A Versatile Combination of Classifiers for Protein Function Prediction

Haneen Altartouri

Institute for Neural Computation
Ruhr-University Bochum, Germany
Email: haneen.altartouri@ini.rub.de

Tobias Glasmachers

Institute for Neural Computation
Ruhr-University Bochum, Germany
Email: tobias.glasmachers@ini.rub.de

***Abstract*—Protein classification problems can be addressed with a wide range of machine learning methods. Top performance is achieved with a variety of methods, and the best method depends on the data set under study. Therefore, a minimal requirement for a general proceeding is to consider multiple classifiers and to tune their hyperparameters. Further highly task-specific performance gains can be achieved through additional measures like feature selection, which is particularly important for high-dimensional descriptors, or with separate classifiers for different clusters. In this paper, we design a versatile classifier with the aim to combine all of the above options, but with robust defaults and fallback options. We demonstrate systematic performance improvements across a wide range of protein prediction problems.**

Keywords–protein classification; feature selection; clustering.

I. INTRODUCTION

In recent years, an increasing number of protein sequences has been extracted through high-throughput sequencing technologies. As a consequence, identifying functions of these sequences became one of the most interesting and challenging topics in bioinformatics [1]. Different computational approaches to predict the functions of protein sequences in an efficient way have been explored.

In most approaches, prediction of protein functions is based on supervised classification algorithms, which construct a learning model determining the relation between the protein sequences and their functions. The trained model can help in predicting the function of the new sequences. For many protein sequence datasets, the predictive accuracy achievable this way is not fully satisfactory. Classification is often easy if the discriminative features are homogeneous for the whole data set. For heterogeneous datasets, we should, therefore, find homogeneous regions and address them with separate classifiers. In particular, tuning feature vectors and hyperparameters specifically for each region can improve the overall performance of the prediction.

Clustering can be used for obtaining homogeneous regions inside a dataset. Clustering is a class of unsupervised learning methods, which group similar data based on their properties without depending on labels. Sequences within a cluster are more similar to each other than sequences in other clusters. In this research, clustering is applied prior to classification to construct meaningful homogeneous sub-datasets in order to improve the performance of the classification.

Several researchers explored combinations of classification and clustering in different applications, such as disease diagnosis [2], text classification [3][4], and network traffic classification [5]. Their results show that this combination can improve predictive performance in many cases. In the field of protein problems, clustering has been used for many years to group proteins into families [6]–[8]. However, the effect of using clustering algorithms to reduce the heterogeneity of the protein datasets in order to improve the performance of supervised prediction of the function of proteins has not yet been studied. In this study, we close this research gap.

Clustering can be used to improve the performance of classification either by reducing the feature representation [9][10], or by extracting structural information form the data. In the second case, clustering is used to discover a structure in the training examples. Some approaches use the clustering information by expanding the feature vectors with new attributes extracted from clusters. For example, Kyriakopoulou et al. [3] have enhanced the text classification performance for the spam detection problem by grouping the training data into clusters and then each cluster contributes one meta-feature to the feature space of the training and testing data. Finally, they used a Support Vector Machine (SVM) classifier [11] to classify the expanded data containing the original features and meta features. Their experimental results demonstrate that the inclusion of meta features improves the classification accuracy. Xiao et al. [2] constructed a clustering-based attribute selection measure from the clustering step. This attribute called hybrid information gain ratio takes, into consideration the class label and the cluster of the sample. They trained a C4.5 decision tree based on this ratio. Their results show that using the new attribute improved the performance of classification for healthcare and disease diagnoses problems.

The most commonly used approach to combine clustering with classification depends on breaking down a complex classification problem into simpler problems using clustering, then training a single classifier on each cluster. Rajamohamed et al. [12] applied k-means and rough k-means to group credit card churn samples into clusters. Then, they divided each cluster into testing and training data to apply a classifier within each cluster. Different classifiers were tested and the results showed that combining the rough k-means with SVMs improved the classification performance compared to using a single classifier. Gaddam et al. [13] combined k-means clustering and the ID3 decision tree learning methods for classifying anomalous and normal activities in a network, an active electronic circuit, and a mechanical mass beam system. In their work, the dataset is divided into k subsets based on the

Copyright (c) IARIA, 2020. ISBN: 978-1-61208-792-4

similarity, then the ID3 classifier is trained on each cluster. The results showed that this hybrid achieved better performance than a single global classifier. Fradkin [14] applied clustering within classes to artificially increase the number of classes, then a multi-class classifier was trained to distinguish between the clusters. The results showed that clustering within classes can improve the classification in many cases.

In this work, we aim at designing a general approach for improving the prediction of protein sequences. To this end, we reduce the heterogeneity of the dataset (if it exists) by constructing meaningful homogeneous regions (sub-datasets), and then handling each sub-dataset separately as a small problem inside a large complex dataset. This allows us to train different classifiers in each sub-dataset, and importantly, to select features and tune the hyper-parameters of these classifiers separately. We also introduce an option to return back to the classifier trained on the whole complex dataset in case of weakness, as proposed in [14]. This is an important mechanism that greatly stabilizes the results and that avoids over-fitting to small clusters. We analyse the features inside each sub-datasets and apply feature reduction to select locally significant features to help in distinguishing sequences that belong to different classes and improve the sub-dataset classifier, which implies improving the overall performance. In contrast, existing hybrid models apply reduction only *before* the clustering to select globally significant features [15][16].

We tested two methods for features reduction, and different classifiers were trained and their hyperparameters tuned. We evaluated the effect of the proposed approach on six protein function prediction problems. Our results show that the proposed approach improves the performance of the prediction in most cases, without degrading performance in other cases.

The remainder of this paper is organized as follows: the next section describes the proposed approach in detail. Section III briefly introduces the benchmarks of this study. In section IV, we present the experimental results and discuss our findings. In section V, we close with conclusions from our work.

II. The proposed Approach

We propose a versatile approach for the classification of protein functions consisting of the following steps:

1) encode variable-length protein sequences with a fixed-length descriptor or feature vector,
2) cluster the dataset into sub-datasets,
3) apply feature reduction inside each sub-dataset
4) train multiple classifiers for each sub-dataset, and
5) decide for each subset which classifier to use.

The proposed approach is summarised in Figure 1, where the number of sub-datasets is 2.

We go through these steps one by one.

A. Representing Protein Sequences

We aim to represent protein sequences in a form that can be easily handled by machine learning algorithms. The main challenge is that sequences can have different lengths. For many learning machines, we need to encode these sequences into a fixed-length descriptor that extracts the relevant features. In this study, we rely on Chou's Pseudo Amino Acid Composition (PseAAC) descriptors for protein sequence encoding [17]. It has been demonstrated that PseAAC descriptors are extremely effective features for protein problems [18]–[22]. PseAAC represents amino acid frequencies and, in addition, it preserves most of the sequence-order information [23][24]. A protein sequence is represented by 20 + λ numerical features. The first 20 features are the occurrence frequencies of the 20 amino acids. The remaining λ descriptors encode the sequence order. For a detailed description of PseAAC, we refer to [17][25].

PseAAC depends on Physico-Chemical Properties (PCPs) of the amino acids to represent the sequence. A PCP is a (scalar) physical or chemical feature of the amino acid. In this work, we used two sets of PCPs. The first set is rather small and it consists of three PCPs used in Chou's work [17]: hydrophobicity, hydrophilicity, and side chain mass. The other set is more rich: it contains fifty non-redundant PCPs of amino acids proposed by Georgiev [26], such as: normalized relative frequency of double bend , pK (-COOH) , relative mutability and flexibility parameter for two rigid neighbors .

B. Clustering the Dataset into Sub-datasets

The second step in this approach is clustering the dataset (D) into sub-datasets (SDs). Clustering is a process of grouping the samples into meaningful clusters, with the aim to identify groups of homologous protein sequences. This way, we break down a complex protein prediction problem into a set of simpler problems. To keep the approach manageable, we applied only one clustering algorithm, which is k-means [27].

K-means is a partition clustering algorithm, where each sample belongs to a unique cluster. It is widely used in bioinformatics [28][29] because it is simple, easy to implement, and reasonably fast. For details on k-means, we refer to [27]. To apply k-means, we need to select the number of clusters (k), which is a hyperparameter of the method. Optimal clustering requires dataset-specific tuning of k [30] and, since there is no method that is guaranteed to find the optimal value for k (determining the right number of clusters is still an open problem in clustering research), we apply an array of pre-defined values for k and study its effect on the overall performance.

C. Reducing Feature Vector Dimensionality

After clustering, we apply feature reduction inside each sub-dataset in order to optimally separate sequences that belong to different classes. Yang et al. [22] showed that applying feature reduction on PseAAC features can improve the performance of protein classification.

Feature reduction is an important step before applying machine learning algorithms if some of these features are irrelevant or redundant [31], and possibly add noise. These redundant and irreverent features do not contribute to the accuracy of a predictive model and sometimes even reduce its performance. Then, removing these features can improve the accuracy of the model, or decrease the size of the feature space without affecting the prediction accuracy [31]. To study the effect of reducing the feature vector, we have tested two reduction techniques: the Recursive Feature Elimination (RFE) algorithm as a feature selection technique, and Principal Component Analysis (PCA) as a feature extraction technique. For more details about these algorithms, please see [32] and [33], respectively.

Copyright (c) IARIA, 2020. ISBN: 978-1-61208-792-4

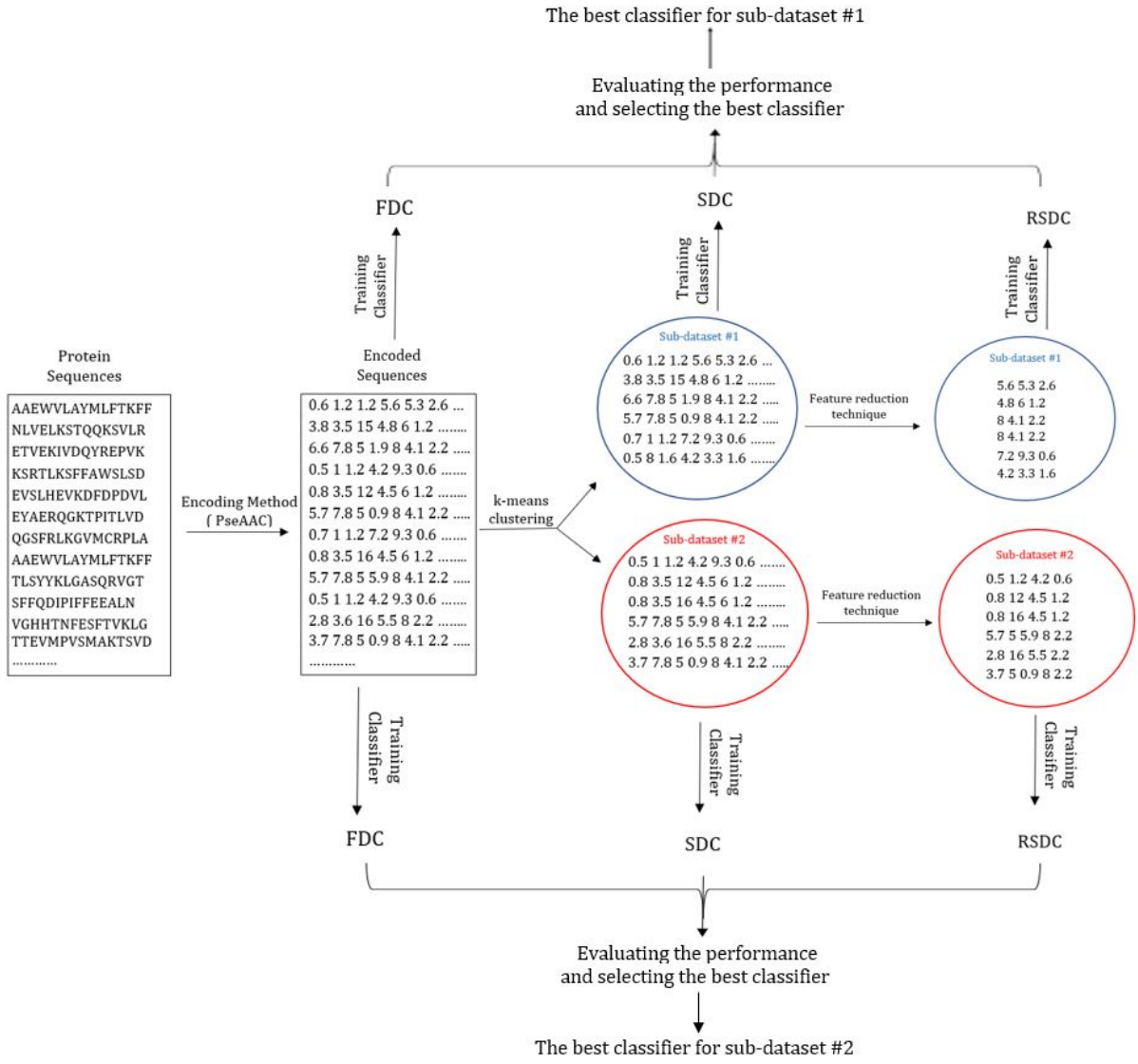


Figure 1. The proposed Approach.

D. Training Classifiers on the Sub-datasets

The central step of this approach is to train a classifier. We apply a set of standard supervised learning algorithms that build a predictive model based on the training data. The model is used to classify new samples (testing data).

After dividing the protein dataset into k simpler problems (sub-datasets) in the clustering step, we train one classifier per sub-dataset, so each classifier focuses on classifying the proteins in a specific region. This step includes hyperparameter tuning, where the tuning procedure employed depends on the classifier at hand.

For some sub-datasets, we cannot train a reliable classifier because there is not enough data. In such a case, we need to use another classifier to process data within this region. Therefore, we resort to the classifier trained on the whole dataset as proposed by [14], which we refer to as the "Full Dataset Classifier" (FDC).

From the previous steps, we have sub-datasets with the full features, and a view on these sub-datsets with reduced features. Therefore, for each sub-dataset, we train two classifers: one using the full feature set, called "Sub-dataset Classifier" (SDC), and the other using the reduced feature set, called "Reduced Sub-dataset Classifier" (RSDC).

E. Classifier Selection

For each sub-dataset, we have up to three classifiers available: FDC, SDC, and RSDC. We select one of them into our predictive model. To this end, we estimate the performance of all three classifiers by means of cross-validation, restricted to the sub-dataset. We select the classifier with the highest AUC (area under the receiver operator characteristic curve). The classifier with the highest AUC is then responsible for classifying all data in the corresponding cluster.

This last step proves to be crucial for achieving high predictive accuracy. Our intuition on the classifier selection step is as follows. Our basic hypothesis is that protein datasets can consist of meaningful sub-datasets, where different feature sets are discriminative in each subset. That logic leads straight

Copyright (c) IARIA, 2020. ISBN: 978-1-61208-792-4

to the construction of the RSDC classifiers. However, using RSDC in all cases does not work well. On the one hand, this is the case if there is no pronounced subset structure in the dataset under study. On the other hand, some clusters represent harder classification problems than others. In the hard clusters, RSDCs suffer from the reduced amount of data. Then, feature reduction and even model training can be unreliable and subject to a high risk of over-fitting, while the global classifier makes better use of surrounding points. Therefore, it is important to provide the robust fallback options of reverting to the SDC or even to the FDC.

When a new sequence is incoming, we measure its similarity with the centroids of the sub-datasets and assign it to the nearest centroid. Then, we apply the chosen classifier based on the previous cases. We want to stress that the decisions in all five design stages, namely how to represent sequences, how many clusters to use, which learning machine to apply and how to set its hyperparameters, which features to reduce, and which classifier to pick for each cluster, are all made based on cross-validation, so that we end up with a single hybrid classifier.

III. Benchmark Datasets

To evaluate the performance of the proposed approach, we have used six protein datasets. Table 1 summarize all these datasets. The first three datasets contain long sequences and the last three contain peptide sequences. All datasets are binary classification problems. We have split each dataset into a training and a testing set.

IV. Experiments and Results

The main aim of our experiments is to demonstrate the benefit of our approach as compared to the FDC, which is a natural baseline. We used the following parameter settings in our experiments. The PseAAC encoding, as described in Section II-A consists of two parts. The weight of the features representing the sequence order was set to $w = 1/2$. The length of the shortest sequence was set to $\lambda = 7$ for peptides and to $\lambda = 30$ for long protein sequences. These settings result in 27 and 50 PreAAC features, respectively.

Since the distribution of the sequences differs from one dataset to another, we have to tune the number of sub-datsets (k) for each dataset. For small datasets (Caspase-3, DNA-binding, and Antioxidant proteins), k is selected from a range of 2 to 7 with a step size of 1, and for the other datasets k is selected from a range of 5 to 30 with a step size of 5.

We have tuned the hyper-parameters for the FDC, SDC, and RSDC classifiers using 5-fold cross-validation repeated 3 times, and we have applied an inner cross-validation for RFE to assess generalization on an independent dataset and avoid over-fitting [40]. Cross-validation was also used for estimating the quality of the classifiers.

To evaluate the performance of the classifiers, we depend on sensitivity (SN), specificity (SP), and Matthew's Correlation Coefficient (MCC) [41]. We also use the Receiver Operating Characteristics (ROC) curve. This curve illustrates the achievable trade-offs between true positive rate and false positive rate. The quality of the ROC curve is measured by computing the area under the ROC curve (AUC) [42]. All values displayed in this research are computed on independent test sets.

A. Selecting the Best Classifier for the Proposed Approach

In our experimental study, we considered the following types of classifiers: SVM [43], Random Forest (RF) [44], Artificial Neural Network (ANN) [45], and eXtreme Gradient Boosting (xGBoost) [46]. We trained FDCs with all four learning machines with protein sequences represented by PseAAC descriptors using 50 PCPs [26] and using 3 PCPs [17]. Figure 2 shows the ROC curves and AUC values of all four classifiers on the six datasets.

For some datasets, there are significant differences in classifier performance and, in some cases, the number of PCPs makes a difference. It can be deduced from Figure 2 that different classifiers work well for different problems, a fact we account for with our approach. SVMs are a solid choice for most datasets using 50 PCPs, while RF is the best choice when using 3 PCPs. Therefore, we restrict ourselves to SVM and RF classifiers. In the following, we present two sets of experiments. First, we assess the effect of splitting the dataset into sub-datasets by comparing SDCs and FDC, then we investigate the effect of feature learning inside each cluster by comparing SDCs and RSDCs.

B. Impact of Training Multiple Classifiers

In order to study the effect of training separate classifiers in some regions within the dataset, we run several experiments varying k (number of sub-datasets). Table II shows a comparison between using FDC only (baseline), and SDCs with the option to use FDC for weak SDCs based on AUC values, as described in Section II-B. The results in the table represent the best cross-validation performance over k and the machine-specific hyperparameters. We observed that, in the most cases, applying SDCs with option to resort to the FDC in a per-cluster manner improves the performance over using FDC only, except for the AMP dataset, where the improvement is very small for both SVM and RF. For three datasets, the effect of using multiple classifiers on the overall AUC is small (1% improvement), while for RNA-binding, Antioxidant, and MHCII datasets, we have achieved a respectable improvement of 3% in the AUC values. For the other metrics, we achieved significant improvements.

The results clearly indicate that classification based on SDCs with the fallback option to the FDC consistently improves over the FDC baseline.

The best result is obtained for the RNA-binding dataset: 3% and 12% improvement for the AUC and MCC values, respectively, by grouping the dataset into 5 sub-datsets and using SVM-SDCs inside 3 groups, while the other 2 groups depend on the SVM-FDC. On the other hand, the best result obtained with an RF is 1% improvement in both AUC and MCC where $k = 5$. For MHCII peptides, we obtained the best result by grouping the dataset into 15 clusters, but using SDCs for 3 groups only. This result indicates that, in some cases, significant improvements are achievable by handling only a few sensitive regions with specific SDCs. Furthermore, it is worth noting that, for most datasets, SVM achieved better improvement than RF inside SDs, except for the Antioxidant dataset.

C. Impact of Reducing the Features Inside Sub-datasets

As detailed in Section II-C, the proposed approach allows to reduce the features separately for each sub-dataset. Figure 3

Copyright (c) IARIA, 2020. ISBN: 978-1-61208-792-4

TABLE I. DATASETS USED FOR THE APPROACH EVALUATION

Dataset	# of Positives	# of Negatives	sequences length
DNA-binding proteins [34]	523 binding proteins	543 non binding proteins	50 - 1323 amino acids
Antioxidant proteins [35]	250 antioxidant	1547 non-antioxidant	31 - 1463 amino acids
RNA-binding proteins [36]	2780 binding proteins	7077 non binding proteins	50 - 8799 amino acids
Antimicrobial peptides (AMP) [37]	869 AMPs	2405 non-AMPs	8 - 103 amino acids
Caspase 3 human substrates [38]	247 cleaved peptides	247 non-cleaved peptides	14 amino acids
Major Histocompa. Complex II (MHCII) [39]	3510 binding peptides	1656 non-binding peptides	9 - 37 amino acids

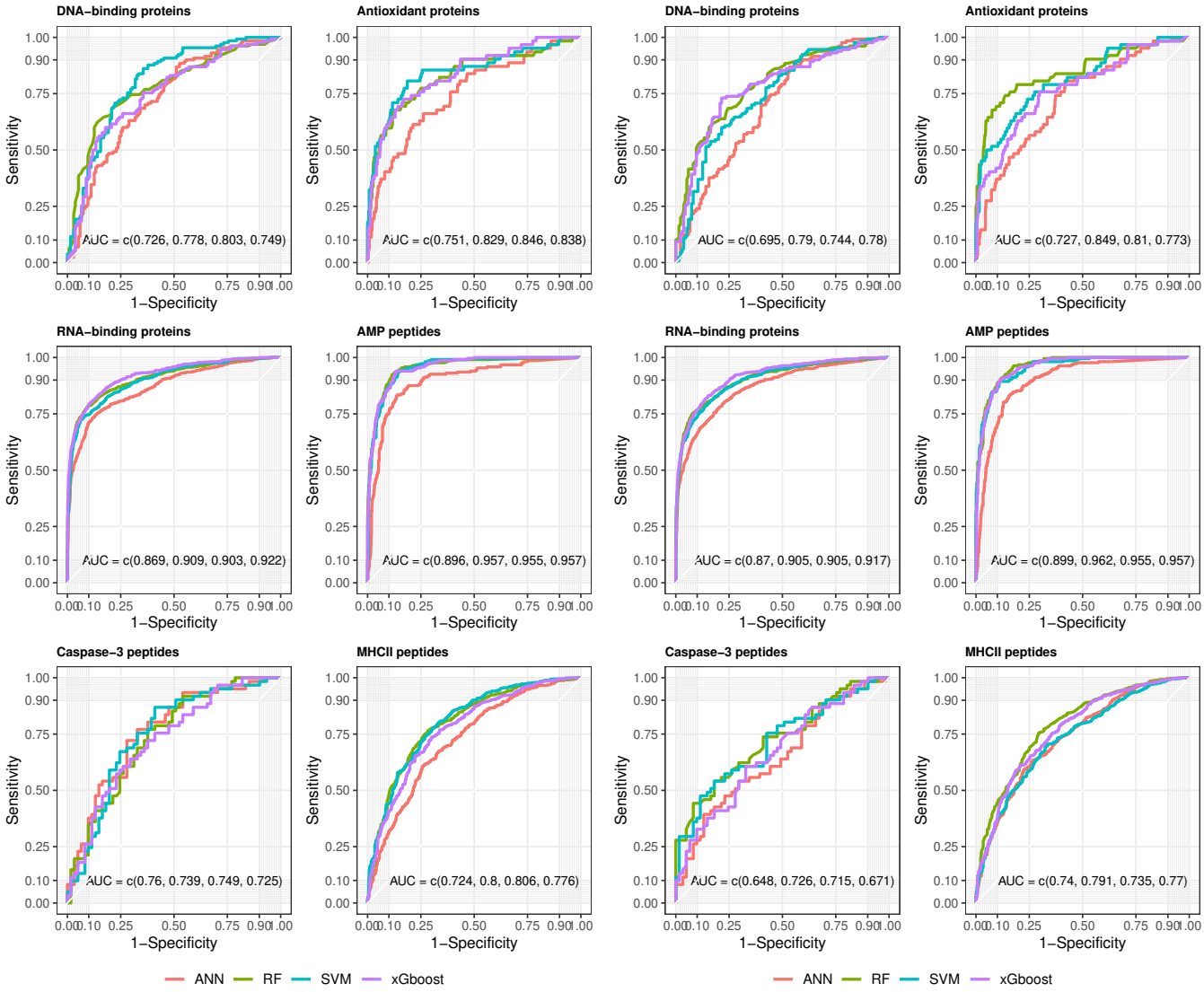


Figure 2. ROC curves for 4 different classifiers: ANN, RF, SVM, and xGBoost (with AUC values in parentheses), using 50 PCPs (columns 1 and 2) and 3 PCPs (columns 3 and 4).

illustrates that different feature sets can be discriminative in different sub-datasets. It shows relative feature importance for two sub-datasets of MHCII at $k = 20$. The importance of the features differs not only between the two sub-datasets, but also from the full dataset. Therefore, applying feature reduction on a per-cluster basis has the potential to improve overall performance.

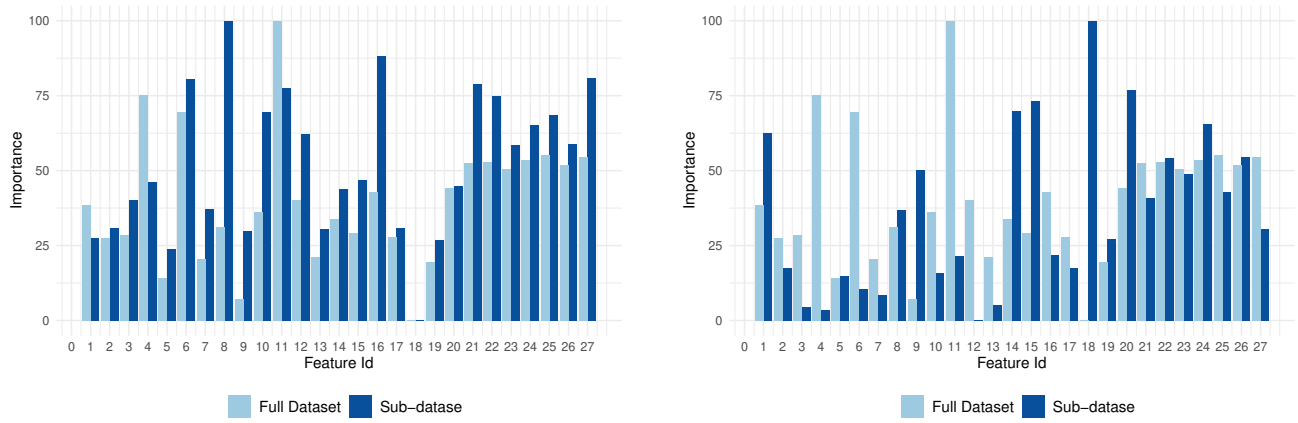
We ran two sets of experiments to study the effect of reducing the features inside the sub-datasets. In the first case, we select the best set of features that maximize the performance of prediction using RFE and, in the second case, we apply PCA to extract a new descriptor to represent the data by selecting principal components that cover at least 95% of the total variance. Like for SDCs, if the resulting RSDC turned out to be unreliable, then we returned back to FDC or SDC based on the cross-validated AUC scores.

Table II shows the best improvement of RSDCs with options of reverting to SDCs or FDC, compared to SDCs with option of reverting to FDC. The results show that RFE outperforms PCA in reducing the features inside the sub-

Copyright (c) IARIA, 2020. ISBN: 978-1-61208-792-4

TABLE II. COMPARISON BETWEEN USING ONLY FDC, SDCs WITH OPTION TO USE FDC IN THE WEAK CLUSTERS, AND RSDCs WITH OPTIONS TO USE FDC OR SDCs IN THE WEAK CLUSTERS FOR 6 BENCHMARKS.

Method	SVM						RF					
	k	AUC	SEN	SPE	MCC	(FDC,SDC, RSDC(algo.))	k	AUC	SEN	SPE	MCC	(FDC,SDC, RSDC(algo.))
DNA-binding proteins												
- FDC only (the baseline)	-	0.8033	0.7769	0.6963	0.4744	-	-	0.7899	0.6692	0.7556	0.4266	-
- SDCs with reverting option to FDC	4	0.8197	0.7769	0.763	0.5398	(2,2,-)	3	0.7859	0.6923	0.7778	0.4721	(2,1,-)
- RSDC with reverting option to FDC or SDCs (RFE, PCA)	2	0.8433	0.8154	0.7481	0.5643	(1,0,1(RFE))	2	0.8348	0.7308	0.8074	0.5401	(0,0,2(PCA))
Antioxidiant proteins												
- FDC only (the baseline)	-	0.8405	0.68	0.8987	0.5193	-	-	0.8493	0.7419	0.8627	0.5032	-
- SDCs with reverting option to FDC	4	0.8591	0.7333	0.8966	0.5545	(2,2,-)	4	0.8706	0.7097	0.9275	0.5991	(3,1,-)
- RSDC with reverting option to FDC or SDCs (RFE, PCA)	4	0.8681	0.7333	0.9009	0.5625	(2,1,1(RFE))	-	no improvement achieved				-
RNA-binding proteins												
- FDC only (the baseline)	-	0.903	0.6331	0.9582	0.6548	-	-	0.9053	0.636	0.9661	0.6727	-
- SDCs with reverting option to FDC	5	0.9301	0.7942	0.9588	0.7788	(2,3,-)	5	0.9136	0.659	0.9644	0.687	(4,1,-)
- RSDC with reverting option to FDC or SDCs (RFE, PCA)	5	0.9412	0.8187	0.961	0.801	(2,0,3(RFE))	10	0.9344	0.705	0.9638	0.721	(5,0,5(RFE))
AMP peptides												
- FDC only (the baseline)	-	0.9552	0.765	0.9418	0.7247	-	-	0.9624	0.7926	0.9484	0.7574	-
- SDCs with reverting option to FDC	5	0.9634	0.788	0.9434	0.7451	(3,2,-)	25	0.9619	0.7926	0.9567	0.7724	(21,4,-)
- RSDC with reverting option to FDC or SDCs (RFE, PCA)	30	0.956	0.8203	0.9401	0.7638	(23,3,4(RFE))	5	0.9741	0.8295	0.9551	0.7967	(2,0,3(RFE))
Caspase 3 peptides												
- FDC only (the baseline)	-	0.7487	0.623	0.7541	0.3803	-	-	0.7263	0.7377	0.5246	0.2685	-
- SDCs with reverting option to FDC	2	0.7474	0.6393	0.7705	0.4134	(1,1,-)	6	0.7417	0.7377	0.5574	0.3	(5,1,-)
- RSDC with reverting option to FDC or SDCs (RFE, PCA)	2	0.7565	0.7541	0.7869	0.5413	(0,0,2(RFE))	-	no improvement achieved				-
MHCII peptides												
- FDC only (the baseline)	-	0.8034	0.7605	0.6981	0.4396	-	-	0.7909	0.7571	0.7029	0.4401	-
- SDCs with reverting option to FDC	15	0.8371	0.7765	0.7488	0.5022	(12,3,-)	30	0.8042	0.7537	0.715	0.4472	(19,11,-)
- RSDC with reverting option to FDC or SDCs (RFE, PCA)	15	0.843	0.7879	0.7536	0.5192	(10,2,3(RFE))	20	0.8475	0.7697	0.7754	0.5182	(8,2,10(RFE))



(a) Sub-dataset #1, where $k = 20$

(b) Sub-dataset #2, where $k = 20$

Figure 3. Comparison between the importance of features inside the full dataset and inside sub-datsets using Gini importance [44]

datasets for most of the cases except for the DNA-binding dataset, where training an RF on PCs improves the overall performance.

Although using the proposed approach can improve the overall performance either using SVM or RF as classifiers inside the sub-datasets, we have achieved the highest performance using the SVM classifier in most cases, except for AMP peptides.

For the DNA-binding dataset, we achieved an improvement of 3% for both AUC and MCC by grouping the dataset into 2 sub-datasets with SVM-RFE for one subset, while the other set uses the FDC. In effect, we enhance the performance by 4% for AUC and 9% for MCC compared to the FDC baseline. On the other hand, for $k = 2$ and using PCA and RF on these two sub-datasets, we achieved 5% and 4% improvement for AUC and MCC compared to the SDC without feature reduction, which corresponds to improvements of 5% and 12% for AUC and MCC compared to the FDC baseline.

For the RNA-binding dataset, we have improved the MCC by 3% using SVM-RFE with a very small improvement in AUC, while we achieved 2% and 4% improvement for AUC and MCC using RF-RFE.

For Antioxidant, we have achieved only 1% improvement for both AUC and MCC using SVM-RFE compared to using

SDCs without features selection, and no improvement using RF-RFE and RF-PCA. This indicates that, unsurprisingly, feature reduction does not help for all problems, and we can just depend on using RF with SDCs.

As mentioned in the previous section, SDCs barely improve over the FDC on the AMP dataset. In contrast, splitting the dataset into 5 clusters and applying RF-RFE inside of three of them improved the baseline by about 4% for MCC, with a very small improvement in AUC.

For Caspase 3, we did not improve the overall AUC of the classifier, but we achieved a significant improvement in the MCC value (about 13% improvement) using SVM-RFE compared to use the SDCs with FDC option. On the other hand, for the MHCII dataset, if we depend on the SVM as a classifier algorithm inside the sub-datasets, feature reduction did not pay off, since the improvement was very small (about 1%). However, in order to achieve similar results with an RF, we need to group the dataset into 20 sub-datasets and apply RF-RFE inside 10 of them.

Going beyond predictive performance, we also analyzed the role of the features selected within the clusters. In most cases, RFE shows that the frequencies of amino acids play an important role in classifying the sequences inside the clusters, while the sequence order has a higher impact on classifying the full dataset. Figure 4 illustrates the rank of the optimal set of features for 9 sub-sets of MHCII out of $k = 20$, compared to the full dataset using RF-RFE. For datasets containing long protein sequences, RFE shows that the optimal sets of features for clusters contain only a bit more than 50% of all available descriptors, and most of theses descriptors represent amino acid frequencies.

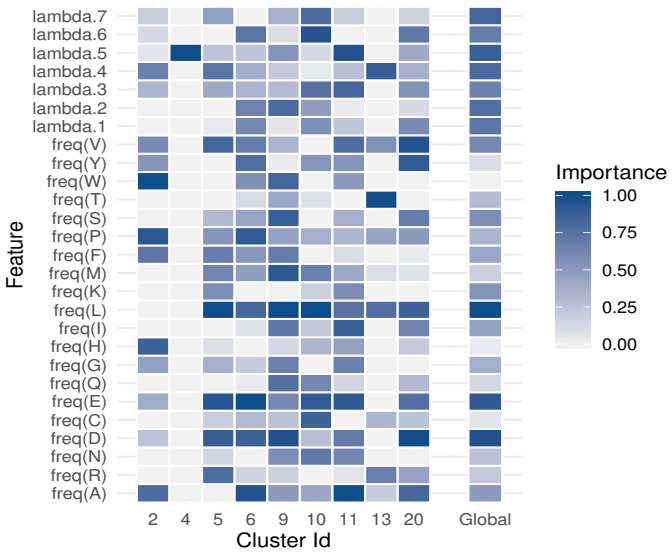


Figure 4. The importance of features based on RF-RFE

V. CONCLUSION

We have studied the effect of exploiting homogeneous sub-datasets inside protein sequence data by training multiple classifiers on sub-datsets. The proposed approach handles each sub-dataset as a separate classification problem that requires tuning the hyper-parameters and finding the best features separately. More hyperparameter choices on smaller datasets can potentially give rise to over-fitting. Therefore, it is imperative for robust performance to allow the classifiers to revert to classifiers trained on all features, and even on the full dataset, as fallback options.

In this study, we have evaluated the performance of SVM and RF classifiers inside the sub-datsets, and RFE and PCA are tested as a reduction feature algorithms. SVM and SVM-RFE achieved good performance for most datasets. The performance of the proposed approach depends on the number of sub-datasets, the encoding method, and for each cluster the classifier with its hyperparameters and the feature reduction method applied. We find that, for different datasets, the best performance is achieved with different approaches. Our approach is sufficiently versatile to account for this finding.

The results indicate that the proposed approach improved the overall performance of function prediction of protein sequences in most cases. Hence, they indicate that many protein sequence datasets suffer from heterogeneity.

REFERENCES

[1] R. Saidi, M. Maddouri, and E. Mephu Nguifo, “Protein sequences classification by means of feature extraction with substitution matrices,” BMC Bioinformatics, vol. 11, 2010, p. 175.

[2] J. Xiao, Y. Tian, L. Xie, and J. Huang, “A hybrid classification framework based on clustering,” IEEE Transactions on Industrial Informatics, 2019, pp. 1–1.

[3] A. Kyriakopoulou and T. Kalamboukis, “Combining clustering with classification for spam detection in social bookmarking systems,” RSDC, 2008.

[4] A. Thomas and M. Resmipriya, “An efficient text classification scheme using clustering,” Procedia Technology, vol. 24, 2016, pp. 1220–1225.

[5] J. Erman, M. Arlitt, and A. Mahanti, “Traffic classification using clustering algorithms,” Proceedings of the 2006 SIGCOMM Workshop on Mining Network Data, MineNet’06, vol. 2006, 2006, pp. 281–286.

[6] A. Krause, J. Stoye, and M. Vingron, “Large scale hierarchical clustering of protein sequences,” BMC Bioinformatics, vol. 6, 2005, p. 15.

[7] W. bang Chen and C. Zhang, “A hybrid framework for protein sequence clustering and classification using signature motif information,” Integrated Computer-Aided Engineering, vol. 16, 2009, pp. 353–365.

[8] L. Szilágyi, L. Medvés, and S. M. Szilágyi, “A modified Markov clustering approach to unsupervised classification of protein sequences,” Neurocomputing, vol. 73, 2010, pp. 2332–2345.

[9] S. Chormunge and S. Jena, “Correlation based feature selection with clustering for high dimensional data,” Journal of Electrical Systems and Information Technology, vol. 5, 2018, pp. 542–549.

[10] X. Zhu et al., “A new unsupervised feature selection algorithm using similarity-based feature clustering,” Computational Intelligence, vol. 35, 2018.

[11] C. Cortes and V. Vapnik, “Support-vector networks,” Machine Learning, vol. 20, no. 3, 1995, pp. 273–297.

[12] R. Rajamohamed and J. Manokaran, “Improved credit card churn prediction based on rough clustering and supervised learning techniques,” Cluster Computing, vol. 21, 2018, pp. 1–13.

[13] S. Gaddam, V. Phoha, and K. Balagani, “K-means+id3: A novel method for supervised anomaly detection by cascading k-means clustering and id3 decision tree learning methods,” Knowledge and Data Engineering, IEEE Transactions on, vol. 19, 2007, pp. 345–354.

[14] D. Fradkin, “Within-class and unsupervised clustering improve accuracy and extract local structure for supervised classification,” PhD thesis, Rutgers, The State University of New Jersey, 2006.

[15] Y. K. Alapati and K. Sindhu, “Combining clustering with classification: A technique to improve classification accuracy,” International Journal of Computer Science Engineering (IJCSE), vol. 5, 2016, p. 3.

Copyright (c) IARIA, 2020. ISBN: 978-1-61208-792-4

[16] H. Malik and J. Kender, "Classification by pattern-based hierarchical clustering." From Local Patterns to Global Models Workshop (ECML/PKDD 2008), Antwerp, Belgium, 2008.

[17] K.-C. Chou, "Prediction of protein cellular attributes using pseudo-amino acid composition," Proteins, vol. 43, 2001, pp. 246–55.

[18] Y. Fang, Y. Guo, Y. Feng, and M. Li, "Predicting dna-binding proteins: Approached from chou's pseudo amino acid composition and other specific sequence features," Amino Acids, vol. 34, 2008, pp. 103–9.

[19] D. Georgiou, T. Karakasidis, J. Nieto, and A. Torres-Iglesias, "Use of fuzzy clustering technique and matrices to classify amino acids and its impact to chou's pseudo amino acid composition," Journal of Theoretical Biology, vol. 257, 2008, pp. 17–26.

[20] P. Wang et al., "Prediction of antimicrobial peptides based on sequence alignment and feature selection methods," PloS One, vol. 6, 2011, p. e18476.

[21] N. Xiaohui et al., "Using the concept of chou's pseudo amino acid composition to predict protein solubility: An approach with entropies in information theory," Journal of Theoretical Biology, vol. 332, 2013, pp. 392—396.

[22] R. Yang, C. Zhang, L. Zhang, and R. Gao, "A two-step feature selection method to predict cancerlectins by multiview features and synthetic minority oversampling technique," BioMed Research International, vol. 2018, 2018, pp. 1–10.

[23] K.-C. Chou and H.-B. Shen, "Euk-mploc: A fusion classifier for large-scale eukaryotic protein subcellular location prediction by incorporating multiple sites," Journal of Proteome Research, vol. 6, 2007, pp. 1728–1734.

[24] ——, "Chou, k.c. & shen, h.b. review: recent progresses in protein subcellular location prediction. anal. biochem. 370, 1-16," Analytical Biochemistry, vol. 370, 2007, pp. 1–16.

[25] K.-C. Chou, "Pseudo amino acid composition and its applications in bioinformatics, proteomics and system biology," Current Proteomics - CURR PROTEOMICS, vol. 6, 2009, pp. 262–274.

[26] A. Georgiev, "Interpretable numerical descriptors of amino acid space," Journal of Computational Biology, vol. 16, no. 5, 2009, pp. 703–723.

[27] J. A. Hartigan and M. A. Wong, "A k-means clustering algorithm," Applied Statistics, vol. 28, no. 1, 1979, pp. 100–108.

[28] T. Chappell, S. Geva, and J. Hogan, "K-means clustering of biological sequences," 22nd Australasian Document Computing Symposium, 2017, pp. 1–4.

[29] A. Bustamam, H. Tasman, N. Yuniarti, Frisca, and I. Mursidah, "Application of k-means clustering algorithm in grouping the dna sequences of hepatitis b virus (hbv)," AIP Conference Proceedings, vol. 1862, 2017, p. 030134.

[30] F. Shahnaz, M. Berry, V. Pauca, and R. Plemmons, "Document clustering using nonnegative matrix factorization," Information Processing & Management, vol. 42, 2006, pp. 373–386.

[31] E. Alpaydin, Introduction to Machine Learning, 2nd ed. The Adaptive Computation and Machine Learning series, Massachusetts Institute of Technology, 2010.

[32] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, "Gene selection for cancer classification using support vector machines," Machine Learning, vol. 46, 2002, pp. 389–422.

[33] I. Jolliffe, Principal Component Analysis, 2nd ed. Springer, New York, 2002.

[34] S. Chowdhury, S. Shatabda, and I. A. Dehzangi, "idnaprot-es: Identification of dna-binding proteins using evolutionary and structural features," Scientific Reports, vol. 7, 2017.

[35] P.-M. Feng, H. Lin, and W. Chen, "Identification of antioxidants from sequence information using naïve bayes," Computational and Mathematical Methods in Medicine, vol. 2013, 2013, p. 567529.

[36] X. Zhang and S. Liu, "Rbppred: predicting rna-binding proteins from sequence using svm," Bioinformatics (Oxford, England), vol. 33, 2016, p. 854–862.

[37] W.-Z. Lin and D. Xu, "Imbalanced multi-label learning for identifying antimicrobial peptides and their functional types," Bioinformatics, vol. 32, 2016, p. btw560.

[38] M. Ayyash, H. Tamimi, and Y. Ashhab, "Developing a powerful in silico tool for the discovery of novel caspase-3 substrates: a preliminary screening of the human proteome," BMC Bioinformatics, 2012.

[39] M. Nielsen and O. Lund, "Nn-align. an artificial neural network-based alignment algorithm for mhc class ii peptide binding prediction," BMC Bioinformatics, vol. 10, 2009, p. 296.

[40] K. Ron and H. J. George, "Wrappers for feature subset selection," Artificial Intelligence, vol. 97, 1997, pp. 273–324.

[41] B. W. Matthews, "Comparison of the predicted and observed secondary structure of t4 phage lysozyme," Biochimica et Biophysica Acta (BBA) - Protein Structure, vol. 405, no. 2, 1975, pp. 442–451.

[42] T. Fawcett, "An introduction to roc analysis," Pattern Recognition Letters, vol. 27, 2006, p. 861–874.

[43] C. Cortes and V. Vapnik, "Support-vector networks," Machine Learning, vol. 20, no. 3, 1995, pp. 273–297.

[44] L. Breiman, "Random forests," Machine Learning, vol. 45, no. 1, 2001, pp. 5–32.

[45] M. A. Nielsen, Neural Networks and Deep Learning. Determination Press, 2015.

[46] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, 2016, pp. 785–794.

Copyright (c) IARIA, 2020. ISBN: 978-1-61208-792-4