A New Robust Molding of Heat and Mass Transfer Process in MHD Based on Adaptive-Network-Based Fuzzy Inference System

AHMAD A. ALHARBI[1], AMR R. KAMEL[2*], SAMAH A. ATIA[3]
[1]Department of Mathematics, Faculty of Science and Arts, Northern Border University, Arar, SAUDI ARABIA
[2]Department of Applied Statistics and Econometrics, Faculty of Graduate Studies for Statistical Research (FGSSR), Cairo University, Giza 12613, EGYPT
[2]Data Processing and Tabulation at Central Agency for Public Mobilization and Statistics (CAPMAS), Nasser City 2086, EGYPT
[3]Department of Mathematical Statistics, Faculty of Graduate Studies for Statistical Research (FGSSR), Cairo University, Giza 12613, EGYPT

Abstract:- This study concerns with the Process intensification deal with the complex fluids in mixing processes of many industries and its performance is based on the flow of fluid, magnetohydrodynamic (MHD) heat and mass transfer. This paper proposes a dynamic control model based on adaptive-network-based fuzzy inference system (ANFIS), weighted logistic regression and robust relevance vector machine (RRVM). Suitable similarity variables are applied to convert the flow equations into higher order ordinary differential equations and solved numerically. The surface-contour plots are utilized to visualize the influence of active parameters on velocity, thermal, nanoparticles concentration and motile microorganism's density. The hybrid-learning algorithm comprised of gradient descent and least-squares method is employed for training the ANFIS. A novel RRVM is presented to predict the endpoint. RRVM solves the problem of sensitivity to outlier characteristic of classical relevance vector machine (RVM), thus obtaining higher prediction accuracy. The key idea of the proposed RRVM is to introduce individual noise variance coefficient to each training sample. In the process of training, the noise variance coefficients of outliers gradually decrease so as to reduce the impact of outliers and improve the robustness of the model. To compare the proposed RRVM and other methods with outliers, the Monte Carlo simulation study has been performed. The simulation results showed that, based on mean squared error (MSE), mean absolute error (MAE), root mean squared error (RMSE) and coefficient of determination (R^2) criteria, the proposed RRVM give better performance than other methods when the data contain outliers. While when the dataset does not contain outliers, the results showed that the classical RVM is more efficient than other methods.

Key-Words:- ANFIS, Heat and Mass Transfer, MHD flow, Monte Carlo Simulation, Outliers, Robust Classification, Robust RVM, Sparsity, Weighted Logistic Regression.

Received: June 25, 2021. Revised: January 15, 2022. Accepted: February 26, 2022. Published: March 30, 2022.

1 Introduction

The magnetohydrodynamic (MHD) heat and mass transfer processes over a moving surface are of interest engineering and geophysical applications such as geothermal reservoirs, thermal insulation, enhanced oil recovery, packed-bed catalytic reactors, cooling of nuclear reactors. Many chemical engineering processes, such as metallurgy and polymer extrusion, require cooling a molten liquid as it is stretched into a cooling system; the fluid mechanical characteristics of the final product are mostly determined by the cooling liquid employed and the velocity of stretching. Some polymer fluids with higher electromagnetic characteristics, such as polyethylene oxide and polyisobutylene solution in cetane, are commonly employed as cooling liquids because their flow may be managed by external magnetic fields to improve the quality of the final product. Many transport processes in the industrial world include simultaneous heat and mass transfer as a result of the combined buoyancy effects of thermal diffusion and chemical species diffusion. This might be due to the fact that the research of combined heat and mass transfer is beneficial in a variety of technological transfer procedures. make a few attempts in this direction. The study of magnetic fields and the movement of electrically conducting fluids in porous media has raised significant concerns [1].

Nomenclature		Greek Symbols	
x, y	space coordinates	T_i	target value
u, v	velocity components	F_i	forecast value
$U_w(x,t)$	stretching sheet velocity	$\bar{T}$	average target value
$B(x,t)$	magnetic field strength	N	number of data
T_w	surface temperature	b	power law index
T_∞	ambient temperature	Γ	material time constant
N_w	surface motile microorganism density	ρ	fluid density
N_∞	ambient motile microorganism density	ν	kinematic viscosity
n	power law index	k^*	mean absorption coefficient
t	time	ξ	similarity variable
T	fluid temperature	l_1	velocity slip factor
C	nanoparticle volume fraction	l_2	thermal slip factor
N	density of motile microorganisms	l_3	concentration slip factor
C_w	surface nanoparticle concentration	l_4	microorganism slip factor
Wc	maximum cell swimming speed	R_r	fuzzy rule
C_∞	ambient concentration	$l(u)$	logistic loss function
q_r	radiative heat flux	Σ	variance matrix
c_p	specific heat	μ	mean value vector
D_B	Brownian diffusion coefficient	w_i	weight associated
D_T	thermophoresis diffusion coefficient	σ	kernel width
D_n	microorganism diffusion coefficient	α_j	unique hyperparameter individually
n	chemotaxis constant	σ^*	Stefan-Boltzmann constant

Toki and Tokis [2] study unstable free convection fluid flows that are incompressible and viscous near a porous infinite plate with arbitrary time dependent heating plate. Senapati et al. [3] published the results of chemical reactions of viscous fluids that are electrically conducting via a porous material in two-dimensional steady free convection flow along a vertical surface with slip flow area.

Moreover, the non-newtonian fluids and their properties play an important role in the intensification of mixing processes in a variety of sectors, including plastics, paper, rubber, food, and minerals. The carreau rheological model is a non-newtonian rheological model in which the constitutive relation holds for both high and low shear rates. Because of its numerous uses in engineering and technology, the Carreau fluid flow has gotten a lot of attention. Several researches on the heat and mass transfer properties of magneto Carreau nanofluids with diverse characteristics such as heat source/sink, thermal radiation, suction/injection, and changing thermal conductivity over a permeable/impermeable stretched sheet have been conducted, see [4,5]. Nanofluids improve heat transmission and can be used to improve the efficiency of heat exchangers and reactors. In nanofluids, bioconvection improves mass transfer, induces microvolume mixing, improves stability, and prevents nanoparticle clustering. Bio-nano cooling systems, microfluidic devices, enhanced energy conservation devices, medical filtration, and microbial fuel cell technologies are all possible uses of bioconvection phenomena in nanofluids.

Understanding MHD is inextricably linked to an understanding of the physical consequences that occur in MHD. Electric current is induced in the conductor as it travels into a magnetic field, and the conductor develops its own magnetic field. The magnetic field lines will be excluded from the conductor because the generated magnetic field seeks to eradicate the original and externally supported field. The induced field enhances the applied field when the magnetic field forces the conductor to move it out of the field. As a result of this procedure, the force lines appear to be pulled together with the conductor. The fluid with complicated movements is the conductor in this article. To comprehend the dynamical impact, we must first understand that when cur rents are created by a conducting fluid moving through a magnetic field, a Lorentz force acts on the fluid and alters its velocity. In MHD, movement affects the field and vice versa. As a result, the theory is significantly non-linear.

The data-processing techniques like artificial neural network (ANN), adaptive-network-based fuzzy inference system (ANFIS) and genetic algorithm (GA) attracted the researchers because of its applications in many non-linear systems. An ANFIS can assist us in determining the best

distribution of membership functions by determining the mapping relation between input and output data via hybrid learning. This inference system is made up of five levels. The node function describes numerous nodes in each tier. Fixed nodes, shown by circles, represent parameter sets that are fixed in the system, whereas adaptive nodes, denoted by squares, represent parameter sets that are modifiable in these nodes. The current layer's input will be the output data from the preceding levels' nodes.

To alleviate the above drawbacks, Tipping [6,7] proposed the relevance vector machine (RVM). The RVM is a Bayesian evidence-based nonlinear probabilistic model. To optimize the hyperparameters of the model and get a sparse solution, it employs the type-II maximum likelihood approach, often known as the "evidence process." For each of the model coefficients, an independent zero-mean Gaussian prior is assumed, as well as an independent Gamma hyper prior for each hyperparameter. After that, a training data set is used to determine posterior distributions of model coefficients and hyperparameters. Initially, the posterior distributions were calculated using the type-II maximum likelihood approach, which is an evidence procedure. A variational inference technique, which maximizes a variational lower bound on the marginal log likelihood, is an alternate strategy for the approximation.

The posterior distributions of several of the model coefficients are strongly peaked around zero, and so those coefficients may be omitted from the final model, thanks to the hierarchical prior structure known as automated relevance determination prior. As a result, we can have a sparse solution. The relevance vectors are the training observations with non-zero coefficient values. The support vector machine (SVM) is another common kernel-based learning technique that also delivers a sparse solution, see [8]. The support vectors in the SVM are the observations that contribute to the final decision boundary. In practice, the RVM offers significant benefits over the SVM. The number of relevance vectors is substantially fewer than the number of support vectors, resulting in a higher degree of sparsity. Second, it generates probabilistic results (e.g., class probability estimates). Finally, model complexity may be controlled automatically, without the need for an extra regularization parameter. However, RVM has a serious weakness that it assumes all of the training samples are coupled with independent Gaussian noise: $\varepsilon \sim N(0, \sigma^2)$. A well-known disadvantage with Gaussian noise model is that it is not robust. The accuracy of the RVM model will be considerably harmed if the training samples are polluted by outliers.

In this paper, a novel robust relevance vector machine (RRVM) is contrived, which posits that each training sample has its own coefficient of noise variance. To discover and eradicate outliers, the coefficients corresponding to outliers will be severely reduced throughout the model training method. To estimate the endpoint carbon content and temperature of molten steel, we use the suggested RRVM as an identifier. Measured data are frequently intermixed with outlying observations in MHD heat and mass transfer processes, although RRVM can lessen the impact of outliers and has strong generalization capacity. As a result, it is appropriate to build the endpoint prediction model.

The remainder of this paper is organized as follows: In Section 2, the literature review of MHD heat and mass transfer processes described. Section 3 presents the The mathematical formulation of the problem. Section 4 introduces the methods ANFIS, Weighted logistic regression with Transformation of the logistic function and RRVM utilized in this paper. RRVM for classification using variational inference are given in Section 5. Section 6 contains the Monte Carlo simulation study. In Section 7, the conclusions are drawn.

2 Modeling Studies: literature review

There is a growing body of research in the topic of nanofluids, and multiple examinations of their thermal conductivities have been carried out to assess the impact of various factors. While experimental work necessitates a significant investment in a well-equipped laboratory and appropriate instruments, which is a significant barrier for some scholars, predictive approaches are increasingly popular for a faster and less expensive view of various influential parameters on desired parameters. Actually, predicting the impact of thermal conductivity of nanofluids is quite difficult, and this has been a focus of intense research for scientists.

Naveed et al. [9] examined MHD BL (boundary layer) unsteady flow above curved stretching surface. Abbas et al. examined numerically radiation impacts on MHD flow above curved stretching surface of nanofluid by assimilating the slip, collective radiation and heat generation effects. Sahoo [10] investigated the mass and heat transfer in MHD flow of viscoelastic fluid via porous media bounded by vacillating plate in slip flow system.

Singh et al. 11 have inspected mass transfer and heat in MHD flow or viscous fluid past a straight up plate in oscillatory velocity suction. Noor et al. [11] used the shooting approach to investigate MHD flow on an inclined surface with heat source/sink effects. MHD fluid flow across a spinning disc was researched by Turkyilmazoglu [12]. He analysed the viscous dissipation and Joule heating components using a spectral numerical integration approach. Chen [13] studied heat and mass transport in MHD free convective flow with Ohmic heating and viscous dissipation using a numerical technique.

In recent years, statistical learning theory has been rapidly developed. It is based on the notion of structural risk reduction and focuses on managing the generalization ability of the learning process, see [14]. The SVM was created based on this notion. It improves processing capabilities by translating data into a high-dimensional space and employing kernel functions. In addition, Müller et al. [15] established a regularization parameter C to adjust the trade-off between model complexity and training error. As a result, SVM has shown to be an effective tool for identifying non-linear systems, with several successful applications, see [16]. SVM has also performed well in the application of steelmaking process control. To forecast the endpoint parameters of electric arc furnace steelmaking, Yuan et al. [17] combined multiple support vector machines with principal component regression. Valyon and Horvth [18] suggested a sparse and robust extension of least-square SVM (LS-SVM) for calculating the quantity of oxygen blasted in Basic oxygen furnace (BOF) steelmaking, and showed that LS-SVM outperformed ANNs. Despite its popularity, however, SVM has a number of major and practical drawbacks. Predictions, for example, are not probabilistic, hence the kernel function must meet Mercer's requirement. Cross validation is required to estimate the error/margin trade-off parameter C, which takes a long time. Furthermore, despite the fact that SVM is a sparse model, the number of support vectors rises linearly with the size of the training sample set. These drawbacks limit the scope of SVM future uses.

On the other hand, several projects have been undertaken to construct robust kernel-based learning algorithms, see Hwang et al. [19,20]. The robust truncated hinge loss SVM was proposed by Wu and Liu [21]. They used the difference convex approach to solve the nonconvex problem through a series of convex sub-problems because the underlying optimization problem comprises nonconvex minimization. However, because they were created using the SVM technique, these studies are unable to provide statistical information such as a class probability. For the logistic regression, Park and Liu [22] used a truncated logistic loss function to remove the effect of outlying observation. Despite the fact that this study can estimate the class probability, it does not provide a sparse solution.

Furthermore, if a dataset contains outliers, a decision boundary derived from the RVM may be severely warped. Because data sets with outliers are regularly encountered in practice, a robust learning algorithm for the RVM that is insensitive to outliers is sought. In this paper, the influence of an outlier on the decision boundaries from the SVM (dotted line), RVM (dashed line), and the new approach, which is dubbed the RRVM (full line), is illustrated using a simulated dataset example in Figures 1-2. Figure 1 represents the decision boundaries obtained by employing the linear kernel, while Figure 2 displays the decision boundaries obtained by radial basis function (RBF) kernel with $\sigma = 2$. For the SVM, the regularization constant C is set to 1. From the figures, it is observed that the decision boundaries from the SVM and RVM are pulled toward to the outlier regardless of the type of kernels.

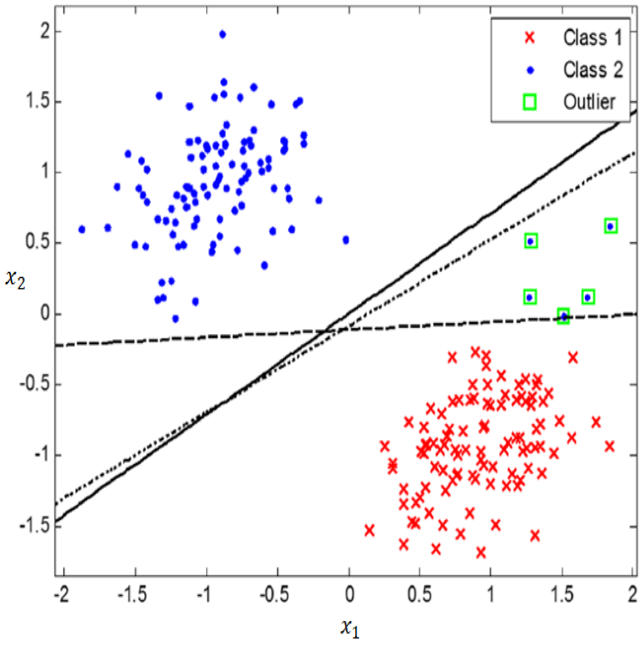

Fig. 1 A simulated dataset with outliers: plots of the decision boundaries from SVM, RVM and RRVM by employing the linear kernel.

An adaptive-network-based fuzzy inference system (ANFIS) is used to generate the values of these control variables, which is based on operator control experience and production data from a steel factory. ANFIS can learn from a set of input-output data and offers competitive computation accuracy.

Combining ANFIS with RRVM, a dynamic control model of MHD heat and mass transport processes is created. In order to achieve the intended control effect, the RRVM model must be well-trained as an identifier to approximate the link between input and output and correctly anticipate the endpoint carbon content and temperature. The simulations in the final section of this paper will demonstrate that RRVM has a high degree of approximation ability and robustness.

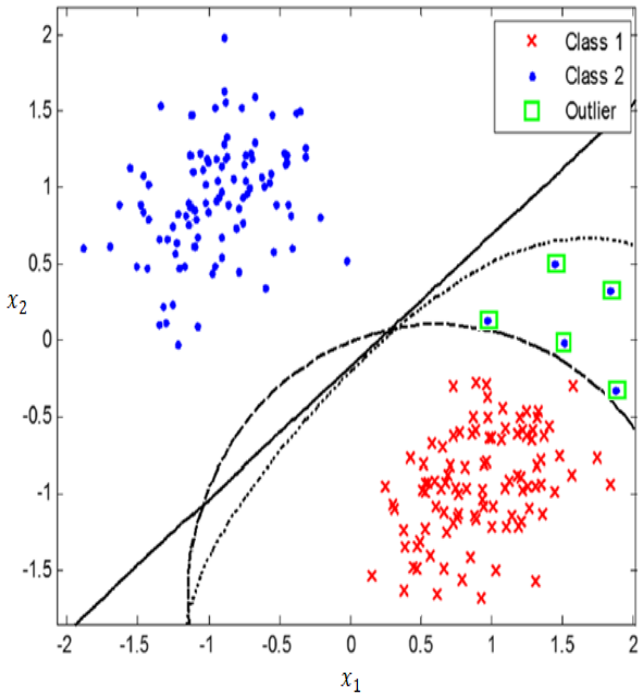

Fig. 2 A simulated dataset with outliers: plots of the decision boundaries from SVM, RVM and RRVM obtained by RBF kernel with $\sigma = 2$.

Finally, a trimmed relevance vector machine (TRVM) was suggested by Yuan et al. [17], which redefined the likelihood function as a trimmed one. During model training, outliers are removed, and a weighted technique is used to determine the trimmed subset. The new technique can detect outliers and improve the model's robustness. Many robust methods are discussed by many papers in several models, see e.g. [23-26].

3 The Mathematical Formulation of the Problem

In this section, our new robust modeling approach is described. Initially, the description on how the MHD heat and mass transfer processes are handled is presented. The adaptation to convective effects is also included.

3.1 Modeling Description

Consider an unsteady 2D flow of a magnetohydrodynamic Carreau nano-fluid containing gyrotactic micro-organisms influenced by a slendering stretching surface in the presence of thermal radiation and multiple slips. The heat transfer and mass transfer features are examined with the effects of Brownian motion and thermophoresis.

The slendering sheet is stretched in the x-direction with velocity $U_w(x,t) = U_0(x+b)^m/1-ct$ and y-axis is normal to the flow, see Figure 3. The surface is assumed to be impermeable ($v_w = 0$) with the thickness $y = A(x+b)^{\frac{1-m}{2}}$ where $m \neq 1$. A uniform magnetic field of strength $B(x,t) = B_0\sqrt{\frac{(x+b)^{m-1}}{1-ct}}$ is imposed in the direction transverse to the flow. The temperature $T_w(x,t)$, nanoparticle concentration $C_w(x,t)$, and density of motile microorganisms $N_w(x,t)$, at the stretching sheet are assumed to be greater than the ambient values T_∞, C_∞ and N_∞, respectively.

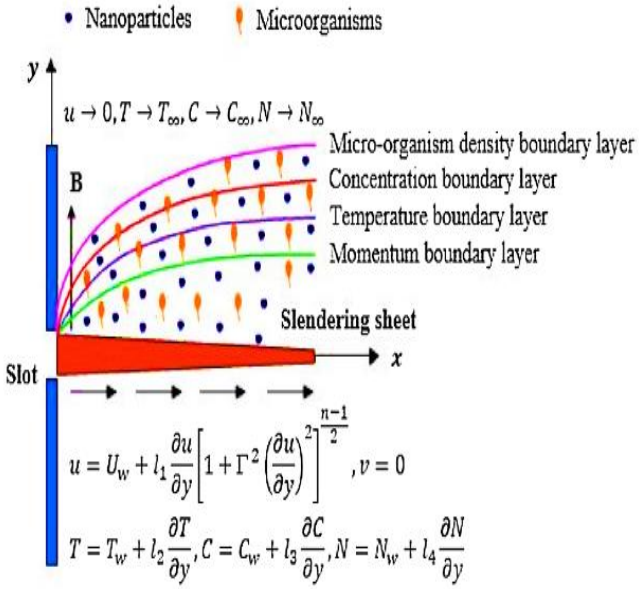

Fig. 3 Schematic form of the physical model

3.2 Boundary Conditions and Governing Equations

Based on the foregoing assumptions, the governing equations for mass, momentum, energy, nanoparticle concentration, and microorganisms are as follows:

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \qquad (1)$$

$$\frac{\partial u}{\partial t}+u\frac{\partial u}{\partial x}+v\frac{\partial u}{\partial y}= \nu\frac{\partial^2 u}{\partial y^2}\left[1+\Gamma^2\left(\frac{\partial u}{\partial y}\right)^2\right]^{\frac{n-1}{2}}+\nu(n-1)\Gamma^2\frac{\partial^2 u}{\partial y^2}\left(\frac{\partial u}{\partial y}\right)^2\times \left[1+\Gamma^2\left(\frac{\partial u}{\partial y}\right)^2\right]^{\frac{n-1}{2}}-\frac{\sigma B^2(x,t)}{\rho}u \quad (2)$$

$$\frac{\partial T}{\partial t}+u\frac{\partial T}{\partial x}+v\frac{\partial T}{\partial y}= \alpha_m\frac{\partial^2 T}{\partial y^2}+\tau\left[\left(\frac{D_T}{T_\infty}\right)\left(\frac{\partial T}{\partial y}\right)^2+D_B\frac{\partial T}{\partial y}\frac{\partial C}{\partial y}\right]-\frac{1}{\rho c_p}\frac{\partial q_r}{\partial y} \quad (3)$$

$$\frac{\partial C}{\partial t}+u\frac{\partial C}{\partial x}+v\frac{\partial C}{\partial y}=D_B\frac{\partial^2 C}{\partial y^2}+\left(\frac{D_T}{T_\infty}\right)\frac{\partial^2 T}{\partial y^2} \quad (4)$$

$$\frac{\partial N}{\partial t}+u\frac{\partial N}{\partial x}+v\frac{\partial N}{\partial y}=D_n\frac{\partial^2 N}{\partial y^2}-\frac{bW_c}{\Delta C}\left[\frac{\partial}{\partial y}\left(N\frac{\partial C}{\partial y}\right)\right] \quad (5)$$

The problem boundary conditions are defined as follows:

$$u=U_w(x,t)+l_1\frac{\partial u}{\partial y}\left[1+\Gamma^2\left(\frac{\partial u}{\partial y}\right)^2\right]^{\frac{n-1}{2}},$$
$$v=0, T=T_w(x,t)+l_2\frac{\partial T}{\partial y}, C$$
$$=C_w(x,t)+l_3\frac{\partial C}{\partial y}, N=N_w(x,t)+l_4\frac{\partial N}{\partial y}aty$$
$$=A(x+b)^{\frac{1-m}{2}}u\to 0, T\to T_\infty, C\to C_\infty, N\to N_\infty \text{ as } y\to\infty \quad (6)$$

where (u,v) denotes the components of velocity along (x,y) directions, t is the time, T, C and N represent the fluid temperature, nanoparticles volume fraction and motile micro-organisms density, Γ and n stand for material time constant and power law index, ρ, ν, σ and α_m are the density, kinematic viscosity, electrical conductivity and thermal diffusivity, c_p is the specific heat, $\tau=(\rho c)_p/(\rho c)_f$ where $(\rho c)_p$ is effective heat capacity of nanoparticles and $(\rho c)_f$ is heat capacity of base fluid, D_B, D_T and D_n indicate the Brownian, thermophoretic and microorganism diffusion coefficients, b and W_c are the chemotaxis constant and maximum cell swimming speed (bW_c - constant), l_1, l_2, l_3 and l_4 signify the velocity, thermal, concentration and microorganism slip factors.

$$T_w(x,t)=T_\infty+T_0\sqrt{\frac{(x+b)^{1-m}}{1-ct}},$$

$$C_w(x,t)=C_\infty+C_0\sqrt{\frac{(x+b)^{1-m}}{1-ct}}, N_w(x,t)$$
$$=N_\infty+N_0\sqrt{\frac{(x+b)^{1-m}}{1-ct}} \quad (7)$$

However, the heat flux q_r can be expressed as;

$$q_r=-\frac{4\sigma^*}{3k^*}\frac{\partial T^4}{\partial y}=-\frac{16\sigma^* T_\infty^3}{3k^*}\frac{\partial T}{\partial y} \quad (8)$$

where σ^* denote Stefan-Boltzmann constant and k^* is mean absorption coefficient, respectively.

In make use of Eq. (8), the Eq. (3) reduced to

$$\frac{\partial T}{\partial t}+u\frac{\partial T}{\partial x}+v\frac{\partial T}{\partial y}= \left(\alpha_m+\frac{16\sigma^* T_\infty^3}{3k^*\rho c_p}\right)\frac{\partial^2 T}{\partial y^2}+\tau\left[\left(\frac{D_T}{T_\infty}\right)\left(\frac{\partial T}{\partial y}\right)^2+D_B\frac{\partial T}{\partial y}\frac{\partial C}{\partial y}\right] \quad (9)$$

4. Methods Utilized

In this section, the methods utilized in the dynamic control model, such as ANFIS, Weighted logistic regression with Transformation of the logistic function, and RRVM, will be presented.

4.1 Adaptive-Network-Based Fuzzy Inference System

Adaptive-network-based fuzzy inference system (ANFIS) is an off-line learning model. It's a type of artificial neural network that uses the Takagi–Sugeno fuzzy inference system as its foundation. In the early 1990s, the approach was developed. It has the potential to capture the benefits of both neural networks and fuzzy logic principles in a single framework because it integrates both. Its inference system is made up of a set of fuzzy IF–THEN rules with the capacity to approximate nonlinear functions through learning. As a result, ANFIS is regarded as a universal estimator. By building a collection of fuzzy if-then rules with appropriate membership functions, it has been widely employed in the modeling and control of nonlinear systems, see [27]. Generally, an ANFIS model consists of five layers. The architecture is shown in Figure 4.

The fuzzy rules extracted from input–output pairs are described as;

R_r: if x_1 is $A_1^{S_1}$ and x_2 is $A_2^{S_2}$ ... and x_n is $A_n^{S_n}$

Then $f_r=f_r(x_1,x_2,\dots,x_n),\ r=1,\dots,K$ (10)

where R_r denotes the r^{th} fuzzy rule, and $\mathrm{A}_1^{\mathrm{S}_1},\dots,\mathrm{A}_\mathrm{n}^{\mathrm{S}_\mathrm{n}}$ are the fuzzy sets associated with the input variables $x_1,\dots,x_n$. Function $f_r=f_r(x_1,x_2,\dots,x_n)$ is the output of the r^{th} fuzzy rule. The different functions of five layers are described as follows:

Layer 1: Input variables are fuzzificated and the membership of $x_l(l=1,\dots,n)$ on different fuzzy sets are calculated according to formula;

$$\mu_l^{s_l} = \mu_{A_1^{s_l}}(x_l) \quad (11)$$

where $\mu_{A_1^{s_l}}(\cdot)$ denotes the membership function of variable x_l on fuzzy sets $A_l^{S_l}$ and $\mu_l^{s_l}$ is the membership degree.

Layer 2: Calculate the confidence degrees of fuzzy rules. As for the r^{th} fuzzy rule, the degree of confidence is calculated as formula;

$$\omega_r = \mu_1^{s_1} \cdot \mu_2^{s_2} \cdots \mu_n^{s_n},\ r = 1, \dots, K \quad (12)$$

Layer 3: All of the confidence degrees are normalized as:

$$\bar{\omega}_r = \omega_r / \left(\sum_{p=1}^{K} \omega_p\right), r = 1, \dots, K \quad (13)$$

Layer 4: Calculate the output of each fuzzy rule according to formula (14). Here Takagi–Sugeno type fuzzy rules are adopted.

$$f_r = p_1^r x_1 + p_2^r x_2 + \cdots + p_n^r x_n + q^r \quad (14)$$

where $p_1^r, p_2^r, \dots, p_n^r$ and q^r are fuzzy consequent parameters which can be determined based on least-square regression.

Layer 5: Calculate the final output of ANFIS. It is the weighted summarization of f_r, and the weight is $\bar{\omega}_r (r = 1, \dots, K)$.

$$y = \sum_{r=1}^{K} \bar{\omega}_r f_r \quad (15)$$

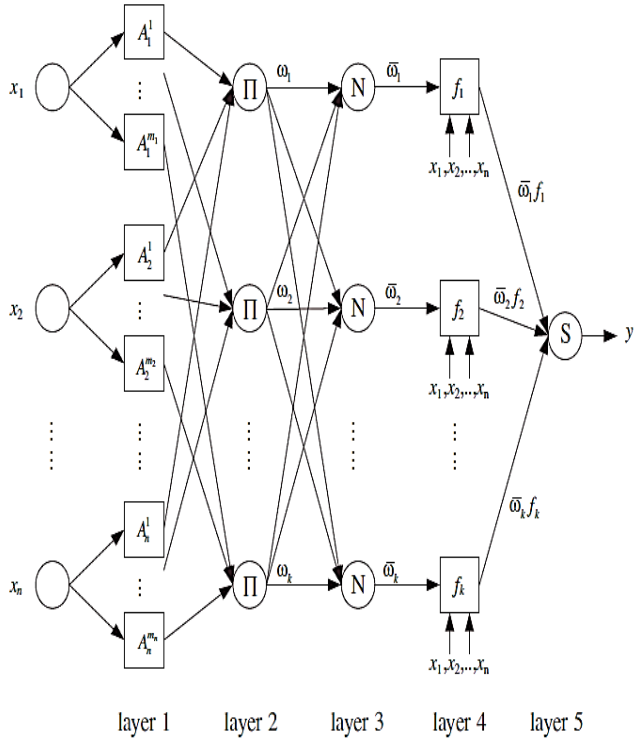

Fig. 4 The architecture of ANFIS

4.2 Weighted Logistic Regression

In this section, we briefly describe a standard logistic regression and its weighted version for achieving the robustness. Consider a data set of N input-target pairs $\{x_i, t_i\}_{i=1}^N$, where x_i represents a d-dimensional input vector and t_i represents class labels: $t_i = 0$ if the i$^{\text{th}}$observation belongs to the first class and $t_i = 1$ if it belongs to the second class. A decision boundary can be defined as a linear combination of M basis functions as follows:

$$f(\phi(x)) = \beta^T \phi(x) = \beta_0 + \sum_{i=1}^{M} \beta_i \phi_i(x) \quad (16)$$

where $\beta = (\beta_0, \beta_1, \dots, \beta_M)^T$ is a vector of model coefficients and $\phi(x) = (1, \phi_1(x), \dots, \phi_M(x))^T$ is a vector of basis functions. By employing some nonlinear basis functions, the decision boundary $f(\phi(x))$ becomes a nonlinear function with respect to x. Some commonly used basis functions are the polynomial kernel, $\phi_i(x) = (1 + \langle x, x_i \rangle)^d$, where the parameter d is the degree of polynomial to be used, and the Gaussian RBF kernel;

$$\phi_i(x) = exp\left\{-\frac{(x-x_i)^T(x-x_i)}{\sigma}\right\},$$

where the parameter σ is the kernel wid th.

In a standard logistic regression, the conditional distribution for t is given by;

$$p(t \mid \beta) = \sigma(\beta^T \phi(x))^t \{1 - \sigma(\beta^T \phi(x))\}^{1-t}$$

where $\sigma(u)$ is the logistic function defined as $\sigma(u) = 1/1 + e^{-u}$. Assuming independent and identically distributed data, the likelihood function can be written as;

$$p(t \mid \beta) = \prod_{i=1}^{N} p(t_i \mid \beta) = \prod_{i=1}^{N} \sigma\left(\beta^T \phi(x_i)\right)^{t_i} \left\{1 - \sigma\left(\beta^T \phi(x_i)\right)\right\}^{1-t_i} \quad (17)$$

The model coefficients β can be estimated by the maximum likelihood approach which can be formulated as the following optimization problem in the loss function framework:

$$\min_{\beta} \sum_{i=1}^{N} l\left\{(2t_i - 1) f\left(\phi(x_i)\right)\right\} \quad (18)$$

where $l(u) = ln\ 1 + e^{-u}$ denotes the logistic loss function. It should be noted that the solution of minimizing the sum of loss functions is equivalent to that of maximizing the log likelihood function, that is;

$$\min_{\beta} \sum_{i=1}^{N} l\{(2t_i - 1)f(\phi(x_i))\} \Leftrightarrow$$
$$\max_{\beta} ln\, p(t \mid \beta) \Leftrightarrow \max_{\beta} \sum_{i=1}^{N} ln\, p(t_i \mid \beta) \quad (19)$$

To obtain a robust classification result, a weighting strategy can be employed to the standard logistic regression model in the loss function framework as follows;

$$\min_{\beta} \sum_{i=1}^{N} w_i l\{(2t_i - 1)f(\phi(x_i))\} \quad (20)$$

where w_i is a weight associated with the i^{th} observation. If a small weight is given to an outlying observation, the effect of an outlier can be reduced and therefore a robust decision boundary can be obtained. Then, one question is raised: how the concept of a weighted loss can be transformed into the maximum likelihood approach. From Eq. (19), the following relationship can be obtained:

$$\min_{\beta} \sum_{i=1}^{N} w_i l\{(2t_i - 1)f(\phi(x_i))\} \Leftrightarrow$$
$$\max_{\beta} \sum_{i=1}^{N} w_i\, ln\, p(t_i \mid \beta) \Leftrightarrow$$
$$\max_{\beta} \sum_{i=1}^{N} ln\, p(t_i \mid \beta)^{w_i} \quad (21)$$

Therefore, the concept of a weighted loss can be dealt with in the maximum likelihood approach by replacing $p(t_i \mid \beta)$ with $p(t_i \mid \beta)^{w_i}$.

To avoid the overfitting problem while considering a complex model, the regularization concept has been used in machine learning. By employing the regularization concept to the original logistic regression, the formulation in Eq. (18) can be extended as follows;

$$\min_{\beta} \sum_{i=1}^{N} l\{(2t_i - 1)f(\phi(x_i))\} + \lambda J(f) \quad (22)$$

where $\lambda > 0$ is a regularization parameter which controls the smoothness of a decision boundary and $J(f)$ denotes a regularization term which represents a penalty for a complex decision boundary.

4.3 Classical Relevance Vector Machine

The relevance vector machine (RVM) is a Bayesian-based probabilistic model. Consider the training samples to be a data collection of input-target pairs $\{x_i; t_i\}_{i=1}^{N}$, where $x_i \in R^n$ signifies an n-dimensional input vector and $t_i \in R^n$ denotes a scalar-measured output. Assume that the objectives are sampled separately from the regression model with extra noise ε_i as follows:

$$t_i = y(x_i; w) + \varepsilon_i \quad (23)$$

where ε_i is assumed to be the mean-zero Gaussian noise with variance σ^2, namely $\varepsilon_i \sim N(\varepsilon_i|0, \sigma^2)$. Similar to SVM, the prediction function $y(x;\ w)$ of RVM is defined as a linear combination of the weighted basis functions:

$$y(x; w) = \sum_{i=1}^{N} w_i K(x, x_i) + w_0 \quad (24)$$

where $K(x, x_i)$ is a basis function, effectively define one basis function for each sample in training data set. The weight parameter vector is defined as $w = [w_0, \dots, w_N]^T$. According to Eq. (23) and the noise assumption of ε_i, we have the Gaussian distribution over t_i with mean $y(x_i; w)$ and variance σ^2, viz., $p(t_i \mid x_i) = N(t_i \mid y(x_i; w), \sigma^2)$. For convenience, a hyperparameter β is defined as $\beta = 1/\sigma^2$. Therefore, the likelihood function of the complete training data set is expressed as;

$$p(t \mid w, \beta) = \left(\frac{\beta}{2\pi}\right)^{N/2} exp\left\{-\frac{\beta}{2} \parallel t - \Phi w \parallel^2\right\} \quad (25)$$

where $t = [t_1, t_2, \dots, t_N]^T$ and $\Phi \in R^{N\times(N+1)}$ defined as $\Phi = [\phi(x_1),\ \phi(x_2), \dots, \phi(x_N)]^T$, which is called design matrix. The definition of $\phi(x_i)$ is

$$\phi(x_i) = [1, K(x_i, x_1), K(x_i, x_2), \dots, K(x_i, x_N)]^T \quad ; i = 1, \dots, N.$$

The goal of RVM training is to figure out what the posterior distribution is over the weight vector w. The prior distribution over w_j $(j = 0, \dots, N)$ should be determined first in order to keep the likelihood function sparse and optimize it. Assume that w_j follows a Gaussian distribution with mean zero and variance a α_j^{-1}, thus the previous distribution over w is;

$$p(w \mid \alpha) = \prod_{j=0}^{N} N\left(w_j \mid 0, \alpha_j^{-1}\right) \quad (26)$$

where α_j is the unique hyperparameter individually associated with each weight parameter w_j in a multivariate Gaussian distribution, and $\alpha = [\alpha_0, \alpha_1, \dots, \alpha_N]^T$ The posterior distribution over w may be estimated using the Bayesian rule and the defined prior distribution Eq. (26) and likelihood function Eq. (25).

$$p(w \mid t, \alpha, \beta) = \frac{p(w|\alpha)p(t|w,\beta)}{p(t|\alpha,\beta)} \quad (27)$$

Since $p(w \mid \alpha)$ and $p(t \mid w, \beta)$ are all Gaussian, the product of these two distributions is also Gaussian. Furthermore, $p(t \mid \alpha, \beta)$ does not include w, so it is considered as a normalization coefficient.

The posterior distribution over w is also Gaussian and can be expressed as:

$$p(w \mid t, \alpha, \beta) = N(w \mid \mu, \Sigma) \quad (28)$$

where μ is the mean value vector and Σ is the variance matrix, which are expressed as formulas (29) and (30), respectively:

$$\Sigma = (\beta \Phi^T \Phi + A)^{-1} \quad (29)$$

$$\mu = \beta \Sigma \Phi^T t \quad (30)$$

where $A = \text{diag}(\alpha_0, \alpha_1, \ldots, \alpha_N)$. The posterior distribution over w are determined by hyperparameters β and α, thus the hyperparameters are optimized by using evidence procedure. The iterative optimization formulas for hyperparameters are;

$$\alpha_j = 1/\left(\mu_j^2 + \Sigma_{jj}\right) = \gamma_j/\mu_j^2,\ j = 0,1,\ldots,N \quad (31)$$

$$\beta = N - \sum_{j=0}^{N} \gamma_j / \| t - \Phi\mu \| \quad (32)$$

where μ_j denotes the j th ellement of vector μ and Σ_{jj} denotes the j th diagonal element of matrix $\Sigma, \gamma_j = 1 - \alpha_j \Sigma_{ij}$. In the process of training, Equations (13) – (16) are calculated iteratively. Most of α_j tend to ward infinity and the corresponding μ_j will tend toward zero. The training stops until all the hyperparameters are convergent or the maximum number of iterations is reached.

Classic RVM is based on the assumption that each training samples noise ε_i is a mean-zero Gaussian distribution with the same variance σ^2 (or hyperparameter β). Measured data is usually tainted by outlying observations in actual applications, making the Gaussiannoise assumption unsustainable. This will weaken the RVM regression model's resilience and diminish its prediction accuracy. To alleviate this problem, researchers have proposed some modified methods. Tipping and Lawrence [28] improved RVM by using the Student-t noise model, which had a larger tail distribution than the Gaussian noise model. The updated technique, on the other hand, was developed using variational approximation, which takes longer to compute.

4.4 Proposed New Modeling

In this section, our new modeling approach is described. Initially, the description on how the MHD heat and mass transfer processes are handled is presented. The above modified strategies are mainly based on variational inference or trimming data set. A proposed modeling of robust relevance vector machine (RRVM) is presented to reduce the impact of outliers and the model can still be implemented by using evidence procedure. Rather than using the same noise variance for all samples, we assume that each training sample has its own noise variance coefficient. The iteration formulae are then deduced using the Bayesian evidence framework to maximize the hyperparameters and noise variance coefficients. Outliers noise variance coefficients will decrease during the optimization process, allowing outliers to be detected and eliminated. The following is a full description of the optimization technique.

In reference to Bayesian weighted linear regression,Ting et al. [29] assume that the individual noise distribution of the i^{th} training sample is:

$$p(\varepsilon_i) = N(\varepsilon_i|0, \sigma^2/\beta_i), \quad i = 1, \ldots, N \quad (33)$$

where σ^2denotes the average variance of all the training samples and β_i denotes the noise variance coefficient of the i^{th} sample. The prior distribution of β_i is assumed to be Gamma distribution, namely

$$p(\beta_i) = \text{Gamma}(a_i, b_i) = \Gamma(a_i)^{-1} b_i^{a_i} \beta_i^{a_i - 1} e^{-b_i \beta_i}$$

with "*gamma function*" $\Gamma(a_i) = \int_0^\infty t^{a_i - 1} e^{-t} dt$. Define the vector $\beta = [\beta_1, \beta_2, \ldots, \beta_N]^T$ and the likelihood function of the complete training sample set will change from Eq. (25) to;

$$p(t \mid w, \beta, \sigma^2) = \quad (2\pi\sigma^2)^{-N/2} |B|^{1/2}$$

$$\times exp\left\{-\frac{1}{2\sigma^2}(t - \Phi w)^T B (t - \Phi w)\right\} \quad (34)$$

where $B = diag(\beta_1, \beta_2, \ldots, \beta_N)$, and $|\cdot|$ is the determinant of matrix. The definitions of t, w and Φ are the same as before. The prior distribution over w is still expressed as Eq. (26). According to Bayesian rule, the posterior distribution of w is computed as;

$$p(w \mid t, \alpha, \beta, \sigma^2) = \frac{p(w|\alpha) p(t|w, \beta, \sigma^2)}{p(t|\alpha, \beta, \sigma^2)} = N(w \mid \mu, \Sigma)$$

where the variance matrix Σ and mean value vector μ can be computed by using following formulas;

$$\Sigma = (A + \sigma^{-2} \Phi^T B \Phi)^{-1}$$
$$= \left(A + \sigma^{-2} \sum_{i=1}^{N} \beta_i \phi(x_i) \phi(x_i)^T\right)^{-1} \quad (35)$$

$$\mu = \sigma^{-2} \Sigma \Phi^T B t$$
$$= \sigma^{-2} \Sigma \left(\sum_{i=1}^{N} \beta_i \phi(x_i) t_i\right) \quad (36)$$

Since the computation formulas of variance matrix and mean value vector are both influenced by α , β and σ^2, these hyperparameters need to be optimized so as to maximize the posterior distribution of w. The marginal likelihood function is computed as follows:

$$p(t \mid \alpha, \beta, \sigma^2) = \int p(t \mid w, \beta, \sigma^2) p(w \mid \alpha) dw$$

$$= (2\pi)^{-N/2} |C|^{-1/2} exp\left\{-\frac{1}{2} t^T C^{-1} t\right\} \quad (37)$$

where $C = \sigma^2 B^{-1} + \Phi A^{-1} \Phi^T$. Equivalently, we can optimize the logarithm of the product of $p(t \mid \alpha, \beta, \sigma^2)$ and $p(\beta)$. Moreover, we maximize this quantity with respect to $\log \alpha, \log \beta$ and $\log \sigma^2$ for convenience of computing. Therefore, the objective to be optimized is;

$$\log p(t \mid \log \alpha, \log \beta, \log \sigma^2) + \sum_{i=1}^{N} \log p(\log \beta_i)$$

Note that $p(\log \beta_i) = \beta_i \cdot p(\beta_i)$ and delete the terms which are independent of α, β and σ^2, we get the objective function;

$$L = -\frac{1}{2}[-\log|\Sigma| - \log|A| + N\log \sigma^2 - \log|B| + \mu^T A\mu + \sigma^{-2}(t - \Phi\mu)^T B(t - \Phi\mu)] + \sum_{i=1}^{N} (a_i \log \beta_i - b_i \beta_i) \quad (38)$$

The optimized value of α, β and σ^2 cannot be obtained in closed form, and have to be re-estimated iteratively. Take the partial derivative of Eq. (25) with respect to $\log \alpha_j (j = 0,1, \dots, N)$, $\log \beta_i (i = 1, \dots, N)$ and $\log \sigma^2$, and rearrange the equations to obtain the iteration formulas of α, β and σ^2 as following;

$$\alpha_j = \frac{1}{\mu_j^2 + \Sigma_{jj}} = \frac{\gamma_j}{\mu_j^2} \quad (39)$$

$$\beta_i = \frac{a_i + 0.5}{b_i + 0.5 \cdot [\sigma^{-2}(t_i - \phi(x_i)^T \mu)^2 + \sigma^{-2} tr(\Sigma \phi(x_i) \phi(x_i)^T)]} \quad (40)$$

$$\sigma^2 = \frac{(t - \Phi\mu)^T B (t - \Phi\mu)}{N - \sum_{j=0}^{N} \gamma_j} \quad (41)$$

where $j = 0, \dots, N, i = 1, \dots, N. \Sigma_{jj}$ is the j th diagonal element of variance matrix $\Sigma, \gamma_j = 1 - \alpha_j \Sigma_{ij}$ and $tr(\cdot)$ denotes the trace of matrix. Finally the iterative formulas for optimization are all obtained. Formulas (35), (36), (39), (40) and (41) are the iterative estimations of Σ, μ and hyperparameters $\alpha_j, \beta_i, \sigma^2$, respectively.

5. RRVM for Classification Using Variational Inference

In classification, it is not possible to directly seek the posterior distributions over the model coefficients since the logistic likelihood function is not suitable to be combined with a Gaussian prior. To resolve this issue, Jaakkola and Jordan [30] introduced a transformed logistic function that is quadratically dependent on the model coefficients in the exponent and used it to assess a logistic regression model with a Gaussian prior over the model coefficients in a Bayesian framework. Bishop and Tipping [31] used these findings to develop an alternate training procedure for the RVM in the context of variational inference.

The following is a lower bound on the logistic function with the functional form of a Gaussian, see [30]. To begin, decompose the log of the logistic function σ(u) as follows:

$$ln\, \sigma(u) = -ln(1 + e^{-u})$$

$$= \frac{u}{2} - ln\left(e^{u/2} + e^{-u/2}\right) \quad (42)$$

Note that the function $f(u) = -ln\, e^{u/2} + e^{-u/2}$ is a convex function with respect to the variable u^2. Since a tangent surface to a convex function is a global lower bound for the function, the global lower bound on $f(u)$ can be obtained with a first order Taylor expansion in the variable u^2 at the point ξ (called a variational parameter in the variational inference framework). That is;

$$f(u) \geq f(\xi) + \frac{\partial f(\xi)}{\partial(\xi^2)}(u^2 - \xi^2)$$

$$= -\frac{\xi}{2} + ln\, \sigma(\xi) + \frac{1}{4\xi} tanh\left(\frac{\xi}{2}\right)(u^2 - \xi^2).$$

Combining this lower bound on $f(u)$ with Eq. (42), the lower bound on the logistic function can be obtained as;

$$\sigma(u) \geq \sigma(\xi) \exp\left\{\frac{u - \xi}{2} - \lambda(\xi)(u^2 - \xi^2)\right\} \quad (43)$$

where $\lambda(\xi) = \frac{1}{4\xi} \tanh\left(\frac{\xi}{2}\right) = \frac{1}{2\xi}\left\{\sigma(\xi) - \frac{1}{2}\right\}$.

The bound has the form of the exponential quadratic function of u, which makes the Bayesian approach analytically tractable.

Again, the conditional distribution for t_i can be written as;

$$p(t_i \mid \beta) = \sigma\left(\beta^T\phi(x_i)\right)^{t_i}\left\{1-\sigma\left(\beta^T\phi(x_i)\right)\right\}^{1-t_i}$$
$$= \left(\frac{1}{1+e^{-\beta^T\phi(x_i)}}\right)^{t_i}\left(1-\frac{1}{1+e^{-\beta^T\phi(x_i)}}\right)^{1-t_i}$$
$$= e^{\beta^T\phi(x_i)t_i}\sigma\left(-\beta^T\phi(x_i)\right)$$

Then, the following relationship holds due to Eq. (43):

$$p(t_i \mid \beta) = e^{\beta^T\phi(\mathrm{x}_i)t_i}\sigma\left(-\beta^T\phi(\mathrm{x}_i)\right)$$
$$\geq \sigma(\xi_i)\exp\left\{\beta^T\phi(\mathrm{x}_i)t_i - \frac{\beta^T\phi(\mathrm{x}_i)+\xi_i}{2} - \lambda(\xi_i)\left(\left(\beta^T\phi(\mathrm{x}_i)\right)^2-\xi_i^2\right)\right\}$$
$$\equiv h(\beta,\xi_i) \quad (44)$$

Therefore, the likelihood function can be written as;

$$p(\mathrm{t}\mid\beta) = \prod_{i=1}^{N} p(t_i\mid\beta) \geq \prod_{i=1}^{N} h(\beta,\xi_i)$$

Consequently, from Eq. (21), the modified likelihood function to downweight outliers is given by;

$$p(\mathrm{t}\mid\beta,\mathrm{w}) = \prod_{i=1}^{N} p(t_i\mid\beta)^{w_i} =$$
$$\prod_{i=1}^{N}\left[\begin{matrix}\sigma\left(\beta^T\phi(\mathrm{x}_i)\right)^{t_i}\\ \left\{1-\sigma\left(\beta^T\phi(\mathrm{x}_i)\right)\right\}^{1-t_i}\end{matrix}\right]^{w_i}$$

$$\geq \prod_{i=1}^{N}\left[\sigma(\xi_i)\exp\left\{\begin{matrix}\beta^T\phi(\mathrm{x}_i)t_i - \dfrac{\beta^T\phi(\mathrm{x}_i)+\xi_i}{2}\\ -\lambda(\xi_i)\left(\left(\beta^T\phi(\mathrm{x}_i)\right)^2-\xi_i^2\right)\end{matrix}\right\}\right]^{w_i}$$

$$= \prod_{i=1}^{N} h(\beta,\xi_i,w_i) \equiv h(\beta,\xi,\mathrm{w})$$

6 Simulation Study

Monte Carlo experiments were performed in the presence of outliers; we use the benchmark and industrial data to evaluate the performance of dynamic control model. To investigate the performance of some models in different situations, different simulation factors will be used. To sum up the above arguments, the whole training procedure of RRVM is as follows:

In practical utilization of this algorithm, we should set the initialization of the priors used in equations (35) - (41). First of all, α and σ^2 can be initialized according to the characteristic of the data set, e.g. $\alpha_j = N/var(t), \sigma^2 = var(t)$, where $var(t)$ is the variance of t. Secondly, the scale parameters a_i and b_i, which are included in β_i is prior distribution Gamma(a_i, b_i), should be selected so that the prior means of β_i are 1 . For example, when the parameters are set as $a_i = 1$ and $b_i = 1$, the noise variance coefficient β_i has a prior mean of $a_i/b_i = 1$ with a variance of $a_i/b_i^2 = 1$. That means we start by assuming the noise distributions of all the samples are Gaussian with the same variance, that is to say, all of the training samples are inliers. By using these values, it shows clearly that the range of β_i is $0 < \beta_i < 1.5$, which could be inferred from Eq. (40). This setting of prior parameter values is generally valid for most applications or data sets. During the process of iteration, the β_i corresponding to outliers will gradually become small.

Eq. (40) reveals that the prediction error $(t_i - \varphi(x_i)^T\mu)^2$ of data point $\{x_i, t_i\}$ is in the denominator. If the prediction error in $\mathrm{t_i}$ is so large that it dominates over other denominator terms, then the corresponding noise variance coefficient β_i of that point will be very small. When the prediction error term in the denominator tends to infinity, the β_i will approach to zero. As can be seen from Eq. (35) and (36), the calculation formulas of Σ and μ of the posterior distribution over w both include a term which is the linear weighted combination of all the samples, and the weight is exactly β_i. If a sample has an extremely small coefficient, it will make smaller contribution to the estimate of Σ and μ. This effect is equivalent to the detection and removal of an outlier if the coefficient of the data sample $\{x_i, t_i\}$ is small enough, which can improve the robustness of the model. After training, RRVM can be used to make prediction based on the posterior distribution over w. For a new input datum x^*, the output is $y^* = \phi(x^*)^T\mu$.

The size of sample set is N =100,150,300 and 400. At first, we investigate the approximation performance of RRVM with the clean training sample set. Then, some outliers generated from standard Gaussian distribution are added into the training sample set. We interfuse $10, 20, 30,$ and 40 outliers with the clean training samples, respectively. To evaluate the generalization performance in terms of the robustness, each data set is randomly divided into the training (60%) and test data sets (40%).O

6.1 Iterative algorithm

Update equations from (35) to (41) given the hyperparameters. The training procedure of the proposed method can be summarized as follows:

Step 1: Initialize the hyperparameters α, β and σ^2 as well as a_i and b_i ; $i = 1, \ldots, N$.

Step 2: Compute the variance matrix Σ and mean value vector μ of posterior distribution over w by the use of equations (35) and (36), respectively.

Step 3: Iteratively optimise the hyperparameters, α, β and σ^2 according to (39) – (41). Many of α_j will trend to infinity during the optimization method (as determined by a big threshold number, such as 10^9). This indicates that μ_j will trend to zero, as would w_i, based on Eq. (39). The model sparsity is obtained by pruning the corresponding basis functions.

Step 4: Check to see if all of the parameters are convergent or if the maximum number of iterations has been achieved. If this is the case, you should cease iterating and training. Return to Step 2 if necessary. The basis functions corresponding to non-zero μ_j are referred to as "relevance vectors" when the training is completed.

Step 5: All Monte Carlo experiments involved replications and all the results of all separate experiments are obtained by precisely the same series of random numbers.

6.2 Error Estimation Methods

For comparison, five other methods are also implemented in the experiment, including one-nearest neighbor (1-NN), k-nearest neighbor (k-NN), SVM, classical RVM and TRVM. To verify the robustness of the proposed method RRVM compared to other classification algorithms, the generalization performance of each method is evaluated in terms of three performance measures which are listed below:

- Mean square error (MSE)
- Mean absolute error (MAE)
- Root mean square error (RMSE)

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N} (T_i - F_i)^2,$$

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N} |(T_i - F_i)| ,$$

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} (T_i - F_i)^2}.$$

Moreover, we used the Coefficient of Determination (R^2), which are defined as;

$$R^2 = 1 - \frac{\sum_{i=1}^{N} (T_i - F_i)^2}{\sum_{i=1}^{N} (T_i - \bar{T})^2},$$

where $T_i F_i, \bar{T}$ and N are the target value, forecast value, average target value and the number of data, respectively.

The RMSE depends on the predicted values, not on how the values fall relative to a threshold or relative to each other. It measures how much predictions deviated from the true target values. Note that smaller values of the MSE, MAE, and RMSE mean the better classification ability of the model, while for the R^2, higher is better.

6.3 Results and Discussions

The SVM is implemented using LIBSVM software [32], and the source code to run Classical RVM is obtained from Tipping's website[1]. Moreover, Hybrid learning algorithm was employed to update the network parameters and optimum model of ANFIS [33] was constructed using the trial-and-error process. Also, the proposed method (RRVM) toolbox of **MATLAB 7.5** is utilized to implement the algorithm.

The number of nearest neighbor k should be chosen for k-NN. In this, simulation study, the training data set is subjected to a five-fold cross validation technique, after which the ideal number of k resulting in the lowest error rate is determined. The SVM and RVM model parameters are optimized using a similar technique. The SVM has two model parameters: the regularization parameter C and the kernel parameter (e.g., the width σ of the kernel function in the case of the RBF kernel), whereas the RVM only has one (the kernel parameter value). The proposed RRVM also has the kernel parameter as a single model parameter. While the parameters of the SVM should be optimized through the cross validation procedure which is computationally demanding, the parameter of the RVM can be selected efficiently by comparing the lower bound values.

[1]Tipping's website: http://www.miketipping.com/.

Table 1: Generalization Performance of Classification Methods for $N = 100$

Method	Measure	Percentages of Outliers				
		0%	10%	20%	30%	40%
1-NN	MSE	2.9658	4.2119	4.7871	5.9100	7.2963
	MAE	3.4365	4.5104	5.5024	6.7931	8.7091
	RMSE	1.7222	2.0523	2.1879	2.4311	2.7012
	R^2	0.9034	0.8722	0.9013	0.8925	0.8860
k-NN	MSE	1.2021	2.8330	3.1983	3.9486	4.8748
	MAE	3.3571	4.1632	5.1848	6.4010	8.2064
	RMSE	1.0964	1.6832	1.7884	1.9871	2.2079
	R^2	0.9126	0.9347	0.9061	0.8694	0.9224
SVM	MSE	1.9404	2.1270	2.9807	3.6799	4.5431
	MAE	2.9488	4.1296	5.0418	6.2244	7.9800
	RMSE	1.3930	1.4584	1.7265	1.9183	2.1314
	R^2	0.9041	0.8305	0.9164	0.9237	0.9039
Classical RVM	MSE	**0.5156**	1.4561	2.5820	3.1877	3.9354
	MAE	**2.2480**	3.9856	4.8214	5.9523	7.6312
	RMSE	**0.7181**	1.2067	1.6069	1.7854	1.9838
	R^2	**0.9930**	0.9287	0.8866	0.9388	0.9378
TRVM	MSE	0.8174	1.0361	1.7075	2.1080	2.6025
	MAE	2.4096	3.3568	4.0352	4.9817	6.3868
	RMSE	0.9041	1.0179	1.3067	1.4519	1.6132
	R^2	0.8959	0.9362	0.9208	0.9612	0.9459
RRVM	MSE	0.6289	**0.9580**	**1.4346**	**1.7711**	**2.1865**
	MAE	2.5381	**2.9485**	**3.3294**	**4.1104**	**5.2698**
	RMSE	0.7930	**0.9788**	**1.1977**	**1.3308**	**1.4787**
	R^2	0.8711	**0.9634**	**0.9886**	**0.9802**	**0.9708**

The best performance for each percentage of outliers is given in bold.

The simulation results are presented in Tables 1 to 3, with different sample size N =100,150,300 and 400, respectively. Each table has five sections represent the percentages of outliers. From Tables 1 to 4, we can summarize the effects of the main simulation factors on MSE, MAE, RMSE and R^2 values for all methods as follows:

- As N increases, the values of MSE, MAE and RMSE are decreases in all situations.
- As percentages of outliers increases, the values of MSE, MAE and RMSE are increases in all situations.

The MSE, MAE, RMSE and R^2 comparison of six methods is listed in Tables 1 to 3. When the training sample set excludes outliers, the MSE, MAE and RMSE of RRVM is very close to that of TRVM but is worse than that of classical RVM. We can conclude that in the absence of outliers classical RVM method is more efficient than other methods, because it has minimum MSE, MAE, RMSE and higher values of R^2. When outliers are added, the approximation performance of classical RVM deteriorates drastically, while TRVM and RRVM can still get good results. With the increase of outlier number, RRVM can obtain better result than classical RVM and TRVM, which demonstrates that RRVM can effectively resist the impact of outliers and has good robustness.

The results show that as the contamination percentage increases, the predictive performances of the classifiers get worse and worse, while the RRVM clearly shows its robustness. In addition, it is shown that the RRVM gives a sparse solution. Furthermore, it is confirmed from Tables 1-2 that the RRVM is competitive with other methods in terms of the computation time since it takes relatively a short time to optimize the model parameters. From Table 3, it is clearly shown that the generalization performances of the RRVM are consistently better than other methods even if the training data set is contaminated by the outliers.

Table 2: Generalization Performance of Classification Methods for $N = 150$

Method	Measure	Percentages of Outliers				
		0%	10%	20%	30%	40%
1-NN	MSE	0.9491	1.3478	1.5319	1.8912	2.3348
	MAE	0.8591	1.1276	1.3756	1.6983	2.1773
	RMSE	0.9742	1.1610	1.2377	1.3752	1.5280
	R^2	0.8925	0.8616	0.8904	0.8817	0.8753
k-NN	MSE	0.3847	0.9066	1.0235	1.2635	1.5599
	MAE	0.8393	1.0408	1.2962	1.6002	2.0516
	RMSE	0.6202	0.9521	1.0117	1.1241	1.2490
	R^2	0.9015	0.9234	0.8952	0.8589	0.9112
SVM	MSE	0.6209	0.6806	0.9538	1.1776	1.4538
	MAE	0.7372	1.0324	1.2604	1.5561	1.9950
	RMSE	0.7880	0.8250	0.9766	1.0852	1.2057
	R^2	0.8932	0.8205	0.9053	0.9125	0.8929
Classical RVM	MSE	**0.1650**	0.4659	0.8263	1.0201	1.2593
	MAE	**0.5620**	0.9964	1.2053	1.4881	1.9078
	RMSE	**0.4062**	0.6826	0.9090	1.0100	1.1222
	R^2	**0.9810**	0.9175	0.8758	0.9274	0.9265
TRVM	MSE	0.2616	0.3316	0.5464	0.6746	0.8328
	MAE	0.6024	0.8392	1.0088	1.2454	1.5967
	RMSE	0.5114	0.5758	0.7392	0.8213	0.9126
	R^2	0.8851	0.9248	0.9096	0.9496	0.9344
RRVM	MSE	0.2013	**0.3066**	**0.4591**	**0.5667**	**0.6997**
	MAE	0.6345	**0.7371**	**0.8324**	**1.0276**	**1.3174**
	RMSE	0.4486	**0.5537**	**0.6775**	**0.7528**	**0.8365**
	R^2	0.8605	**0.9517**	**0.9766**	**0.9684**	**0.9590**

The best performance for each percentage of outliers is given in bold.

Graphically, we illustrate the MSE and RMSE values for different methods in all cases with different main factors by 3D graphs are shown in Figures 5 and 6, when $N = 400$. Figures 5 and 6 illustrate the effect of outliers on the decision boundaries obtained from the SVM, Classical RVM, TRVM and the RRVM. Note that the SVM does not provide such probabilistic information. From the figures, it can be observed that the SVM and Classical RVM are not robust to the outliers, i.e. the decision boundaries are distorted by a few outliers. In contrast to them, the TRVM and RRVM is more insensitive to outliers since it reduces the effect of outliers by giving a small weight to them. In terms of the sparsity, the RRVM preserves the sparsity, i.e. the number of non-zero coefficient is small enough, although the training data set contains outliers, see Abonazel [34] for more details to 3D graphs using R software.

7 Conclusions

In this paper, we propose the robust RVM based on an ANFIS and weighting scheme, which is insensitive to outliers and simultaneously maintains the advantages of the original RVM. Given a prior distribution of weights, weight values are determined in a probabilistic way and computed automatically during training. Our theoretical result indicates that the influences of outliers are bounded through the probabilistic weights. Also, a guideline for determining hyperparameters governing a prior is discussed. For comparison, five other methods are also implemented in the experiment, to verify the robustness of the proposed method RRVM compared to other classification algorithms. The simulation results showed that, based on MSE, MAE, RMSE and R^2criteria, the proposed RRVM give better performance than other methods when the data contain outliers. While when the dataset does not contain outliers, the results showed that the classical RVM is more efficient than other methods.

Table 3: Generalization Performance of Classification Methods for $N = 300$

Method	Measure	Percentages of Outliers				
		0%	10%	20%	30%	40%
1-NN	MSE	0.3037	0.4313	0.4902	0.6052	0.7471
	MAE	0.2148	0.2819	0.3439	0.4246	0.5443
	RMSE	0.5511	0.6567	0.7001	0.7779	0.8644
	R^2	0.8817	0.9051	0.9268	0.8710	0.8647
k-NN	MSE	0.1231	0.2901	0.3275	0.4043	0.4992
	MAE	0.2098	0.2602	0.3241	0.4001	0.5129
	RMSE	0.3509	0.5386	0.5723	0.6359	0.7065
	R^2	0.8906	0.9122	0.9402	0.8485	0.9002
SVM	MSE	0.1987	0.2178	0.3052	0.3768	0.4652
	MAE	0.1843	0.2581	0.3151	0.3890	0.4987
	RMSE	0.4458	0.4667	0.5525	0.6139	0.6821
	R^2	0.9203	0.8939	0.9419	0.9137	0.9179
Classical RVM	MSE	**0.0528**	0.1491	0.2644	0.3264	0.4030
	MAE	**0.1405**	0.2491	0.3013	0.3720	0.4769
	RMSE	**0.2298**	0.3861	0.5142	0.5713	0.6348
	R^2	**0.9691**	0.9064	0.9469	0.9162	0.9153
TRVM	MSE	0.0837	0.1061	0.1749	0.2159	0.2665
	MAE	0.1506	0.2098	0.2522	0.3114	0.3992
	RMSE	0.2893	0.3257	0.4182	0.4646	0.5162
	R^2	0.9394	0.9368	0.9525	0.9381	0.9231
RRVM	MSE	0.0644	**0.0981**	**0.1469**	**0.1814**	**0.2239**
	MAE	0.1586	**0.1843**	**0.2081**	**0.2569**	**0.3294**
	RMSE	0.2538	**0.3132**	**0.3833**	**0.4259**	**0.4732**
	R^2	0.9398	**0.9402**	**0.9648**	**0.9567**	**0.9474**

The best performance for each percentage of outliers is given in bold.

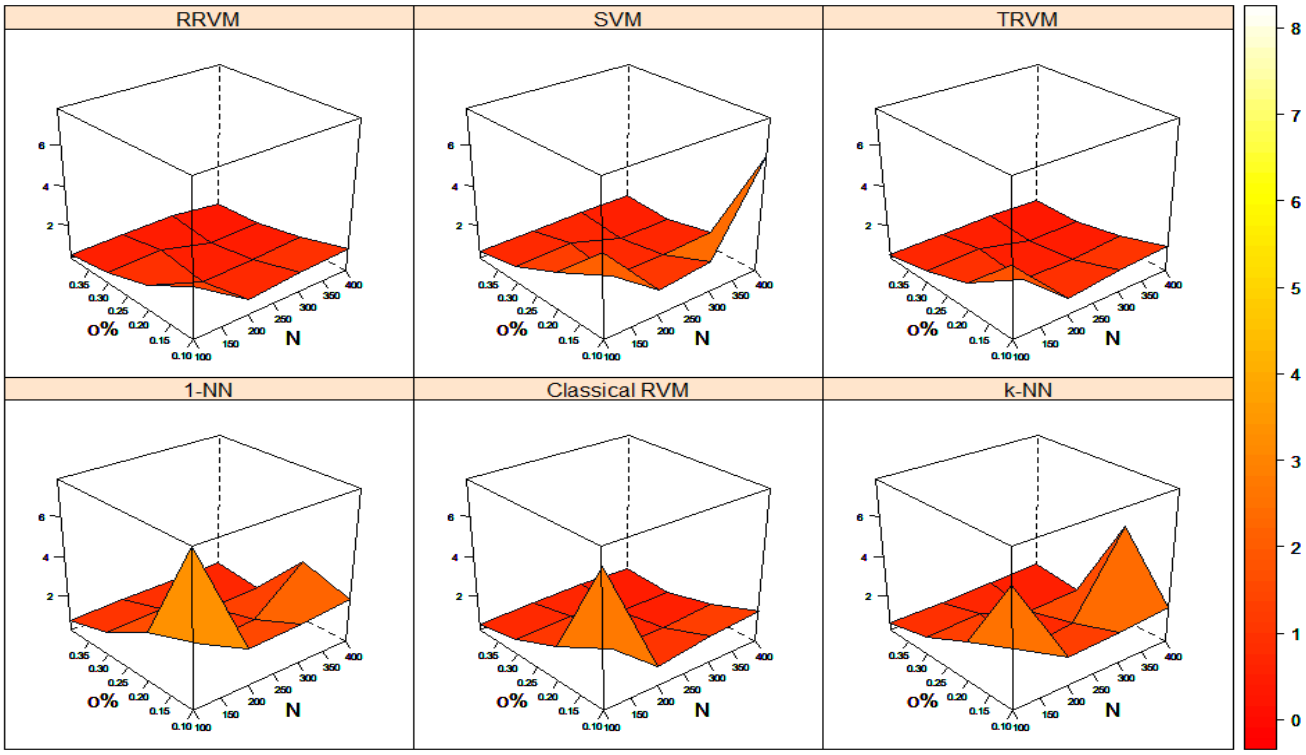

Fig. 5 The MSE values for all methods with different percentages of outliers when $N = 400$

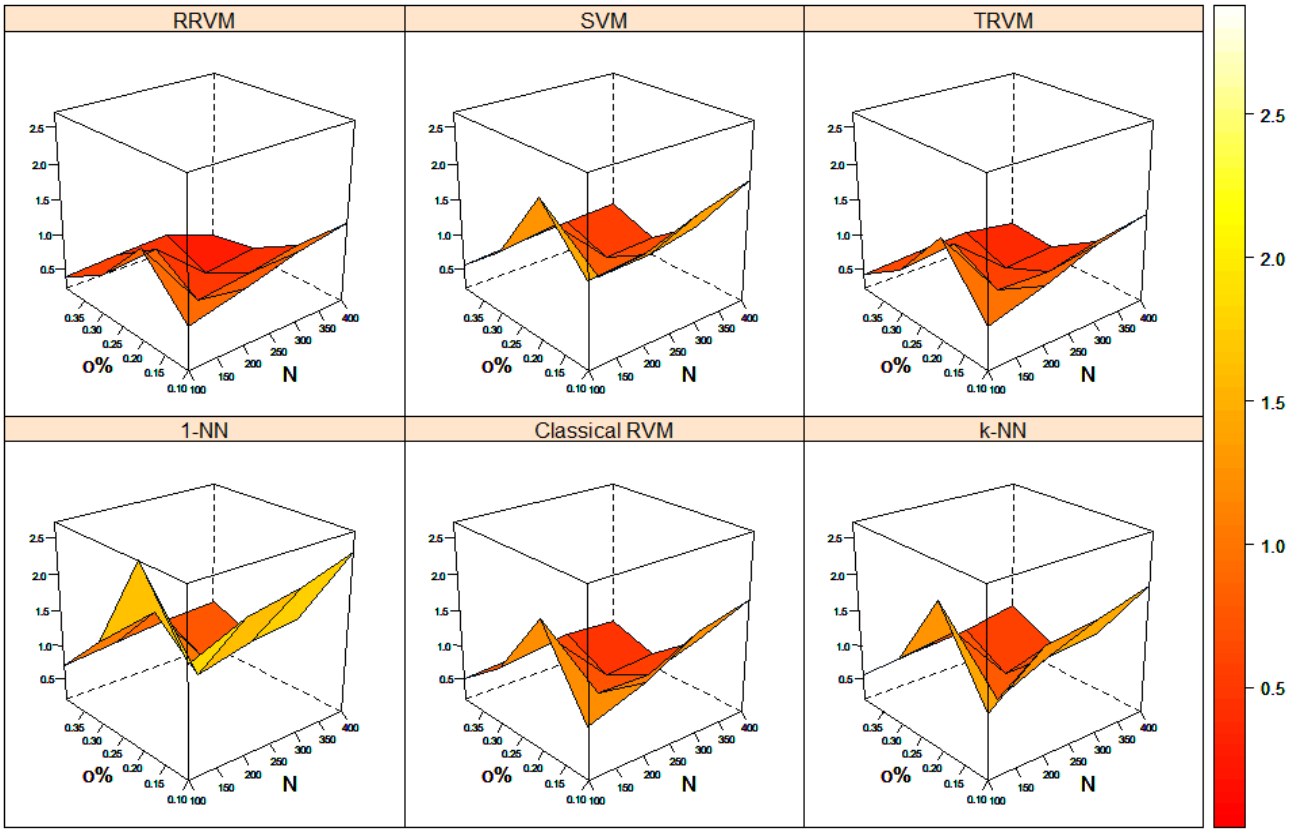

Fig. 6 The RMSE values for all methods with different percentages of outliers when $N = 400$

References:

[1] Das, S. S., Biswal, S. R., Tripathy, U. K., & Das, P. (2011). Mass transfer effects on unsteady hydromagnetic convective flow past a vertical porous plate in a porous medium with heat source. *Journal of Applied Fluid Mechanics*, 4(4), 91–100.

[2] Toki, C. J., & Tokis, J. N. (2007). Exact solutions for the unsteady free convection flows on a porous plate with time- dependent heating. *ZAMM- Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik: Applied Mathematics and Mechanics*, *87*(1), 4-13.

[3] Senapati, N., Dhal, R. K., & Das, T. K. (2012). Effects of chemical reaction on free convection MHD flow through porous medium bounded by vertical surface with slip flow region. *American Journal of Computational and Applied Mathematics*, *2*(3), 124-135.

[4] Khan, M., & Azam, M. (2017). Unsteady heat and mass transfer mechanisms in MHD Carreau nanofluid flow. *Journal of Molecular Liquids*, *225*, 554-562.

[5] Eid, M. R., Mahny, K. L., Muhammad, T., & Sheikholeslami, M. (2018). Numerical treatment for Carreau nanofluid flow over a porous nonlinear stretching surface. *Results in physics*, *8*, 1185-1193.

[6] Tipping, M. E. (2001). Sparse Bayesian learning and the relevance vector machine. *Journal of machine learning research*, *1*(Jun), 211-244.

[7] Tipping, M. E. (2000). The relevance vector machine. in advances in neural information processing systems. *vol*, *12*, 652-658.

[8] Lee, K., Kim, N., & Jeong, M. K. (2014). The sparse signomial classification and regression model. *Annals of Operations Research*, *216*(1), 257-286.

[9] Naveed, M., Abbas, Z., & Sajid, M. (2016). Hydromagnetic flow over an unsteady curved stretching surface. *Engineering Science and Technology, an International Journal*, *19*(2), 841-845.

[10] Sahoo, S. N. (2013). Heat and mass transfer effect on MHD flow of a viscoelastic fluid through a porous medium bounded by an oscillating porous plate in slip flow regime. *International Journal of Chemical Engineering*, *2013*.

[11] Noor, N. F. M., Abbasbandy, S., & Hashim, I. (2012). Heat and mass transfer of thermophoretic MHD flow over an inclined radiate isothermal permeable surface in the presence of heat source/sink. *International Journal of Heat and Mass Transfer*, *55*(7-8), 2122-2128.

[12] Turkyilmazoglu, M. (2012). MHD fluid flow and heat transfer due to a stretching rotating disk. *International journal of thermal sciences*, *51*, 195-201.

[13] Chen, C. H. (2004). Combined heat and mass transfer in MHD free convection from a vertical surface with Ohmic heating and viscous dissipation. *International journal of engineering science*, *42*(7), 699-713.
[14] Vapnik, V. N. (2000). *The nature of statistical learning theory* (second ed.). New York, USA: Springer science & business media.
[15] Muller, K. R., Mika, S., Ratsch, G., Tsuda, K., & Scholkopf, B. (2001). An introduction to kernel-based learning algorithms. *IEEE transactions on neural networks*, *12*(2), 181-201.
[16] Zhang, R., & Wang, S. (2008). Support vector machine based predictive functional control design for output temperature of coking furnace. *Journal of Process Control*, *18*(5), 439-448.
[17] Yang, B., Zhang, Z., & Sun, Z. (2007). Robust relevance vector regression with trimmed likelihood function. *IEEE Signal Processing Letters*, *14*(10), 746-749.
[18] Valyon, J., & Horváth, G. (2009). A sparse robust model for a Linz–Donawitz steel converter. *IEEE Transactions on Instrumentation and measurement*, *58*(8), 2611-2617.
[19] Hwang, S., Jeong, M. K., & Yum, B. J. (2013). Robust relevance vector machine with variational inference for improving virtual metrology accuracy. *IEEE Transactions on Semiconductor Manufacturing*, *27*(1), 83-94.
[20] Hwang, S., Kim, D., Jeong, M. K., & Yum, B. J. (2015). Robust kernel-based regression with bounded influence for outliers. *Journal of the Operational Research Society*, *66*(8), 1385-1398.
[21] Wu, Y., & Liu, Y. (2007). Robust truncated hinge loss support vector machines. *Journal of the American Statistical Association*, *102*(479), 974-983.
[22] Park, S. Y., & Liu, Y. (2011). Robust penalized logistic regression with truncated loss functions. *Canadian Journal of Statistics*, *39*(2), 300-323.
[23] Abonazel, M., & Rabie, A. (2019). The impact of using robust estimations in regression models: An application on the Egyptian economy. *Journal of Advanced Research in Applied Mathematics and Statistics*, 4(2), 8-16.
[24] Abonazel, M., & Gad, A. A. E. (2020). Robust partial residuals estimation in semiparametric partially linear model. *Communications in Statistics-Simulation and Computation*, 49(5), 1223-1236.
[25] Youssef, A. H., Kamel, A.R. & Abonazel, M. R (2021). Robust SURE estimates of profitability in the Egyptian insurance market, *Statistical journal of the IAOS*, (Preprint), 1-13 (2021). DOI:10.3233/SJI-200734.
[26] Kamel, A.R. (2021). *Handling outliers in seemingly unrelated regression equations model*, MSc thesis, Faculty of graduate studies for statistical research (FGSSR), Cairo University, Egypt.
[27] Melin, P., & Castillo, O. (2005). Intelligent control of a stepping motor drive using an adaptive neuro–fuzzy inference system. *Information Sciences*, *170*(2-4), 133-151.
[28] Tipping, M. E., & Lawrence, N. D. (2005). Variational inference for Student-t models: Robust Bayesian interpolation and generalised component analysis. *Neurocomputing*, *69*(1-3), 123-141.
[29] Ting, J. A., D'Souza, A., & Schaal, S. (2007, April). Automatic outlier detection: A Bayesian approach. In *Proceedings 2007 IEEE International Conference on Robotics and Automation* (pp. 2489-2494). IEEE.
[30] Jaakkola, T. S., & Jordan, M. I. (2000). Bayesian parameter estimation via variational methods. *Statistics and Computing*, *10*(1), 25-37.
[31] Bishop, C. M., & Tipping, M. (2013). Variational relevance vector machines. *arXiv preprint arXiv:1301.3838*, available at: https://arxiv.org/ftp/arxiv/papers/1301/1301.3838.pdf
[32] Chang, C. C., & Lin, C. J. (2011). LIBSVM: a library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*, *2*(3), 1-27.
[33] Jang, J. S. (1993). ANFIS: adaptive-network-based fuzzy inference system. *IEEE transactions on systems, man, and cybernetics*, *23*(3), 665-685.
[34] Abonazel, M. R. (2018). A practical guide for creating Monte Carlo simulation studies using R. *International Journal of Mathematics and Computational Science*, *4*(1), 18-33.