

Evaluation of Machine Learning Hyperparameters Performance for Mice Protein Expression Data in Different Situations

Cafer Tayyar Bati^{1*} , Gazel Ser² 

^{1*} Van Yuzuncu Yil University, Institute of Science, Department of Animal Science, Van, Turkey. (e-mail: cafertayyarbati@gmail.com).

² Van Yuzuncu Yil University, Faculty of Agriculture, Department of Animal Science, Van, Turkey. (e-mail: gazelser@gmail.com).

ARTICLE INFO

Received: Jan., 27. 2021

Revised: Jul., 07. 2021

Accepted: Dec., 29. 2021

Keywords:

Artificial neural networks

Hyperparameter

Machine learning

Mice

Protein

Support vector machines

Corresponding author: *Cafer Tayyar Bati*

ISSN: 2536-5010 | e-ISSN: 2536-5134

DOI: <https://doi.org/10.36222/ejt.869094>

ABSTRACT

In this study, the aim was to assess the effect and significance of hyperparameters in four different datasets containing different values for observation numbers and variable counts with the machine-learning methods of support vector machines and artificial neural networks. With this aim, a dataset comprising 15 repeats of 77 protein levels from 38 healthy and 34 down syndrome mice was used. A total of 138 different models and model classification performance criteria were obtained from the datasets in the study comprising combinations of hyperparameters in machine-learning methods. Comparison of the models used criteria like accurate classification percentage, kappa statistic, mean absolute error and square root of mean error squares. According to performance criteria, the first dataset with 1080 observations x 77 variables had 71.30% accurate classification percentage for assumed parameters with the support vector machines polynomial kernel function, while changing the hyperparameter variables increased this rate to 99.44%. Similarly, the second dataset had 50.65% accurate classification percentage with the artificial neural network single hidden layer 2 neuron model, while changing the hyperparameter values increased this rate to 90.46%. In conclusion, in situations with low variable and observation numbers, the machine learning methods were determined to display lower performance. However, in datasets, it is very important for classification performance in artificial neural networks and support vector machines, especially polynomial and radial basis function kernel functions, to set hyperparameters according to the dataset. In situations with low variable numbers, especially, the effect of hyperparameters was determined to gain importance.

1. INTRODUCTION

With machine learning methods algorithms developed for complicated and large data can perform classifications with a high degree of accuracy [1]. In procedures for processing and assessing data at increasing scales, machine learning methods provide a transition from parametric methods to nonparametric methods. Due to machine learning, computers may be programmed to optimize a process based on previous experience or datasets. Thus, dataset classifications can be performed effectively in a short time, a model may be created at the end of this process and this model may be used for estimations about the future and auditing purposes [2].

Machine learning algorithms require adjustment before operation and include ‘hyperparameters’ with no clear defaults acceptable across a wide range of applications. Examples of these hyperparameters that require determination for these algorithms are the depth of a decision tree, number

of trees in the forest and number of neurons in each layer of an artificial neural network. These parameters have critical importance for machine learning because different hyperparameters generally result in performances with a significant degree of difference [3]. The ideal settings for hyperparameters do not just determine the performance in the training process, but more importantly, determines the quality of the prediction models which emerge [4,5]. Currently, though there are many studies about the importance of hyperparameters in the machine learning field [6-8], studies about the variability displayed in the effect of hyperparameters linked to factors like the structure and dimension of the dataset are limited [9]. In addition, it has been stated in many studies [10-13] on machine learning that artificial neural networks and support vector machines are very successful methods for classification and estimation, as well as the presence of more than one hyperparameter in methods has been effective in the preference of methods in the

study. In this context, this study aimed to assess the effect of hyperparameters in datasets in four different situations where observation numbers and variable numbers have different values.

2. MATERIAL AND METHOD

The ‘mice protein expression’ dataset used in the study was obtained from the UCI machine learning data repository [14]. This dataset created by Higuera *et al.* [15] contains 15 repeated measurements for 77 protein levels in 38 healthy and 34 Down syndrome mice. The dataset is divided into 8 classes according to the genotype of the mice (control (c), trisomy (t)), behavior (context-shock (CS), shock-context (SC)) and treatment (memantine (m), saline (s)). In total, there are 1080 observations for each protein. This dataset was chosen for the study due to its containing several features together like sufficient observations (1080), high numbers of variables (77), repeated measures (15) and sufficient class numbers (8).

2.1. Data analysis scenario

Four different datasets were created (Table 1), with different combinations of hyperparameters (Table 2) used (with grid search) and 138 different models were generated. Additionally, 10-fold cross-validation was used against overfitting situations that may occur in the models. In addition, the percentage of correct classification, kappa statistic, mean absolute error and root mean square error performance criteria were obtained (Table 3). Analyses were performed in the Weka (Waikato Environment for Knowledge Analysis, Version 3.8.1) program [16].

TABLE 1
DATA SETS USED IN THE STUDY

Data set	Number of Observations	Variable Number	Explanation
Data set I	1080	77	Original data set
Data set II	1080	9	By applying principal component analysis to the first data set,
Data set III	72	77	Deleting 14 repeated measurements of each individual in the first data set,
Data set IV	72	9	By applying principal component analysis to the third data set,

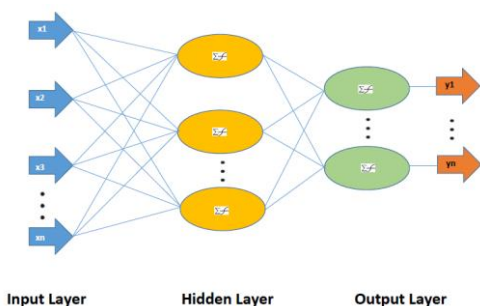


Figure 1. Multilayer perceptron

HYPERPARAMETERS USED IN THE STUDY		
Algorithm	Hyper Parameters*	
Support Vector Machines	Linear Kernel	C
	Polynomial Kernel	C and D
	Radial Basis Function Kernel	C and γ
Artificial Neural Networks	Hidden layers	
	The number of neurons in each layer	

* C: Editing parameter (1, 10, 100), d: Polynomial degree (2, 3, 5), γ : Kernel size (0.01, 0.1, 1, 10).

TABLE 3
PERFORMANCE METRICS [17]

Performance Metrics	Notation *
Correct classification percentage	$Accuracy = \frac{(T_p + T_n)}{(T_p + F_p + F_n + T_n)}$
Kappa statistics	$\mathcal{K} = \frac{(P_o - P_e)}{(1 - P_e)}$
Average absolute error	$MAE = \frac{1}{n} \sum_{t=1}^n x_t - \hat{x}_t $
Square root mean square error	$RMSE = \sqrt{\frac{\sum_{i=1}^n (x_t - \hat{x}_t)^2}{n}}$

* $T_p + T_n$: Number of correctly classified data, $F_p + F_n$: Number of incorrectly classified data, P_o : Accepted rate, P_e : Expected rate

2.2. Methods used in the study

Machine learning is an artificial intelligence application completing learning and development operations from available data without openly programming the computer. Machine learning searches for patterns in data based on data provided for training and applies these patterns to test data in order to make better decisions in the future. The purpose of these methods is to ensure computers learn automatically without human intervention. Artificial neural networks consider calculation units in a learning model as resembling human neural cells, mimicking human neural systems for machine learning tasks [18]. The greatest task of artificial neural networks is to create machines with artificial intelligence mimicking the architecture of the human neural system for calculations. The currently-used multilayer perceptron (Figure 1) comprises an input, hidden layers and output layer, different from a single-layer perceptron. Linked to the complexity of the problem to be solved, the number of hidden layers and the number of neurons in these layers may vary [19]. In a multilayer perceptron, information in the dataset is sent to the input layer, here some analyses are performed and then data is transferred to the next layer and here the number of layers changes according to the complexity (data dimensions, number of variables, relationships between variables, etc.) and dimensions of the dataset to be studied, with the condition that the number of layers is at least one. As the layers are interconnected, the output value obtained from one layer is organized as the input for another layer. The neurons in the layer act in connection with each other. The number of neurons is determined linked to the complex structure of the dataset. The final layer comprises an output layer where information or data from the input and hidden layers are operated. Figure 1 shows a multilayer perceptron model. The oval shapes in Figure 1 represent neurons, organized into output layers and hidden layers. Signals

TABLE 2

(values) from the input are subjected to a weighted total activation function (transfer function). In the figure, the curve shows the activation function (sigmoid) for each neuron. Support vector machines (SVM) are controlled machine learning algorithms that can be used for both classification and regression problems [20]. The basic function of this algorithm is to draw margins between two or more groups found on a plane and separate the groups from each other. This algorithm works with the logic that the drawn margin should be at the most distant point from the group members [21]. Currently, SVMs are used successfully for many real-world problems like classification of text and images, bioinformatic classification and detecting characters in handwriting [22]. SVM finds the data points in different classes and attempts to draw lines between them. The chosen data points are called support vectors, while the boundaries are called hyperplanes. The algorithm pays attention to each data pair until the closest pair in each class is found and a straight line (or plane) is drawn between them. In situations where input data can be linearly separated creating a hyperplane is simpler. However, generally, the classification regions overlap and it appears that no single hyperplane can complete the boundary function. In this situation, support vector machines create kernel functions to reflect the data at higher dimensions and complete the classification process [22].

The decision function for a two-class problem that can be linearly separated can be written by Equation 1 [21].

$$f(x) = \text{sign}(\sum_{i=1}^n \alpha_i y_i(x \cdot x_i) + b) \quad (1)$$

In situations where two classes cannot be clearly divided along a broad margin, the ξ_i parameter symbolizing crossed margins enters the target function and the aim is to minimize this parameter. In this situation when classes cannot be divided in a linear way, Equation 2 minimizes this. This equation is called the soft margin. Additionally, the C parameter in the equation is an equilibrium parameter balancing margin and misclassification rates. The larger the C value, the larger the importance given to the error; in other words, the method is more sensitive to variations in data compared to the available data [23].

$$P = \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n \xi_i, \quad (y_i(w x_i + b) \geq 1 - \xi_i \text{ ve } \xi_i \geq 0) \quad (2)$$

In nonlinear support vector machines, in situations where it is not possible to perform linear classification, data is converted to a higher dimension based on the principle of dividing two classes with a hyperplane. The conversion to higher dimensions uses functions with various features. These functions are called kernel functions [23, 24].

The inner product is used for linear classifiers ($K(x_i, x_j) = x_i^T x_j$). If each data point is mapped to the higher-dimensional

attribute space by some transformations ($\Phi: x \rightarrow \varphi(x)$), then the inner product can be shown by Equation 3.

$$K(x_i, x_j) = \varphi(x_i)^T \varphi(x_j) \quad (3)$$

In Equation 3, $K(x_i, x_j)$ is the kernel function and in low-dimensional input space, it is sufficient to calculate this function instead of the high dimensional inner product [25].

The kernel functions used in the study are given in Table 4.

TABLE 4
KERNEL FUNCTIONS USED IN THE STUDY

Kernel	Notation
Linear	$k(x_i, x_j) = x_i^T x_j$
Polynomial	$k(x_i, x_j) = (1 + x_i^T x_j)^d$
Radial Basis Function	$k(x_i, x_j) = e^{-\gamma \ x_i - x_j\ ^2}, \quad \gamma > 0, \quad \gamma = 1/(2\sigma^2)$

3. RESULT AND DISCUSSION

The study examined 138 different models created from four different datasets with combinations of hyperparameters in machine learning methods. The results for the most successful, least successful and default models for the datasets and algorithms related to these models are given in Tables 5, 6, 7, and 8.

According to Table 5, the model obtained by changing the parameters of the default parameter model for the linear kernel function was not observed to cause a large change. Additionally, in terms of the use of the C parameter in the linear kernel, increasing the value of this parameter did not contribute to improving the model performance; in fact, it lowered model performance in some datasets (II and IV). This situation may be expected considering the low number of variables in these datasets. The C parameter is an equilibrium parameter balancing margin and misclassification rates, so considering large C values are more sensitive to variation in data [23], the use of small C values is required for lower variations. Additionally, more complicated decision curves may be obtained attempting to comply with all points in the data with large C values, so it should be considered that generalizing these decision curves to test data may be difficult [26]. Additionally, when the confusion matrix for model 1.1 is investigated, it appears 1045 mice were accurately classified (97% accuracy rate). However, it was determined that this model classified 8 mice in the memantine treatment group (c-CS-m) as the saline treatment group (c-CS-s) and 7 mice inversely as memantine group in spite of being in the saline treatment group.

TABLE 5
LINEAR KERNEL FUNCTION RESULTS FOR SUPPORT VECTOR MACHINES

Data Set	C ¹	Model No	Correctly Classified Instances	Accuracy (%)	Kappa Statistic	Mean Absolute Error	Root Mean Squared Error	Confusion Matrix ²
I	1	1.1*	1045	96.76	0.963	0.008	0.09	<pre> a b c d e f g h <-- 139 0 8 0 3 0 0 0 a 0 150 0 0 0 0 0 0 b 7 0 122 0 3 1 2 0 c 1 0 0 134 0 0 0 0 d 2 0 2 0 129 0 1 1 e 0 1 0 0 1 133 0 0 f 0 0 0 0 2 0 103 0 g 0 0 0 0 0 0 0 135 h </pre>
	10	1.2	1056	97.78	0.975	0.006	0.07	
II	1	2.1*	730	67.59	0.629	0.081	0.285	<pre> a b c d e f g h <-- 60 1 37 8 29 0 11 4 a 0 134 0 0 1 13 0 2 b 23 2 57 0 39 0 14 0 c 0 3 0 104 0 14 0 14 d 0 0 16 0 100 0 13 6 e 0 36 0 25 0 74 0 0 f 9 0 6 0 14 0 74 2 g 0 2 0 0 0 1 5 127 h </pre>
	100	2.3	691	63.98	0.588	0.09	0.300	
III	1	3.1*	42	58.33	0.523	0.104	0.323	<pre> a b c d e f g h <-- classified as 5 0 3 0 0 0 1 1 a = c-CS-m 0 9 0 0 0 1 0 0 b = c-SC-m 4 0 1 0 3 0 1 0 c = c-CS-s 0 0 0 7 0 1 0 1 d = c-SC-s 0 0 4 0 3 0 2 0 e = t-CS-m 0 3 0 2 0 4 0 0 f = t-SC-m 1 0 1 0 1 0 4 0 g = t-CS-s 0 0 0 0 0 0 0 9 h = t-SC-s </pre>
	10	3.2	43	59.72	0.539	0.100	0.317	
IV	1	4.1*	39	54.17	0.475	0.115	0.339	<pre> a b c d e f g h <-- classified as 3 0 5 0 0 0 1 1 a = c-CS-m 0 9 0 0 0 1 0 0 b = c-SC-m 4 0 1 0 2 0 2 0 c = c-CS-s 0 1 0 6 0 1 0 1 d = c-SC-s 0 1 4 0 4 0 0 0 e = t-CS-m 0 3 0 1 0 5 0 0 f = t-SC-m 2 0 0 0 1 0 3 1 g = t-CS-s 0 1 0 0 0 0 0 8 h = t-SC-s </pre>
	10	4.2	38	52.78	0.460	0.118	0.344	

¹C: Editing parameter, ²a=c-CS-m, b=c-SC-m, c=c-CS-s, d=c-SC-s, e=t-CS-m, f=t-SC-m, g=t-CS-s, h=t-SC-s, *: Default model.

According to Table 6, the model appeared to provide 71% success for classification with the default settings for dataset I, but changing the parameters increased the model performance to 99%. Additionally, changing the hyperparameter values for all datasets, especially dataset II, obtained very successful models, showing the effect of the hyperparameters is very significant. Additionally, when the confusion matrix for model 2.5 which accurately classified 616 mice is investigated, several of the most pronounced

errors in the model were classifying 85 mice from the saline treatment group (c-CS-s) as in the memantine treatment group (c-CS-m) and classifying 16 mice with context-shock behavior features as shock-context. Additionally, considering the use of the C parameter in the polynomial kernel in all datasets, the best performance was obtained when the C value was 100; additionally, the best performance was generally obtained when the polynomial degree was 3 (2nd degree in dataset III).

TABLE 6
POLYNOMIAL KERNEL FUNCTION RESULTS FOR SUPPORT VECTOR MACHINES

Data Set	C ¹	D ²	Model No	Correctly Classified Instances	Accuracy (%)	Kappa Statistic	Mean Absolute Error	Root Mean Squared Error	Confusion Matrix ³
I	1	2	1.4	745	68.98	0.64	0.08	0.28	
		3	1.5*	770	71.30	0.67	0.07	0.27	
I	100	3	1.11	1074	99.44	0.99	0.001	0.04	<pre> a b c d e f g h <-- 150 0 0 0 0 0 0 0 0 a 0 150 0 0 0 0 0 0 0 b 3 0 130 0 1 0 1 0 0 c 0 0 0 135 0 0 0 0 0 d 0 0 0 0 135 0 0 0 0 e 0 0 0 0 0 135 0 0 0 f 0 0 0 0 0 0 105 0 0 g 0 0 0 0 0 0 1 134 0 h </pre>
II	1	3	2.5*	616	57.04	0.506	0.107	0.328	<pre> a b c d e f g h <-- 121 0 0 25 0 0 1 3 1 a 0 95 0 33 0 22 0 0 0 b 85 0 31 16 1 0 2 0 0 c 0 0 0 134 0 1 0 0 0 d 35 0 4 0 93 0 3 0 0 e 0 15 0 82 0 38 0 0 0 f 55 0 7 3 14 0 26 0 0 g 3 6 0 46 0 2 0 78 0 h </pre>
		5	2.6	487	45.09	0.365	0.137	0.370	
III	1	2	3.4	22	30.56	0.198	0.174	0.417	
		3	3.5*	30	41.67	0.328	0.146	0.382	
III	100	2	3.10	43	59.72	0.538	0.107	0.317	<pre> a b c d e f g h <-- classified as 5 0 2 0 1 0 1 1 a = c-CS-m 0 9 0 0 0 1 0 0 b = c-SC-m 4 0 1 0 3 0 1 0 c = c-CS-s 0 1 0 6 0 1 0 1 d = c-SC-s 2 0 1 0 5 0 0 1 e = t-CS-m 0 3 0 1 0 5 0 0 f = t-SC-m 1 0 1 0 1 0 3 1 g = t-CS-s 0 0 0 0 0 0 0 9 h = t-SC-s </pre>
IV	1	3	4.5*	14	19.44	0.078	0.201	0.449	
		5	4.6	10	13.89	0.011	0.215	0.464	
IV	100	3	4.11	38	52.78	0.458	0.118	0.344	<pre> a b c d e f g h <-- classified as 7 0 3 0 0 0 0 0 a = c-CS-m 0 6 0 2 0 2 0 0 b = c-SC-m 3 0 3 0 2 0 1 0 c = c-CS-s 0 0 0 7 0 2 0 0 d = c-SC-s 2 0 1 0 3 2 1 0 e = t-CS-m 0 2 0 4 0 3 0 0 f = t-SC-m 3 0 1 0 1 0 2 0 g = t-CS-s 1 0 0 0 0 1 0 7 h = t-SC-s </pre>

¹C: Editing parameter, ²D: Polynomial degree, *: Default model, ³a=c-CS-m, b=c-SC-m, c=c-CS-s, d=c-SC-s, e=t-CS-m, f=t-SC-m, g=t-CS-s, h=t-SC-s

When the confusion matrix data for model number 3.21 given in Table 7 is investigated, it was determined that this dataset comprised 77 protein level measurements from 72 mice after removing repeated measures from the original dataset and that the model accurately classified 44 mice. However, it was

determined that this model classified 3 mice with trisomy genotype (t-SC-m) as control genotype (c-SC-m). Additionally, the model classified 2 mice in the trisomy genotype and memantine treatment group (t-SC-m) as control genotype and saline treatment group (c-SC-s).

TABLE 7
RBF KERNEL FUNCTION RESULTS FOR SUPPORT VECTOR MACHINES

Data Set	C ¹	Gamma ²	Model No	Correctly Classified Instances	Accuracy (%)	Kappa Statistic	Mean Absolute Error	Root Mean Squared Error	Confusion Matrix ³
I	1	0.01	1.13*	713	66.02	0.61	0.085	0.292	
	10	0.1	1.18	1080	100.0	1.0	0.0	0.0	<pre> a b c d e f g h <-- 150 0 0 0 0 0 0 0 a 0 150 0 0 0 0 0 0 b 0 0 135 0 0 0 0 0 c 0 0 0 135 0 0 0 0 d 0 0 0 0 135 0 0 0 e 0 0 0 0 0 135 0 0 f 0 0 0 0 0 0 105 0 g 0 0 0 0 0 0 0 135 h </pre>
II	1	0.01	2.13*	638	59.07	0.53	0.102	0.319	
	100	0.1	2.22	1064	98.52	0.983	0.003	0.061	<pre> a b c d e f g h <-- 149 0 0 0 0 0 0 1 a 0 146 0 0 0 4 0 0 b 2 0 131 0 0 0 2 0 c 0 0 0 134 0 1 0 0 d 0 0 0 0 135 0 0 0 e 0 4 0 0 0 131 0 0 f 1 0 0 0 1 0 103 0 g 0 0 0 0 0 0 0 135 h </pre>
III	1	0.01	3.13*	18	25	0.131	0.188	0.433	
	100	0.01	3.21	44	61.11	0.555	0.097	0.312	<pre> a b c d e f g h <-- classified as 6 0 2 0 0 0 1 1 a = c-CS-m 0 8 0 0 0 2 0 0 b = c-SC-m 4 0 2 0 2 0 1 0 c = c-CS-s 0 0 0 7 0 1 0 1 d = c-SC-s 2 0 0 0 4 0 2 1 e = t-CS-m 0 3 0 2 0 4 0 0 f = t-SC-m 2 0 0 0 1 0 4 0 g = t-CS-s 0 0 0 0 0 0 0 9 h = t-SC-s </pre>
IV	1	0.01	4.13*	18	25	0.131	0.188	0.433	
	100	0.1	4.22	43	59.72	0.539	0.101	0.317	<pre> a b c d e f g h <-- classified as 5 0 2 0 2 0 0 1 a = c-CS-m 0 8 0 0 0 1 0 1 b = c-SC-m 2 0 2 0 3 0 2 0 c = c-CS-s 0 2 0 5 0 1 0 1 d = c-SC-s 1 0 1 0 4 0 1 2 e = t-CS-m 0 1 0 1 0 7 0 0 f = t-SC-m 3 0 0 0 1 0 3 0 g = t-CS-s 0 0 0 0 0 0 0 9 h = t-SC-s </pre>
	10	4.24	9	12.5	-0.013	0.219	0.468		

¹C: Editing parameter, ²Gamma (γ): Kernel size, *: Default model. ³a=c-CS-m, b=c-SC-m, c=c-CS-s, d=c-SC-s, e=t-CS-m, f=t-SC-m, g=t-CS-s, h=t-SC-s

According to Table 7, the success rate for the default model for dataset I was 66%, and changing the parameters increased this rate to 100%. Additionally, all kernel functions used for dataset I displayed good performance, with the most successful model determined to be RBF (Tables 5, 6, 7). When the results for the support vector machines for the first dataset are assessed, especially adjusting the polynomial and RBF kernel parameter values according to the dataset was determined to significantly increase model performance. Additionally, changing the hyperparameter values in the RBF kernel function obtained very successful models among kernel functions used for dataset II; in this context, the most successful kernel was determined to be RBF (Model no: 2.22, accurate classification rate 99%, Table 7). When the results for the C parameter in the RBF kernel are investigated, the C value was 100 in the most successful models; additionally,

when the gamma value was 0.1 (gamma 0.1, dataset III) the most successful models were obtained in general. Based on these results, it was determined that the hyperparameter dataset was very important especially for polynomial and RBF kernel functions in support vector machines. Parallel to our study results, Van Rijn and Hutter [27] compared the performance of a variety of algorithms on several datasets and stated that the same hyperparameters were typically important for several datasets in these studies and that gamma and C parameters were very important for SVM. Additionally, when results for other datasets apart from the original dataset are investigated, it appears that the classification performance for models created with these datasets were lower compared to the default models. Machine learning algorithms require many data in order to learn the basic structure of data during the training process. The low dimensions of the dataset affect the

model performance and lower the chance of performing good generalization of the model. For this reason, higher amounts of training data in machine learning ensure higher performance [28, 29]. Therefore, this issue may be shown as the cause of the low default model performance in these data sets. Additionally, it was determined that the most successful models created with support vector machines were created with the RBF kernel. Additionally, the results obtained from the linear kernel for dataset I and II were determined to be very close to the RBF kernel results. This situation may be assessed as due to the excess number of variables in these two models

because the study by Hsu et al. [9] reported that in situations with high variable (feature) numbers, it is necessary to use the linear kernel. In their study, they stated that a large number of features also may not require data to be matched to a higher dimensional space and may improve non-linear mapping performance. They stated that RBF was as good as the linear kernel in this dataset; however, this function contains two parameters (C, gamma) and this increases operation load, while the use of the linear kernel containing only the C parameter is appropriate.

TABLE 8
RESULTS OBTAINED FROM ARTIFICIAL NEURAL NETWORKS

Data Set	Hidden Layer	Neuron Number	Model No	Correctly Classified Instances	Accuracy (%)	Kappa Statistic	Mean Absolute Error	Root Mean Squared Error	Confusion Matrix ¹
I	1	42	1.30*	1074	99.44	0.99	0.005	0.033	<pre> a b c d e f g h <-- 149 0 0 0 1 0 0 0 a 0 150 0 0 0 0 0 0 b 1 0 132 0 2 0 0 0 c 0 0 0 134 0 0 0 1 d 0 0 1 0 134 0 0 0 e 0 0 0 0 0 135 0 0 f 0 0 0 0 0 0 105 0 g 0 0 0 0 0 0 0 135 h </pre>
	2	2	1.31	567	52.5	0.45	0.139	0.266	
II	1	8	2.28*	918	85	0.828	0.049	0.179	<pre> a b c d e f g h <-- 103 0 16 1 20 0 10 0 a 1 133 0 0 0 15 0 1 b 12 0 114 0 1 1 7 0 c 0 0 0 112 0 9 0 14 d 0 0 7 0 121 0 5 2 e 0 19 0 4 0 112 0 0 f 3 0 4 0 4 2 90 2 g 0 2 0 0 0 0 0 133 h </pre>
	2	10	2.34	977	90.46	0.891	0.031	0.139	
III	1	10	3.29	50	69.44	0.650	0.102	0.242	<pre> a b c d e f g h <-- classified as 6 0 3 0 1 0 0 0 a = c-CS-m 0 8 0 2 0 0 0 0 b = c-SC-m 3 0 3 0 2 0 1 0 c = c-CS-s 0 0 0 9 0 0 0 0 d = c-CS-s 2 0 1 0 5 0 1 0 e = t-CS-m 0 1 0 1 0 7 0 0 f = t-SC-m 1 0 0 0 0 0 5 1 g = t-CS-s 0 1 0 0 0 1 0 7 h = t-SC-s </pre>
	2	42	3.30*	48	66.67	0.618	0.095	0.244	
IV	1	4	4.26	35	48.61	0.411	0.151	0.309	<pre> a b c d e f g h <-- classified as 5 0 2 1 0 0 2 0 a = c-CS-m 0 4 0 1 0 5 0 0 b = c-SC-m 2 0 3 0 4 0 0 0 c = c-CS-s 0 2 0 5 0 1 0 1 d = c-SC-s 2 0 0 0 6 0 0 1 e = t-CS-m 0 2 0 2 0 5 0 0 f = t-SC-m 3 0 0 1 2 0 1 0 g = t-CS-s 0 1 0 0 0 2 0 6 h = t-SC-s </pre>
	2	8	4.28*	31	43.06	0.347	0.150	0.322	
	2	2	4.30	10	13.89	0.001	0.219	0.331	

¹a=c-CS-m, b=c-SC-m, c=c-CS-s, d=c-SC-s, e=t-CS-m, f=t-SC-m, g=t-CS-s, h=t-SC-s, *: Default model.

Table 8 presents the results obtained from the artificial neural network. Accordingly, the classification performance for dataset IV was low, which again is assessed as linked to the low observation numbers and variable numbers in the dataset [28,29]. Additionally, the most successful model in this dataset was a single hidden layer and 4 neuron model number 4.26 (49% accuracy). When the confusion matrix for this model is investigated, 35 mice were accurately classified in the model; however, 5 mice with control genotype (c-SC-m) were classified as trisomy genotype (t-SC-m). Additionally, the most successful model for dataset I was model number 1.30 with 99% accurate classification rate. This model comprises a single hidden layer and there were 42 neurons in the hidden layer. This model is also the default model and the total number of neurons in the hidden layer is equal to the mean of the sum of the number of independent variables and the number of classes. This situation shows the importance of the appropriate setting of hyperparameters to the dataset in terms of model performance. An advantage of the Weka program used in the study is that the number of neurons in the hidden layer is automatically adjusted according to the dataset by taking as the default value [16]. The adjustment process generally causes better hyperparameter settings compared to the default values. Studies by Koch *et al.* [4] reported that even in situations where default settings provide good results, the hyperparameter adjustment process performs intuitive verification of these settings and has significant value in ensuring the construction of a model with higher accuracy.

4. CONCLUSION

According to the results obtained in our study;

- For all datasets, especially in polynomial and RBF kernel function support vector machines and artificial neural networks, the arrangement of hyperparameters according to the dataset is very important for classification performance,
- In situations with low numbers of variables and observations, machine learning methods displayed lower performance,
- In situations with a low number of variables, the effect of hyperparameters can be said to gain greater importance.

Generally, we recommend the following strategies for training of new networks due to these results;

Observation number > variable number

- Performance of all kernels used in support vector machines are close to each other,
- The effect of hyperparameters in polynomial and RBF kernels are more important,
- Support vector machines are more successful than artificial neural networks.

Observation number >> variable number

- RBF and polynomial kernels are more successful than the linear kernel,
- The effect of hyperparameters in all kernels is much more important,
- Support vector machines are more successful than artificial neural networks.

Observation number < variable number

- All kernel performances used in support vector machines are close to each other,

- The effect of hyperparameters in polynomial and RBF kernels are much more important,
- Artificial neural networks are more successful than support vector machines.

When researchers apply machine learning methods, they need to focus on the most important hyperparameters for the dataset in different situations, determine situations where hyperparameters are more important and significant, and decide on methods accordingly.

ACKNOWLEDGEMENT

This study was presented as a presentation at the Karadeniz 3rd International Applied Sciences Congress (25-26 April 2020, Ordu, Turkey) and printed as an abstract in the congress booklet.

REFERENCES

- [1] A. Gandomi, M. Haider, "Beyond the hype: Big data concepts, methods, and analytics", *International Journal of Information Management*, vol. 35, pp. 137-144, 2015.
- [2] A. Oussous, F.Z. Benjelloun, A.A. Lahcen, and S. Belfkih, "Big data technologies: A survey", *Journal of King Saud University Computer and Information Sciences*, vol. 30, pp. 431-448, 2018.
- [3] B. Wang, N.Z. Gong, "Stealing hyperparameters in machine learning", *IEEE Symposium on Security and Privacy*. DOI 10.1109/SP.2018.00038
- [4] P. Koch, B. Wujek, O. Golovidov, and S. Gardner, "Automated Hyperparameter tuning for effective machine learning", SAS514-2017. SAS Institute Inc. USA. 2017.
- [5] G. Ser, C. T. Bati, "Determining the best model with deep neural networks: Keras application on mushroom data", *YYU J. AGR. SCI.* vol. 29, no.3, pp. 406-417, 2019.
- [6] M. Claesen, B.D. Moor, "Hyperparameter search in machine learning" *The XI Metaheuristics International Conference*. arXiv:1502.02127v2 [cs.LG] 6 Apr 2015.
- [7] L.C. Padierna, M. Carpio, A. Rojas, H. Puga, R. Baltazar, and H. Fraire, "Hyper-parameter tuning for support vector machines by estimation of distribution algorithms." In *Nature-Inspired Design of Hybrid Intelligent Systems*. pp.787-800, 2017.
- [8] P. Probst, A.L. Boulesteix, and B. Bischl, "Tunability: Importance of Hyperparameters of Machine Learning Algorithms", *J. Mach. Learn. Res.*, vol. 20, no. 53, pp.1-32, 2019.
- [9] C.W. Hsu, C.C. Chang, C.J. Lin, "A practical guide to support vector classification", <http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf>. Date of access: 12.10.2020
- [10] M. Prevolnik, D. Škorjanc, M. Čandek-Potokar, and M. Novič, "Application of artificial neural networks in meat production and technology", *Computer and Information Science*, vol. 11, pp. 223-240, 2011.
- [11] S. Pour Hamidi, M.R. Mohammadabadi, M. Asadi Foozi, and H. Nezamabadi-pour, "Prediction of breeding values for the milk production trait in Iranian Holstein cows applying artificial neural networks", *Journal of Livestock Science and Technologies*, vol. 5, no.2, pp. 53-61, 2017.
- [12] K. Liakos, S.P. Moustakidis, G. Tsiotra, T. Bartzanas, D. Bochtis, and C. Parisses, "Machine Learning Based Computational Analysis Method for Cattle Lameness Prediction", *HAICTA* pp. 128-139, 2017.
- [13] H. Bisgin, T. Bera, H. Ding, H. G. Semey, L. Wu, Z. Liu, and W. Tong, "Comparing SVM and ANN based machine learning methods for species identification of food contaminating beetles", *Scientific reports*, vol. 8, no.1, pp.1-12, 2018.
- [14] UCI Database, 2020. <https://archive.ics.uci.edu/ml/datasets/Mice+Protein+Expression#>
- [15] C. Higuera, K.J. Gardiner and K.J. Cios, "Self-Organizing Feature Maps Identify Proteins Critical to Learning in a Mouse Model of Down Syndrome", *PLoS ONE* vol. 10, no:6, e0129126. [Web Link] [journal.pone.0129126](https://doi.org/10.1371/journal.pone.0129126), 2015.
- [16] E. Frank, A.H. Mark, H.W. Ian, "The WEKA Workbench. Online Appendix for Data Mining: Practical Machine Learning Tools and Techniques", Morgan Kaufmann, Fourth Edition, 2016.

- [17] C. Aydin, "Classification of Fire Station Needs Using Machine Learning Algorithms", *European Journal of Science and Technology*, no.14, pp. 169-175, 2018.
- [18] C.C. Aggarwal, "Neural networks and deep learning", Springer, 2018.
- [19] S. Haykin, "Neural networks and learning machines", Pearson Education Canada, 2008.
- [20] J. Rasmi, <https://www.hackerearth.com/blog/developers/simple-tutorial-svm-parameter-tuning-python-r/>. Date of access: 13.09.2020
- [21] T. Kavzaoglu, I. Colkesen, "Investigation of the effects of kernel functions on the classification of support vector machines and application images", *Journal of Map*, vol. 44, pp. 73-82, 2010.
- [22] C. Dawson, <https://towardsdatascience.com/a-guide-to-svm-parameter-tuning-8bfe6b8a452c>. Date of access: 04.08.2020
- [23] L. Igual, S. Seguí, "In Introduction to Data Science" pp. 1-4. Springer, Cham, 2017.
- [24] A. Moore, "Support vector machines", <https://www.autonlab.org/media/tutorials/svm15.pdf>. Date of access: 04.01.2020.
- [25] A. Guran, M. Uysal, and O. Dogrusoz, "Effects of support vector machine parameter optimization on sentiment analysis", *DEU Faculty of Engineering Journal of Engineering Sciences*, vol. 16, no. 48, pp. 86-93, 2014.
- [26] R. Pupale, <https://towardsdatascience.com/https-medium-com-pupalerushikesh-svm-f4b42800e989>. Date of access: 11.11.2020
- [27] J. N. Van Rijn, F. Hutter, "Hyperparameter importance across datasets", In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 2367-2376, 2018.
- [28] T. Shaikhina, D. Lowe, S. Daga, D. Briggs, R. Higgins, and N. Khovanova, "Machine learning for predictive modelling based on small data in biomedical engineering", *IFAC-PapersOnLine*, vol. 48, no. 20, pp. 469-474, 2015.
- [29] B. Xu, W. Wang, G. Falzon, P. Kwan, L. Guo, Z. Sun, and C. Li, "Livestock classification and counting in quadcopter aerial images using Mask R-CNN", *International Journal of Remote Sensing*, pp. 1-22, 2020.

BIOGRAPHIES

Cafer Tayyar Bati obtained his BSc degree from Van Yüzüncü Yıl University, Department of Mathematics in 2007, and his master's degree from Van Yüzüncü Yıl University, Faculty of Agriculture, Department of Animal Science/Biometry and Genetics in 2017. He started his Ph.D. studies in the same department in 2017. He is still continuing his Ph.D. education and working as a Mathematics teacher at Ipekyolu Anatolian High School. He carries out machine learning studies in the field of animal husbandry.

Gazel Ser graduated from Van Yüzüncü Yıl University, Faculty of Agriculture, Department of Food Engineering in 2001. She completed his master's degree in 2004 and his Ph.D. in 2011 at Van Yüzüncü Yıl University, Institute of Science, Department of Animal Science, Biometrics and Genetics. She was appointed as an assistant professor to Van Yüzüncü Yıl University, Faculty of Agriculture, Department of Animal Science in 2013, where she worked as an expert. She received the title of Associate Professor as of 2017 and still works as an Associate Professor in the same department.