Discovering critical proteins in the learning process in a Down Syndrome model of mouse through machine learning

Xhoena Polisi (xpolisi@epoka.edu.al)
Epoka University
Ali Osman Topal
University of Luxembourg
Arban Uka
Epoka University

Research Article

Keywords: Down syndrome, Ts65Dn mouse model, machine learning

Posted Date: April 16th, 2021

DOI: https://doi.org/10.21203/rs.3.rs-418223/v1

Discovering critical proteins in the learning process in a Down Syndrome model of mouse through machine learning

Xhoena Polisi[1,*], Ali Osman Topal[2,*], and Arban Uka[1,*]

[1]EPOKA University,Computer Engineering Department,Tirana, 1032 Albania
[2]University of Luxembourg, Computer Science, Esch-sur-Alzette, 4365, Luxembourg
[*]xpolisi@epoka.edu.al, aliosman.topal@uni.lu , auka@epoka.edu.al

ABSTRACT

Caused by an extra copy of the human chromosome21 (Hsa21), Down syndrome produces an intellectual disability that is still unknown and requires further research in order to have a better perception. One research conducted in this area of study has analysed different protein levels of the Ts65Dn mouse model of DS. Many researchers are trying to find the critical proteins that categorize the mice classes accurately by using machine learning. In this study, we expand the problem by trying to find the critical proteins that affect different types of learning. The protein subsets are found using forward feature selection method, ReliefF respectively and four different supervised learning algorithms are used. The experimental results are compared with previous related work, and demonstrated that the proposed method outperforms, or is comparable to, its competitors in term of accuracy. Then, a thorough analysis is done to identify the critical proteins for each learning case, by lowering the number to 9 critical proteins that can help in a better categorization of the mice. We hope that our work withhelp the scientists on their further research on finding a treatment that may help the learning process and ease the intellectual disability caused by Down Syndrome.

Introduction

Down Syndrome (trisomy 21) (DS) is caused by an extra copy of the human chromosome21 Hsa21. The central feature of DS is the impaired intellectual function and the intellectual disability (ID)[1][2][3][4][5][6]. Currently, Hsa21 is estimated to contain 234 protein-coding genes[7]. It also shares many features with Alzheimer's Disease (AD)[3][8][9], such as the deposition of both amyloid plaques and neurofibrillary tangles[10]. On their work, Ahmed et al.[11][12] and Costa et al.[13] have shown that mouse model Ts65Dn, a Down Syndrome model of mice expresses a learning recovery capability when using Context Fear Shocking and a pharmaceutical drug used on treatment of AD called memantine. More recent studies are focused on investigating the effects of memantine on the learning process[14][15][16][17] and Smalheiser, Neil R.[18] proposes ketamine as a neglected therapy for AD. Since DS is invariably leading to 44 early-onset AD[19] and a deep analysis of dementia phenomenon in DS[20], it is crucial to understand what are the proteins affecting the learning process. Ahmed et al.[12] measured 85 prontei levels in the hippocampus and cortex of the mice model Ts65Dn and normal mice to identify how these proteins affect the learning process. The dataset was published online by [21], where the total number of proteins is 77. Since then, different analysing methods are used to study the relationship between the proteins and the learning process. The first study performed by Ahmed was done by using statistical models. Later, Higuera et al.[21] applied unsupervised learning Self Organizing Maps (SOM) algorithm in order to identify the critical proteins rather than statistical models. However, since then the problem has been treated more as a classification problem, rather than a clustering problem as thought by Higuera et al. On their paper, Eicher et al.[22] used linear Support Vector machines (SVM) in order to identify the proteins that can classify two different classes of mice. In[23] B. Feng used adaptive boosted Decision Tree (AdaBoost) method as forward feature selection to identify the most correlated proteins, then they used Random Forest (RF), SVM, and Decision Tree (DT) algorithms for classification. On their paper, Kulan et al.[24] have used the same procedure steps as B.Feng et al. by only changing the feature selection method. They thought that Naïve Bayes would discriminate the protein better than AdaBoost, and they were right, since they got a higher accuracy than the previous work. On their recent paper, Kulan et al.[24] compared their results with the ones obtained from Higuera et al. related to finding the critical proteins that affect three types of different learning, such as successful, rescued and failed learning. They used again Naïve Bayes for feature selection in order to identify the correlated proteins, and their results were higher than the previous ones.

In this work, we aim to find the critical proteins by applying a different forward feature selection method. The subset of features was selected from 77 protein expression levels obtained from the hippocampus and the cortex of normal and

Ts65Dn trisomic mice. After feature selection, RF, SVM, Neural network (NN), and K Nearest Neighbour (KNN) classification algorithms were applied in order to find the critical proteins in two different problems: multiclass classification and learning process. For multiclass classification, we compared our results with B. Feng et al. and Kulan et al.[24] works in which AdaBoost and Naïve Bayes were used for feature selection. Regarding the learning process, we compared our results with Higuera et al work with Kulan et al.[25] that used SOM and supervised learning respectively. Our results show that the selected protein subsets have a higher accuracy in classification then the previous related models. The subset of proteins selected in our work help researchers have a better understanding of the protein involvement on the learning process, providing solid grounds on drug development for treating the ID.

The rest of paper is designed as follows: the second section is dedicated to the related works; third section describes the materials and methods used in this paper. Results are shown on the fourth section and a deep analysis is done on the discussion part at the fifth section. We conclude our findings in the sixth section.

Related Works

Ahmed et al[11 12] measured the expression levels of 84 protein in the hippocampus and the cortex of normal mice and trisomic mice. By doing so, they hoped to provide a better understating of their effect on the learning process. Memantine is known to help improve the learning process on AD patients[13 26], so they thought of analysing its effect also on DS. They also used Context Fear Shocking (CFS) and Shock Context (SC) during the learning process. By applying 3 LME statistical analysis, they showed that when exposed to either CFC or memantine, the level of proteins in hippocampus and cortex changes significantly. Also, a study related to the protein dynamics of the trisomic mice was done. At the end of their work, they concluded that the protein expression level of normal and trisomic mice are very different.

Meanwhile, Higuera et al[21] posit that machine learning method could give a better understanding than the work previously done by Ahmed et al. They saw the problem as a clustering problem, so they applied unsupervised learning in order to identify the critical proteins related to the learning process. By applying SOM on 77 proteins out of 85, they found a set of specific clusters for each class with at least 80% of its information coming from one mouse. The Wilcoxon rank-sum test was used to find the different proteins that are described as critical between two classes.

However, Eicher et al[22] disagreed with the view of Higuera et al of treating this problem as a clustering one. Rather, they proposed to view it as a classification problem, where the relevant proteins among different classes need to be selected. Also, compared to the validation of the accuracy, it would be easier to apply different validation method such as cross validation, accuracy, precision etc, rather than visual representation. However, they agreed with Higuera et al. for using the Wilcoxon rank-sum test for statistical analysis. So, they used linear SVM for classification between two classes of mice and finding the discriminating proteins. By using this method, they obtained a higher accuracy than the previous reported values.

B. Feng et al[23] used forward feature selection to reduce the number of proteins from 77 to 30 by applying AdaBoost learning method. They applied different classification methods, such as Random Forest, Decision Trees, and SVM to classify normal and trisomic mice. Their obtained results were compared with the classification results when using all the features. An increment of accuracy with their reduced protein subset (98% by applying Random Forest) was shown compared to the previous models built from the original protein data-sets.

On the other hand, Kulan et al[24] argues that Naïve Bayes feature selection method could provide a better protein subset rather than AdaBoost used by B.Feng et al. Same as the previous work, they reduced the dataset from 77 proteins to 30 and applied Random Forest, Deep Neural Network, and SVM for classification. They compared their results with B. Feng et al, and showed that their method gives a higher accuracy, by reaching 99% accuracy using DNN.

Sara et al.[27] proposed a quantitative approach to investigate protein expression in order to identify important differences in protein levels in mice exposed to CFC by using machine learning feature selection algorithms. Four different selection models are used such as Fisher score, Chi score, correlation-based approach and Deep feature selection (D-DFS) proposed by Yifeng et al.[28]. SVM classification method and 5 fold validation method are used in order to check the performance of each set. Their results showed that Sod1, CaNA, Ubiquitin, and nNOS proteins have the highest impacts on learning process. They selected 15 proteins with a two hidden layer process and the highest accuracy they have reached was 93%. On the trisomic case, their accuracy was 99%. D-DFS method outperformed the other methods.

On their recent paper, Kulan et al.[25] treated the data-set differently than just multiclass classification. They wanted to find the correlated proteins in successful learning, rescued learning and failed learning, mentioned previously by Ahmed et al, and Higuera et al. Naïve Bayes is used as forward feature selection and SVM, Random Forest, Deep Neural Network, and Gradient Boosted Tree as classification models. They compared their results with Higuera et al. work and showed that their method has a higher accuracy than the previous work. Also, they made an analysis of the protein subset selected from each method. They concluded that their subsets contain proteins that are more critical related to the learning process than the ones on Higuera et al. work.

Methods

Exploring the dataset

The dataset used in this paper UCI (University of California Irvine) Machine Learning Repository dataset, publicly available at **[21]**. It consists of 77 numerical values (proteins modifications expression levels) and 3 categorical values such as genotype, behaviour, and treatment. In total, there are 72 mice where 38 of them are control mice and 34 are trisomic (Down syndrome) mice. For each mouse, 15 measurements are collected, thus 570 measurements in total for control mice and 510 measurements in total for trisomic mice. The dataset contains 1080 samples where each measurement can be considered as independent variables. As mentioned before, there are two different classes for genotype: control and trisomic. There are also two different classes for the behaviour such as context-shock CS (stimulated to learn) and shock-context SC (not stimulated to learn). For treatment also there are two different classes where the first group of mice were injected with memantine and the second group did not get any treatment. Table1 shows the summary of the mice classes.

		# of mice
c-CS-s	control mice, stimulated to learn, injected with saline	9
c-CS-m	control mice, stimulated to learn, injected with memantine	10
c-SC-s	control mice, not stimulated to learn, injected with saline	9
c-SC-m	control mice, not stimulated to learn, injected with memantine	10
t-CS-s	trisomy mice, stimulated to learn, injected with saline	7
t-CS-m	trisomy mice, stimulated to learn, injected with memantine	9
t-SC-s	trisomy mice, not stimulated to learn, injected with saline	9
t-SC-m	trisomy mice, not stimulated to learn, injected with memantine	9

Table 1. Summary of classes on the dataset

Data Preprocessing

Missing data Most of the related works have used the mean in order to fill the missing values. We wanted to try a new method by filling the missing values with the most frequent value of each mouse class. To compare these methods two datasets are prepared: D1 where the missing values are filled with the most frequent value method and D2 where the missing values are filled with the mean method.

Data Normalization In order to prevent the influence of the proteins with higher value, there is a need for data normalization. Higuera et al.[21] has applied max-min normalization, which in fact does not preserve the range. Because of this, Kulan et al.[25] propose the Z-score normalization by subtracting mean of values from each value and divided by standard deviation in the end. On this paper, a different normalization technique in which each value is normalized in the range of $[-1,1]$ is used.

Feature Selection

Since a comparison of our work with other related works will be done in the end, the number of features selected is the same with the one used on the literature. As mentioned before, the comparison will be done in two different problems. The first problem is multi class classification of all the data. Our results will be to compared with Kulan et al.[24] and B.Feng et al[23] for this problem. They have reduced the dimension of the data from 77 proteins (excluding the categorical values) to 30 proteins. B.Feng et al has used AdaBoost learning method and Kulan et al. has used Naïve Bayes classifier for feature selection. In order to compare our results, in this work also 30 features are selected out of 77 proteins. In order to go a little bit deeper, we tried to do multiclass classification by reducing even more the dimensions to 11, 10, and 9 features. The second problem is related to finding the critical proteins subset related to normal learning, rescued learning, and failed learning. Our results will be compared with the ones in Higuera et al.[21] and Kulan et al.[25]. Higuera et al. applied SOM (Self Organizing map) for such cases and Kulan et al. used forward feature selection using Naïve Bayes classifier. For successful learning they have selected 11 features. For rescued learning they have selected 9 features and for failed learning they have selected 9 features. We have used the same number of features for each class.

The feature selection is done by Relief-based method ReliefF. Relief-based algorithms (RBAs), a unique family of filter-style feature selection algorithms are the only individual evaluation filter algorithms capable of detecting feature dependencies[29]. They use the information taken from the nearest neighbours in order to speculate the feature weights. By doing this, no search is involved on trying to combine the features, but they try to find the features that affect indirectly. Moreover, RBAs have an asymptotic time complexity of O ($instances^2 * features$) (quadratic function) giving them a feature that make them

advantageous compared to the other algorithms. This feature makes them relatively fast and may save many computational efforts.

0.1 Classification

After feature selection, different algorithms comparable with the other works based on our problems are selected. We used four different classifiers such as KNN (K-Nearest Neighbour), NN (Neural network), SVM (Support Vector Machines) and RF (Random forests). KNN is not used in any of the work compared, but it is added by us in order to check the accuracy of this algorithm with our methods. The software used for classification is Orange Inc[30].

KNN (K-nearest neighbour) KNN[30] is a supervised machine learning algorithm that can be used for both classification[31,32,33,34,35,36] and regression predictive problems[37,38]. However, it is generally used in classification problems rather than regression. It is one of the simplest classification algorithm, taking in consideration the distance of the to-be-classified object with its k neighbours. The KNN algorithm assumes that similar things tend to stay closer to each other. In other words, similar things have a short distance among themselves. In order to identify the similar objects, KNN calculates the distance between the objects, by classifying them according to their shortest distances. Even though it is a very simple algorithm, it is still being used in classification problem and gives highly competitive results. On this work, we have used K (the number of nearest neighbours) as 5 and Manhattan distance as input parameters.

NN (Neural Network) Neural network[39] is another method that is used for protein classification[40,41,42,43]. NN finds correlation between inputs and outputs by mapping them, through a process done in multiple hidden layers. These layers are made by nodes. A node is the place where the computations, such as assigning different weights to inputs, are executed. Then, these computations are passed through an activation function. By using the training process, they can adapt themselves to the data and change the weights properly without any explicit intervention. They are very robust towards noise and missing values and achieve higher results when more layers are added. But this at a cost of increase of computational times, which can make them slow. In our case, we have a neural network with 100 neurons in hidden layer. The activation function is a rectified linear units (ReLu) function[44] and the solver is Adam solver. Maximum number of iteration is 200.

SVM (Support Vector Machines) SVMs[45,46] classification is done by determining a decision plane which discriminates best a set of object dataset of different classes. The data are mapped into kernel, which usually is a higher dimension. This is very useful especially on the cases where the data are very difficult to separate on lower dimensions. The distance between decision plane and nearest data point from either part of the plane is known as the margin. The margin should be higher in order to have a better classification. The classification of an object is done by checking the margin values between the object and the classes. SVM works well on small datasets, however it is not very robust on noisy data with too many overlapping objects. It is found a practical method for protein classification[47,48,49,50]. In our work, the kernel is radial basis function RBF[51] and the maximum number of iteration is 100.

RF (Random Forest) Random forest[52] (strong learners) is a collection of many decision trees (weak learners) selected from a random subset of training set. For each classification result, a vote is taken from each tree. In the end, the classification result may either be the average or the mode of the voting from each individual tree. Random forest is especially robust to missing values. It has been used in classification problems[53,54,55,56] and regression problems[57,58,59]. Our parameters are: the number of trees is 10 and subsets smaller than 5 should not split.

Validation method

Cross validation method [62] is used to check how accurately the predicted models will perform in practice. Here, k-fold cross validation is used for evaluation. The dataset is split randomly into k folds (subsets) of equal sizes. Then the model is trained and tested k times. The accuracy is the ratio between the correct classifications over the total number. In our case, sometimes 5 fold validation is used, sometimes 10 fold validation and sometimes random validation, and random sampling, meaning every sample is selected randomly by a random seed

Results

Multiclass classification problem

As mentioned above, for this problem we are going to compare our results with Kulan et al. [25] and B.Feng et al [24]. For classification technique 10-fold cross validation result. This is also the same method used by the other authors. We have added extra random sampling where 90% of the data is used for training and 10% for testing. The precision results are reported on Table 2. As we can see from the table, our results (in percentage) are higher than the ones reported on the other works, except for Random Forest. Comparing D1 with D2, we have a difference of 0.01% in precision.

Method	B.Feng et al.	Kulan et al.	D1		D2	
			10 fold	Random	10 fold	Random
KNN	-	-	**99.8**	**99.8**	**99.8**	**99.9**
NN	-	99	**99.4**	**99.5**	**99.4**	**99.4**
RF	98	98.2	97.9	97.6	97.9	96.7
SVM	93.3	97.1	**99.8**	**99.9**	**99.9**	**99.8**

Table 2. Results compared to other works

In order to find the critical proteins, we reduced the dimensionality of the data even more: to 11, 10 and 9 respectively. Here we used only random sampling for result validation. On Table3 we can see results for D1 with 11 features. We still have a high precision, 99.5% from KNN which is greater than the results reported on the related works with 30 features.

Method	AUC(%)	CA(%)	F1(%)	Precision(%)	Recall(%)
KNN	100	99.5	99.5	99.5	99.5
RF	99.8	95.5	95.5	95.5	95.5
NN	99.8	94.4	94.3	94.3	94.4
SVM	99.7	92.8	92.7	92.8	92.8

Table 3. D1 results with 11 features

On Table4, we have reported the results taken from D2 with 11 features. As we can see, KNN results of D2 compared with D1 have a very small difference compared to the difference of other methods results. In fact, they are higher than D1. Compared to other related work, we still have a higher precision (99.5% from KNN and 98.3% from NN).

Method	AUC(%)	CA(%)	F1(%)	Precision(%)	Recall(%)
KNN	100	99.4	99.4	99.5	99.4
NN	99.9	98.2	98.2	98.3	98.2
SVM	99.9	97.1	97.1	97.2	97.1
RF	99.8	95.7	95.7	95.8	95.7

Table 4. D2 results with 11 features

The difference of precision accuracy from 30 features is only 0.5%. So, in case we have a reduced dataset with this subset of features, we would still have a high accuracy on classification. Table 5 shows the results taken from D1 when selecting 10 features. The accuracy of the other methods starts dropping, but still KNN accuracy (99.4%) is higher than the related work with 30 features. We have only a 0.1% difference on the precision accuracy taken from 11 features.

Method	AUC(%)	CA(%)	F1(%)	Precision(%)	Recall(%)
KNN	**100**	**99.4**	**99.4**	**99.4**	**99.4**
NN	100	96.7	96.7	96.7	96.7
RF	99.9	96.4	96.4	96.4	96.4
SVM	99.6	89.9	89.5	91.2	89.9

Table 5. D1 results with 10 features

On the other hand, we see an increase on the accuracy of KNN (99.6% or 0.1% greater than 11 features or 0.2% greater than D1 results) when selecting only 10 features (see Table6). This means that the proteins subset used for classification in this case are more critical and more important. We notice an increase on the accuracy of NN and SVM, but a decrease on the accuracy of RF compared to D1 results.

The last results to be shown for this problem are the results taken after selecting only 9 features. Table7 shows the results taken from D1 dataset. Again, we see that KNN classification accuracy is greater than the ones reported on the related work (99.1%) compared to the best result Kulan et al. [25] reported from NN when selecting 30 features (99%). This is an important notice since we have a greater accuracy when the number of features is very small compared to the dataset (9 out of 77 proteins) and smaller than the 30 features that the previously related work have reported.

Method	AUC(%)	CA(%)	F1(%)	Precision(%)	Recall(%)
KNN	100	99.6	99.6	99.6	99.6
NN	99.9	97.2	97.2	97.3	97.2
SVM	99.9	96.6	96.6	96.6	96.6
RF	99.8	94.7	94.7	94.8	94.7

Table 6. D2 results with 10 features

Method	AUC(%)	CA(%)	F1(%)	Precision(%)	Recall(%)
KNN	100	99.1	99.1	99.1	99.1
NN	99.8	96.1	96.1	96.1	96.1
SVM	99.8	95.7	95.7	95.8	95.7
RF	99.8	92.7	92.6	93.1	92.7

Table 7. D2 results with 10 features

However, the other methods have a lower accuracy than KNN. SVM, NN and RF report a lower accuracy when the dimensions are reduced, which is also true for KNN, but the difference from dimensionality reduction on KNN is smaller than on the other methods. Table8 shows the results of D2 with 9 features selected. Again, we notice the same trend on these results as the ones when selecting 10 features for NN, SVM and RF. The first two report an increase in the accuracy, whereas RF report a decrease in the accuracy.

Method	AUC(%)	CA(%)	F1(%)	Precision(%)	Recall(%)
KNN	99.1	99.1	99.1	99.1	99.6
NN	99.7	97.1	97.1	97.2	97.1
SVM	99.7	94.8	94.8	94.9	94.8
RF	99.7	94.3	94.3	94.3	94.3

Table 8. D2 results with 9 features

As a conclusion, we may say that applying ReliefF for feature selection and KNN method for classification will yield us a better accuracy than the other methods so far, even when selecting a very small number of features (12% of the proteins in the dataset).

Critical proteins subset of multiclass classification problem

The results showed an increase on the precision accuracy, with KNN algorithm performing best. This means that our method for finding the critical proteins is proved to be effective. The first run was done with the selection of 30 proteins, then with 11, 10 and 9 features. We notice that the accuracy of D1 and D2 differs, which means that there should be different protein subsets selected as features. We start by comparing the subsets when selecting 11 features. On Table 9 we see the proteins subsets selected as 11 features on D1 and D2. Out of 11 proteins, 2 are different in any case: H3MeK4_N and H3AcK18_N for D1 and AcetylH3K9_N and APP_N for D2. The rest is the same. Depending on the method used, either D1 or D2 will give the highest accuracy. For example, KNN on D1 gives the accuracy 99.5% whereas on D2 gives a lower accuracy of 99.4%. However, SVM on D1 gives the accuracy 92.8% whereas on D2 gives a higher accuracy of 97.1%.

Let's see the protein subsets determined when selecting 10 features. Again, we see from the subsets shown on Table 10 that the subsets differ from 2 proteins: PKCA_N and H3MeK4_N on D1 and Ubiquitin_N and Braf_N on D2 respectively. The rest is all the same. If we compare the accuracy, we have an increase on accuracy on D2 for KNN, NN and SVM but a decrease for RF.

The last comparison is between the subsets of proteins selected with 9 features. Even on these subsets, only two proteins differ from subsets (see Table 11) and all the rest are the same. On D1 we have protein BRAF_N and H3MeK4_N as different, whereas on D2 we have APP_N and pGSK3B_N. The accuracy of classification changes from the method used and from the dataset.

If we compare the subsets of 11, 10 and 9 features in order to find the most critical proteins that need to be in every feature vector, the comparison is done on Table 12. From these subsets we found the 9 critical proteins which are CaNA_N, pPKCG_N, SOD1_N, pCAMKII_N, S6_N, H3MeK4_N, pP70S6_N, and APP_N.

As a conclusion, we may say that the pre-processing method that we use for filling the missing values affects the proteins subsets selected as features for classification. KNN algorithm has given the highest accuracy so far, where the difference on

D1 protein subset	D2 protein subset
CaNA_N	CaNA_N
pPKCG_N	pPKCG_N
pPKCAB_N	pCAMKII_N
SOD1_N	SOD1_N
pCAMKII_N	pPKCAB_N
S6_N	S6_N
H3MeK4_N	pP70S6_N
pP70S6_N	Tau_N
H3AcK18_N	**AcetylH3K9_N**
Tau_N	Ubiquitin_N
Ubiquitin_N	**APP_N**

Table 9. Protein subsets selected as 11 features on D1 and D2

D1 protein subset	D2 protein subset
CaNA_N	CaNA_N
pPKCAB_N	pCAMKII_N
SOD1_N	pPKCG_N
pCAMKII_N	SOD1_N
pPKCG_N	pPKCAB_N
S6_N	S6_N
pP70S6_N	pP70S6_N
PKCA_N	**Ubiquitin_N**
H3MeK4_N	APP_N
APP_N	**BRAF_N**

Table 10. Protein subsets selected as 10 features on D1 and D2

D1 protein subset	D2 protein subset
CaNA_N	CaNA_N
pPKCG_N	pPKCG_N
pCAMKII_N	pCAMKII_N
SOD1_N	SOD1_N
pPKCAB_N	pPKCAB_N
S6_N	pP70S6_N
pP70S6_N	S6_N
BRAF_N	**APP_N**
H3MeK4_N	**pGSK3B_N**

Table 11. Protein subsets selected as 9 features on D1 and D2

the accuracy from different pre-processing method is very small compared to the other methods. Using ReliefF for feature selection proved to be an effective method for finding the critical proteins. After merging the proteins selected on D1 and D2 for 11, 10 and 9 features, we found 9 most critical proteins. Even using these 9 features we have an accuracy of 99.1% for multiclass classification given from KNN.

Successful learning, Rescued learning and Failed learning

As mentioned above, the second problem is related to finding the critical proteins subset related to normal learning, rescued learning and failed learning. Higuera et al.[21] and Kulan et al.[25] have done some work for this problem. Higuera et al. applied SOM or known differently as Kohonen map approach in order to identify the proteins subset that make the most critical contribution on learning. The authors treated the problem as a clustering problem in order to identify these proteins. Kulan et al. however treated this as classification problem and used Naïve Bayes for feature selection. They used DNN, GBT (Gradient Boosted Tree), RF and SVM for classification. The validation is done using 5 fold cross validation. We are going to compare only the results with DNN, RF and SVM and we add also KNN as a classifier. For comparison purposes, we also used 5 fold

11 features (D1&D2)	10 features (D1&D2)	9 features (D1&D2)	Common proteins
CaNA_N	CaNA_N	CaNA_N	**CaNA_N**
pPKCG_N	pPKCAB_N	pPKCG_N	**pPKCG_N**
pPKCAB_N	SOD1_N	pCAMKII_N	**pPKCAB_N**
SOD1_N	pCAMKII_N	SOD1_N	**SOD1_N**
pCAMKII_N	pPKCG_N	pPKCAB_N	**pCAMKII_N**
S6_N	S6_N	S6_N	**S6_N**
H3MeK4_N	pP70S6_N	pP70S6_N	**H3MeK4_N**
pP70S6_N	PKCA_N	BRAF_N	**pP70S6_N**
H3AcK18_N	H3MeK4_N	H3MeK4_N	**APP_N**
Tau_N	APP_N	APP_N	
Ubiquitin_N	Ubiquitin_N	pGSK3B_N	
AcetylH3K9_N	BRAF_N		
APP_N			

Table 12. Comparison of protein subsets for 11, 10 and 9 features

cross validation and added 10 fold and random sampling as other validation methods. Table 13 shows the accuracy results of our methods and the comparison with the other methods for successful learning. By checking the results taken from 5 fold cross validation, we notice an increase on the accuracy using D1 and D2. We also notice that when using 10 fold cross validation, we have greater accuracy, where KNN has the highest with 99.8%. If we compare D1 and D2 results, they differ depending on the method. On KNN, NN and SVM we see an increase on the accuracy, whereas RF changes from the validation method. Still, comparing the results means that our subset has found more critical proteins than the other methods.

	Higuera et al.	Kulan et al.	D1(%)			D2(%)		
			10 fold	5 fold	Random	10 fold	5 fold	Random
KNN	-	-	**99.8**	**99.2**	**99.5**	**99.8**	**99.8**	**99.8**
NN	96.7	97.2	99.1	**97.3**	98.4	99.5	**98.8**	98.9
RF	90.2	96.3	97.2	**98.1**	95.8	98.1	**96.5**	97.7
SVM	96.1	98.1	98.6	**98.1**	98.7	98.6	**98.6**	99.5

Table 13. Successful Learning accuracy comparison results

For rescued learning, 9 features are selected. The results of our method and the comparison with the other related work are shown on Table 14. We may notice that using D1, we have a higher accuracy for all the methods compared to the others using 5 fold cross validation. This differs when using D2, since SVM accuracy is lower than the previous work, and NN reports the same accuracy. Comparing D1 results with D2, we see that the results of D1 are greater than the results of D2. This means that the subset selected from D1 has more critical proteins than D2. The highest accuracy is given by KNN with 99.5%.

	Higuera et al.	Kulan et al.	D1(%)			D2(%)		
			10 fold	5 fold	Random	10 fold	5 fold	Random
KNN	-	-	**99.8**	**99.2**	**99.5**	**99.8**	**99.8**	**99.8**
NN	95.4	97.1	97.9	**97.3**	98.4	97.6	97.1	96.3
RF	88.3	94.6	97.1	**98.1**	95.8	98.1	**96.3**	96.1
SVM	92.1	97.1	98.1	**98.1**	98.7	94.9	94.4	93.9

Table 14. Rescued Learning accuracy comparison results

For failed learning, 10 features are selected. The accuracy results are reported on Table 15. The first thing that we notice is that using KNN for classification we have an accuracy of 100%, meaning that all the data are classified accordingly. This means that the subset chosen in this case is the best one so far that can classify the data 100% correctly. Comparing the results taken from 5 fold cross validation, we see that there is an increase on the accuracy, with the biggest difference (9.7%) on RF from 89.2% to 98.9% taken using D2. KNN and SVM give the best results, whereas RF in average reports the lowest.

	Higuera et al.	Kulan et al.	D1(%)			D2(%)		
			10 fold	5 fold	Random	10 fold	5 fold	Random
KNN	-	-	**100**	**100**	**100**	**100**	**100**	**98.9**
NN	92.1	92.6	99.3	**99.3**	100	99.3	99.3	99.6
RF	85.9	89.2	99.6	**98.1**	100	98.1	**98.9**	99.6
SVM	91	92.6	100	**99.6**	100	99.3	99.3	100

Table 15. Failed Learning accuracy comparison results

Critical proteins for Successful learning, Rescued learning and Failed learning

Comparing our method results with the other related work, we have an increase on the accuracy compared with the others. We noticed also that the accuracy varies on D1 and D2 for different methods for each type of learning, meaning again that we have different protein subsets for each dataset. We compare them in order to find the critical proteins for each type of learning. On Table 16, we show the proteins subsets selected for successful learning on D1 and D2. We notice that they differ only on 1 protein selected as feature: D1 differs on H3AcK18_N and D2 differs on ADARB1_N. This explains also why we have different accuracy for different methods.

D1 protein subset	**D2 protein subset**
SOD1_N	SOD1_N
CaNA_N	CaNA_N
pPKCG_N	pPKCG_N
pCAMKII_N	pCAMKII_N
pPKCAB_N	pPKCAB_N
pNUMB_N	S6_N
pP70S6_N	pNUMB_N
S6_N	pP70S6_N
Ubiquitin_N	pGSK3B_N
pGSK3B_N	Ubiquitin_N
H3AcK18_N	**ADARB1_N**

Table 16. Proteins subsets selected on successful learning on D1 and D2

Now let's compare our proteins with the ones provided by Higuera et al. and Kulan et al.[24] on their respective work. The comparison is shown on Table 17, where we notice that for each group there are only five proteins same with the ones found on our method. On Kulan et al. protein subset for example, the common proteins are SOD1_N, Ubiquitin_N, pGSK3B_N, S6_N and CaNA_N. On Higuera et al. protein subsets, the common proteins found are SOD1_N, pNUMB_N, pGSK3B_N, S6_N and CaNA_N. The difference in the accuracy values is explained also by the low number of the same proteins found as critical on the other methods. We may say that these proteins subsets found with our methods are more critical in order to have a successful learning, and really affect the ability to have a successful learning.

Rescued Learning proteins found as feature subsets had a fixed length of 9 proteins. Again, because of the varying accuracy of D1 and D2 in different methods, the feature subsets should differ with each other. They differ only on one protein: D1 differs on pPKCAB_N and D2 differs on pP70S6. The other proteins are the same. The protein subset of each dataset are shown on Table 18.

We compare our protein subset with the ones of the other related work in the following Table 19. We notice fewer same proteins than for the successful learning. With the subset of Kulan et al.[24] for example, there are only two common proteins BRAF_N and SOD1_N. With the Higuera et al. protein subset instead there are 3 common proteins: DYRK1A_N, pERK_N, and BRAF_N. Even though there are more common proteins with Higuera et al. subset than Kulan et al., the accuracy of Kulan et al. is higher than Higuera et al. But looking the accuracy of our algorithms in order to identify the proteins that are important on rescued learning, we may say that our subset has found the critical proteins.

Failed learning has a fixed protein subset of length 10. By using different pre-processing methods, we have different proteins subsets. This was also shown above by a varying accuracy among the datasets D1 and D2. On Table 20, the protein subsets are shown and it can be seen that they differ only by 1 protein. On D1 we have the protein BAD_N present, whereas on D2 we have the protein pRSK_N instead.

The proteins that we have found seem to be the critical proteins, because of the high accuracy that we get on classification.

Protein subset (D1 & D2)	Kulan et al. [26]	Higuera et al.
SOD1_N	**SOD1_N**	DYRK1A_N
CaNA_N	**Ubiquitin_N**	ITSN1
pPKCG_N	**pGSK3B_N**	pERK
pCAMKII_N	**S6_N**	BRAF
pPKCAB_N	**CaNA_N**	**SOD1_N**
S6_N	IL1B	**pNUMB_N**
pNUMB_N	BAX	**pGSK3B_N**
pP70S6_N	pNR2A	CDK5
pGSK3B_N	BDNF	**S6_N**
Ubiquitin_N	pJNK	GFAP
H3AcK18_N	pCFOS	**CaNA_N**
ADARB1_N		

Table 17. Critical proteins on successful learning

D1 protein subset	D2 protein subset
Tau_N	Tau_N
BRAF_N	pPKCG_N
AcetylH3K9_N	BRAF_N
SOD1_N	AcetylH3K9_N
pERK_N	pERK_N
CaNA_N	SOD1_N
pPKCAB_N	**pP70S6_N**
pPKCG_N	DYRK1A_N

Table 18. Proteins subsets selected on rescued learning on D1 and D2

Protein subset (D1 & D2)	Kulan et al.	Higuera et al.
Tau_N	**BRAF_N**	**DYRK1A_N**
pPKCG_N	S6	**pERK_N**
BRAF_N	CDK5	**BRAF_N**
AcetylH3K9_N	BDNF	CDK5
pERK_N	pCREB	RRP1
SOD1_N	PKCA	GFAP
DYRK1A_N	**SOD1_N**	GluR3
CaNA_N	PSD95	P3525
pPKCAB_N	pNR2A	Ubiquitin
pP70S6_N		

Table 19. Critical proteins on rescued learning

But let's compare this subset with the ones of Higuera et al. and Kulan et al.[24] work. By observing the subsets shown on Table 21, we notice that there are only 2 common proteins with Kulan et al. subsets, pPKCAB_N and pCAMKII_N respectively. With Higuera et al.'s subset instead there is only one common protein pS6_N. The fact that very few proteins are in common explains the difference on the accuracy between our work and their work.

Now, after finding the protein subsets for each type of learning, we started comparing them. We wanted to compare them in order to identify the critical proteins among the different type of learning. We were surprised to notice that the different proteins that were among the subsets of different types of learning were of the same size 3. Table 22 shows these critical proteins. For successful learning, the critical proteins that are not found on the subsets of the other types of learning are pNUMB_N, Ubiquitin_N and ADARB1_N. For rescued learning the different proteins are BRAF_N, pERK_N and DYRK1A_N. And the last, for failed learning the different proteins are ARC_N, BAD_N, pRSK_N. It is important to mention that some of these proteins are among the common proteins found with the other works protein subsets such as Ubiquitin_N, pNUMB_N for successful learning and BRAF_N, pERK_N, DYRK1a_N for rescued learning. It is important to mention that the rescued learning critical proteins are also the common proteins of our subset with Higuera et al.'s subset. There is no critical protein

D1 protein subset	D2 protein subset
pPKCG_N	pPKCG_N
pPKCAB_N	pPKCAB_N
pP70S6_N	ARC_N
BAD_N	pS6_N
AcetylH3K9_N	pP70S6_N
ARC_N	pGSK3B_N
pS6_N	AcetylH3K9_N
pCAMKII_N	**pRSK_N**
H3AcK18_N	pCAMKII_N
pGSK3B_N	H3AcK18_N

Table 20. Proteins subsets selected on failed learning on D1 and D2

Protein subset (D1 & D2)	**Kulan et al.[26]**	**Higuera et al.**
pPKCG_N	P38	pNRI
pPKCAB_N	**pPKCAB_N**	APP
ARC_N	CAMKII	MTOR
pS6_N	**pCAMKII_N**	P38
pP70S6_N	GluR3	NR2B
pGSK3B_N	DSCRI	RAPTOR
AcetylH3K9_N	mNOS	**pS6_N**
pCAMKII_N	BAX	Tau
H3AcK18_N	pCFOS	Glur3
BAD_N	ERK	EGRI
pRSK_N		

Table 21. Critical proteins on failed learning

found among the common proteins for the failed learning. These subsets may help researchers advance their work by studying them in detail.

Successful Learning	**Rescued Learning**	**Failed Learning**
pNUMB_N	BRAF_N	ARC_N
Ubiquitin_N	pERK_N	BAD_N
ADARB1_N	DYRK1A_N	pRSK_N

Table 22. The proteins that are different among the subsets of different types of learning

Analysis of the methods used

The relationship of the pre-processing with the methods used

We mentioned earlier that by applying two different method to fill the missing values (with the most frequent and the mean respectively), we had a two different datasets D1 and D2. We also noticed when presenting the accuracy results that there is a varying accuracy between different methods and different datasets. So, we would like to observe how the pre-processing step that we choose in the beginning affects our classification results in the end. We start analysing this relationship by comparing the results of the multiclass classification with 9, 10, and 11 features respectively. The compared values are area under curve (AUC), cumulative accuracy (CA), F1, precision and recall. The first comparison was done for KNN and NN algorithms. As can be seen on Figure 1 and 2, the values vary among different datasets. In some cases D1 gives the highest values, sometimes D2. So we can conclude that these methods are not very much affected by the pre-processing step used to fill the missing values.

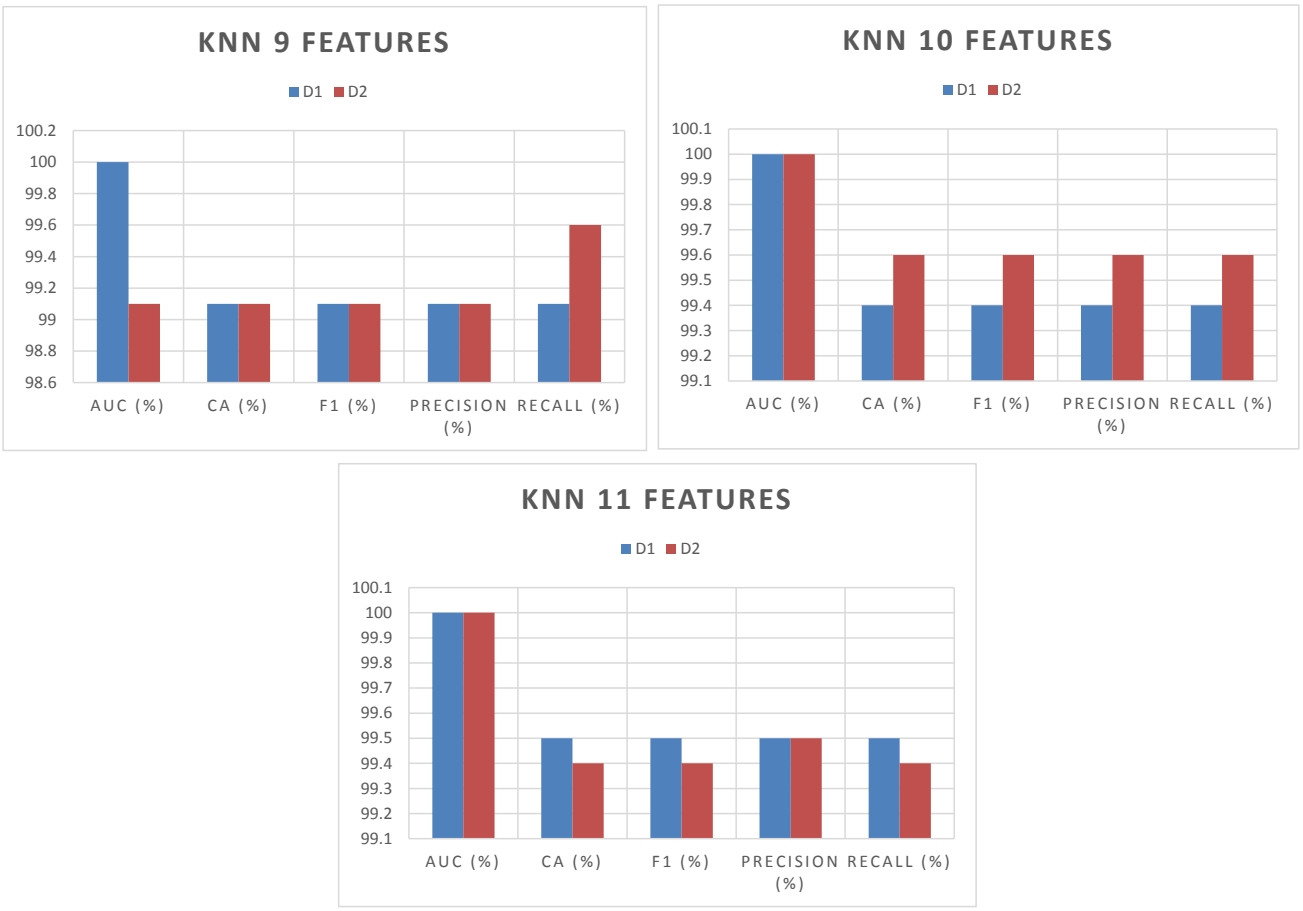


Figure 1. Comparing the accuracy results between D1 and D2 for KNN with 9, 10, and 11 features

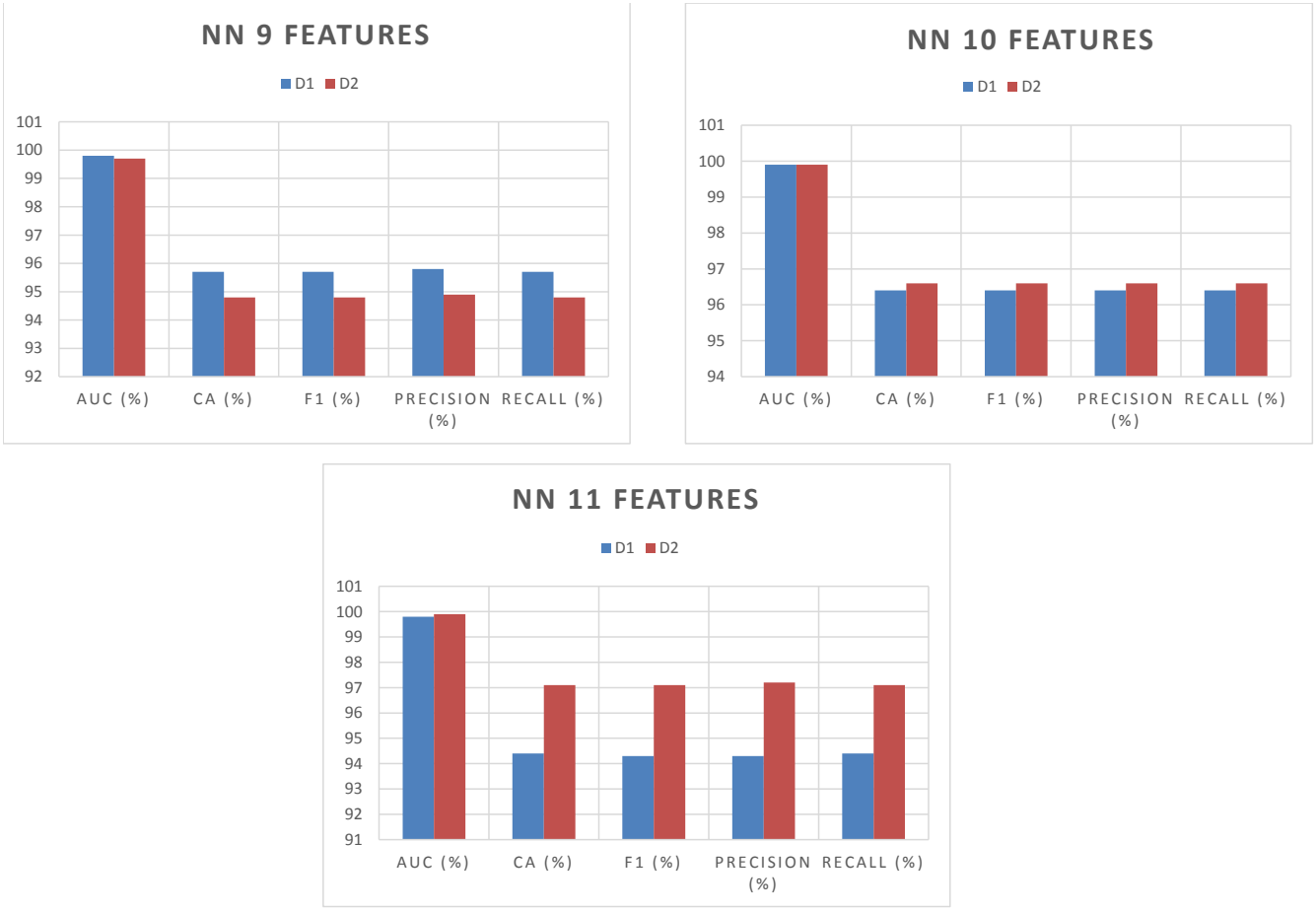


Figure 2. Comparing the accuracy results between D1 and D2 for NN with 9, 10, and 11 features

The next to be compared are RF and SVM for the same conditions. On Figure 3 and 4 the respective graphs are shown. By observing the graphs, we notice that in contrary to KNN and NN, there is a trend for RF and SVM. The accuracy values are greater for D2 than D1, which means that for these algorithms the pre-processing steps has an influence on the accuracy of the classification. If we would like to use RF and SVM for classification, it is better to use as a pre-processing step to fill the missing values by using the mean, instead of mode.

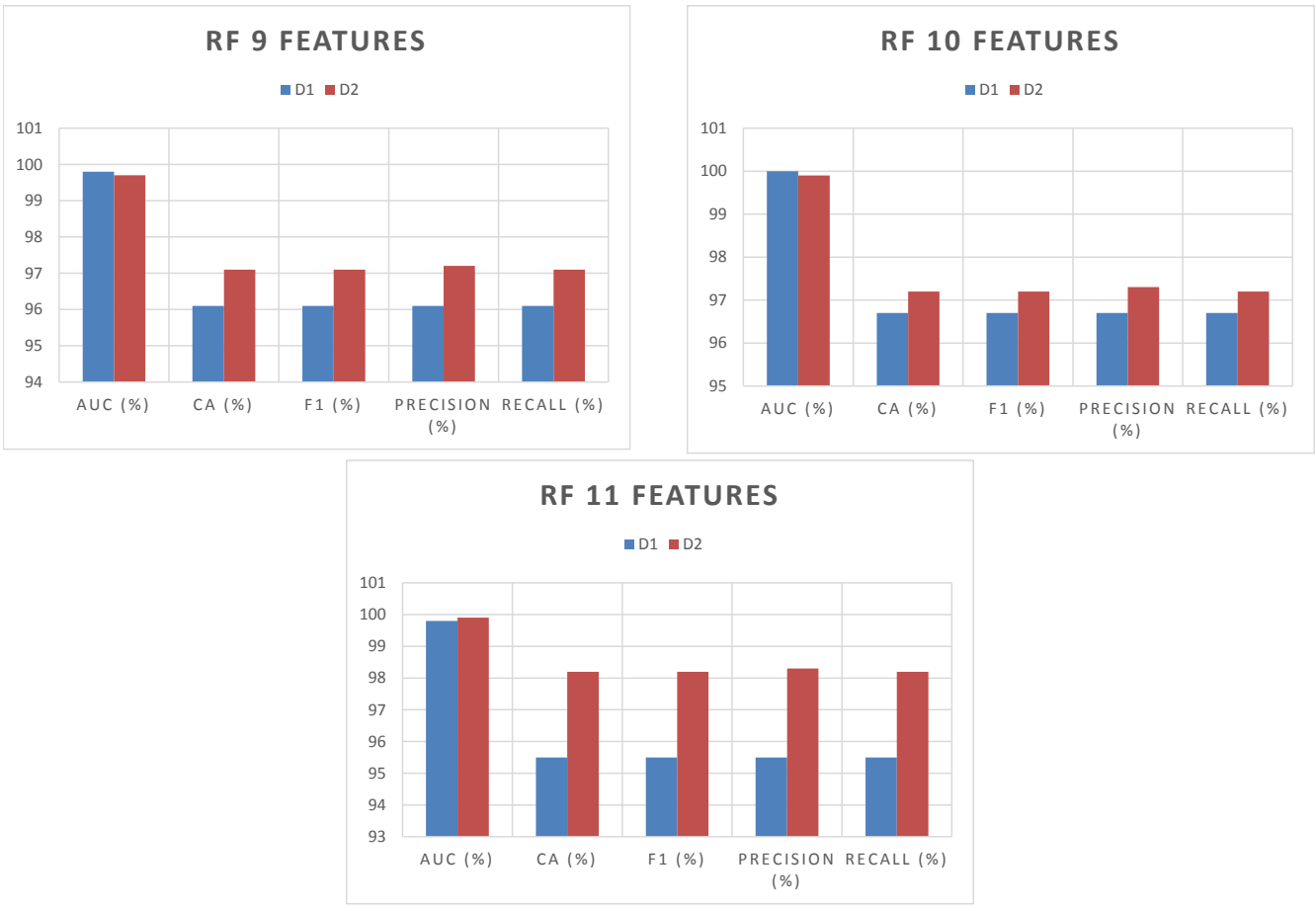


Figure 3. Compared accuracy results between D1 and D2 for RF with 9, 10 and 11 features

We can conclude that KNN and NN is not affected by the pre-processing step to fill the missing values. RF and SVM instead are affected by this step and the accuracy is higher when filling the missing values with the mean value instead of the mode. So, depending on the classification algorithm, depends also the pre-processing value that is going to be used to fill the missing values.

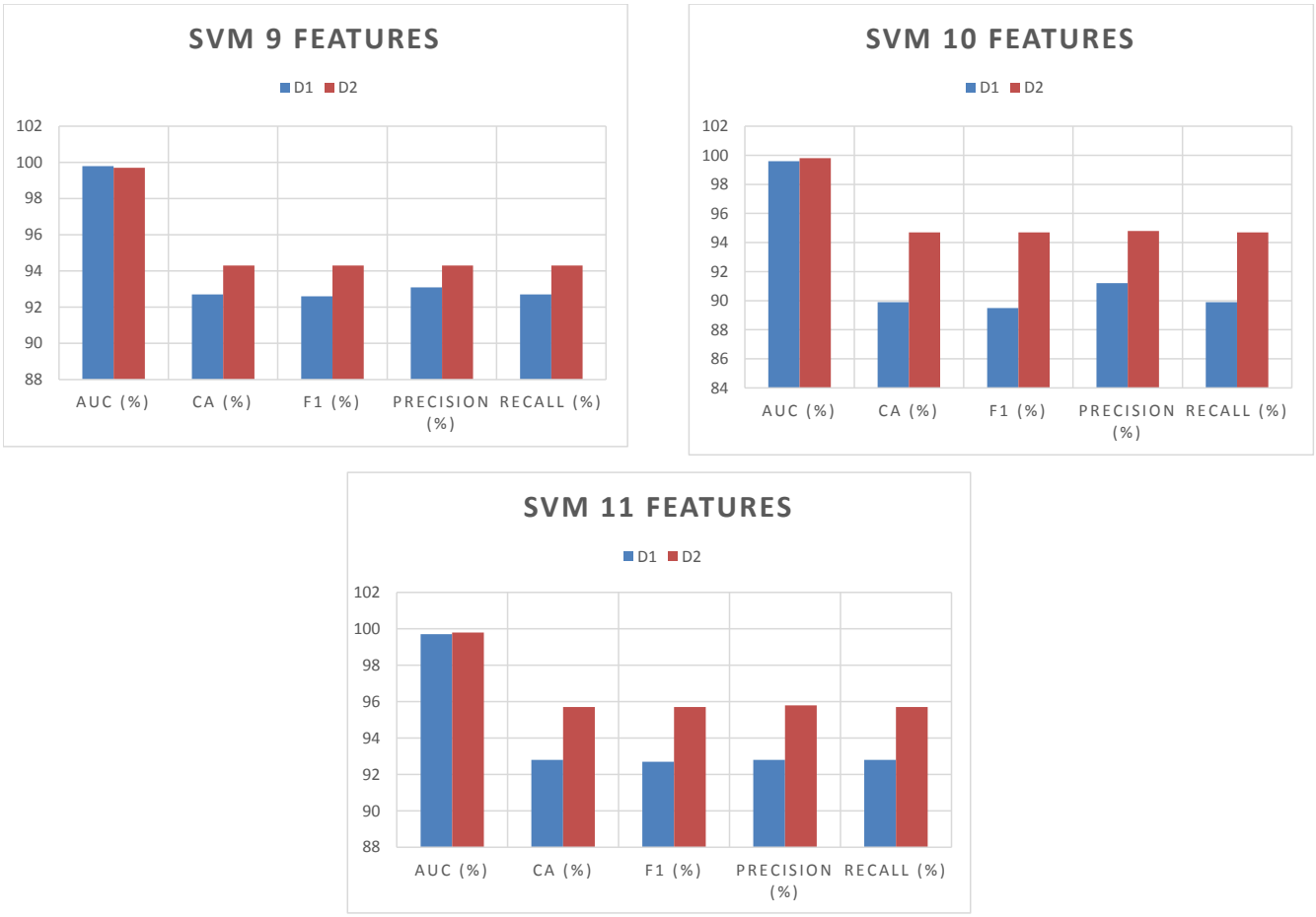


Figure 4. Compared accuracy results between D1 and D2 for SVM with 9, 10 and 11 features

The relationship between the validation method and the algorithm used for D1 and D2

The next observation is done to find the relationship between the validation method and the algorithm used for the different datasets D1 and D2. For this case, the results of different types of learning are taken for comparison. The validation methods that are going to be compared are 10 fold, 5 fold and random sampling.

The first algorithms to be compared are KNN and NN. The graphs are shown on Figure 5 and 6. As it can be noticed, we see a trend among the validation method, the algorithm and the dataset used. For successful learning, D2 gives the highest value for all types of validation method. On rescued learning instead, it gives lower values than D1. On failed learning, the values are the same for 10 fold and 5 fold, but lower than D1 for random sampling.

Figure 5. Compared results of KNN with different types of validation methods for D1 and D2

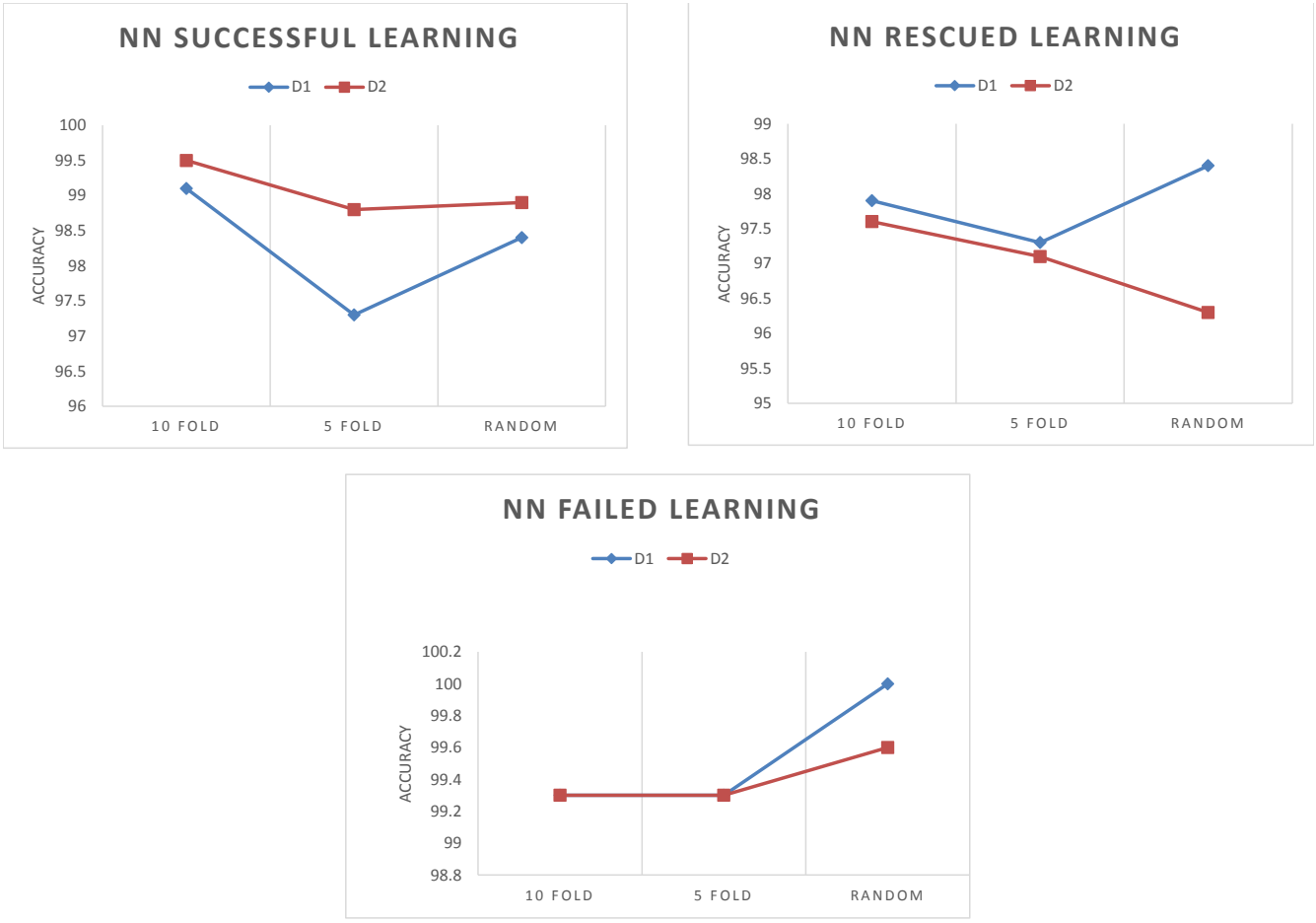


Figure 6. Compared results of NN with different types of validation methods for D1 and D2

RF and SVM are also compared and the respective graphs are shown in Figure 7 and 8. By looking at them, one can not determine a clear relationship between the validation method, the algorithm used and the dataset. But we may say that for RF algorithm it is noticed that the values of D1 and D2 for different type of validation method are the opposite of each other. Hence, whenever D1 is giving higher accuracy, D2 is lower and vice versa. SVM instead shows a different perspective. In successful learning the value of D1 and D2 are the same for 10 fold, and then D2 gives higher values. Whereas on rescued learning for each validation method D2 gives lower values. On failed learning, it gives lower values for 10 fold and 5 fold and the same value for random sampling.

Figure 7. Compared results of RF with different types of validation methods for D1 and D2

We need to keep in mind that the accuracy of the algorithms is also related to the feature subset they are using for classification. But we may conclude that we see the same trend for KNN and NN in the relationship of the validation method and the dataset used, whereas on RF and SVM there is not a clear relationship that can be described.

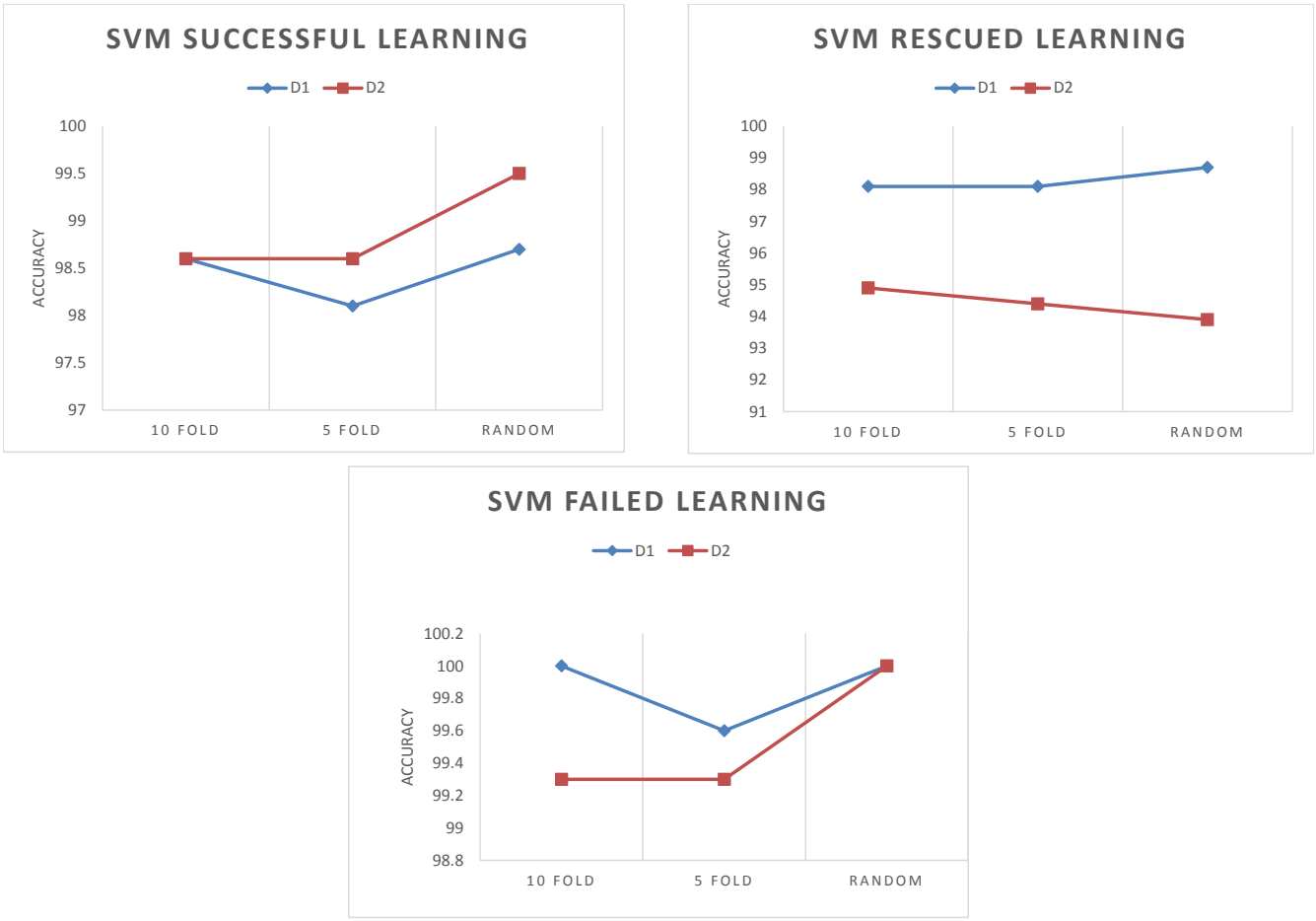


Figure 8. Compared results of SVM with different types of validation methods for D1 and D2

Conclusion

The aim of this paper was to identify the critical proteins that discriminate the classes among the mice more accurately and the critical proteins that are related to different types of learning. Two different methods are used to fill the missing values including the use of the mean of each class and the most frequent values. The algorithms used for classification are RF, SVM, NN and KNN. Data normalization is performed by normalizing all the values on the interval [-1, 1]. Forward feature selection using ReliefF is used to identify the protein subsets. The results obtained from the classification are compared to the other related works. For multiclass classification problem, protein subsets of different lengths are used such as 30, 11, 10 and 9. The first subset is used in order to compare our method with the results reported on other related papers, B.Feng et al. and Kulan et al. [25] respectively. It was shown that our method has a higher accuracy than their methods except for RF, with the highest achieved from KNN as 99.8%. Even when the protein subset length is lowered to 11, 10 and 9 features respectively, KNN showed a higher accuracy compared to the highest accuracy reported on these papers (99.1%). The same procedure is followed in order to identify the critical proteins related to different types of learning such as successful learning, rescued learning and failed learning. We compared our classification results with the related work. Our method shows a higher accuracy than the ones reported by Higuera et al. and Kulan et al. [26] The algorithm with the best performance was again KNN with an accuracy of 99.8% accuracy for successful learning, 99.5% for rescued learning and 100% for failed learning. Furthermore, we compared the protein subsets of each case and each problem in order to identify the critical proteins. We found the nine critical proteins which are CaNA_N, pPKCG_N, SOD1_N, pCAMKII_N, S6_N, H3MeK4_N, pP70S6_N, and APP_N for multiclass classification problem. For different types of learning, we found critical proteins for each type. In the end, we discovered the proteins that vary for each type. For successful learning, the critical proteins that are not found on the subsets of the other types of learning are pNUMB_N, Ubiquitin_N and ADARB1_N. For rescued learning the different proteins are BRAF_N, pERK_N and DYRK1A_N. And the last, for failed learning the different proteins are ARC_N, BAD_N, pRSK_N. The pre-processing method that is used to fill the missing values, may or may not affect the accuracy of the classification, this depending on the

algorithm. The subset of proteins selected provides have a better understanding of the protein involvement on the learning process that may lead to a better drug development for treating the ID. Finally, we may conclude that our goal to identify the critical proteins is achieved and we hope that this study will help scientists to achieve their goal of finding a treatment that may help the learning process and ease the intellectual disability caused by Down syndrome.

References

1. Wester Oxelgren, U. *et al.* More severe intellectual disability found in teenagers compared to younger children with down syndrome. *Acta Paediatr.* **108**, 961–966 (2019).
2. Antonarakis, S. E. Down syndrome and the complexity of genome dosage imbalance. *Nat. Rev. Genet.* **18**, 147 (2017).
3. Karmiloff-Smith, A. *et al.* The importance of understanding individual differences in down syndrome. *F1000Research* **5** (2016).
4. Sheehan, R. *et al.* Dementia diagnostic criteria in down syndrome. *Int. journal geriatric psychiatry* **30**, 857–863 (2015).
5. Eady, N. *et al.* Author's reply to: Difficulties of diagnosing and managing dementia in people with down syndrome. *The Br. J. Psychiatry* **213**, 669–669 (2018).
6. Tovar, Á. E., Westermann, G. & Torres, A. From altered synaptic plasticity to atypical learning: A computational model of down syndrome. *Cognition* **171**, 15–24 (2018).
7. Ahlfors, H. *et al.* Gene expression dysregulation domains are not a specific feature of down syndrome. *Nat. communications* **10**, 2489 (2019).
8. Wiseman, F. K. *et al.* A genetic cause of alzheimer disease: mechanistic insights from down syndrome. *Nat. Rev. Neurosci.* **16**, 564–574 (2015).
9. Annus, T. *et al.* The pattern of amyloid accumulation in the brains of adults with down syndrome. *Alzheimer's & Dementia* **12**, 538–545 (2016).
10. Di Domenico, F. *et al.* Restoration of aberrant mtor signaling by intranasal rapamycin reduces oxidative damage: Focus on hne-modified proteins in a mouse model of down syndrome. *Redox biology* 101162 (2019).
11. Ahmed, M. M. *et al.* Protein profiles associated with context fear conditioning and their modulation by memantine. *Mol. & Cell. Proteomics* **13**, 919–937 (2014).
12. Ahmed, M. M. *et al.* Protein dynamics associated with failed and rescued learning in the ts65dn mouse model of down syndrome. *PloS one* **10**, e0119491 (2015).
13. Costa, A. C., Scott-McKean, J. J. & Stasko, M. R. Acute injections of the nmda receptor antagonist memantine rescue performance deficits of the ts65dn mouse model of down syndrome on a fear conditioning test. *Neuropsychopharmacology* **33**, 1624 (2008).
14. Eady, N. *et al.* Impact of cholinesterase inhibitors or memantine on survival in adults with down syndrome and dementia: clinical cohort study. *The Br. J. Psychiatry* **212**, 155–160 (2018).
15. Sinai, A. *et al.* Predictors of age of diagnosis and survival of alzheimer's disease in down syndrome. *J. Alzheimer's Dis.* **61**, 717–728 (2018).
16. Zhou, X. *et al.* Memantine improves cognitive function and alters hippocampal and cortical proteome in triple transgenic mouse model of alzheimer's disease. *Exp. neurobiology* **28**, 390–403 (2019).
17. Cipriani, G., Danti, S., Carlesi, C. & Di Fiorino, M. Aging with down syndrome: the dual diagnosis: Alzheimer's disease and down syndrome. *Am. J. Alzheimer's Dis. & Other Dementias* **33**, 253–262 (2018).
18. Smalheiser, N. R. Ketamine: a neglected therapy for alzheimer disease. *Front. aging neuroscience* **11**, 186 (2019).
19. Jiang, Y. *et al.* Lysosomal dysfunction in down syndrome is app-dependent and mediated by app-βctf (c99). *J. Neurosci.* **39**, 5255–5268 (2019).
20. Rafii, M. S. & Santoro, S. L. Prevalence and severity of alzheimer disease in individuals with down syndrome. *JAMA neurology* **76**, 142–143 (2019).
21. Higuera, C., Gardiner, K. J. & Cios, K. J. Self-organizing feature maps identify proteins critical to learning in a mouse model of down syndrome. *PloS one* **10**, e0129126 (2015).
22. Eicher, T. & Sinha, K. A support vector machine approach to identification of proteins relevant to learning in a mouse model of down syndrome. In *2017 International Joint Conference on Neural Networks (IJCNN)*, 3391–3398 (IEEE, 2017).

23. Feng, B., Hoskins, W., Zhou, J., Xu, X. & Tang, J. Using supervised machine learning algorithms to screen down syndrome and identify the critical protein factors. In *International Conference on Intelligent and Interactive Systems and Applications*, 302–308 (Springer, 2017).

24. Kulan, H. & Dag, T. Using machine learning classifiers to identify the critical proteins in down syndrome. In *Proceedings of the 2018 2nd International Conference on Computational Biology and Bioinformatics*, 51–54 (ACM, 2018).

25. Handan, K. & Tamer, D. In silico identification of critical proteins associated with learning process and immune system for down syndrome. (2019).

26. Di Santo, S. G., Prinelli, F., Adorni, F., Caltagirone, C. & Musicco, M. A meta-analysis of the efficacy of donepezil, rivastigmine, galantamine, and memantine in relation to severity of alzheimer's disease. *J. Alzheimer's Dis.* **35**, 349–361 (2013).

27. Abdeldayem, S. S. & Elhefnawi, M. Deep feature selection for identification of essential proteins of learning and memory in mouse model of down syndrome. *BioRxiv* 333849 (2018).

28. Li, Y., Chen, C.-Y. & Wasserman, W. W. Deep feature selection: theory and application to identify enhancers and promoters. *J. Comput. Biol.* **23**, 322–336 (2016).

29. Urbanowicz, R. J., Meeker, M., La Cava, W., Olson, R. S. & Moore, J. H. Relief-based feature selection: introduction and review. *J. biomedical informatics* **85**, 189–203 (2018).

30. Demšar, J. *et al.* Orange: data mining toolbox in python. *The J. Mach. Learn. Res.* **14**, 2349–2353 (2013).

31. Hu, L.-Y., Huang, M.-W., Ke, S.-W. & Tsai, C.-F. The distance function effect on k-nearest neighbor classification for medical datasets. *SpringerPlus* **5**, 1304 (2016).

32. Chen, C.-H., Huang, W.-T., Tan, T.-H., Chang, C.-C. & Chang, Y.-J. Using k-nearest neighbor classification to diagnose abnormal lung sounds. *Sensors* **15**, 13132–13158 (2015).

33. Dongardive, J. & Abraham, S. Protein sequence classification based on n-gram and k-nearest neighbor algorithm. In *Computational Intelligence in Data Mining—Volume 2*, 163–171 (Springer, 2016).

34. Zuo, Y.-C. *et al.* Discrimination of membrane transporter protein types using k-nearest neighbor method derived from the similarity distance of total diversity measure. *Mol. bioSystems* **11**, 950–957 (2015).

35. Rosa, A. C. *et al.* Analyzing cysteine site neighbors in proteins to reveal dimethyl fumarate targets. *Proteomics* **19**, 1800301 (2019).

36. Rahman, M. M. & Bhuiyan, M. I. H. A classification scheme for predicting the subcellular localization of the apoptosis proteins using composition features and multiscale entropy. In *2018 10th International Conference on Electrical and Computer Engineering (ICECE)*, 345–348 (IEEE, 2018).

37. Song, Y., Liang, J., Lu, J. & Zhao, X. An efficient instance selection algorithm for k nearest neighbor regression. *Neurocomputing* **251**, 26–34 (2017).

38. Tanveer, M., Shubham, K., Aldhaifallah, M. & Ho, S. S. An efficient regularized k-nearest neighbor based weighted twin support vector regression. *Knowledge-Based Syst.* **94**, 70–87 (2016).

39. Hansen, L. K. & Salamon, P. Neural network ensembles. *IEEE Transactions on Pattern Analysis & Mach. Intell.* 993–1001 (1990).

40. Ding, C. H. & Dubchak, I. Multi-class protein fold recognition using support vector machines and neural networks. *Bioinformatics* **17**, 349–358 (2001).

41. Diplaris, S., Tsoumakas, G., Mitkas, P. A. & Vlahavas, I. Protein classification with multiple algorithms. In *Panhellenic Conference on Informatics*, 448–456 (Springer, 2005).

42. Fout, A., Byrd, J., Shariat, B. & Ben-Hur, A. Protein interface prediction using graph convolutional networks. In *Advances in Neural Information Processing Systems*, 6530–6539 (2017).

43. Pärnamaa, T. & Parts, L. Accurate classification of protein subcellular localization from high-throughput microscopy images using deep learning. *G3: Genes, Genomes, Genet.* **7**, 1385–1392 (2017).

44. Nair, V. & Hinton, G. E. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, 807–814 (2010).

45. Suykens, J. A. & Vandewalle, J. Least squares support vector machine classifiers. *Neural processing letters* **9**, 293–300 (1999).

46. Hsu, C.-W., Chang, C.-C., Lin, C.-J. *et al.* A practical guide to support vector classification. (2003).

47. Leslie, C., Eskin, E. & Noble, W. S. The spectrum kernel: A string kernel for svm protein classification. In *Biocomputing 2002*, 564–575 (World Scientific, 2001).

48. Eskin, E., Weston, J., Noble, W. S. & Leslie, C. S. Mismatch string kernels for svm protein classification. In *Advances in neural information processing systems*, 1441–1448 (2003).

49. Li, D., Ju, Y. & Zou, Q. Protein folds prediction with hierarchical structured svm. *Curr. Proteomics* **13**, 79–85 (2016).

50. Manavalan, B., Shin, T. H. & Lee, G. Pvp-svm: sequence-based prediction of phage virion proteins using a support vector machine. *Front. microbiology* **9**, 476 (2018).

51. Meyer, D. & Wien, F. T. Support vector machines. *The Interface to libsvm package e1071* **28** (2015).

52. Liaw, A., Wiener, M. *et al.* Classification and regression by randomforest. *R news* **2**, 18–22 (2002).

53. Díaz-Uriarte, R. & De Andres, S. A. Gene selection and classification of microarray data using random forest. *BMC bioinformatics* **7**, 3 (2006).

54. Qi, Y., Klein-Seetharaman, J. & Bar-Joseph, Z. Random forest similarity for protein-protein interaction prediction from multiple sources. In *Biocomputing 2005*, 531–542 (World Scientific, 2005).

55. Yan, K., Xu, Y., Fang, X., Zheng, C. & Liu, B. Protein fold recognition based on sparse representation based classification. *Artif. intelligence medicine* **79**, 1–8 (2017).

56. Ismail, H. D., Saigo, H. & KC, D. B. Rf-nr: Random forest based approach for improved classification of nuclear receptors. *IEEE/ACM Transactions on Comput. Biol. Bioinforma. (TCBB)* **15**, 1844–1852 (2018).

57. Arun, K. & Langmead, C. J. Structure based chemical shift prediction using random forests non-linear regression. In *Proceedings Of The 4th Asia-Pacific Bioinformatics Conference*, 317–326 (World Scientific, 2006).

58. Yu, D.-J. *et al.* Disulfide connectivity prediction based on modelled protein 3d structural information and random forest regression. *IEEE/ACM transactions on computational biology bioinformatics* **12**, 611–621 (2014).

59. Wang, C. & Zhang, Y. Improving scoring-docking-screening powers of protein–ligand scoring functions using random forest. *J. computational chemistry* **38**, 169–177 (2017).

Acknowledgements

This project has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement No:760921 (PANBioRA).

Author contributions statement

Xh.P. did all the experiments, A.O.T. and A.U. guided and analysed the results. All authors reviewed the manuscript.

Additional information

Competing interests (mandatory statement).

The author(s) declare no competing interests.

Figures

Figure 1

Comparing the accuracy results between D1 and D2 for KNN with 9, 10, and 11 features

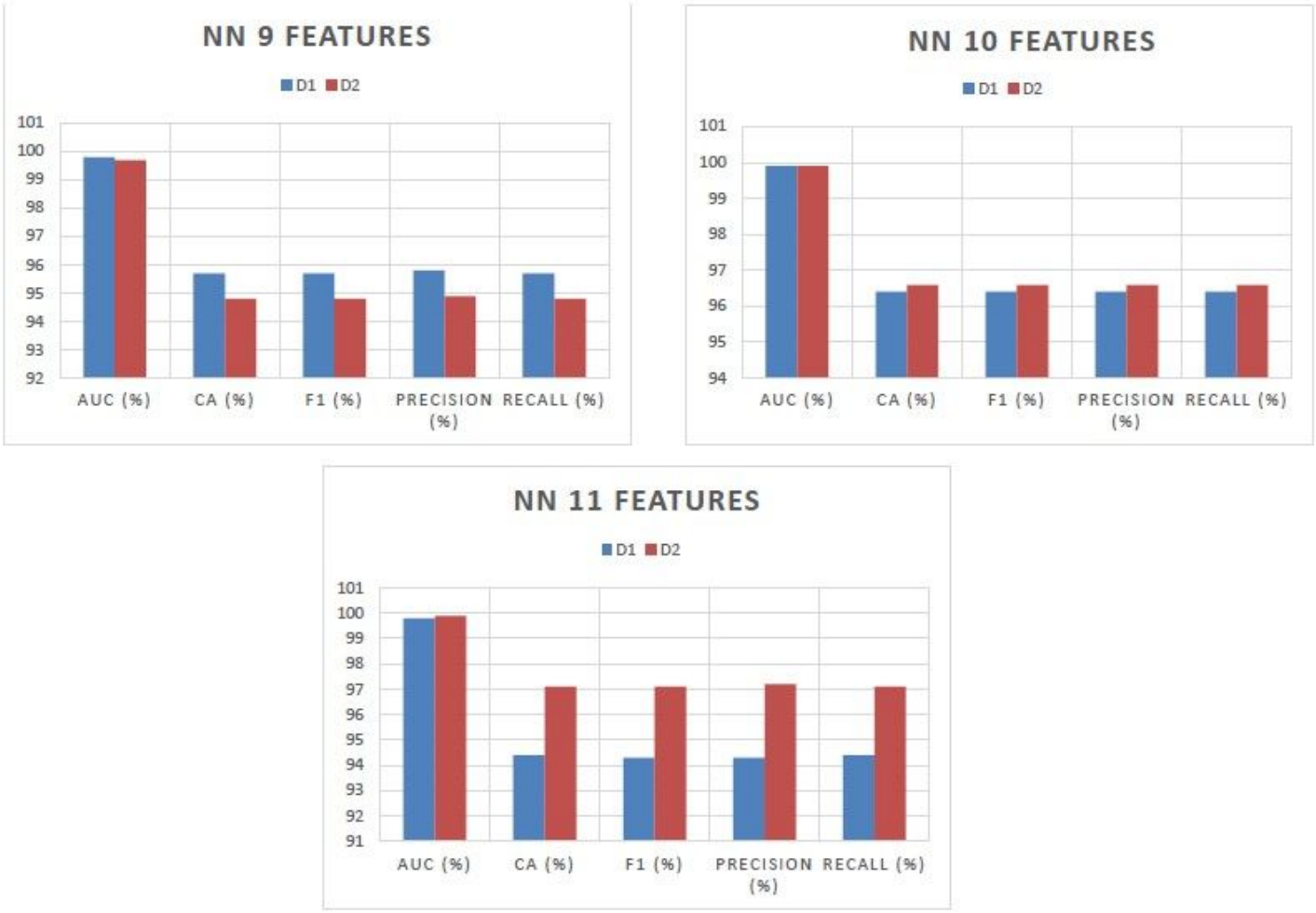


Figure 2

Comparing the accuracy results between D1 and D2 for NN with 9, 10, and 11 features

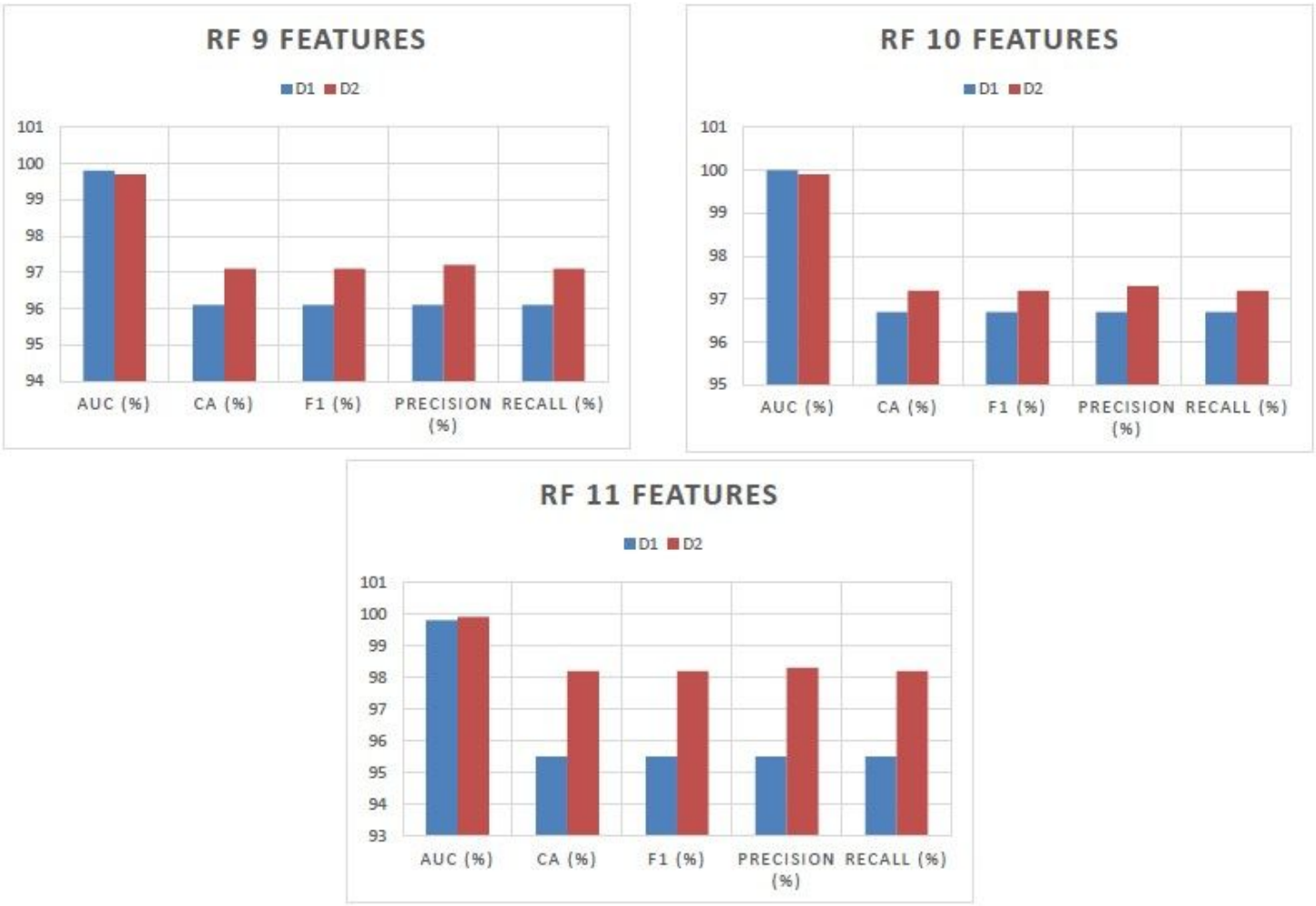
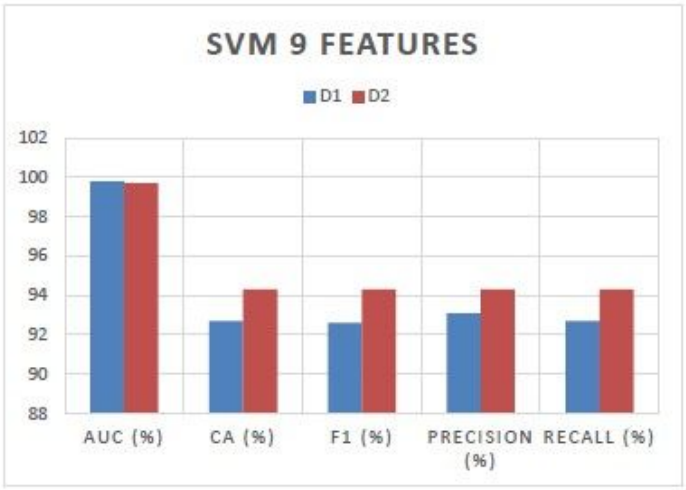
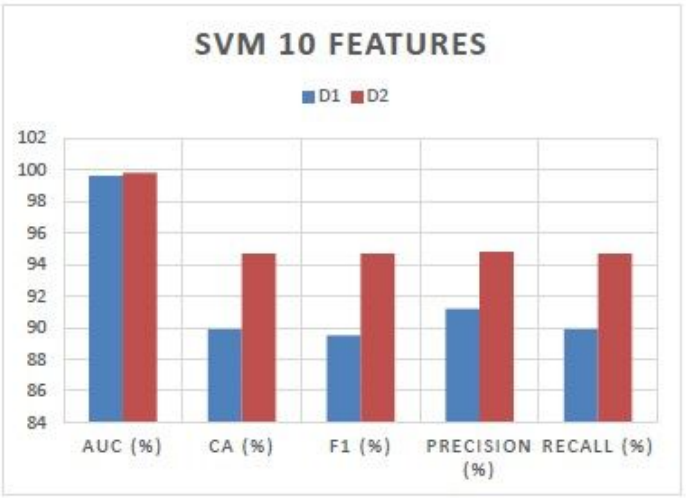


Figure 3

Compared accuracy results between D1 and D2 for RF with 9, 10 and 11 features





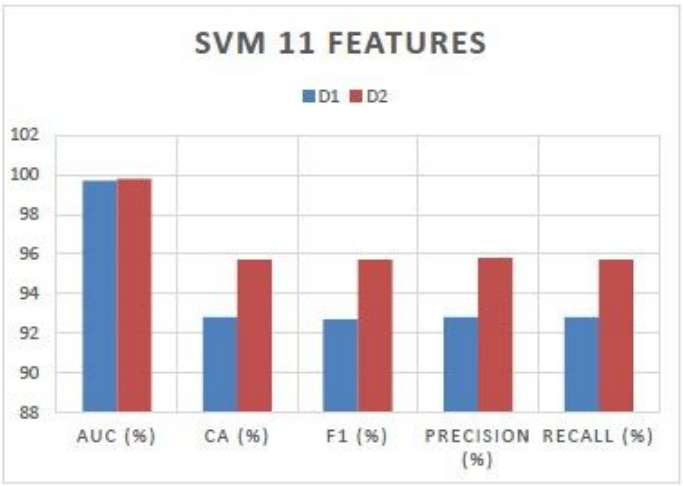


Figure 4

Compared accuracy results between D1 and D2 for SVM with 9, 10 and 11 features

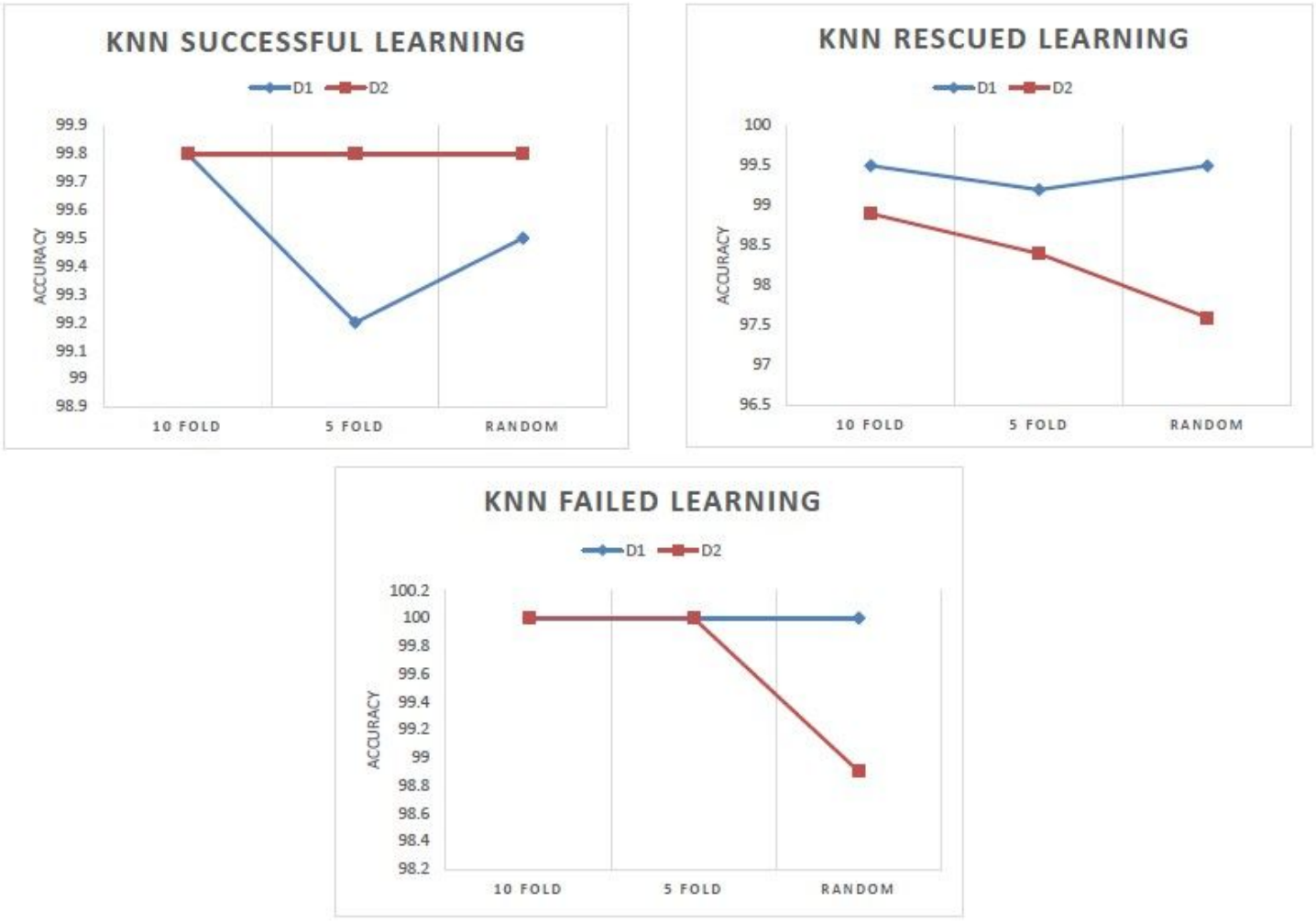


Figure 5

Compared results of KNN with different types of validation methods for D1 and D2

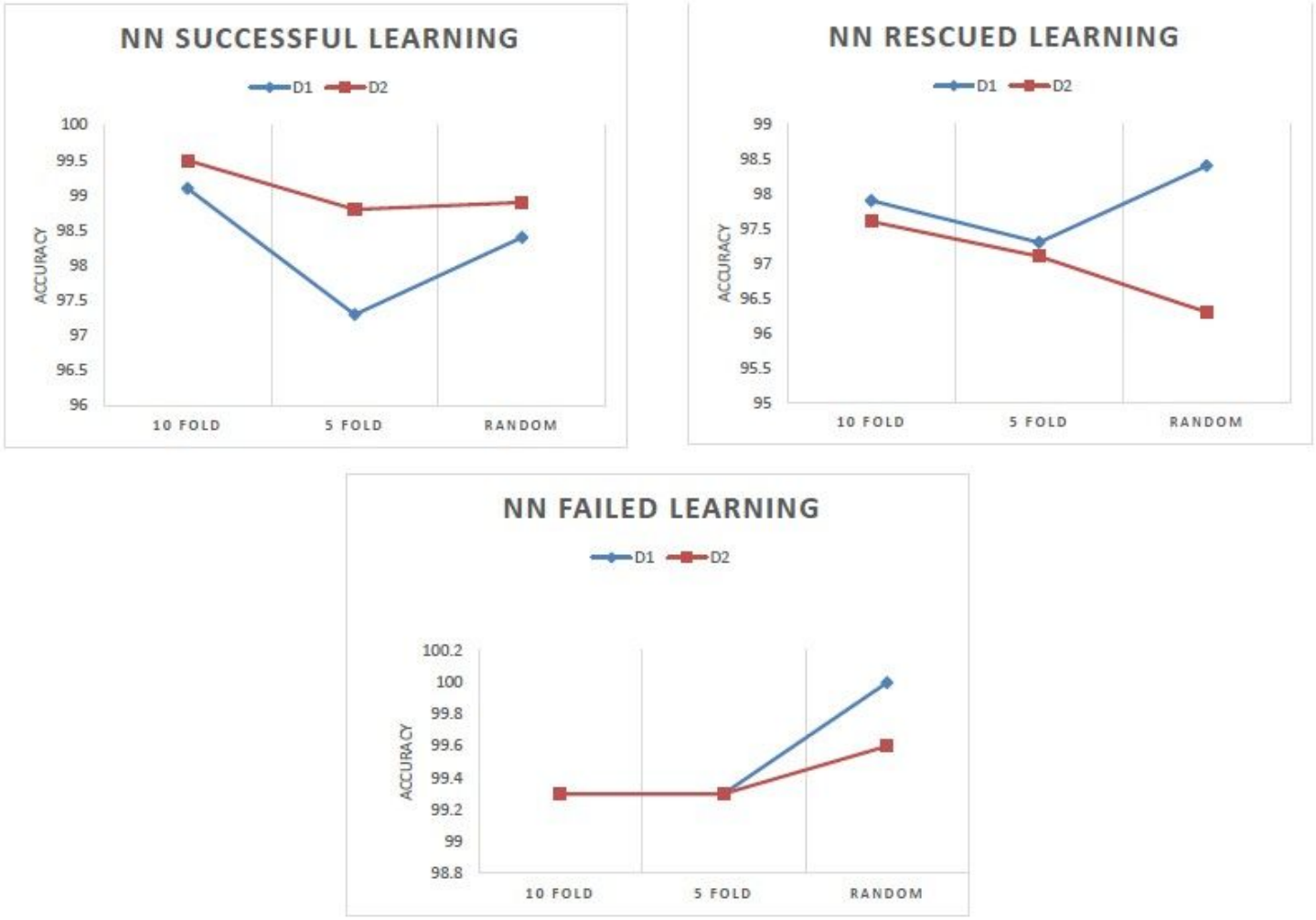


Figure 6

Compared results of NN with different types of validation methods for D1 and D2

Figure 7

Compared results of RF with different types of validation methods for D1 and D2

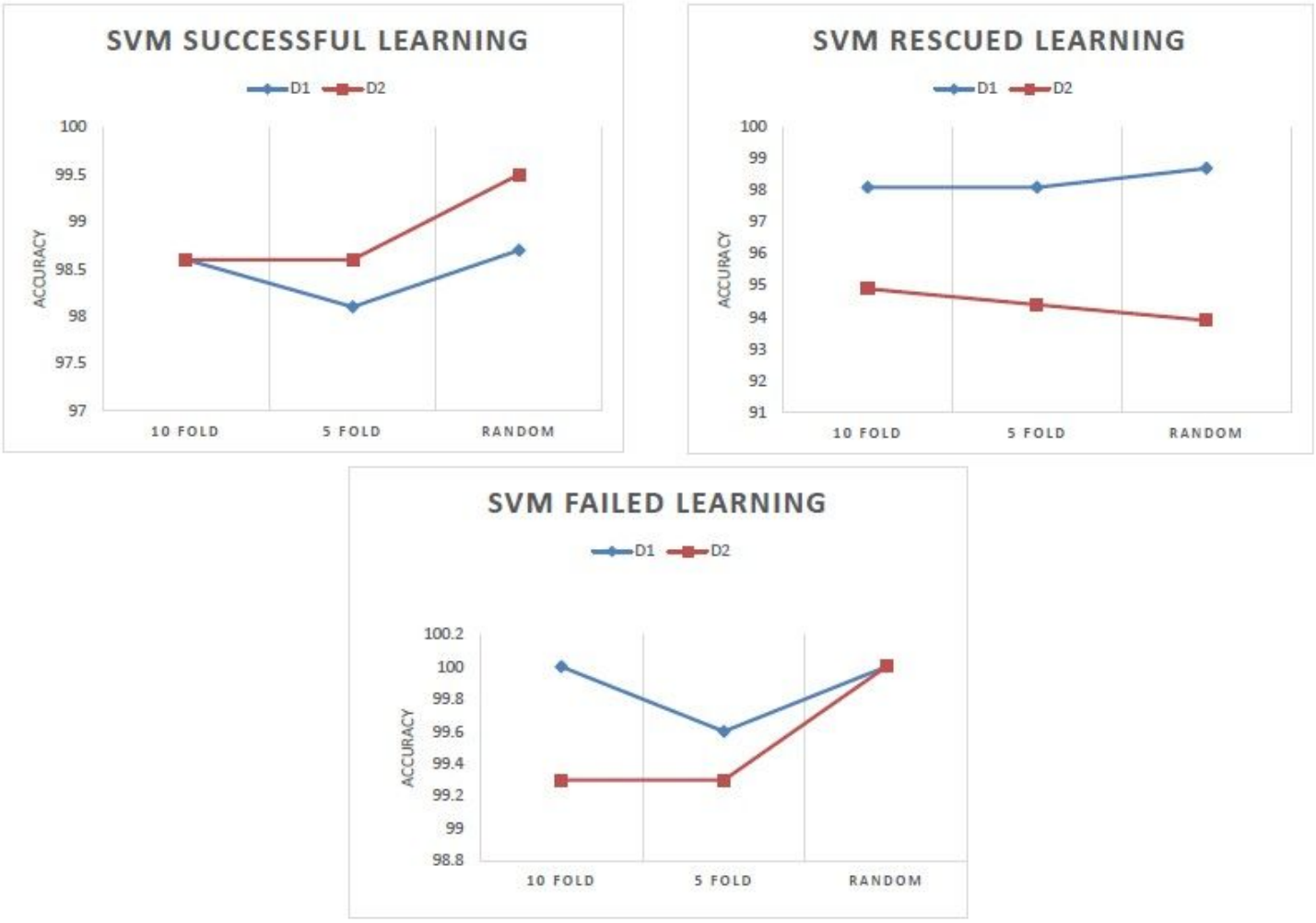


Figure 8

Compared results of SVM with different types of validation methods for D1 and D2