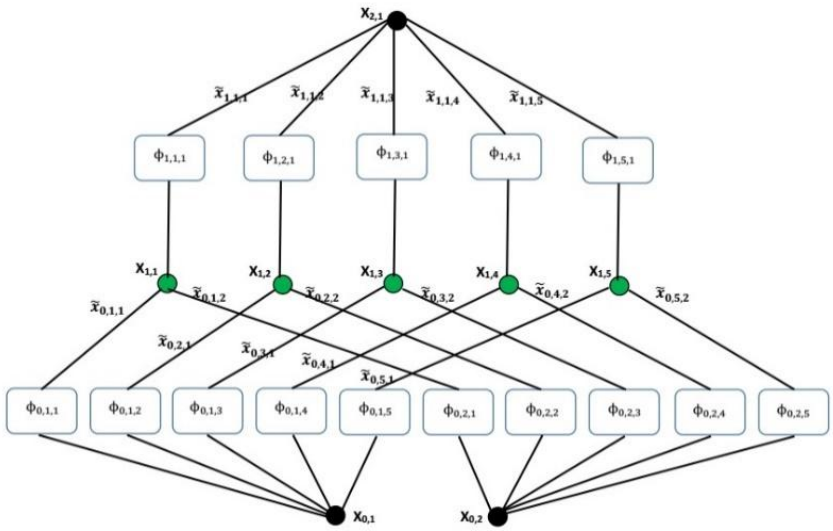


Figure 1
An example of the representation of activation functions flowing through the network in the KANs.

Here, the activation functions are parameterized as a B-spline that allows switching between coarse-grained and fine-grained grids. A KAN structure can be represented as a matrix of 1D functions as follows [29]:

$$\Psi = \{\phi_{l,p,q}\},\ p = 1, \dots, n_{in},\quad q = 1, \dots, n_{out}, \tag{7}$$

where the functions $\phi_{p,q}$ have learnable parameters. The shape of KAN can be expressed an integer array as $[n_0, n_1, \dots, n_l]$ where n_l represents the number of neurons in layer l. While the ith neuron in the lth layer is shown with the index (l, i), the activation value of this neuron is represented by $x_{l,i}$. There are $n_l n_{l+1}$ activation functions between the consecutive layers. The activation function connecting neurons (l, i) and $(l + 1, j)$is shown as follows:

$$\phi_{l,i,j},\ l = 0, \dots, L-1,\quad i = 1, \dots, n_l,\ \ j = 1, \dots, n_{l+1}. \tag{8}$$

In KANs, the activation functions appear as on the edges rather than nodes. Specifically, the pre-activation of $\phi_{l,i,j}$ is $x_{l,i}$, while the post-activation of $\phi_{l,i,j}$ is $\tilde{x}_{l,i,j}$. The terms $\tilde{x}_{l,i,j}$ are then summed to determine the activation value $x_{l+1,i}$ at the $l + 1, i$ −neuron as in Eq. (9):

$$x_{l+1,i} = \sum_{i=1}^{n_l}(\tilde{x}_{l,i,j}) = \sum_{i=1}^{n_l} \phi_{l,i,j}(x_{l,i}). \tag{9}$$

x_{l+1} can be written in a matrix form with the function matrix (Ψ_l) corresponding to the lth KAN layer as

$$x_{l+1,i} = \begin{bmatrix} \phi_{l,1,1}(.) & \phi_{l,1,2}(.) & \cdots & \phi_{l,1,n_l}(.) \\ \phi_{l,2,1}(.) & \phi_{l,2,2}(.) & \cdots & \phi_{l,2,n_l}(.) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_{l,n_{l+1},1}(.) & \phi_{l,n_{l+1},2}(.) & \cdots & \phi_{l,n_{l+1},n_l}(.) \end{bmatrix} \tag{10}$$

Then, a general KAN is defined as a composition of L layers:

$$KAN(\mathbf{x}) = (\Psi_{L-1} \circ \Psi_{L-2} \circ \cdots \circ \Psi_1 \circ \Psi_0)\mathbf{x}. \tag{11}$$

Replacing linear weights in KANs with learnable univariate functions placed on the edges connecting neurons allows the network to learn more complex transformations. Thus, increasing the network's capacity to approximate high-dimensional functions.

KAN based feature extraction is an optimization based simple and effective approach compared to literature methods, i.e. unlike PCA, which is limited to linear projections and variance maximization, the proposed KAN layer enables nonlinear and adaptive feature transformations driven by the data. Compared to autoencoder-based approaches, which require additional neural network training and architectural design choices, the KAN-based layer provides an explicit and interpretable nonlinear feature construction with lower computational complexity.

4 Adolescent Identity Search Algorithm (AISA)

Adolescent Identity Search Algorithm (AISA) is a powerful meta-heuristic optimization method inspired by the identity search process of adolescents [30]. During identity discovery, adolescents can form their own identity by observing and reasoning about the behaviors of their peer group. Three different behaviors are taken into consideration in AISA. *Case 1:* Adolescents can identify and imitate the best features in their peer group. The best features ($\boldsymbol{\chi}^*$) are found by using an orthogonal function approach. *Case 2:* If the adolescent's identity is formed by imitating a role model ($\boldsymbol{\chi}^{\mathrm{rm}}$) with a high-level of power, the individual with the most appropriate fitness value in the peer group can be chosen as the role model. *Case 3:* If the adolescent adopts a negative identity feature ($\boldsymbol{\chi}^{\mathrm{q}}$), this is used to make the algorithm exploratory. To define these behaviors in AISA, an unconstrained fitness function $\boldsymbol{f}(.)$ to be minimized is written by Eq. (12)

$$\begin{aligned} &\min \boldsymbol{f}(.) \\ &\underline{b}_j \le \chi_j \le \overline{b}_j, \;\; j = 1, 2, \ldots, n \end{aligned} \tag{12}$$

In AISA, a random initial adolescent population ($\boldsymbol{\chi}$) is formed within the boundaries of the solution space with an artificial peer group, where $\underline{b}_j$ and $\overline{b}_j$ are the jth lower and upper bound terms, respectively. Each adolescent identity vector defines the jth identity feature of the adolescent as $\chi_j^i = \overline{b}_j + U(0,1)_j \times \left(\overline{b}_j - \underline{b}_j\right)$.

$U(0,1)$ is a random number. The general form of the fitness vector of the solution population is written as in Eq. (13):

$$\boldsymbol{f}(\chi) = \begin{bmatrix} f_1(\chi_1^1, \chi_2^1, \dots, \chi_n^1) \\ \vdots \\ f_N(\chi_1^N, \chi_2^N, \dots, \chi_n^N) \end{bmatrix}_{N\times 1} \tag{13}$$

where N and n represent adolescents and each adolescent's identity, respectively. Due to their ability in recurrence processes, Orthogonal Chebyshev Polynomials are suitable for function approximation applications, so in AISA, identities are mapped via $\{T_k(\chi)\}_{k=0,1,2,\dots}$ where k is the degree of Chebyshev polynomials [30, 31]. Then, the weighting parameters are estimated using the regressor matrix by following

$$\widehat{w} = (\zeta^T\zeta)^{-1}\zeta^T f, \tag{14}$$

and the regressor matrix is defined by Eq. (15):

$$\zeta = \begin{bmatrix} T_1(\hat{\chi}_1^1) & \dots & T_k(\hat{\chi}_1^1) & \dots & T_k(\hat{\chi}_n^1) \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ T_1(\hat{\chi}_1^N) & \dots & T_k(\hat{\chi}_1^N) & \dots & T_k(\hat{\chi}_n^N) \end{bmatrix}_{N\times(n+k)} \tag{15}$$

where $\hat{\chi}_j^i$ is the normalized jth identity feature value of the ith adolescent. The partial fitness value for each $\hat{\chi}_j^i$ is obtained using $\hat{f}_j^i = \zeta_j^i w^i$. The best identity vector of the current population can be found as $\chi_j^* = \chi_j^{m^j}$ where $m^j = \arg min_l\{\hat{f}_j^i \mid l = 1, 2, \dots, N\}$, $\forall j$. Assuming that each adolescent randomly selects only one of the cases in each iteration, the possible behaviors are summarized as follows:

$$\chi_{new}^{\mathrm{i}} := \begin{cases} Case\ 1: & \chi^i - r_1(\chi^i - \chi^*),\ r_4 \le \frac{1}{3} \\ Case\ 2: & \chi^i - r_2(\chi^p - \chi^{rm}),\ \frac{1}{3} \le r_4 \le \frac{2}{3} \\ Case\ 3: & \chi^i - r_3(\chi^i - \chi^q),\ \frac{2}{3} \le r_4 \end{cases} \tag{16}$$

where r_1, r_2 and r_4 are random number in the range of $[0,1]$. r_3 defines a row vector of uniformly distributed numbers in the range of $[0,1]$. The complexity of the adolescent identity search optimization algorithm generally varies depending on the data structure used, search strategy, and problem size. The high-performance applications of AISA in the literature, especially in complex and multidimensional problems such as control and image processing, have enabled it to make many theoretical and practical contributions compared to other alternative optimization algorithms. The pseudocode of the AISA is given in Algorithm 1.

Algorithm 1: AISA Optimization

Define N, k, max_{iter};
Initialize adolescent population $x_j^i, (i = 1, \ldots, N; j = 1, \ldots, n)$;
Evaluate objective function for each adolescent;
while *stopping criterion is not satisfied or* $t < max_{iter}$ **do**
  Form the normalized input matrix $\hat{\boldsymbol{X}}$;
  Define the regressor matrix $\boldsymbol{\zeta}$;
  Compute the weighting vectors $\hat{\boldsymbol{w}}$;
  Define the partial fitness matrix $\hat{\boldsymbol{F}}$;
  Find the best set of identity feature vector $\boldsymbol{\chi}^*$;
  for $i = 1$ *to* N **do**
    Update $r_4 \sim U(0, 1)$;
    if $r_4 \leq \frac{1}{3}$ **then**
      Update $r_1 \sim U(0, 1)$;
      $\mathbf{x}_{new}^i = \mathbf{x}^i - r_1(\mathbf{x}^i - \mathbf{x}^*)$;
    else if $r_4 > \frac{1}{3} \&\& r_4 \leq \frac{2}{3}$ **then**
      Update $r_2 \sim U(0, 1)$;
      Find the role model $(\mathbf{x}^{rm})$;
      Randomly choose one of the adolescents $p | p \neq rm$;
      $\mathbf{x}_{new}^i = \mathbf{x}^i - r_2(\mathbf{x}^p - \mathbf{x}^{rm})$;
    else
      Update $r_3 \sim U(0, 1)$;
      Generate the negative identity vector $(\mathbf{x}^q)$;
      $\mathbf{x}_{new}^i = \mathbf{x}^i - r_3(\mathbf{x}^i - \mathbf{x}^q)$;
    end
    Apply the boundary control mechanism and updating mechanism
  end
end
return *the best optimal solution*

5 Feature Extraction Through Kolmogorov-Arnold Network

The proposed feature extraction approach uses Kolmogorov-Arnold Networks as a pre-processing layer before the SVM model. In fact, functionally the two structures are intertwined since the input data is first transformed through a set of optimized KAN functions, which serve as feature extractors. These functions are designed to capture complex nonlinear relationships in the data, providing a flavored feature representation for the SVM model. The KAN model consists of an internal transformation layer, where input variables are processed through basis functions such as trigonometric or polynomial functions. The extracted features are then passed through an exponential activation function, further enhancing the representation capability of the model. These transformed features become the input to the SVM classifier, improving its ability to discern complex patterns in the data. The proposed model is named KAN ε −SVM shown in Figure 2.

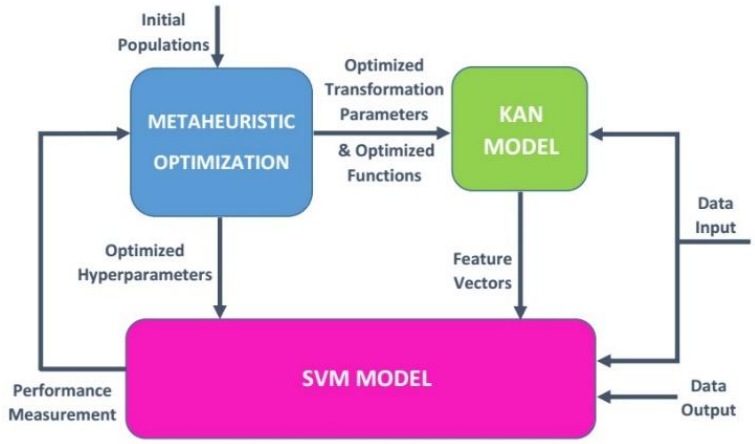


Figure 2
The proposed KAN based SVM design where KAN functions are optimized with AISA.

A major point of this integration is that while KAN provides as the feature transformation mechanism, its external layer functions effectively behaves as the kernel functions of the SVM. This means that the final decision boundary is shaped not only by the SVM model but also by the structured transformations introduced by KAN functions. By doing that the synergy between KAN-based feature extraction and SVM kernel functions enables the development of a more powerful classification/regression framework with improved generalization capabilities and lower computational requirements compared to deep learning models. By optimizing the KAN transformation layer through the AISA, the proposed method ensures that the most effective basis functions and scaling parameters are selected. The transformation layer also achieves feature extraction process that enhances the classification performance of the SVM model while maintaining efficiency in real-time applications. The proposed feature extraction approach qualifies KANs as a pre-processing layer before the SVM model. The transformation of input data in KAN functions is defined by Eq. (17)

$$\phi_i(x) = \sum_{j=1}^{n} w_{ij} \Psi\left(\alpha_j x_j + b_j\right), \tag{17}$$

where w_{ij} are optimized weights, α_j and b_j are scaling and shifting parameters updated through the AISA, and x_j represents the input variables. $\Psi(x)$ is the optimized basis function of the KAN layer. The extracted features are then passed through an exponential activation function of the SVM model, $z_i = e^{\phi_i(x)}$, further enhancing the representation capability using kernel-space of the SVM model. In the nonlinear kernel-space, the constraint optimization of SVM parameters expected to build optimal hyperplanes for the problem definition. It might be

considered that the KAN layer can be considered is an internal trigonometric kernel-space of the input data. Then, the SVM classification output is defined by Eq. (18)

$$f(x) = \sum_{i=1}^{p} (\alpha_i - \alpha_i^*)\, K(z_i, z) + b, \tag{18}$$

where $K(z_i, z)$ is the SVM kernel function computed over the transformed feature space z. It means that the final decision hyperplane is shaped not only by the SVM model but also by the structured transformations introduced by KAN functions. Together KAN-based feature extraction and SVM kernel functions enable the development of a more powerful classification model with improved generalization capabilities and lower computational requirements compared to deep learning models with less number of parameters. In general, by optimizing the KAN transformation layer through the AISA algorithm, the proposed model ensures that the most effective basis functions and scaling parameters are selected. In this study, the population size of AISA is fixed to 50 candidates, the maximum number of iterations is set to 100, and all optimization parameters are constrained within the interval [−5, +5], respectively. The KAN-based feature extraction model proposed in this study is a two-layer nonlinear transformation structure where each layer uses 15 pre-defined nonlinear basis functions. These basis functions are constructed using trigonometric, hyperbolic, exponential, and logarithmic mappings. Each basis function is parameterized by a scalar coefficient, and all coefficients are optimized using the AISA algorithm during training. Unlike classical spline-based KAN formulations, explicit spline interpolation is not used; instead, parametric analytic nonlinear functions are adopted to construct the extended feature space in a computationally efficient manner.

6 Computational Results

This section presents the numerical and real-time application results of SVM models on benchmark datasets, both in classification and regression. In addition, an experimental barn system has been introduced, and its identification results are shown, respectively. The main goal is to present the improved results using the proposed feature extraction layer. It is known that the most important design problem of the SVM model is the selection of hyperparameters. The selected parameters directly affect its performance. Therefore, the hyperparameters ε, C and σ are optimized with AISA optimization in addition to the functions of the feature extraction layer. As a result, optimal hyperparameters and a KAN layer have been found for the given data. In this study, the authors wanted to investigate how the signals of the layer are analyzed with important features such as the mean and variance of the central frequency and energy, the ratio of negative and positive correlation coefficients, the principal components and the number of points in the K-means clusters [32].

The model training procedure follows a fixed and reproducible pipeline. All input features are first normalized, after which the KAN-based nonlinear feature extraction layer is applied to construct an extended feature space. The model is trained using the predefined training set, while hyperparameter selection and early stopping are performed on the validation set. The final performance is then reported on the independent test set using standard evaluation metrics. The computational cost of the proposed framework mainly depends on the KAN-based feature extraction and the AISA optimization stages. The complexity of the KAN layer scales with the number of input features and the number of nonlinear basis functions, while the computational load of AISA is determined by the population size and the maximum number of iterations. In the experiments, fixed population and iteration settings are used, and the overall runtime remains practical on standard computing hardware. In the experiments, a fixed population size and a pre-defined maximum number of iterations were used for the AISA algorithm. n_{χ}, representing the population size, and max_{iter}, representing the maximum number of iterations, were set to 50 and 100, respectively. To ensure numerical stability during the search process, the optimization parameters were limited within the specified upper bounds ($\overline{b} = 5$) and lower bounds ($\underline{b} = -5$). These settings were kept constant throughout the experiments to ensure reproducibility and fair comparison of results.

6.1 Classification Results

6.1.1 Breast Cancer Classification

The Wisconsin Diagnostic Breast Cancer (WDBC) dataset is open-source data at the UCI Machine Learning repository and was created at the University Hospital of California [33]. It has 569 samples, of which 212 were malignant and 357 were benign. 32 input features are recorded to diagnose it as benign or malignant. Some of the input features are the radius, texture, perimeter, area, smoothness, compactness, concavity, concave points, symmetry, fractal size for each cell nucleus, etc. The features are normalized to $[0,1]$ and then classified to diagnose. In the literature, better classification accuracy can be seen using different models for this dataset, but the main goal here is to present the advantages of the proposed feature extraction method instead of comparing them.

6.1.2 PIMA Diabet Classification

One of the datasets most commonly used to compare machine learning models in the literature is the Pima Indian diabetes dataset [34], which contains data from 768 female diabetics. These data were collected in Phoenix, Arizona, from patients at least 21 years of age, of whom 65% were diagnosed as non-diabetic and 35% as

having diabetes. There are eight different fields and one output set in the dataset. The features are as follows: i) the number of pregnancies of the woman ranges from 0 to 17, ii) blood glucose value that ranges from 0 to 199, iii) blood pressure that ranges from 0 to 122, iv) skin thickness in mm that ranges from 0 to 99, v) the amount of insulin in the blood is 0 to 846 μU/ml, vi) the body mass index (BMI) shows the weight-to-height ratio, vii) the diabetes pedigree function (DPF) of the body is 0.078 to 2.42; viii) the age ranges from 21 to 81; and the output data is 0 or 1, non-diabetic or diabetic. The input features are normalized to $[0,1]$.

The classification results with conventional $\varepsilon-$SVM (with only hyperparameter optimization) and the proposed $\varepsilon-$SVM with KAN (with hyperparameter and KAN layer optimization) are presented in Table 1.

Table 1
The performances of classification.

Model/Accuracy	Breast Cancer Classificattion	Pima Indian Diabet Classification
$\varepsilon-$SVM	%94.69	%80.28
$\varepsilon-$SVM with KAN	%99.17	%83.12

6.2 Regression Results

The datasets are partitioned into training, validation, and testing subsets using a 60%–20%–20% split. For the regression experiments, approximately three years of sunspot data (around 1,100 samples) and 30 minutes of data collected from the experimental setup (approximately 1,800 seconds) are used. This data partitioning strategy is applied same throughout the experiments to ensure fair evaluation and reproducibility of the results.

6.2.1 Experimental Barn Identification

Heat stress in livestock farming refers to the situation in which animals are exposed to a high temperature and humidity of the environment that exceeds their ability to regulate body temperature. If the physiological changes in animals resulting from heat stress are affected, meat and milk production may decrease significantly. Therefore, the use of high-level automation solutions is preferred to ensure optimal environmental conditions in the stables. In [36], a scaled-down experimental barn model has been designed to develop modeling and control methods to achieve improved environmental conditions to prevent heat stress in livestock farming. The designed experimental barn is shown in Figure 3.

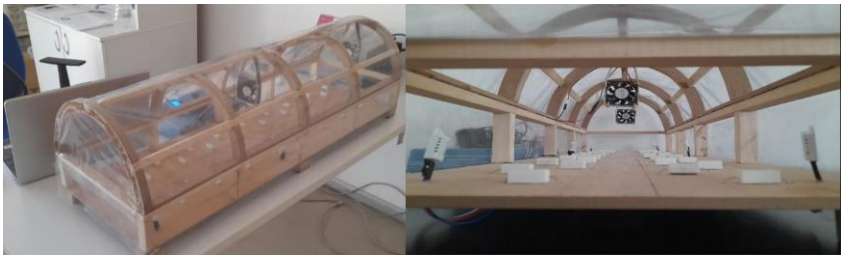
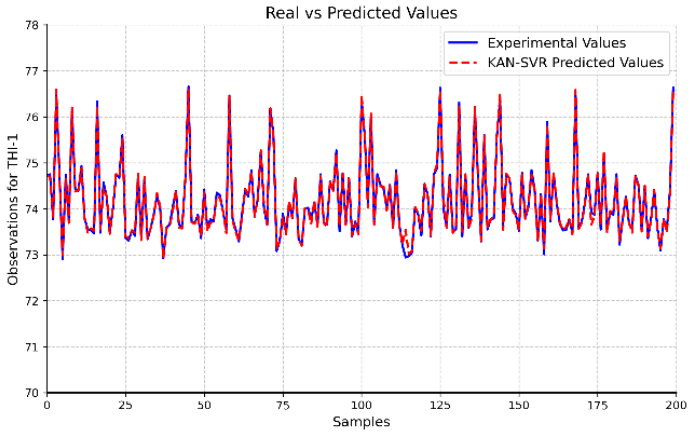


Figure 3
The experimental barn: outside view (left), internal view (right).

The temperature-humidity index (THI), a linear function of temperature and humidity, is a general measure of heat stress. Temperature and humidity values in cow barns are measured in different ways in different parts of the barn due to reasons such as the angle of incidence of sunlight and the number of animals in the barn. The cooling fans must provide air to the entire barn while the cooling process is taking place. By reading the temperature and humidity values from the sensors in the designed experimental barn, ventilation fans are activated when necessary, thus keeping the THI value of the environment at the optimum level. The collected data are used to model the THI value of the barn [36]. For the application of the proposed model, THI values of the first and second regions are being identified and results are shown in Figure 4, respectively.



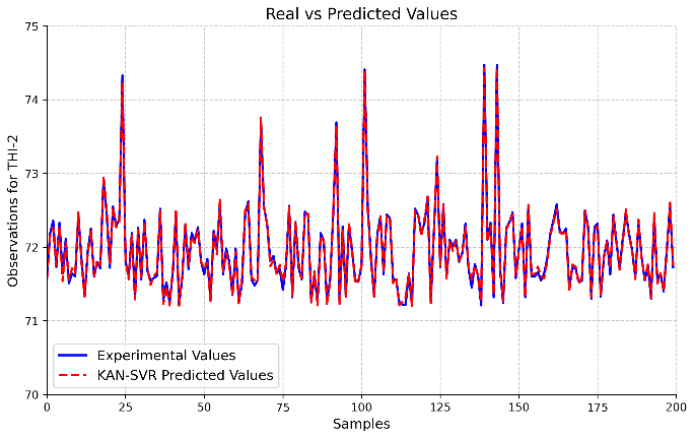


Figure 4

Temperature-humidity index identification of experimental barn.

6.2.2 Sunspot Time Series Prediction

Fluctuations in solar activity can cause many variable effects on solar systems, such as solar flares and magnetic storms. These effects are observed as significant differences in the Earth's space magnetosphere and ionospheric plasma density. In addition, many issues that concern living things, such as radar systems, power grids, radio communications, telecommunications or climate, can be affected by these changes [35]. For the reasons mentioned, estimating and forecasting solar activity has become important. The sunspot number time series is the only direct record used to observe the long-term evolution of the solar cycle and the long-term impact of the Sun on the Earth's environment. In literature studies, the International Sunspot Number time series, which includes solar activity data between 1749 and 2025, is generally used. In this paper, we considered a sunspot time series dataset organized as annual and monthly on the SILSO website (www.sidc.be/silso/datafiles) to evaluate the effectiveness of the proposed method. This highly nonlinear time series is difficult to identify with linear models. In simulations, a total of 3 years of daily-based data have been used, but only 200 of the predictions have been seen in detail.

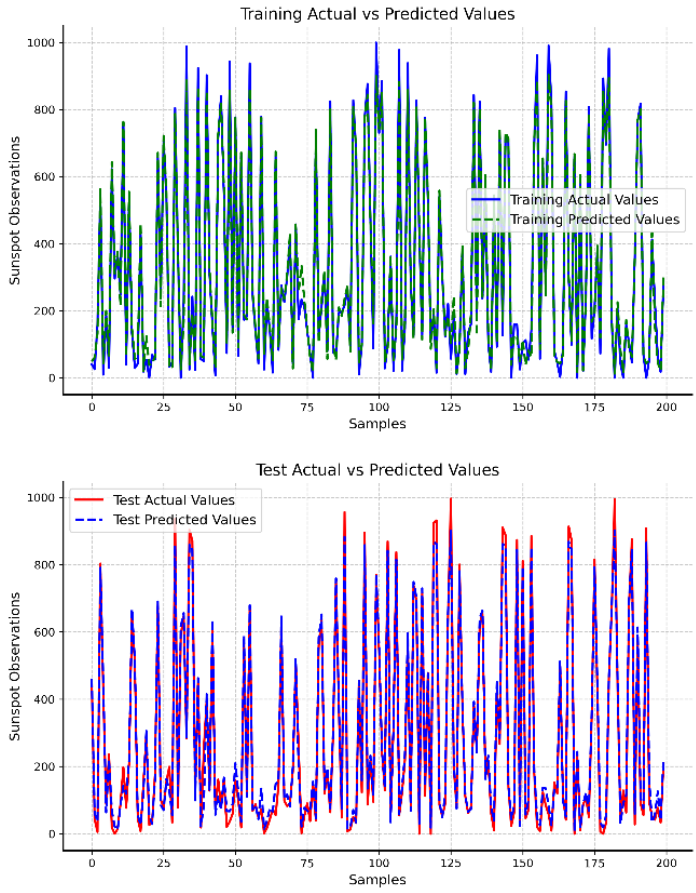


Figure 5
Sunspot time series prediction using KAN based SVM model.

The system identification results of this data are illustrated in Figure 5. For the regression results, the root mean squared error (RMSE), the mean absolute error (MAE), the mean squared error (MSE) and R-squared (R^2) values of the identification datasets are given in Table 2 where significant improvements in performance are observed. Compared to the experimental barn system, the performance improvement on the sunspot time series prediction is more limited, as sunspot time series is highly nonlinear, noisy, and exhibits weaker short-term temporal dependency, making accurate prediction considerably more difficult. In fact, a similar difference exists on the classification part due to the limited data and difficulty of the Pima Indian Diabet dataset seen in Table 1.

Table 2
The model performances of system identification.

Performance (RMSE, MAE, MSE, R^2)	**Sunspot Time Series**	**Experimental Barn System**
ε −SVM	RMSE: 16.13, MAE: 11.77, MSE: 0.062, R^2: 0.9003	RMSE: 0.4488, MAE: 0.3864, MSE: 0.00856, R^2: 0.8316
ε −SVM with KAN	RMSE: 15.12, MAE: 11.65, MSE: 0.029, R^2: 0.9121	RMSE: 0.1818, MAE: 0.1580, MSE: 0.00185, R^2: 0.9724

6.3 Feature Modification with KAN Layer

Table 3 and Table 4 illustrate the analysis results of the input and feature layers for the breast cancer dataset and the identification of the dynamic system, respectively. They show how a simple feature layer changes the signal characteristics and increases the performance of both classification and regression. In Table 3, the optimized functions have first increased the frequency ranges and changed the amplitude of the input signals. Therefore, there is a significant increase in the mean and variance of the signal energies. Then, when we observe the ratio of positive and negative numbers of correlation coefficients between the data, it is observed that the KAN basis function layer separates the data. Moving away from zero provides information by increasing the positive and negative correlations. It is observed that the irregularity slightly increases in the data measured by the Shannon entropy for classification. When the principal components are analyzed, the variance ratio of the first value is increased and the number of variances in the first 95 percent confidence interval is observed. Finally, K-means clustering has been applied for 3 clusters. It has been observed that some of the data are transformed to another space and the distribution is changed.

Table 3
Signal properties of input and feature layers for Breast Cancer Dataset.

Signal Property/Layer	Input Layer	Feature Layer
Energy *(mean, variance)*	(661, 6.57e4)	(4.4e4, 2.2e10)
Correlation Analysis *(ratio of (-) and (+) values)*	(13%, 87%)	(41%, 59%)
Central Frequency *(mean, variance)*	(1.9e-3, 1.3e-5)	(9.0e-3, 9.4e-4)
Shannon Entropy	2.80	2.82

Principle Components	(52.77, 5)	(56.10,7)
K-means Clustering	(121, 165, 283)	(196, 366, 7)

In Table 4, some signal properties varied in a similar way, while others changed in the opposite direction. The reason is that in dynamic system identification, some inputs are past values of the output and there is a direct functional relationship between input and output. For this reason, the feature functions of dynamic system identification are described in detail below.

Table 4
Signal properties of input and feature layers for Dynamic System Identification.

Signal Property/Layer	Input Layer	Feature Layer
Energy *(mean, variance)*	(1.3e3, 8.04e4)	(694.6, 7.1e4)
Correlation Analysis *(ratio of (-) and (+) values)*	(0%, 100%)	(50%, 50%)
Central Frequency *(mean, variance)*	(4.3e-3, 6.27e-8)	(3.3e-3, 7.8e-7)
Shannon Entropy	2.52	2.48
Principle Components	(98.6, 3)	(91.4,5)
K-means Clustering	(358, 413, 229)	(229, 322, 449)

Since the mathematical model of the dynamic system is known for the generation of data, the question mark was whether the feature functions would be similar to the system functions or not. The SVM model is assumed to be a second-order, dynamic, casual, nonlinear and discrete-time system, and it has input and feature vectors as in Eq. (19):

$$\hat{y}(k) = SVM\big(\mathbf{x}(k), u(k), y(k-1), y(k-2)\big) \tag{19}$$

where $\mathbf{x}(k)$ is the feature vector of the inputs such that in the end of optimization, its KAN functions are found by Eq.(20).

$$\mathbf{x}(k) = \begin{bmatrix} \tanh\big(u(k)+y(k-1)\big) \\ \sin\Big(\mathrm{mod}\big(\mathrm{y(k-1)},\mathrm{y(k-2)}\big)\Big) \\ \mathrm{mod}\big(u(k),u(k)\big)^2 \\ \cos\Big(3*\max\big(\mathrm{u(k)},\mathrm{y(k-2)}\big)\Big) \\ \max\big(\mathrm{u(k)},\mathrm{y(k-2)}\big)^2 \\ \tanh\Big(\mathrm{mod}\big(\mathrm{u(k)},\mathrm{u(k)}\big)\Big) \\ \sin\Big(\mathrm{mod}\big(\mathrm{u(k)},\mathrm{y(k-1)}\big)\Big) \\ 2*\Big(\min\big(u(k),u(k)\big)\Big)^2-1 \\ 2*\Big(\min\big(y(k),u(k)\big)\Big)^2-1 \\ 4*\Big(\max\big(y(k-2),y(k-2)\big)\Big)^3 \\ -3*\max\big(y(k-2),y(k-2)\big) \end{bmatrix}^T \tag{20}$$

where two of them are producing zero values. Since the original system mathematical model was trigonometric and highly nonlinear, some features were found to be trigonometric, and others were nonlinear polynomial. The finally optimized hyperparameters are found as $C = 1088$, $\varepsilon = 3.8\mathrm{e}-5$, and $\sigma = 0.155$, respectively. When the hyperparameters are only optimized without feature layer, they are found as $C = 526$, $\varepsilon = 9.99\mathrm{e}-5$, and $\sigma = 0.0295$, respectively.

7 Conclusions

This paper proposed a simple and effective feature extraction method to improve the regression and classification performance of the SVM model. This approach contributes to the field by providing advanced feature extraction capability previously seen in deep learning models while increasing the ability of SVM to deal with complex data structures. The use of KAN increased generalization ability and interpretability such that the internally created trigonometric kernel space features allowed the SVM model to successfully learn complex relationships by providing the extraction of meaningful features from the data. In addition, KAN's rapid applicability and low computational cost have made it possible to conduct real-time signal processing experiments. As a result of analyzing the signal properties in the input and feature layers, it has been shown that the signal properties change significantly and provide information for the SVM model. In future work, new approaches will be developed to improve these signal features in deep learning and to understand which signal feature improves performance when using different data.

References

[1] Hou HR, Meng QH, Zeng M, et al (2017) Improving classification of slow cortical potential signals for bci systems with polynomial fitting and voting support vector machine. IEEE Signal Processing Letters 25(2):283–287

[2] Yang L, Feng W, Wu Y, et al (2023) Radar-infrared sensor fusion based on hierarchical features mining. IEEE Signal Processing Letters

[3] Li K, Luo J, Hu Y, et al (2020) A novel multi-strategy de algorithm for parameter optimization in support vector machine. Journal of Intelligent Information Systems 54:527–543

[4] Dalal AA, Cai Z, Al-qaness MA, et al (2023) Etr: Enhancing transformation reduction for reducing dimensionality and classification complexity in hyperspectral images. Expert Systems with Applications 213:118971

[5] Mauceri S, Sweeney J, Nicolau M, et al (2021) Feature extraction by grammatical evolution for one-class time series classification. Genetic Programming and Evolvable Machines 22(3):267–295

[6] Eldin SS, Mohammed A, Eldin AS, et al (2021) An enhanced opinion retrieval approach via implicit feature identification. Journal of Intelligent Information Systems 57:101–126

[7] Comak E, Gunduz, G. (2025). Comparative analysis of performances of an improved particle swarm optimization and a traditional particle swarm optimization for training of neural network architecture space. Acta Polytechnica Hungarica, 22(5).

[8] Gupta U, Gupta D (2023) Least squares structural twin bounded support vector machine on class scatter. Applied Intelligence 53(12):15321–15351

[9] Li Y, Xie X (2025) Two novel deep multi-view support vector machines for multiclass classification. Applied Intelligence 55(2):1–17

[10] Liu Z, Wang Y, Vaidya S, et al (2024) Kan: Kolmogorov-arnold networks. arXiv preprint arXiv:240419756

[11] Viktoratos I, Tsadiras A (2025) Advancing real-estate forecasting: A novel approach using kolmogorov–arnold networks. Algorithms 18(2):93

[12] Liu M, Geißler D, Nshimyimana D, et al (2024) Initial investigation of kolmogorov-arnold networks (kans) as feature extractors for imu based human activity recognition. In: Companion of the 2024 on ACM International Joint Conference on Pervasive and Ubiquitous Computing, pp 500–506

[13] Bodner AD, Tepsich AS, Spolski JN, et al (2024) Convolutional kolmogorov-arnold networks. arXiv preprint arXiv:240613155

[14] Seydi ST, Sadegh M, Chanussot J (2025) Kolmogorov-arnold network for hyperspectral change detection. IEEE Transactions on Geoscience and Remote Sensing

[15] Jamali A, Roy SK, Hong D, et al (2024) How to learn more? exploring kolmogorov–arnold networks for hyperspectral image classification. Remote Sensing 16(21):4015

[16] Wu Z, Lu H, Paoletti ME, et al (2025) Kacnet: Kolmogorov-arnold convolution network for hyperspectral anomaly detection. IEEE Transactions on Geoscience and Remote Sensing

[17] SS S, AR K, KP A, et al (2024) Chebyshev polynomial-based kolmogorov-arnold networks: An efficient architecture for nonlinear function approximation. arXiv preprint arXiv:240507200

[18] De Carlo G, Mastropietro A, Anagnostopoulos A (2024) Kolmogorov-arnold graph neural networks. arXiv preprint arXiv:240618354

[19] Shuai H, Li F (2025) Physics-informed kolmogorov-arnold networks for power system dynamics. IEEE Open Access Journal of Power and Energy

[20] Long Q, Wang B, Xue B, et al (2025) A genetic algorithm-based approach for automated optimization of kolmogorov- arnold networks in classification tasks. arXiv preprint arXiv:250117411

[21] Koenig BC, Kim S, Deng S (2024) Kan-odes: Kolmogorov–arnold network ordinary differential equations for learning dynamical systems and hidden physics. Computer Methods in Applied Mechanics and Engineering 432:117397

[22] Rigas S, Papachristou M, Papadopoulos T, et al (2024) Adaptive training of grid-dependent physics-informed kolmogorov-arnold networks. IEEE Access

[23] Wang Z, Zainal A, Siraj MM, et al (2025) An intrusion detection model based on convolutional kolmogorov-arnold networks. Scientific Reports 15(1):1917

[24] Mohri M, Rostamizadeh A, Talwalkar A (2018) Foundations of machine learning. MIT press

[25] Barcelo´ P, Monet M, Pe´rez J, et al (2020) Model interpretability through the lens of computational complexity. Advances in neural information processing systems 33:15487–15498

[26] Yarotsky D (2017) Error bounds for approximations with deep relu networks. Neural networks 94:103–114

[27] Schmidt-Hieber J (2021) The kolmogorov–arnold representation theorem revisited. Neural networks 137:119–126

[28] Ibrahum ADM, Shang Z, Hong JE (2024) How resilient are kolmogorov–arnold networks in classification tasks? a robustness investigation. Applied Sciences 14(22):10173

[29] Abueidda DW, Pantidis P, Mobasher ME (2025) Deepokan: Deep operator network based on kolmogorov arnold networks for mechanics problems. Computer Methods in Applied Mechanics and Engineering 436:117699

[30] Bogar E, Beyhan S (2020) Adolescent identity search algorithm (aisa): A novel metaheuristic approach for solving optimization problems. Applied Soft Computing 95:106503

[31] Cetin M, Bahtiyar B, Beyhan S (2019) Adaptive uncertainty compensation-based nonlinear model predictive control with real-time applications. Neural Computing and Applications 31:1029–1043

[32] Hayes MH (1996) Statistical digital signal processing and modeling. John Wiley & Sons

[33] Wolberg MStreet (1992) Breast cancer wisconsin (diagnostic) data set. Tech. rep., UCI Machine Learning Repository [http://archive. ics. uci. edu/ml/]

[34] Kaggle (2023) Pima indians diabetes database. Tech. rep., https://www.kaggle.com/datasets/uciml/pima-indians- diabetes-database (accessed Jun. 22, 2023)

[35] Hayes LA, O'Hara OS, Murray SA, et al (2021) Solar flare effects on the earth's lower ionosphere. Solar Physics 296(11):157

[36] Beyhan S (2023) A scaled-down model of a dairy barn to imitate a livestock building for modelling and control of environmental conditions. Journal of Thermal Biology 114:103571