IEC
Institute of Electronics
and Computer

Journal of Artificial Intelligence and Systems, 2022, 4, 33-49
https://iecscience.org/journals/AIS
ISSN Online: 2642-2859

Symbolic Regression Based Feature Extraction of Shallow Neural-Networks for Identification and Prediction

Selami Beyhan[1, *]

[1] Izmir Demokrasi University, Electrical and Electronics Engineering, Uckuyular Dist., Karabaglar 35140, Izmir, Turkey

[*] Corresponding author's E-mail: selami.beyhan@idu.edu.tr

How to cite this paper: Selami Beyhan (2022). Symbolic Regression Based Feature Extraction of Shallow Neural-Networks for Identification and Prediction. Journal of Artificial Intelligence and Systems, 4, 33–49. https://doi.org/10.33969/AIS.2022040103.

Received: July 20, 2022
Accepted: October 5, 2022
Published: October 14, 2022

Abstract

This paper proposes a feature extraction method to improve the performance of shallow neural-network models with less number of parameters to apply especially on embedded system design at remote applications. Feature extraction method is designed using fuzzy c-means clustering based fuzzy system design cascaded a layer of symbolic operators and functions, respectively. During the training stage of neural-networks, symbolic operators and functions are selected using random-learning theory with the unity internal weights such that based on the prediction performance, optimal sequences are recorded for feature extraction to be utilized on testing phase. Extracted features are here used to empower the single-layer neural-network (SLNN) with sigmoid hyperbolic activation functions, functional-link neural- network (FLNN) with Chebyshev polynomials and Pi-Sigma higher-order neural-network (PSNN) with sigmoid activation functions, respectively. The internal and output parameters of the appended shallow neural-networks are optimized using batch optimization methods. Proposed regression models are first tested on identification of an artificial discrete-time dynamic system and real-time inverted pendulum then also for prediction of the sunspot time-series and traffic density estimation. As a result, the prediction performance of shallow neural networks is improved to be used in future applications.

Keywords

Symbolic regression, clustering, shallow neural-network, time-series prediction, system identification

1. Introduction

Recently, the accuracy of machine learning models have been increased through various feature extraction capabilities. The touchstone step has been taken with deep neural-networks and stochastic gradient-descent methods. Hence, recurrent and convolution filter based deep models can perform feature extraction for auto-encoding and image classification successfully. The current disadvantages of deep models are parameter optimization with long computational-time, repeatability [1] and implementation problem [2] with too many parameters. Therefore, deep neural-networks are mostly running on the computer-based environments. If the decision-making is possible to perform by connecting to the computer remotely, model complexity is not a problem. However, if the communication platform does not satisfy enough data transfer, and time-delay exists on the decision making, there is still required to design and utilize a simple, efficient and implementable model [3, 4].

With the development of deep structures, artificial neural-networks with less number of hidden layers and parameters are recalled as shallow neural-networks. They have less number of parameters compared to the deep structures. Thus, they need less computational- time for parameter optimization and less memory-space in the embedded system design. The approximation capability of single-layer models are limited when they are used in their original form. Based on this fact that feature extraction has long become an important discipline used by machine learning scientists to improve the performance of designed models. Especially with the increase in the application of single-layer artificial neural- networks, it has gained more importance. Basically, it is placed as a layer in front of the designed structure and the hidden features are extracted from the input data. Many feature extraction methods have been introduced and applied in the literature [5, 6].

Symbolic regression is originally known as genetic programming which is constructed by genetic algorithm [7] to approximate analytically all kinds of input-output problem such as regression, classification, clustering, decision making etc. The selection of symbolic functions and design of the functional model are based on the genetic algorithm. Therefore, any input-output data based implicitly given problem can be solved using a set of analytical functions and operators. The combination of various functions in the nested form with analytical operators provides high approximation power for the applications. Simple design, accurate regression capability and fast implementation advantages are the main reasons to prefer. Recently, OLS-based genetic programming was introduced for system diagnosis based on regression [8]. As the value of feature extraction methods has increased in the last few years, symbolic regression based feature has also been introduced and used in modelling [9] and control studies [10] and prediction [11]. Recently, extreme learning machine model-based symbolic regression is introduced and applied for the modelling of nonlinear systems [12–14].

This paper proposes a fuzzy clustering and symbolic regression based feature extraction methodology for the efficient design of shallow artificial-neural networks. Single-layer artificial-neural networks (SLNN), functional-link neural-networks (FLNN), Pi-Sigma neural-networks (PSNN) can be accepted as main shallow neural networks. The main goal is to obtain accurate single-layer models with less number of parameters and implementable structure to be able to run in the remote front-end devices. At the same time, the type of symbolic functions are selected randomly from a predefined sets which is based on the extreme learning machines. In feature extraction process, different characteristics of the symbolic functions are used to find out qualified features.

2. Shallow Neural-Network Models

This section introduces basic single-layer artificial neural-network (SLNN), functional-link artificial neural-network (FLNN) and higher-order neural-network (HONN) models that are especially selected to increase their accuracy by using the proposed feature extraction methodology.

2.1. Single-Layer Neural-Networks

SLNN is the basic building block of general neural-networks. Most of the regression and classification problems at smooth and moderate levels can be solved with enough data and suitable optimization of parameters. Fundamentally, the parameters are optimized to learn the input-output behaviour of the problem therefore they imitate the learning mechanism of the human neurons. SLNN model with multi-input and multi-output form is shown in Figure 1. Assume that

(x, y) input-output vectors,

$$\hat{y} = w_o f(w_i^T x + b) \quad (1)$$

Where w_o, w_i and b are the design parameters to optimize. The nonlinear function $f(\cdot)$ is the activation function that is usually selected as sigmoid and tangent hyperbolic functions etc. The sigmoid function is $f(x) = 1/(1+e^{-x})$ and tangent hyperbolic function is defined as $f(x) = (e^x - e^{-x})/(e^x + e^{-x})$ respectively.

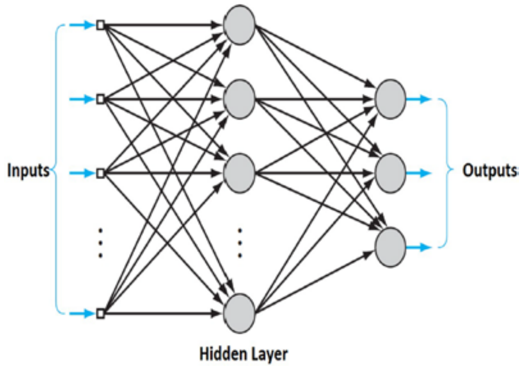


Figure 1 SLNN model

2.2. Functional-Link Neural-Networks

FLNN is a simple and efficient function approximator due to its implementation and accurate function approximation capability. Fundamentally, according to the function approximation theory, FLNN can represent any nonlinear function by a small approximation error using enough number of basis functions. In fact, FLNNs are similar to the SLNNs however the initial layer parameters of the FLNNs are unity. A multi-input single-output FLNN model is illustrated in Figure 2. The input-output (x, y) relation is given as,

$$y = w_o T(x) \quad (2)$$

Where w_o is the design parameters to optimize using least-squares estimation (LSE) etc. A polynomial basis function $T(\cdot)$ is used as the activation function of a FLNN. Chebyshev and Legendre polynomial basis are efficient to be used as activation function $T(\cdot)$. The first five functions for the Chebyshev polynomials of first kind are given as $T(x) = \left[1,\ x,\ 2x^2-1,\ 4x^3-3x,\ 8x^4-8x^2+1\right]$ where as first five functions of Legendre polynomials are defined as

$L(x) = \left[1,\ x,\ \frac{1}{2}\left(3x^2-1\right),\ \frac{1}{2}\left(5x^3-13x\right),\ \frac{1}{8}\left(35x^4-30x^2+3\right)\right]$ respectively.

2.3. Pi-Sigma Neural-Networks

High-order neural networks are also single-layered ANN models, and have likewise fast learning capacity, strong approximation, large storage capacity, high fault tolerance, and accurate mapping capability [15]. Recently, a new Pi-Sigma neural-network has been proposed and used for the prediction of time series with optimization of internal weights in [16].

In the PSNN model, the multiplication of the linear combinations of the network inputs construct the output of the network. The linear combination of the network inputs shows the degree of the PSNN. When the problem definition is highly nonlinear and complex, in order to model the input-output behaviour, there is need a relatively large degree. The large degree of PSNN provides better prediction results, but needs more computational time of training due to the over-fitting. For standard regression data, small number of the linear combination might be enough for the output modelling. The PSNN model with n inputs and m outputs are illustrated in Figure 3. The linear combination of the inputs are calculated by tunable W_i parameter and biases parameter vector B is added to calculate the outputs of the hidden neuron cells. The jth neuron output of the hidden-layer is calculated as,

$$h_k = f_h\left(\sum_{i=1}^{n} W_{ik} x_i + B_k\right), k = 1 \ldots H \quad (3)$$

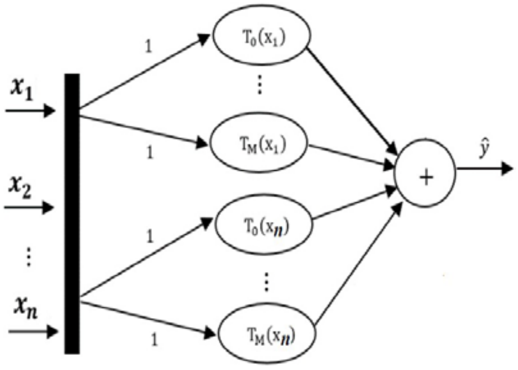


Figure 2 FLNN model

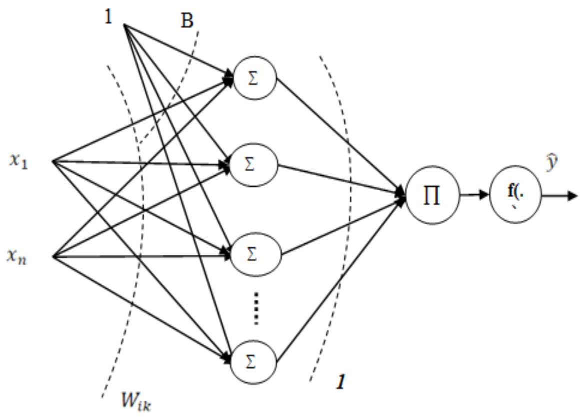


Figure 3 PSNN model

Where f hidden activation function is selected as a linear activation function as $f_h(x) = x$. However, the outputs are calculated using logistic activation function $f_h(x) = 1/(1+\exp(-x))$ as,

$$h_k = f_h(\prod_{k=1}^{H} h_k) = \frac{1}{1+\exp(-\prod_{k=1}^{H} h_k)}, r = 1 \ldots m \quad (4)$$

In Figure 3, the neuron activation functions of hidden layer use linear combination of previous features then summation passes through a linear activation function. The outputs of the hidden layer neurons are multiplied to construct the output of the network where these outputs are passed through a nonlinear activation function such as logistic, tangent hyperbolic etc. The most important difference is that the w_o weight parameters of the PSNN are fixed constant and not trained. Input data of the designed models are normalized to $[0,1]$ interval and LSE optimization is used to optimize output parameters.

3. Proposed Feature Extraction

Proposed feature extraction has two parts such as fuzzy clustering based fuzzy inference and symbolic regression layers. It is considered that the modelling performance of shallow models can be much more improved when it is used as a feature extraction layer for time series prediction and system identification.

3.1. Fuzzy Preprocessing

The fuzzy clustering is an unsupervised learning method that provides cluster centres and membership values of each input data pair. Without normalization of the input data, the cluster centres can be found and each input will have a degree of membership between [0, 1]. If fuzzy inference methods are used based on these membership values, there is obtained fuzzy clustering based fuzzy system design. Normally when the output centres of the singleton membership functions are determined by an optimization method, the fuzzy system design is completed. However, as known from the conventional fuzzy neural networks, the model can be enlarged to be suitable for different purposes. Therefore, in the present work, fuzzy system model is first continued by a symbolic regression model to be able to extract features then a decision layer is added as an output layer to the complete models. Symbolic function model is explained in next section. Fuzzy C-means clustering [17] determines the values of centres and corresponding memberships as,

$$\mu_{ik}=\frac{1}{\sum_{j=1}^{C}\left(\frac{\|x_k-c_i\|}{\|x_k-c_j\|}\right)^{2/(m-1)}},1\le i\le C;c_i=\frac{\sum_{i=1}^{N}(\mu_{ik})^{m_{xk}}}{\sum_{i=1}^{N}(\mu_{ik})^{m}},1\le k\le N \quad (5)$$

To optimize a cost function,

$$J_m=\sum_{i=1}^{N}\sum_{j=1}^{C}(\mu_{ik})^m\|x_k-c_j\|^2,1<m<\infty \quad (6)$$

Where $x_k,1<=k<=N$ is input value, $\mu(\cdot)$ is a membership function, $c_i,1<=i<=C$ are the obtained centers, $J_m,1<m<\infty$ cost function corresponding to the each m batch iteration. The iteration is ended when $\|U^{j+1}-U^j\|<\varepsilon$, where the U matrix is structured with memberships of input values. ε is a small constant to stop clustering. Mamdani-type fuzzy rule base is designed based on the clustering where jth rule is,

$R^j: IF x_i$ is A_1^j and ... and x_i is A_M^j THEN y is B^j (7)

The fuzzy basis function vector is designed using product inference engine, singleton fuzzifier and center average defuzzifier as,

$$\phi_j(x_i)=\frac{\prod_{i=1}^{N}\mu_{A_i^j}(x_i)}{\sum_{j=1}^{C}\prod_{i=1}^{N}\mu_{A_i^j}(x_i)} \quad (8)$$

Cluster centres are used for Gaussian membership functions with unity standard-deviation as,

$$\mu_{A_i^j}(x_i)=e^{-\frac{1}{2}\frac{(x_i-c_j)^2}{\sigma_j^2}} \quad (9)$$

For the ith input data and jth cluster centre. Fuzzy clustering based fuzzy system design is in fact the first feature extraction layer. Fuzzy inference transforms the input space to a nonlinear space with automatically found centres that provides to extract the hidden features according to the centre values such that each input can have a representation level at the output. Then, outputs of fuzzy basis functions pass through a symbolic function layer given in next section.

3.2. Symbolic Regression Layer

Symbolic regression layer has a one-hidden layer structure such that there is an analytical operator and function randomly chosen in the training step. These analytical operators and functions have very different characteristics taken from a predefined sets. Main difference of the proposed symbolic regression layer from the conventional symbolic regression is that the analytical operators and functions are determined randomly from predefined sets not using genetic algorithm etc. The second difference is that the structure of the symbolic regression is fixed here but conventional symbolic regression has too much nested operators and functions. The symbolic regression layer is here only provides features but these features passes through a decision layer. The output parameters are optimized by one-step estimation method. Therefore, optimal regression model is obtained with less complex structure with optimal estimated parameters. In addition, the random selection of operators and functions provides unbiased features at each training step.

Table 1. Some operators and functions

Operator	Function
$+$	x^2
$-$	$\sqrt{x}$
$\times$	$\cos(x)$
$min(x1, x2)$	$\tanh(x)$
$\max(x1, x2)$	$1/(1+e^{-x})$
$\mathrm{mod}(x1, x2)$	$\max\{0, x\}$

3.3. Design of Shallow Networks with Feature Extraction

The proposed feature extraction is located in front of the SLNN and FLNN models. The fuzzy clustering and fuzzy design parts are performed and initial features extracted at one step. However, the optimal design analytical operators and functions are based on the random learning theory. The type of operator between two inputs and the symbolic function that they passed through are determined randomly then the features are extracted using the training data. After that the network parameters are optimized. The SLNN model needs to design both input and output layer parameter where input layer are parameters are non-linearly formulated therefore complete set of parameters are optimized using Levenberg-Marquardt optimization. However, FLNN model has only output parameters to design therefore LSE method is used to optimize.

Algorithm 1 Design of shallow networks with feature extraction.

Begin: Training
Find: cluster centres using fuzzy clustering
Design: fuzzy system using Gaussian MFs
Construct: first feature matrix
for Number of random trials **do**
  Select: random operators and functions
  Construct: second feature matrix
  while up to stopping criterian **do**
    -LM optimization: SLNN and PSNN design
    -LSE method: FLNN design
  end while
  Record: MSE and optimized network parameters
end for
Find: optimal MSE values
Save: optimal symbolic operators and functions, and parameters
End: Training

Begin: Testing
Construct: first feature matrix via cluster centres and fuzzy system
Construct: second feature matrix via optimal operators and functions
Obtain: output values with optimized network parameters
Record: testing MSE values.

23: **End: Testing**

4. Computational Results

The advantages of proposed feature extraction method is presented with various application results. An artificial nonlinear system and inverted pendulum output identification are achieved as identification applications, respectively. In the regression part, sunspot time series prediction and traffic density estimation are performed for the time-series prediction. Prediction of sunspot time series and traffic flow estimation are relatively difficult problems due to the unpredictable or fast change of the dynamics. However, input-output data of the system identification problem is produced from a casual and deterministic system, therefore it is relatively easy problem. However, testing parts have different dynamics that makes difficult to predict future behaviour. The performance results with and without the feature extraction method are given in Table 2 and in Table 3, respectively. MATLAB software is used for designing and producing the application results of the proposed methods. For identification, the models are assumed as nonlinear auto-regressive moving average with exogenous input (NARMAX) form $\hat{y}(k) = Model(u(k), y(k-1))$, whereas for time series prediction in NARMA form as $\hat{y}(k) = Model(y(k-1), y(k-2))$ where two passed values are used for one-step ahead predictions.

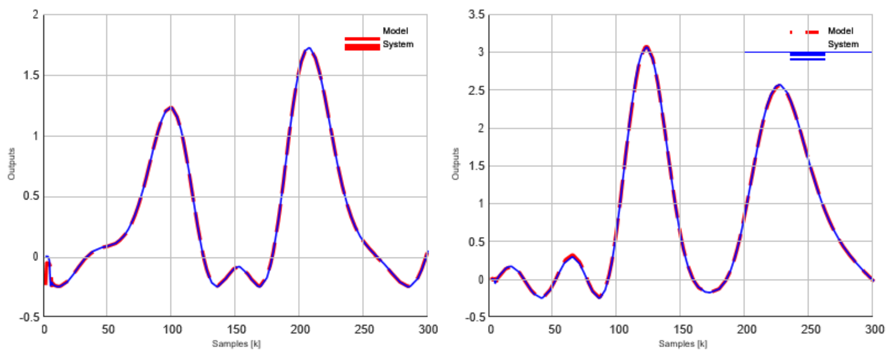
4.1. Nonlinear Dynamic System Identification

Identification of nonlinear systems is crucial problem for the control theory. The prediction of unknown system behaviour is used to monitor unmeasured dynamics and may also be used to produce a suitable control signal in predictive manner. In literature, there are many benchmark datasets for testing and comparing the developed models. The mathematical dynamics of a nonlinear, dynamic, casual, discrete-time and time-delayed system are given as,

$$
\begin{aligned}
u(k) &= sin(2\pi k / 300), \\
x(k) &= 0.4 sin(0.9u(k)) + cos(2.3u(k)) sin(1.2u(k)) - cos(2u(k)) (training), \\
x(k) &= cos(1.3u(k)) sin(2u(k)) - cos(0.6u(k)) + 0.9 sin(2.5u(k)) (testing), \\
q(k) &= x(k-D) + 0.35q(k-1), \\
y(k) &= q(k) + q(k)^2
\end{aligned}
\tag{10}
$$

Where (u(k), y(k)) is the input-output pair and (x(k), q(k)) are internal dynamics of the system where D is the time delay of the internal dynamics. The difficulty comes with time delay D and different behaviour of internal dynamics in the training and testing parts.

Figure 4 presents the identification results corresponding to the training and testing data sets. The data sets are produced by using Eq. (10). In fact, this nonlinear system is one of the benchmark systems such that it was already utilized in [18] where the test performance obtained as 2.2e-4MSE or 14.8e-3RMSE. As given in Table 2, the RMSE performance is also improved by using FLNN model.



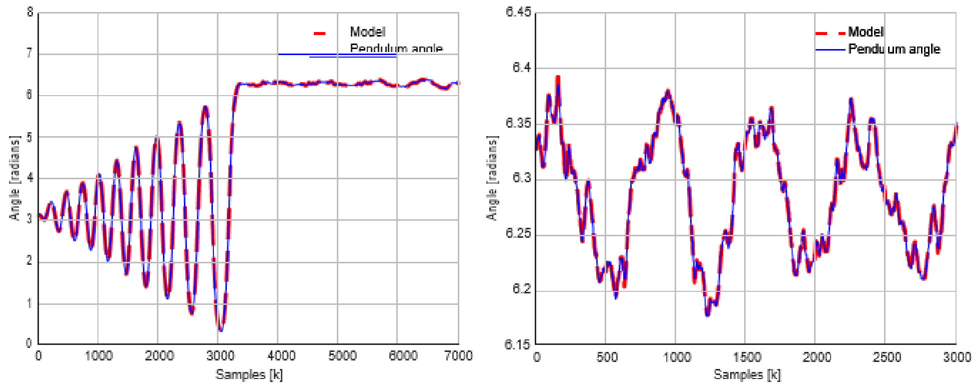
(a) Training results. (b) Testing results.

Figure 4 Identification of a nonlinear dynamic system

Figure 5. Inverted pendulum system

4.2. Inverted Pendulum Identification

The control of inverted pendulum is also one of the benchmark problems due to the nonlinear and unstable dynamics [19]. A suitable control voltage is applied to the chart table to hold the pendulum end-effector to the up-right position. Experimental system and applied control voltage are seen in Figure 5. Designed models are used to approximate the input output relationship of the pendulum so that pendulum dynamics are assumed as unknown. The past values of control voltages and angle positions are used as inputs of the models but future values of the angle positions are used the output of the designed models. The input-output data is produced from the experimental system that is used in control laboratory for testing the control approaches.



(a) Training data. (b) Testing data.

Figure 6. Identification of an inverted pendulum

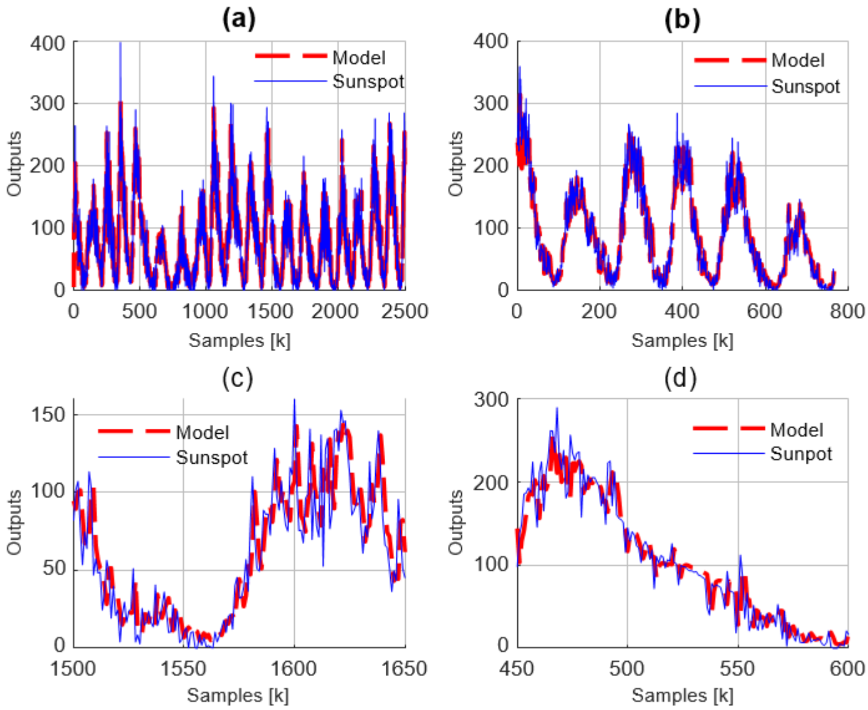


Figure 7 Prediction of sunspot time series

4.3. Sunspot Prediction

Sunspot data is one of the well-known benchmark datasets [20]. It is one of the benchmark time-series data to predict since there is no any input-output relationship for the future values. From 1749 to 2021, there exist 3265 sunspot data which is monthly recorded. 2500 values of the time series data is used for training the designed models whereas remaining 765 of the data is used for testing. Prediction results are shown in Figure 7, respectively.

Figure 7(a) and Figure 7(c) represents the training data predictions. Due to the large number of the data in Figure 7(a), a part of the data is detailed in Figure 7(c). The comparative results of the models are given in the tables. The sunspot time series data is downloaded from a public site.

4.4. Traffic Flow Density Prediction

Traffic flow or density estimation has recently become an important problem of the crowded cities. Based on the accurate prediction of traffic density, the traffic signalization can be automated instantaneously. Fine automatization of the traffic lights enhance the quality of people and reduce the energy consumption in general.

Even though the data has a periodicity, the magnitude of the density at periods varies in a highly nonlinear form. There are numerous unpredictable reasons that traffic density can vary everyday at the same traffic lights. The traffic density data is also downloaded from a public site.

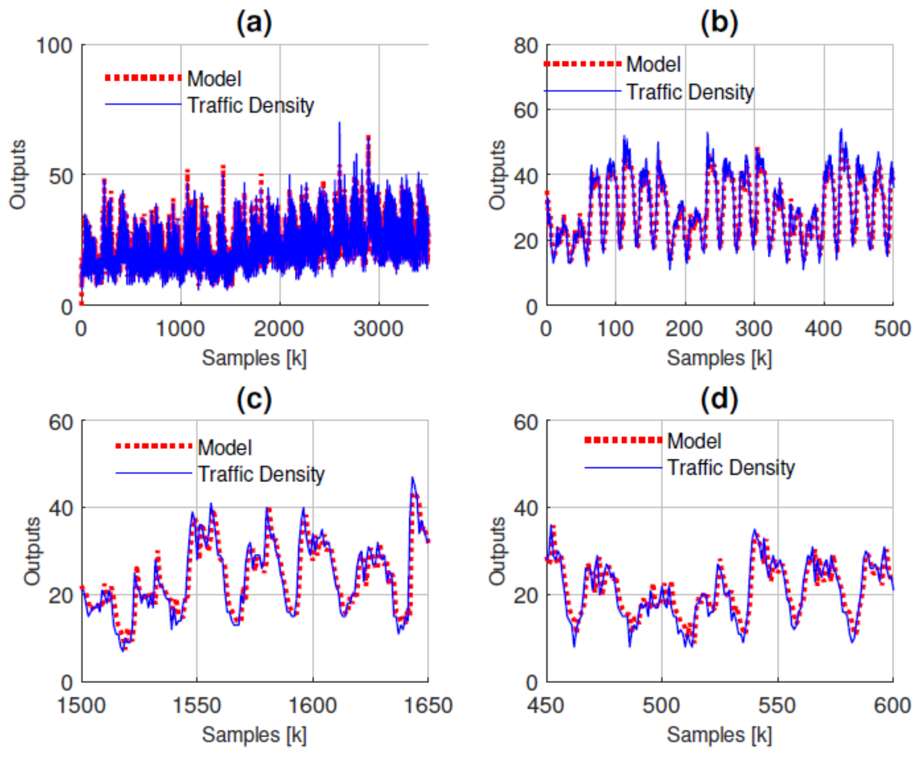


Figure 8 Prediction of traffic flow density

Figure 8 presents the predictions of traffic density results. Figure 8(a) and Figure 8(c) show training data predictions where some parts of the data zoomed in to examine the application results in detail. Figure 8(b) and Figure 8(d) illustrate the prediction results of testing data.

5. Discussion and Conclusion

In this study, a new feature extraction method is proposed for better approximation of shallow neural-network models for the system identification and regression. In general, we can say that the method is successful by looking at the performance results. However, as with all developed methods, we can criticize the advantages and disadvantages of this method from different perspectives. The proposed feature extraction methodology can be evaluated using the design steps, the computational load and the performance results.

First of all, the proposed feature extraction method consists of known and novel parts. The known part is the fuzzy inference part based on fuzzy clustering, the proposed part is a new symbolic regression layer. Here, it is seen that the symbolic regression part is designed differently based on the studies in the literature. The symbolic regression part is based on the random learning method as in extreme learning machines, unlike the symbolic regression methods in the literature. The random learning method has made many of regression methods very successful so that the symbolic regression part is designed based on random learning. The fuzzy inference part is obtained by data clustering and passing through Gauss membership functions in cases where the data is unknown. The symbolic regression part is the extraction of features with the help of randomly assigned operators and functions based on a batch learning. These features will then be applied as an input to the appended shallow neural-network model. Therefore, a two-stage approximation method is proposed. If the proposed method is criticized in terms of computational load, as expected, a little more time is required for the design compared to the methods designed in one step in the literature. But even with low computer configuration this is only in the order of seconds and minutes. If we compare it with the deep learning methods in the literature, which take a long time, the design is completed in a much shorter time. In terms of performance, it is observed that there is a dramatic performance change in the shallow models designed with feature extraction and in cases where it is not. For some data, it was observed that the performance did not increase much. This can be considered by considering the data and the shallow model. In general, the advantage of the developed method can be said that the regression performance increases significantly, as well as the disadvantage that the computational load increases compared to one-step designs.

Table 2. Performance of models with feature extraction

Method/RMSE	Dynamic System Identification	Inverted Pendulum Identification	Sunspot Prediction	Traffic Density Estimation
SLNN	20.6e-3 88.7s	6.9e-3 123.2s	16.88(4.23%) 352.7s	3.00(4.29%) 218.8s
FLNN	8.1e-3 11.40s	2.0e-3 63.90s	16.81(4.21%) 53.06s	2.8(4.0%) 122.2s
PSNN	91e-3 43.06s	78e-3 153.1	16.89(4.23%) 161.06	3.00(4.29%) 101.4

Table 3. Performance of models without feature extraction

Method/RMSE	Dynamic System Identification	Inverted Pendulum Identification	Sunspot Prediction	Traffic Density Estimation
SLNN	34.8e-3 4.05s	7.2e-3 37.34s	17.02 (4.27%) 77.7s	3.37 (4.8%) 146.4s
FLNN	24.6e-3 0.062s	4.61e-3 0.31s	16.93 (4.25%) 0.058s	5.81 (8.3%) 0.070s
PSNN	0.13 1.66s	0.10 51.07s	17.43 (4.84%) 39.26	4.5 (6.42%) 24.81s

In general, these shallow neural-networks are simple models for the difficult regression or identification problems. Therefore, in this paper, the main purpose is to improve their performance to be able to apply in the industry not to compare with advanced models. Anyone in this field can use and improve the performance of their much advanced models. It is also expected that different versions of the proposed feature extraction method will be conveniently used in the prediction of different regression problems.

Conflicts of Interest

The authors declare that they do not have any conflict of interest regarding the work.

References

[1] S. S. Alahmari, D. B. Goldgof, P. R. Mouton, and L. O. Hall, "Challenges for the repeatability of deep learning models," IEEE Access, vol. 8, pp. 211 860–211 868, 2020.

[2] X. Feng, Y. Jiang, X. Yang, M. Du, and X. Li, "Computer vision algorithms and hardware implementations: A survey," Integration, vol. 69, pp. 309–320, 2019.

[3] A. Ananthakrishnan and M. G. Allen, "All-passive hardware implementation of multilayer perceptron classifiers," IEEE Transactions on Neural Networks and Learning Systems, 2020.

[4] H. Younes, A. Ibrahim, M. Rizk, and M. Valle, "Efficient fpga implementation of approximate singular value decomposition based on shallow neural networks," in 2021 IEEE 3rd International Conference on Artificial Intelligence Circuits and Systems (AICAS). IEEE, 2021, pp. 1–4.

[5] J. Yan, X. Guo, S. Duan, P. Jia, L. Wang, C. Peng, and S. Zhang, "Electronic nose feature extraction methods: A review," Sensors, vol. 15, no. 11, pp. 27 804–27 831, 2015.

[6] S. Sachar and A. Kumar, "Survey of feature extraction and classification

techniques to identify plant through leaves," Expert Systems with Applications, vol. 167, p. 114181, 2021.

[7] J. R. Koza and J. R. Koza, Genetic programming: on the programming of computers by means of natural selection. MIT press, 1992, vol. 1.

[8] J. Madár, J. Abonyi, and F. Szeifert, "Genetic programming for the identification of nonlinear input- output models," *Industrial & engineering chemistry research*, vol. 44, no. 9, pp. 3178–3186, 2005.

[9] J. Kubalík, E. Derner, and R. Babuška, "Multi-objective symbolic regression for physics-aware dynamic modeling," *Expert Systems with Applications*, vol. 182, p. 115210, 2021.

[10] A. Diveev, S. Konstantinov, and A. Danilova, "Solution of the optimal control problem by symbolic regression method," Procedia Computer Science, vol. 186, pp. 646–653, 2021.

[11] G. H. Yamashita, F. S. Fogliatto, M. J. Anzanello, and G. L. Tortorella, "Customized prediction of attendance to soccer matches based on symbolic regression and genetic programming," Expert Systems with Applications, vol. 187, p. 115912, 2022.

[12] B. E. Köktürk-Güzel and S. Beyhan, "Symbolic regression based extreme learning machine models for system identification," *Neural Processing Letters*, vol. 53, no. 2, pp. 1565–1578, 2021.

[13] F. O. de Franca and M. Z. de Lima, "Interaction-transformation symbolic regression with extreme learning machine," Neurocomputing, vol. 423, pp. 609–619, 2021.

[14] S. Beyhan, "Fuzzy emulated symbolic regression for modelling and control of markov jump systems with unknown transition rates," IEEE Transactions on Circuits and Systems II: Express Briefs, 2021.

[15] Y. Shin and J. Ghosh, "The pi-sigma network: An efficient higher-order neural network for pattern classification and function approximation," in IJCNN-91-Seattle international joint conference on neural networks, vol. 1. IEEE, 1991, pp. 13–18.

[16] E. Egrioglu and E. Bas, "Modified pi sigma artificial neural networks for forecasting," Granular Computing, pp. 1–5, 2022.

[17] S. Beyhan and M. Alci, "Fuzzy functions based arx model and new fuzzy basis function models for nonlinear system identification," Applied Soft Computing, vol. 10, no. 2, pp. 439–444, 2010.

[18] M. Alci and M. H. Asyali, "Nonlinear system identification via laguerre network based fuzzy systems," Fuzzy sets and systems, vol. 160, no. 24, pp. 3518–3529, 2009.

[19] M. no. 33-936S Ed01 122006, "Manual on digital pendulum control experiments," 2011.

[20] Y. Xixi, D. Fengqian, and L. Chao, "Time series prediction based on high-order intuitionistic fuzzy cognitive maps with variational mode decomposition," Soft Computing, vol. 26, no. 1, pp. 189–201, 2022.