Hindawi
Security and Communication Networks
Volume 2022, Article ID 5933086, 7 pages
https://doi.org/10.1155/2022/5933086

Research Article

Providing a Consistent Method to Model the Behavior and Modelling Intrusion Detection Using A Hybrid Particle Swarm Optimization-Logistic Regression Algorithm

Mahdi Ajdani,[1] **Azad Noori**,[2] **and Hamidreza Ghaffary**[1]

[1]*Department of Computer Science, Ferdows Branch, Islamic Azad University, Ferdows, Iran*
[2]*Department of Computer Engineering, Technical and Vocational University (TVU), Tehran, Iran*

Correspondence should be addressed to Hamidreza Ghaffary; ajdanighaffary@gmail.com

Received 29 June 2021; Revised 29 April 2022; Accepted 10 May 2022; Published 7 June 2022

Academic Editor: AnMin Fu

An intrusion detection system is a collection of instruments, methods, and documentation that aid in identifying, determining, and reporting unwanted or illegal network activity. Intrusion detection systems are built as software and hardware systems, each with its own set of benefits and drawbacks. Because of the intrusion detection system's nonlinearity and nonstationary, the correctness of traditional methods, such as regression analysis and neural networks, was limited. In this research, a regression-based prediction model is proposed to handle an intrusion detection behavior problem. To develop an effective regression model, the parameters must be carefully adjusted. This present research introduces a hybrid methodology called real-value particle swarm optimization (RPSO) algorithm regression, which uses real-value particle swarm optimization algorithms to find the optimal parameters. Then, it uses the best parameters to build the regression models. The method is used to forecast the data related to an intrusion detection behavior from the VirusTotal dataset. Due to the root mean square error (RMSE) 0.0234 and the mean absolute percentage error (MAE) 1.845, the experimental results show that RPSO performs best the standard regression and backpropagation (BP) neural network models (MAPE). It was proved that the RPSO model is a practical method to recognize the behavior of the intrusion detection system feature.

1. Introduction

Infiltration is a collection of illicit behaviors that jeopardize the secrecy of a resource or access to it. An intrusion detection system is a collection of instruments, methods, and documentation that aid in identifying, determining, and reporting unwanted or illegal network activity. Intrusion detection systems are built as software and hardware systems, each with its own set of benefits and drawbacks. Hardware systems have the advantages of speed and accuracy, as well as the lack of security breaches by intruders. However, the simplicity of software use, the flexibility to adapt to software settings, and the differences between different operating systems provide software systems with more generality. In general, such systems are better options. Growing access to data and processing them faster while also increasing the data volume and the requirement to supply data from many sources via computer networks result in forming hazardous sources via system flaws. One of the primary strategies for assuring the network and computer system security is intrusion detection and prevention (IDS). Numerous intrusion detection systems are available to identify attacks, and the key problem raising the system's efficiency. Existing intrusion detection analyzes all network packet parameters to examine and recognize attack patterns, even when some of these parameters are useless. Utilizing all of the parameters improves the efficiency of the long detection and system detection processes. The massive amount of data is the key issue in the intrusion detection system. However, due to the huge amount of warnings produced by such systems and the development of erroneous signals, such systems are unable to handle and analyze the created

warnings. In different datasets, the number of properties with the highest accuracy in recognizing intrusion is determined empirically. However, choosing attributes may result in the loss of certain data. As a result of the variation in network data, one problem is estimating the ideal amount of features. Accordingly, numerous models were created, such as neural networks, fuzzy models, and Monte Carlo simulations,. A software package that controls the behavior of the intrusion detection system and permits the user to perform simulations owing to the design system approach is a property in the intrusion detection system [1]. In an intrusion detection system feature, traditional regressions are considered a model achieved in many scenarios for basin users. Because the behavior and regression of the intrusion detection system are not certain and are subject to the conditions controlling the intrusion detection system, it is required to enhance the feature using a nondeterministic model such as artificial intelligence. Compared to the standard modelling method, particle swarm optimization programming is also a model that does not take into account any type of a postulated form. In the regression approach, the model's structure is predefined (e.g., quadratic and linear regression), and then, the model's parameters are calculated. When using particle swarm optimization programming to create the equation or formula between input and output variables, the key benefit is the capacity to automatically choose input variables that are beneficial in the model while disregarding the ones that are not [2]. As a result, it is commonly referred to as particle swarm optimization modification since it can essentially minimize the dimensions of input variables and for the relationships that it displays, it is feasible to specify the change in various programming environments. In other words, the behavior of other factors is considered by modifying each parameter's graph that affects the system. Furthermore, logistic regression is an effective learning machine as a result of statistical learning theory and a concept of structural risk minimization that has been effectively applied to nonlinear system modelling Kalita [3]. Under similar training conditions, logistic regression outperforms artificial neural networks in terms of reliability and performance. Despite its strong characteristics, logistic regression is restricted in academic studies and industrial applications because the user must adequately set many parameters [4–6]. To build the model, the logistic regression parameters must be carefully adjusted. If logistic regression parameters are selected inappropriately, they will lead to underfitting or overfitting. Furthermore, differing parameter regulations may result in large changes in performance [7]. Choosing the best parameters is thus an essential stage in logistic regression design. However, there are no universal guidelines available to assist in the selection of such criteria. [8–10]. Many recent advancements in machine learning were arguably only possible given the exponential growth of datasets [11–18]: a decade ago, these models would learn from datasets with hundreds of entries and these same applications can now benefit from hundreds of thousands of entries, making these techniques more viable. While this was positive, it also meant that optimization was needed to make algorithms perform tasks in adequate time [14–18]. Logistic regression operates on each feature to compute via regression and their coefficients. With a big enough dataset, the complete set of features will slow down this process and, as such, there needs to be a way to select features that more directly impact the output while discarding duplicates and/or noise. This problem, aptly called curse of dimensionality, is mitigated by the feature selection [14–17] technique, of which there are several types of solutions, depending on circumstances. As such, filter methods rank features by a score to either keep or remove from the dataset; wrapper methods select a subset of features from the search space, measure their accuracy, and aim to, by the end, select the best subset of features; embedded methods execute during creation of the model [13–15]. As a result, in this study, we propose a hybrid approach of logistic regression with real-value particle swarm optimization (RPSO) algorithm regression developed by using an RPSO to specify the logistic regression free parameters, and thus, the generalization ability and forecasting accuracy are improved. The method is used to predict the behavior of intrusion detection system features. In addition, for comparison, the standard logistic regression model and a BP neural network were studied. The experimental findings demonstrate that the mentioned proposed method can improve predictive accuracy and generalization capability [19–21].

2. Materials and Methods

2.1. PSO (Particle Swarm Optimization Algorithm). Particle motion is based on modelling of mass behavior applied to simulate the movements of a group of birds and fish and is considered as one of the highest usual techniques of metaheuristic optimization. The cumulative movement of particles explores the search space by using the population factor, which includes alternative solutions to the study problem. Each member of the population has an adaptive speed (displacement) in the cumulative motion of particles, which causes it to move in the search space. Furthermore, each has a memory, i.e., the ideal position in an area. They recall such results, so each member advances in two directions: towards the best scenario they have encountered and to the best situation the best member they have faced in their proximity [8]. The simplicity of the PSO approach, which includes only two equations of the position and velocity, where the coordinates of each particle indicates a probable response concerning two vectors, is one of its primary features and popularity. Let f: $\mathbb{R}n \longrightarrow \mathbb{R}$ denotes the cost function, which is better to be minimized. The process accepts a representative solution as an explanation as a vector of real numbers and outputs an actual number indicating the objective function value of the supplied candidate solution. It is unknown what the gradient of f is. It is aimed at discovering a solution such that $f(a) \leq f(b)$ for all b in the search space, implying that a is the global minimum. Let S denotes the number of particles in the swarm, each with a position in the search-space $xi \in \mathbb{R}n$ and a velocity

$vi \in \mathbb{R}n$. Suppose that pi and g represent the most common part of the particle i and the position of the whole swarm, respectively [5].

2.2. Logistic Regression. The correlation coefficient is frequently used to represent the strength of a linear relationship between two quantitative variables. Furthermore, we employ the regression model to demonstrate the link model between them. Meanwhile, a template for predicting the dependent variable (Y) resulting from the independent variable (X) is produced. However, both the independent and dependent variables are minor in the generated model. The regression approach also takes into account the continuity requirement of these values. We might assess the association between an independent and dependent variable (with continuous and qualitative values), respectively [9]. The standard linear regression method will not work in this scenario, and "logistic regression" should be employed instead. Instead of a linear relationship, we require a function ranging from 0 to 1 to identify the model of the relationship between a dependent and an independent variable. The logistic regression approach makes use of a procedure known as the "logistic function." As a result, this regression approach is known as logistic regression. This function is introduced in the next section, and its corresponding diagram (Figure 1) is shown concerning the parameters in the image.

2.3. Optimization of the Logistic Regression Model Based on RPSO. The performance of logistic regression generalization (estimation accuracy) is dependent on the log parameter being correctly chosen [9–11]. However, there are no common recommendations for selecting these characteristics. Most researchers still use a standard method (trial and error) based on the grid algorithm; first, generating numerous logistic regression models based on a set of different parameters, and then, their testing on a validation set to determine the ideal parameters. This procedure, however, takes a lot of time. We attempted to apply it, but we could not converge it in the perfect world [12]. As a result, we used a real value particle swarm optimization (RPSO) algorithm to find the best logistic regression parameters in order to increase prediction performance. The values of logistic regression parameters are directly encoded with the actual value data on the chromosome in the proposed RPSO-logistic regression model. We use the RPSO evolution method to dynamically improve the values of logistic regression parameters and then use the optimized parameters to develop an optimal logistic regression model to advance the prediction.

The framework for optimizing logistic regression parameters with a simple value particle swarm optimization algorithm is shown in Figure 2, and it is summarized in the following steps.

Step 1 (particle code): the parameters of logistic regression for particle creation are directly randomly coded [5, 7]. The domain is (1, 100), which are (0.0001, 0.01), and (1, 100) and (0, 1), respectively. There are 100 particles in the population.

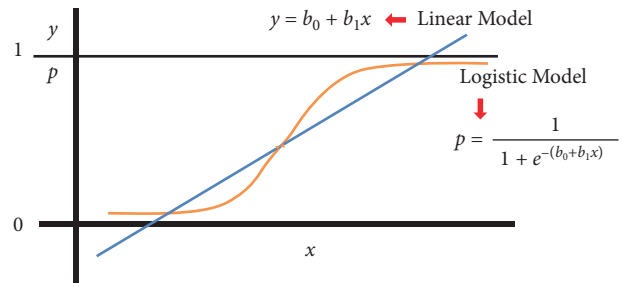

Figure 1: Logistic regression [8].

Step 2 fitness definition: the fit of the training dataset is simple to compute; however, it is prone to being overly suitable. To solve this problem, a cross-validation method can be used. To overcome the overly appropriate phenomena, a fivefold validation approach is applied in this situation [10]. The regression function is built with a set of provided parameters and a training set of four subsets. The root mean square error (RMSE) under the last subset is used to assess the performance of the parameter set. The preceding process is performed five times so that each subset is validated just once. The expected generalization error for the training sets is the average RMSE over the five experimental trials.

Step 3 (particle swarm optimization algorithm operators): a standard convergence is employed in the operators to choose excellent particles for reproduction. For particle exchange between two spaces, a single-point convergence is chosen at random: the probability of forming a new particle per pair is 0.5. The coefficient is subject to a convergence action that determines whether a particle will proceed to the next generation. Each new population's particle has a chance of convergence of 0.02.

Step 4 (stop criteria): if the new population lacks a termination condition, steps 3-4 are repeated indefinitely until the models are satisfied with the least amount of model error.

2.4. Principle of a Neuro-Fuzzy GMDH Network. The GMDH model is one of the machine learning approaches based on the polynomial theory of complex systems. From this network, the most significant input parameters, the number of layers, the number of neurons of middle layers, and optimal topology design of the network are defined automatically [13] Therefore, the GMDH network includes those active neurons known as a self-organized model. The structure of the GMDH network is configured through the training stage with the polynomial model, which produces the minimum error between the predicted value and the observed output [14].

3. Collecting Data and the Behavior of Preprocessing of the Intrusion Detection System

3.1. Data Acquisition. We used the VirusTotal dataset to evaluate the model [12]. Suricata [12], a signature-based

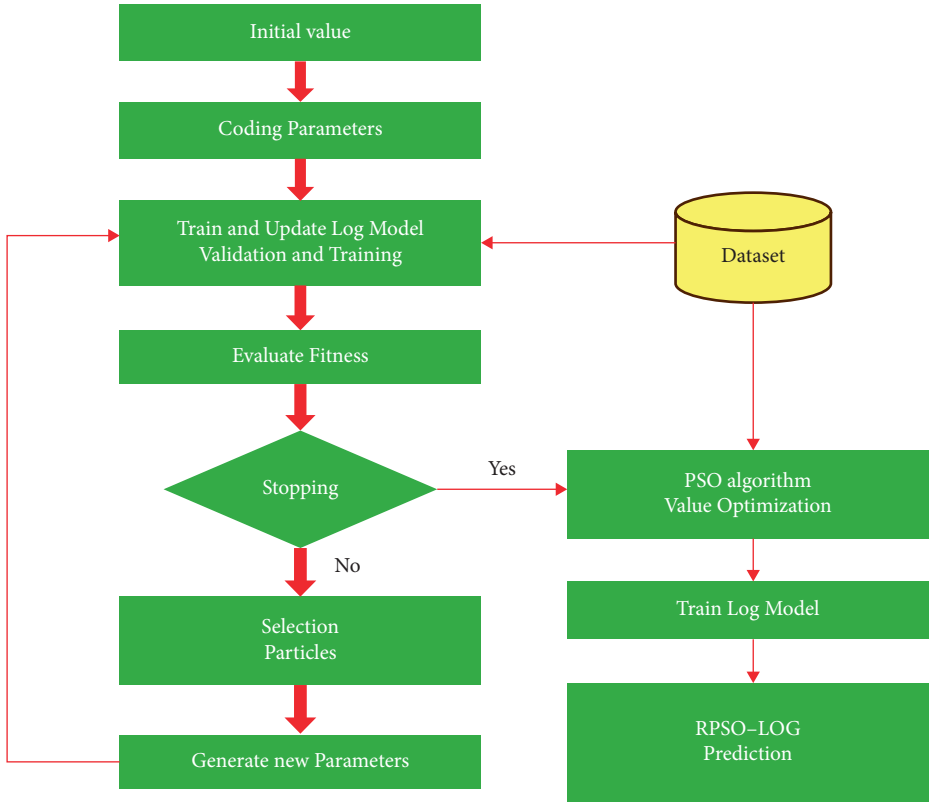

Figure 2: RPSO-logistic regression model [9].

network intrusion detection system (NIDS), matched data on high severity alerts via the VirusTotal API. This procedure ensured that legitimate transmission was free of harmful data. It should be noted that the data were equipped with the final composition of the dataset following feature extraction. The dataset used in this investigation included 2.5 million scans performed between 2012 and 2015. The dataset was chosen at random from this website. Given that the data contain one million items, the suggested approach for processing a large amount of data was used. In addition, the amount of required RAM was explored. Scalability was not considered for static and dynamic analysis costs in this method; instead, it was obtained from the VirusTotal dataset. Because the data required 770 GB of space and RAM, it was compressed by decreasing size and it was reduced to 4.2 GB. Processing requires less memory because the data are stored in columns. Feature extraction is also incorporated into the software in a condensed version. Using this method showed that with its natural features and affordable memory, it could be an appropriate method of scaling industrial applications. Except for basic feature data, label data, feature vector, and predictions, all necessitate memory. The vector file and the feature matrix are used to hold feature vectors, and their sizes for the dataset mentioned above were calculated to be 13.8 GB and 1.5 GB, respectively. The feature matrix file is compressed to save memory and network resources, while the vector file is left unaltered for easy access by machine learning algorithms. The file responsible for label storage requires less than one gigabyte of memory, and it contains labels and other data that are kept uncompressed in order to avoid costly calculations and make data analysis quicker. The proposed method took one hour; however, there is a need for much more time and resources to run this program for online data to be performed every week and month. Furthermore, feature vectors should be updated daily and weekly, and they should be reevaluated after each period. This software was created on a 5-core computer, and if there was a way to boost resources so that the processor was a 40-core processor, it could have been completed 8 times less than the current method. We can also enable scalability this way. Using static feature vectors, on the other hand, can reduce the computational load. Similarly, applying machine learning techniques can help the process and significantly compensate for the scalability aspect. Furthermore, the scalability factor in terms of complexity might be impressive among machine learning algorithms.

The data for intrusion detection system behavior prediction were divided into two parts: the first 600 datasets of intrusion detection system feature datasets were used for RPSO-logistic regression modelling training, while the last 120 datasets were used as testing data to examine an RPSO-logistic regression prediction performance.

3.2. Preprocessing the Data Sharing and Normalization, In General, Reduces the Network Speed and Accuracy by Introducing Raw Data into the Algorithm. To avoid such circumstances and normalize the data value, the input data

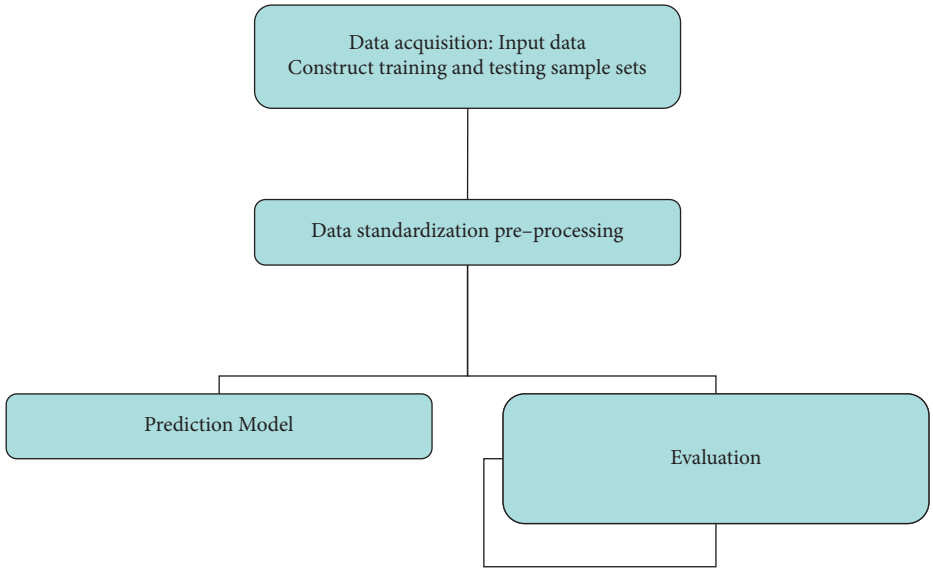

Figure 3: The sketch map of the relationship between spatial resolution and local variance. Data collection, data standardization, preprocessing, forecasting, testing findings, and application are all parts of the feature and intrusion detection system behavior forecasting system.

should be standardized before training the neural network, i.e., all data should be equalized between 0 and 1. Following normalization, the data were separated into two categories: training and testing, in order to create a network model. The model uses training data to determine the link between inputs and observed outputs.

3.3. Designing the Intrusion Detection Feature Forecasting Model. There is no standard approach for determining the free parameters of the logistic regression model. As a result, the suggested RPSO-logistic regression model optimizes logistic regression parameter values dynamically. Initially, fivefold cross-validation and RPSO were used for searching, resulting in superior combinations of the logistic regression parameters when the fivefold cross-validation value is at its lowest. Then, using RPSO-logistic regression the intrusion detection system Feature forecasting model is built. Figure 3 depicts the structure of the aquaculture intrusion detection system behavior forecasting system as a result of PSO-LOGISTIC.

4. Results and Discussion

4.1. Result. During the study, the aim to forecast the behavior of the intrusion detection system was chosen based on the most critical elements affecting the model. To anticipate our behavior based on the RPSO-logistic regression model, the values of the current monitoring feature were obtained as input parameters of the RPSO-logistic regression model. Then, the degrees of the control are compared and examined.

The classic neural network logistic regression and BP methods were also used to evaluate and compare the performance of the RPSO-logistic regression combo strategy. The learning rate for the BP neural network is 0.086, and the

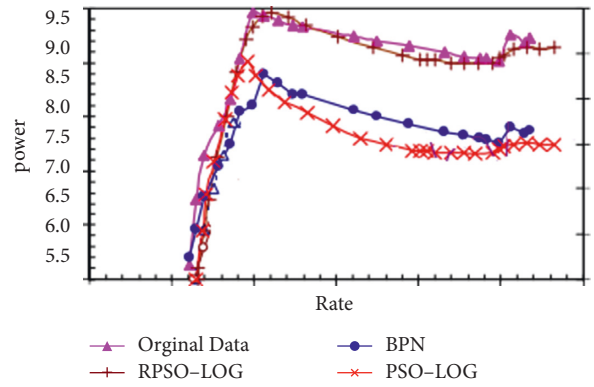

Figure 4: Model prediction.

sigmoid activation function: 5000,000 training courses are also confirmed as a terminating condition. For reliable estimation, the mean absolute error (MAPE) and the root mean square error (RMSE) were predicted. The prediction model's performance improves as the value of these mistakes decreases. Figures 4 and 5 show the forecasted results. Figure 4 depicts the expected results, with the abscissa representing time training of the sample sampling). Figure 5 depicts the outcomes of a half-hour prediction. Table 1 illustrates the parameters of the feature barrier behavior parameters for three different techniques.

It can be demonstrated that RPSO-logistic regression generalizes and predicts the validation process better than logistic regression and BP neural network approaches for both the RMSE and MAPE. The forecast value is better because of the indicators of the behavior of the two-barrier feature. Generally, the RPSO-logistic regression prediction model performs admirably. This is due to the fact that RPSO-logistic regression uses the notion of structural risk reduction rather than minimizing empirical risk, resulting in

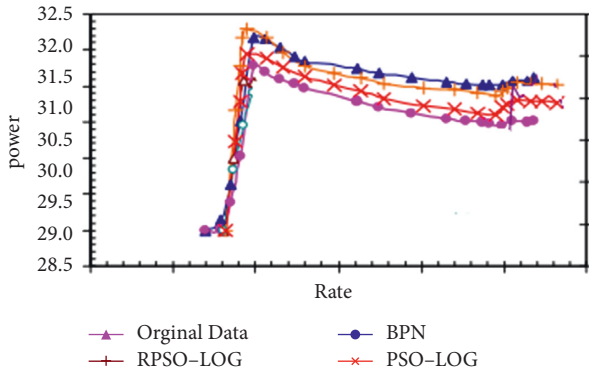

Figure 5: Feature prediction.

Table 1: Performance measure (features) and comparison of the performance of several techniques (model).

	Training stage			
	R	RMSE	MPAE	MAE
AI models				
BPN-RPSO-log	0.99	0.023	3.68	0.037
RPSO-log	0.99	0.051	5.91	0.055
PSO-log	0.98	0.061	6.23	0.082
Log	0.97	0.285	29.96	0.294
	Testing stage			
	R	RMSE	MPAE	MAE
AI models				
BPN-RPSO-log	0.96	0.029	4.84	0.035
RPSO-log	0.93	0.053	1.84	0.046
PSO-log	0.92	0.065	2.43	0.068
Log	0.90	0.305	22.76	0.322

great generalization for small sample sizes. No consensus exists on how the data series should be distributed; it may also be vulnerable to change in the data's thinking. As a result, when the information exhibits a significant fluctuation, the RPSO-logistic regression prediction error is also small. The present research demonstrates that the RPSO-logistic regression algorithm outperforms the neural network BP and classic logistic regression methods in predicting the behavior of the intrusion detection system feature.

4.2. Conclusion. Predicting the intrusion detection system feature behavior is critical because it allows for early alerts of changes in the intrusion detection system feature behavior. The proposed strategy here is to predict using an RPSO-logistic regression combination technique, in which an actual particle swarm optimization algorithm is used to find the suitable logistic regression parameters. The particle swarm optimization approach entails keeping a population of particles that can propose potential solutions to the problem. Based on actual trials with data from primary intrusion detection systems, the hybrid logistic regression technique with the particle swarm optimization algorithm can provide credible information to anticipate the behavior of large-scale feature intrusion detection systems. In addition, the experiment results demonstrate that using an artificial intelligence methodology to anticipate the performance of nonlinear series issues is extremely appropriate. The RPSO-logistic regression prediction approach can assist in reducing economic losses caused by difficulties in the intrusion detection system feature behavior. On the other hand, the function of the particle swarm optimization technique is difficult to alter during the training process of the RPSO-logistic regression model for a range of problems, and different types of rates and mutations need to be modified. As a result, how to apply improved techniques to update the right characteristics and parameters of the suggested model is an essential future development direction.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] H.-G. Han, Y.-N. Guo, and J.-F. Qiao, "Nonlinear system modeling using a self-organizing recurrent radial basis function neural network," *Applied Soft Computing*, vol. 71, pp. 1105–1116, 2018.

[2] Z. F. Hussain, H. R. Ibraheem, M. Alsajri et al., "A new model for iris data set classification based on linear support vector machine parameter's optimization," *International Journal of Electrical and Computer Engineering*, vol. 10, no. 1, p. 1079, 2020.

[3] D. J. Kalita, V. P. Singh, and V. Kumar, "A survey on SVM hyper-parameters optimization techniques," in *Social Networking and Computational Intelligence*, pp. 243–256, Springer, Singapore, 2020.

[4] B. Ma and Y. Xia, "A tribe competition-based genetic algorithm for feature selection in pattern classification," *Applied Soft Computing*, vol. 58, pp. 328–338, 2017.

[5] A. Meier and O. Kramer, "Predictive Uncertainty estimation with Temporal Convolutional networks for dynamic evolutionary optimization," in *Proceedings of the International Conference on Artificial Neural Networks*, pp. 409–421, Springer, Munich, Germany, September 2019.

[6] M. Neshat, M. Tabatabi, E. Zahmati, and M. Shirdel, "A hybrid fuzzy knowledge-based system for forest fire risk forecasting," *International Journal of Reasoning-Based Intelligent Systems*, vol. 8, no. 3-4, pp. 132–154, 2016.

[7] L. Oneto, F. Bisio, E. Cambria, and D. Anguita, "Statistical learning theory and ELM for big social data analysis," *Ieee CompUTATionAl inTelliGenCe mAGAzine*, vol. 11, no. 3, pp. 45–55, 2016.

[8] J. S. Raj and J. V. Ananthi, "Recurrent neural networks and nonlinear prediction in support vector machines," *Journal of Soft Computing Paradigm (JSCP)*, vol. 1, no. 1, pp. 33–40, 2019.

[9] M. Rong, D. Gong, and X. Gao, "Feature selection and its use in big data: challenges, methods, and trends," *IEEE Access*, vol. 7, pp. 19709–19725, 2019.

[10] G. Samigulina and Z. Massimkanova, "Development of smart-Technology for forecasting Technical state of Equipment based on modified particle swarm algorithms and Immune-network modeling," in *Proceedings of the International Conference on Computational & Experimental Engineering and Sciences*, pp. 283–293, Tokyo, Japan, March 2019.

[11] P. Tsirikoglou, S. Abraham, F. Contino, C. Lacor, and G. Ghorbaniasl, "A hyperparameters selection technique for support vector regression models," *Applied Soft Computing*, vol. 61, pp. 139–148, 2017.

[12] I. R. A. Hamid, N. S. Khalid, N. A. Abdullah, N. H. A. Rahman, and C. C. Wen, "Android Malware classification using K-means Clustering algorithm," *IOP Conference Series: Materials Science and Engineering*, vol. 226, no. 1, Article ID 012105, 2017.

[13] A. G. Ivakhnenko, "Polynomial theory of complex systems," *IEEE Trans SystMan Cybern, SMC*, vol. 1, Article ID 364e378, 1971.

[14] M. Najafzadeh, G.-A. Barani, and M. R. H. Kermani, "Abutment scour in clear-water and live-bed conditions by GMDH network," *Water Science and Technology*, vol. 67, no. 5, pp. 1121–1128, 2013d.

[15] M. Najafzadeh, "Neuro-fuzzy GMDH based particle swarm optimization for prediction of scour depth at downstream of grade control structures," *Engineering Science and Technology, an International Journal*, vol. 18, no. 1, pp. 42–51, 2015.

[16] M. Najafzadeh and A. Zahiri, "Neuro-fuzzy GMDH-based evolutionary algorithms to predict flow discharge in straight compound channels," *Journal of Hydrologic Engineering*, vol. 20, no. 12, Article ID 04015035, 2015.

[17] M. Najafzadeh, F. Homaei, and S. Mohamadi, "Reliability evaluation of groundwater quality index using data-driven models," *Environmental Science and Pollution Research*, vol. 29, no. 6, pp. 8174–8190, 2022.

[18] M. Najafzadeh and G. Oliveto, "Riprap incipient motion for overtopping flows with machine learning models," *Journal of Hydroinformatics*, vol. 22, no. 4, pp. 749–767, 2020.

[19] L. Acerbi and W. Ji, "Practical Bayesian optimization for model fitting with Bayesian adaptive direct search," pp. 1836–1846, 2017, https://arxiv.org/abs/1705.04405.

[20] L. D. Chambers, *Practical Handbook of Particle Swarm Optimization Algorithms: Complex Coding Systems*, CRC Press, Boca Raton, Florida, 2019.

[21] A. Gitoee, A. Faridi, and J. France, "Mathematical models for response to amino acids: estimating the response of broiler chickens to branched-chain amino acids using support vector regression and neural network models," *Neural Computing & Applications*, vol. 30, no. 8, pp. 2499–2508, 2018.