Volume 9, Issue 10, October – 2024 International Journal of Innovative Science and Research Technology
ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24OCT497

SVOH: Rigorous Selection Approach for Optimal Hyperparameter Values

Kopoin NDiffon Charlemagne[1] ; Koffi Dagou Augustin[1] ; Zouneme Boris Stéphane[2]
[1] Ecole Supérieure Africaine des TIC, Abidjan-Treichville, Côte d'Ivoire
[2] Université Nangui Abrogoua, Abidjan, Côte d'Ivoire

Abstract:- The problem we address in this paper is a model selection problem. We consider the k-fold cross-validation (KCV) technique, applied to the Gaussian support vector machine (SVM) classification algorithm. In the cross-validation process, the value of k for the number of subsets is generally chosen and set aprioristically (without any experiment). However, the value of k affects the choice of the best compromise between the estimation error and the approximation error of the model. In this way, the k value of the number of subsets can severely influence the optimal values of the SVM classifier's hyperparameters and consequently affect the performance of the selected model and its ability to generalize.

In this work, we propose a rigorous approach for finding the values of the hyperparameters of the Gaussian SVM known as SVOH (Selection of Optimal Hyperparameter Values) in a context of protein-protein interaction (PPI) prediction, where it is necessary to classify the pairs of proteins that interact together and those that do not interact together. The proposed approach considers the *k* value of the number of subsets as an influential parameter of the model and therefore performs learning to find an optimal value of *k*.

Keywords:- *Machine Learning, Model Selection, Cross-Validation, Prediction of Protein-Protein Interactions*

I. INTRODUCTION

The support vector machine (SVM) is one of the most algorithms for classification tasks, parti-cularly in the classification of protein-protein interactions [1]–[3]. SVM belongs to the field of artificial neural networks (ANN) [4] but is characterised by the solid foundations of statistical learning theory. SVMs are learned by searching a set of para-meters obtained by solving a constrained quadratic convex programming problem (CCQP), for which a number of efficient techniques have been deve-loped. The search for optimal parameters does not, however, complete the learning process, because there is a set of additional variables, hyperparameters, which must be set to achieve optimal classification performance, e.g. for ANNs, the hyperparameter is the number of hidden nodes. In the Gaussian SVM framework, these are the regulator parameters C and γ. This setting is not trivial and is an open research problem [5]–[8]. The process of finding the best hyperparameters is generally referred to in the machine learning literature as the model selection phase [9] and is strictly linked to the evaluation of the SVM's generalisation capacity or, in other words, the error rate that the SVM can achieve on new data (unknown data). In fact, it is common practice to select the optimal SVM (i.e. the optimal hyperparameters) by choosing the one with the lowest generalisation error. The methods for carrying out the model selection phase can be divided into two categories according to [9] : theoretical methods [10] and methods based on resampling techniques [11].

Theoretical methods provide in-depth information about classification algorithms but are often inapplicable and incalculable to be of any practical use. On the other hand, as mentioned by [5], Practitioners have found procedures based on resampling techniques, which work well in practice but offer no theoretical guarantee of generalisation error. One of the most popular resampling techniques is the k-fold cross-validation (KCV) procedure [8], which is simple, effective and reliable. The KCV technique consists of dividing a data set into k independent subsets. All but one of these subsets is used to form a classifier, while the remaining subset is used to evaluate the generalisation error. After training, it is possible to calculate an upper limit on the generalisation error for each of the trained classifiers.

In the literature, the choice of the value of k is fixed at 5 or 10. Choosing a fixed value of subsets for cross-validation can produce a model with a high bias and variance [9]. Cross-validation takes the average of several estimates of the retention risk corresponding to different splits of data. In [5] we can check that the value of k influences the stability of the mean error. Still according to [9], Model selection performance with cross-validation is gene-rally optimal when the variance is as low as possible. This variance generally decreases as the number k of subsets increases, with a fixed training sample size n. When k is fixed, the variance of the cross-validation also depends on n. In fact, in [8], we can see that the value of γ depends strongly on the training set used. The choice of k therefore influences the variance of the cross-validation estimator and, according to [6], [12], can have a significant impact on the search for the optimal values of the hyperparameters.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24OCT497

In the following section, we present a new approach called SVOH for selecting the optimal va-lues of the hyperparameters of the Gaussian SVM. We first present the problem to be solved and then show how the SVOH algorithm works.

II. PROPOSED APPROACH

A. Problem to be Solved

In order not to fix the value of k when searching for the optimal values of the Gaussian SVM hyperparameters using *k*-fold cross-validation, we propose to use the procedure of considering several possible combinations of subsets into which the original training set can be divided. The aim is to choose a better cross-validation estimation procedure, one that predicts the lowest bias and variance, allowing a good combination of hyperparameters (C,γ) to be identified so that the SVM classifier can have a low generalisation error and predict unknown data with a higher accuracy rate.

For the proposed approach, we will consider the number *k* as a hyperparameter as in [6], which can take any value in the set $k \in \{i,...,10\}$, $i \geq 3$. The smallest value of *k* is set to 3 because for each subset, the training data must be greater than 60% of the training set, as shown by [13]. Here, we set the highest value of k at 10 to remain within the margin set by empirical methods. This limited choice of test values from k to 10 also means that, in cases where the training set is large, the technique is not very computationally intensive. Assuming that there are q parameters, and that each of them has m distinct values, its computational com-plexity increases exponentially at a rate of $O(m^q)$ as shown by [14], [15]. In addition, in [6], we can see that more than 10 different databases have produced an optimal value *k* less than 10. The number of parameters to optimise for our case therefore becomes the triple (C, γ ,k), given that our *f* decision function uses a Gaussian kernel which itself operates with the parameter pair (C, γ).

B. Functioning of the SVOH Algorithm

Let {C} and {γ} correspond respectively to the set of values for parameter C and the set of values for parameter γ. Let D_Z be our trai-ning set of *Z* observations and *f* our SVM model obtained with the hyperparameters (*C*, γ), D_{ZE} the $Z(k\text{-}1)/Z$ subsets of the training set after subdivision into *k*-subsets and D_{ZS} , the Z/k subsets remaining reserved for the test after subdivision. The algorithm takes as input D_Z, {*C*} and {γ}. For each *k* subdivisions, $k \in \{3,10\}$, of the training set *Dz*, the algorithm trains a *f* classifier using the values of {C} and {γ} on D_{ZE}, then evaluates on D_{ZS} the correctness rate of *f*. Finally, the algorithm selects the best triplet (C, γ, k) that gave a higher correctness rate. A pseudo-code of the SVOH algorithm is shown below.:

➢ *SVOH Algorithm*

Input : D_Z : learning set
{*C*}: set of values for *C*,
{γ}: set of values for γ

Output : {*k**, *C**, γ*}
1 : **for** $C \in \{C\}, \gamma \in \{\gamma\}, k \in \{3,10\}$ **do** :
2 : $f=\emptyset$
3 : D_{ZE}, D_{ZS} = subdivision (D_Z, *k*)
4 : f_E = SVM (D_{ZE}, *C*, γ)
5 : E_r=evaluating the accuracy rate (f_E, D_{ZS})
6 : $f = f \cup \{ E_r \}$
7 : **end for**
8 : {*k**, *C**, γ* }= the best accuracy rate for *f*
9 : **Retourn** {k*,C*,γ* }

III. MATERIALS AND METHODS

A. Learning Data

In this work, the data used for experimentation come from the work of kopoin et al. [3], [16]. Kopoin et al. used the *BP* (Bigram physicochemical) feature extraction technique to produce numerical data from three protein-protein interaction (PPI) reference datasets [17]. PPI refers to whether two proteins interact. In the case of interaction, we speak of positive PPI and in the opposite case, ne-gative PPI. Firstly, HPRD PPI data [1] consisting of a set of 10,000 samples divided into 5,000 positive PPI pairs and 5,000 negative PPI pairs as in [2], [18], [19]. S. Cerevisiae PPI datasets [20], [1] consis-ting of a set of 11188 samples (with 5,594 positive pairs and 5,594 negative pairs) and H. Pylori PPI datasets [21] consisting of 2,496 samples (1,458 po-sitive pairs and 1,458 negative pairs) were also used. Four other PPI datasets also used for interaction prediction were used to test the SVOH approach. The first is the Homo Sapiens (H. Sapiens) dataset. This dataset is collected from the HPRD database as described in [22]. It contains 8,161 protein pairs, including 3,899 positive PPI pairs and 4,262 negative PPI pairs. The second is the Escherichia coli (E. coli) dataset, consisting solely of 6,594 positive pairs [23]. The third is the data set named C. elegans [24] which contains 4,013 positive pairs. Finally, the fourth set is named M. musculus and contains 313 positive pairs [25].

B. SVM Algorithm

The PPI prediction phase, which starts with an optimal SVM, is obtained by selecting the optimal hyperparameters, i.e. those that allow the SVM to have the lowest generalisation error.

Consider a learning set $\mathcal{Z} = \{(x_i, y_i), i \in [1,n]\}$ where for each vector $x \in \mathbb{R}^p$ is assigned a value $\in \{-1,+1\}$. The relationship between x and y is encapsulated in an unknown distribution $p(x,y)$, which is the source of the data. The aim of learning is to find a function $f: \mathbb{R}^d \to y_f \subset \mathbb{R}$ which approximates this relationship. The SVM algorithm [26] can be used for this purpose, where the classifier is identified during the hyperparameter search phase by solving the following convex quadratic problem:

$$max \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i\alpha_j \cdot y_i y_j \cdot h(x_i, x_j)$$

$$subject\ to\ 0 \leq \alpha_i \leq C,\ i = 1,..n$$

$$\sum_{i=1}^{n} \alpha_i y_i = 0$$

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24OCT497

where the α_i are the Lagrange multipliers and $\mathcal{C}$, one of the hyperparameters, which controls the trade-off between margin and misclassification error, and $h(x_i, x_j)$, the kernel function. The kernel considered here is the Gaussian kernel. The Gaus-sian kernel is derived from the RBF (Radial Basic Function) and depends on the Euclidean distance between the vectors in the starting space. It is defined as follows:

$$h(x_i, x_j) = \exp\left(\frac{\| x_i - x_j \|^2}{2\gamma^2}\right)$$

with γ an additional hyperparameter which determines the extent of the influence of a single training example [18]. Solving the convex quadratic problem yields a classifier defined as follows :

$$f(x) = \sum_{i=1}^{n} \alpha_i y_i h(x_i, x_j) + b$$

where *b* represent the bias.

The two hyperparameters C and γ are therefore the influential parameters of the SVM classifier, allowing it to estimate the generalization error.

C. Bigram Physicochemical Method

The Bigram Physicochemical (BP) method is a feature extraction method based on protein sequences. The BP method calculates the bigram of two amino acids (frequency of two amino acids) u-sing the values of a distance function obtained from the values of the hydrophobic and hydrophilic pro-perties of the amino acids [27] in a matrix called the physicochemical matrix (MSP).

Consider a protein *P* composed of *L* amino acid residues :

$$R_1 R_2 R_3 \dots R_{L-1} R_L$$

The value of the bigram between amino acids *i* and *j*, represented by the frequency of occurrence of the transition from the amino acid at position *i* to the amino acid at position *j*, is calculated as follows:

$$BP_{i,j} = \sum_{k=1}^{L-1} C_{k,i} \times C_{k+1,j} \quad , \quad 1 \le i \le 20; 1 \le j \le 20$$

where , $C_{k,i}$ is the value of the MSP in row *k* and column *i* and $C_{k+1,j}$ the value of the MSP in row *k* +1 and column j, calculated as follows:

$$C_{k,i} = \frac{1}{j} f(R_i, R_j)$$

with

$$f(R_i, R_j) = H_1^*(R_i) \times H_1^*(R_j) + H_2^*(R_i) \times H_2^*(R_j) \; ; \quad 1 \le i \le L, \quad 1 \le j \le 20$$

where $H_1(R_i)$ and $H_2(R_i)$ are respectively the correlated hydrophobicity and hydrophilicity functions of amino acid *i* and are obtained as follows :

$$\begin{cases} H_1(R_i) = \dfrac{H_1^0(R_i) - \varphi_1}{\sqrt{\sum_{i=1}^{20} [H_1^0(R_i) - \varphi_1]^2 / 20}} \\ H_2(R_i) = \dfrac{H_2^0(R_i) - \varphi_2}{\sqrt{\sum_{i=1}^{20} [H_2^0(R_i) - \varphi_2]^2 / 20}} \end{cases}$$

with $H_1^0(R_i)$ the hydrohobicity value of amino acid *i*, similarly $H_2^0(R_i)$, the hydrophilic value of the amino acid *i*, φ_1 and φ_2 are respectively the average of the hydrophobicity and hydrophilicity values of the 20 amino acids.

The BP method applied to a protein sequence generates a 400-*D* vector as follows:

$$V_{BP} = [\Phi_1, \Phi_2, \Phi_3, \dots, \Phi_\mu, \dots, \Phi_\Psi]^T$$

where Ψ=r×s=400 is the dimensionality of the cha-racteristic vector V_{BP}.

To represent the pair of proteins, the vector of each protein are concatenated, resulting in a final 800-*D* vector.

IV. RESULTS

The IPP HPRD datasets were used as training data, while the other two datasets, S. Cerevisiae and H. Pylori were used as test data.

A. Evaluation Metrics Used

To evaluate the robustness of the proposed approach, we used the metrics generally used to measure the performance of a classifier. We used the following measures: Accuracy (Acc), Precision (Pre), Sensitivity (Sen) and AUC. Some of these measures are defined as follows:

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \; ,$$

$$Pre = \frac{TP}{TP + FP}$$

$$Sen = \frac{TN}{TN + FN}$$

TP (true positive) is the number of predicted positive PPIs, i.e., interact really, FP (false positive) is the number of predicted positive PPIs, but are negative really, *TN* (true negative) is the number of PPIs predicted negative, and which are negative really, and *FN* (false negative) is the number of PPIs predicted negative, but are positive really. The ROC curve and AUC value graphically illustrate the performance of a binary classification system.

B. Train Results

The training was conducted on HPRD data and consisted in searching using the SVOH algorithm for an optimal value of k, C and γ among a grid of potential values informed in Table 1. We used the accuracy rate as a performance evaluation metric to find the optimal hyperparameter values. The generalisability of the trained model is assessed on the S. Cerevisiae and H. Pylori datasets.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24OCT497

Table 1: Range of Hyperparameter Values

Hyperparameter	*Grid values*
C	{1;3;10;32;50;100}
γ	{10^{-4}; 10^{-3}; 10^{-2}; 10^{-1}; 1}
k	{3;4;5;6;7;8;9;10}

Application of the SVOH algorithm yielded the following optimal hyperparameter values : (C*, γ*, *k**) = (32; 0.01; 7). Table 2 shows the best performance values of the average correctness rate for different combinations of the (C, γ, k) triplet.

Table 2: Results of the Accuracy Rate after Application of the SVOH Approach

k	*(C, γ)*	*Acc (%)*
3	(10 ; 0.1)	91.92
4	(50 ; 0.01)	92.36
5	(100 ; 0.001)	92.70
6	(10 ; 0.001)	91.13
7	(32 ; 0.01)	93.69
8	(32 ; 0.001)	92.36
9	(100 ; 0.1)	92.21
10	(100 ; 0.01)	92.49

The results in Table 2 show that for values of $k \in \{3;4;5;6\}$, the accuracy rates are between 92% and 93%. From $k = 7$, the accuracy rates are much higher and lie between 92% and 94%. On the whole, the accuracy rates are sensibly equal, however, for a number k = {5 ; 7 ; 10} where the values 5 and 10 represent the a priori values, the model formed with a number k = 7 of subsets obtains the best accuracy score with 93.69% against 92.70% for k = 5 and 92.49% for k = 10. These first results show that the best performance of the SVM model is obtained on the triplet (k, C, γ) = (7; 32; 0.01).

In Table 3, we compare the scores obtained for the subdivision values $k = 7$ determinate by the SVOH approach with those obtained for the values $k = \{5;10\}$, which are the values generally applied, for the other metrics precision, sensitivity, and AUC.

Table 3: Results for Other Metrics

k	*Pre (%)*	*Sen (%)*	*Auc (%)*
5	92.90	92.15	96.36
7	94.09	93.16	97.88
10	92.87	92.67	95.58

The scores obtained for a priori values of the number of subsets are approximately the same in all metrics. For a subdivision k = 5 of the training set, we obtain as hyperparameter values (C, γ) = (100, 0.001). The scores obtained in the accuracy, sensitivity and AUC metrics are 92.90%; 92.15% and 96.36% respectively. For a subdivision k = 10, we obtain hyperparameter values (C,γ)=(10;0.01). The scores obtained for the various metrics in Table 3 are 92.87%, 92.67% and 95.38% respectively. However, the rates obtained for a subdivision k =7 with hyperparameter values (C,γ) = (32; 0.01) in the accuracy, sensitivity and AUC metrics are 94.09%, 93.15% and 97.88% respectively. In addition, although the difference between the different rates is not very large, we note that subdividing the training set into 7 subsets improves the accuracy rate by around 1% compared with the rates obtained with the a priori subdivision values (see Table 2). We also observe a better score in the accuracy and sensitivity metrics, with an average performance greater than 0.7%, than those obtained by the a priori values (table 3). The results show that the best rates are obtained with a k = 7 subdivision, i.e. the one determinated by the SVOH approach.

C. Other Results

➢ *Results with other PPI Datasets*

Table 4 shows the results obtained on data sets other than the training data.

Table 4 : Results on Different PPI Datasets

PPI Data	*(k^*, C^*, γ^*)*	*Acc (%)*
H. sapiens	(4;32;0.01)	90.92
E. coli	(5;10;0.001)	90.36
C. elegans	(7;50;0.001)	88.49
M. musculus	(6;10;0.1)	74.43

The results in Table 4 indicate that the hyperparameter triplet values (k*, C*,γ*) that achieve the best performance on the H. sapiens, E. coli, C. elegans and M. musculus datasets are (4; 32; 0.01), (5; 10; 0.001), (7; 50; 0.001) and (6;10; 0.1), respectively. We can see that, apart from the E. coli data where performance is obtained with an a priori value for the subdivision of the training set (k = 5), the other data sets show performance for subdivision values that differ from the usual values. These results show that the number of subdivisions of the training set is important for finding the optimal values of the SVM classifier's hyperparameters.

➢ *Results Obtained with the ANN Algorithm.*

The architecture of an artificial neural network (ANN) [28] is a multi-layer stack of simple modules. The input layer receives the data, then the in-formations of the data are transformed in a non-linear way through several hidden layers. The average gradient [29] is calculated and the weights are adjusted accordingly, before the final outputs are calculated in the output layer. For example, consider learning an artificial neural network with λ-hidden layers, where each layer computes H^{α}, $\alpha \in [1,\ldots, \lambda]$. The first layer considers the network inputs, while the last layer returns the outputs H^{λ} as an a posteriori probability. Let $\{N^1, \ldots, N^{\alpha}, \ldots, N^{\lambda}\}$, the number of neurons for each layer. The intermediate layers return $H^{\alpha} = \{h_i^{\alpha}\}$ where h_i^{α} represents the output of the i-th neuron in the H^{α}. This output is determined according to the following expression:

$$h_i^{\alpha} = f^{\lambda}\left(\sum_{i=1}^{N^{\alpha-1}} \omega_{i,j}^{\alpha} \times h_j^{\alpha-1} \times b^{\alpha-1}\right);$$
$$\forall i \in \{1, \ldots, N^{\alpha}\}, \qquad \forall j \in \{1, \ldots, N^{\alpha-1}\}$$

where $\omega_{i,j}^{\alpha}$ represent weights and $b^{\alpha-1}$ the bias (one per layer) and f^{λ}, a non-linear function applied to the sum of the weights.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24OCT497

The ANN architecture we have adopted here is a network architecture with an input layer, two hidden layers and an output layer. $N = 800$ is taken as the input to the neural network. The process of training the neural network consisted of adjusting the network parameters (the weights) according to the learning algorithm until the network error function reached a minimum. We used the sigmoid function as the network activation function, which is recommended in the case of binary classification [30]. It is defined according to the following equation:

$$h(x) = \frac{1}{1 + e^{-x}}$$

where x represents the input to the downstream layer. Such a function varies the values of the evaluations from 0 to 1 and is generally used to produce a Bernoulli distribution. For an output $h(x)<0.5$, the network will classify it as a negative interaction and for an output $h(x)>0.5$, it will be classed as a positive interaction.

Various hyperparameters of an ANN, such as the number of hidden layers, the number of hidden nodes, the transfer functions, the learning rate, the batch size, and many others, can affect the rate of convergence and therefore the quality of the solution. As the number of configurations and hyperparameters increases exponentially, it is impossible to try them all in practice. We therefore recommend optimising the most important hyperparameters, such as learning rate and batch size. This means exploring different values while keeping all the other hyperparameters constant. For our test, we have chosen only the learning rate and the batch size as parameters to tune. The grid search technique was applied over a range of values used by other authors to find the optimal values of the learning rate and batch size parameters. If we denote by τ the learning rate and by ϑ the batch size, the different values retained are those used in the literature: $\tau \in \{0.5; 0.1; 0.01; 0.001\}$ and $\vartheta \in \{128;100;64;50;32;16\}$. Table 5 shows the results obtained in the search for optimal values (k^*, τ^*, ϑ^*).

Table 5 : Performance Obtained after Application of SVOH in the Case of ANNs

k	***(τ,ϑ)***	***Acc (%)***
3	(0.5 ; 0.01)	90.28
4	(0.01 ; 32))	92.74
5	(0.01 ; 100)	91.88
6	(0.001 ; 100)	91.58
7	(0.01 ; 50)	90.49
8	(0.001 ; 128)	87.75
9	(0.01 ; 64)	90.45
10	(0.001 ; 32)	88.79

This table indicates that contrary to the classical values of subdivision of the training set, $k = 5$ or $k = 10$, the best performance in the correctness metric (92.74%) of the ANN classifier on HPRD data is obtained with the triple $(k,\tau,\vartheta)=(4; 0.01; 32)$. We can say that the ANN classifier obtains the best hyperparameter values on several subdivisions of the training set different from the usual values.

V. DISCUSSION

The main technique used in this study is SVOH for the rigorous search of optimal values of hyperparameters of the Gaussian SVM classifier. The particularity of this approach is that it considers the number k of subdivisions of the training set as a hyperparameter. Experimental results with both the SVM classifier and the ANN classifier confirmed the relevance of the choice of the value of the number k because, after tests on the HPRD, H. Pylori and S. Cerevisiae IPP datasets, the accuracy rate displayed using SVOH proved to be superior to the accuracy rates displayed using the usual values (5 or 10). Cerevisiae datasets, the accuracy rate displayed using SVOH proved to be superior to the accuracy rates displayed using the usual values (5 or 10).

VI. CONCLUSION

In this work, we proposed an approach for finding optimal values for SVM hyperparameters using the k-fold cross-validation technique. This approach makes it possible to automatically arbitrate between the percentage of data used to train a classifier and the rigour of the estimated error, by considering the number of subsets as a hyperparameter to be adjusted during the model selection phase. While the number of subsets k is generally fixed in practice, we have shown, by means of tests on well-known reference data sets, that the proposed approach enables the SVM to achieve superior performance. The results with the ANN algorithm also show that the approach can be applied to any type of machine learning algorithm.

REFERENCES

[1]. X.-Y. Pan, Y.-N. Zhang, et H.-B. Shen, « Large-Scale Prediction of Human Protein−Protein Interactions from Amino Acid Sequence Based on Latent Topic Features », *J. Proteome Res.*, vol. 9, nº 10, p. 4992-5001, oct. 2010, doi: 10.1021/pr100618t.

[2]. W. Ma, Y. Cao, W. Bao, B. Yang, et Y. Chen, « ACT-SVM: Prediction of Protein-Protein Interactions Based on Support Vector Basis Model », *Sci. Program.*, vol. 2020, p. e8866557, juill. 2020, doi: 10.1155/2020/8866557.

[3]. C. N. Kopoin, N. T. Tchimou, B. K. Saha, et M. Babri, « A Feature Extraction Method in Large Scale Prediction of Human Protein-Protein Interactions using Physico-chemical Properties into Bi-gram », in *2020 IEEE International Conf on Natural and Engineering Sciences for Sahel's Sustainable Development - Impact of Big Data Application on Society and Environment (IBASE-BF)*, Ouagadougou, Burkina Faso: IEEE, févr. 2020, p. 1-7. doi: 10.1109/IBASE-BF48578.2020.9069594.

[4]. M. W. Gardner et S. R. Dorling, « Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences », *Atmos. Environ.*, vol. 32, nº 14-15, p. 2627-2636, 1998.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24OCT497

[5]. D. Anguita, A. Ghio, S. Ridella, et D. Sterpi, « K-Fold Cross Validation for Error Rate Estimate in Support Vector Machines. », in *DMIN*, 2009, p. 291-297.
[6]. D. Anguita, L. Ghelardoni, A. Ghio, L. Oneto, et S. Ridella, « The'K'in K-fold Cross Validation. », in *ESANN*, 2012, p. 441-446.
[7]. J. D. Rodriguez, A. Perez, et J. A. Lozano, « Sensitivity Analysis of k-Fold Cross Validation in Prediction Error Estimation », *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, nº 3, p. 569-575, mars 2010, doi: 10.1109/TPAMI.2009.187.
[8]. Y. Bengio et Y. Grandvalet, « No unbiased estimator of the variance of k-fold cross-validation », *J. Mach. Learn. Res.*, vol. 5, nº Sep, p. 1089-1105, 2004.
[9]. S. Arlot et A. Celisse, « A survey of cross-validation procedures for model selection », *Stat. Surv.*, vol. 4, nº none, p. 40-79, janv. 2010, doi: 10.1214/09-SS054.
[10]. V. Vapnik, *The Nature of Statistical Learning Theory*. Springer Science & Business Media, 2013.
[11]. J. L. Rodgers, « The bootstrap, the jackknife, and the randomization test: A sampling taxonomy », *Multivar. Behav. Res.*, vol. 34, nº 4, p. 441-456, 1999.
[12]. C.-W. Hsu, C.-C. Chang, et C.-J. Lin, *A practical guide to support vector classification*. Taipei, 2003.
[13]. G. A. Y. Laura, « Algorithme de descente du gradient stochastique », 2015.
[14]. J. A. A. Brito, F. E. McNeill, C. E. Webber, et D. R. Chettle, « Grid search: an innovative method for the estimation of the rates of lead exchange between body compartments », *J. Environ. Monit.*, vol. 7, nº 3, p. 241-247, févr. 2005, doi: 10.1039/B416054A.
[15]. L. Yang et A. Shami, « On hyperparameter optimization of machine learning algorithms: Theory and practice », *Neurocomputing*, vol. 415, p. 295-316, nov. 2020, doi: 10.1016/j.neucom.2020.07.061.
[16]. C. N. Kopoin, A. K. Atiampo, B. G. N'Guessan, et M. Babri, « Prediction of Protein-Protein Interactions from Sequences using a Correlation Matrix of the Physico-chemical Properties of Amino Acids », *Int. J. Comput. Sci. Netw. Secur.*, vol. 21, nº 3, p. 41-47, mars 2021, doi: 10.22937/IJCSNS.2021.21.3.6.
[17]. J. R. Bock et D. A. Gough, « Predicting protein–protein interactions from primary structure », *Bioinformatics*, vol. 17, nº 5, p. 455-460, mai 2001, doi: 10.1093/bioinformatics/17.5.455.
[18]. Y. E. Göktepe et H. Kodaz, « Prediction of Protein-Protein Interactions Using An Effective Sequence Based Combined Method », *Neurocomputing*, vol. 303, p. 68-74, août 2018, doi: 10.1016/j.neucom.2018.03.062.
[19]. Z.-H. You, Y.-K. Lei, L. Zhu, J. Xia, et B. Wang, « Prediction of protein-protein interactions from amino acid sequences with ensemble extreme learning machines and principal component analysis », *BMC Bioinformatics*, vol. 14, nº S8, p. S10, mai 2013, doi: 10.1186/1471-2105-14-S8-S10.
[20]. Z.-H. You, S. Li, X. Gao, X. Luo, et Z. Ji, « Large-Scale Protein-Protein Interactions Detection by Integrating Big Biosensing Data with Computational Model », BioMed Research International. Consulté le: 5 janvier 2019. [En ligne]. Disponible sur: https://www.hindawi.com/journals/bmri/2014/598129/abs/
[21]. J. Martin, « Prédiction de la structure locale des protéines par des modeles de chaˆınes de Markov cachées », PhD Thesis, Citeseer, 2005.
[22]. C.-H. Huang, H.-S. Peng, et K.-L. Ng, « Prediction of Cancer Proteins by Integrating Protein Interaction, Domain Frequency, and Domain Interaction Data Using Machine Learning Algorithms », BioMed Research International. Consulté le: 28 mai 2018. [En ligne]. Disponible sur: https://www.hindawi.com/journals/bmri/2015/312047/abs/
[23]. M. Riley, « Functions of the Gene Products of Escherichia coli », *MICROBIOL REV*, vol. 57, p. 91, 1993.
[24]. X.-T. Huang, Y. Zhu, L. L. H. Chan, Z. Zhao, et H. Yan, « An integrative C. elegans protein–protein interaction network with reliability assessment based on a probabilistic graphical model », *Mol. Biosyst.*, vol. 12, nº 1, p. 85-92, 2016.
[25]. Y. Z. Zhou, Y. Gao, et Y. Y. Zheng, « Prediction of Protein-Protein Interactions Using Local Description of Amino Acid Sequence », in *Advances in Computer Science and Education Applications*, vol. 202, M. Zhou et H. Tan, Éd., Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, p. 254-262. doi: 10.1007/978-3-642-22456-0_37.
[26]. V. N. Vapnik, « An overview of statistical learning theory », *IEEE Trans. Neural Netw.*, vol. 10, nº 5, p. 988-999, 1999.
[27]. C. J. van Oss, « Hydrophobicity and hydrophilicity of biosurfaces », *Curr. Opin. Colloid Interface Sci.*, vol. 2, nº 5, p. 503-512, 1997.
[28]. P. Wira, « Réseaux de neurones artificiels: architectures et applications », *Cours En Ligne Univ. Haute-Alsace*, 2009.
[29]. L. Bottou, « Stochastic gradient descent tricks », in *Neural networks: Tricks of the trade*, Springer, 2012, p. 421-436.
[30]. X. Cao, W. Zhang, et Y. Yu, « A Bootstrapping Framework With Interactive Information Modeling for Network Alignment », *IEEE Access*, vol. 6, p. 13685-13696, 2018, doi: 10.1109/ACCESS.2018.2811721.