ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 978-1-7281-6327-7/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICASSP49357.2023.10097057

SEQUENTIAL DATUM–WISE JOINT FEATURE SELECTION AND CLASSIFICATION IN THE PRESENCE OF EXTERNAL CLASSIFIER

Sachini Piyoni Ekanayake[1]*, Daphney–Stavroula Zois*[1]*, Charalampos Chelmis*[2]

[1]Department of Electrical and Computer Engineering, University at Albany, State University of New York, Albany, NY

[2]Department of Computer Science, University at Albany, State University of New York, Albany, NY

Emails: {sekanayake, dzois, cchelmis}@albany.edu

ABSTRACT

We introduce a supervised machine learning framework for sequential datum–wise joint feature selection and classification. Our proposed approach sequentially acquires features one at a time during testing until it decides that acquiring more features will not improve label assignment. At that point, and in contrast to prior art, it assigns a label to the example under consideration by selecting between a simple internal and a more powerful external classifier. Easy–to–classify examples are handled by the internal classifier, which assigns labels based on the lowest expected misclassification cost. On the other hand, difficult–to–classify examples are forwarded to the external classifier to be assigned a label based on the acquired features. We demonstrate the performance of the proposed approach compared to existing methods using six publicly available datasets. Experiments indicate that the proposed approach improves accuracy up to 50% with respect to existing sequential methods, while acquiring up to 85% less number of features on average.

***Index Terms*—** supervised classification, instance–wise feature selection, dynamic programming, inaccurate oracle, high–performance classifier

1. INTRODUCTION

Traditional supervised classification adopts a batch–wise approach, where all features are readily available and used for determining the label of an example [1]. Unfortunately, in many real–world applications (e.g., medical diagnosis, insurance assessments), this is not the case, i.e., features are not freely available. For instance, in medical diagnosis, features come at a cost (i.e., due to the acquisition process), and each example includes different features [2]; nonetheless, accurate classification is critical and time–sensitive. Therefore, making accurate predictions using the most informative features is essential. In such a setting, feature selection and classification have received considerable attention [3–7].

This material is based upon work supported by the National Science Foundation under Grants ECCS–1737443 & CNS–1942330.

Standard classification methods (e.g., Support Vector Machines (SVM), Naive Bayes (NB)) consider all features to be available during training and testing. In contrast, offline feature selection methods (e.g., L1–norm based feature selection (Lasso)) select a sub–set of features during training and use them during testing. Due to their high performance, such methods are widely used in practice [8, 9], however, they base their decisions on the same set of features irrespective of the example under consideration. To accommodate prohibitively large feature spaces, *streaming* feature selection methods [10, 11] perform feature selection during training as features sequentially arrive one at a time. During testing, all examples are classified using a standard classification method using the *same* subset of features. *Static instance–wise* approaches [12, 13] perform datum–wise classification, but access all features during testing for that purpose. In contrast, *dynamic instance–wise* feature selection methods [5, 6] adaptively select different features to classify each data instance during testing. A major drawback of the latter approaches is inherently their classification mechanism; labels are assigned based on the smallest expected misclassification cost defined in terms of the posterior probability of the label of the example under consideration given the information provided by the already acquired features. Such mechanism may work well for some examples, but not for all, leading to considerable performance degradation.

To address the above issue, we present a method that decides between the use of a simple internal classifier (based on the expected misclassification cost) and a more powerful external one. The proposed method uses the posterior probability to assess how difficult it is to classify the example under consideration. As features are sequentially acquired during testing, the respective posterior probability is updated accordingly, and used to continue or terminate the feature acquisition process. Then, difficult–to–classify examples are forwarded to the external classifier, while the rest are handled by the internal classifier. The performance of the proposed approach is assessed on six real–world datasets and compared with five existing approaches. We observe that it achieves a good balance between accuracy and average number of acquired fea-

tures. Further, we notice that the inclusion of an external classifier improves accuracy by better handling difficult–to–classify examples, while saving on feature acquisition costs.

2. PROBLEM STATEMENT

In a typical supervised classification setting, the objective is to learn a model that maps an example described by a K–dimensional feature vector $[x_1, \ldots, x_K]^T$, where x_k represents the value of the kth feature, to a label $y \in \{1, \ldots, N\}$. The basic assumption here is that the feature vector is given in full. We instead consider the same problem under a slightly different context. Specifically, each feature x_k is not *freely available*, rather it is *sequentially* acquired during *testing* by expending cost $c_k > 0, k = 1, \ldots, K$. Inherently, acquiring less features can save on acquisition costs. On the other hand, we may not have adequate information to make a reliable label assignment. Thus, it is necessary to *jointly* determine the *subset of features* based on which each example is to be classified, and its *predicted label*.

We start by introducing two random variables to facilitate the description of our proposed framework. We define random variable $R \in \{0, \ldots, K\}$ to indicate the *last feature* acquired before assigning a label to the current example with 0 representing the case that no features were acquired. For example, $R = 2$ indicates the acquisition of features x_1 and x_2. We also define random variable $D_R \in \{1, \ldots, N\} \cup \{\mathcal{E}\}$ to indicate the *decision* reached about the current example based on R features. Specifically, when $D_R \in \{1, \ldots, N\}$, the current example is assigned one out of N labels by an *internal* classifier (c.f. Section 3.2). Otherwise, when $D_R = \mathcal{E}$, the current example is forwarded to an *external* classifier (e.g., SVM) to be classified using the R acquired features.

There are two main benefits of having access to an external classifier. First, classification of examples that would require the acquisition of large number of features could potentially be classified by the external classifier using less features. Second, irrespective of the number of features used by the internal classifier for a given example, classification may be inaccurate; such difficult–to–classify examples may be better handled by the external classifier.

In order to learn a model that simultaneously determines the subset of features based on which each example is to be classified, and its predicted label, we propose to optimize the following cost function:

$$J(R, D_R) = \mathbb{E}\Bigg\{ \sum_{k=1}^{R} c_k + \sum_{j=1}^{N} \sum_{i=1}^{N} M_{ij} P(D_R = j, y = i) + \sum_{i=1}^{N} M_{i\mathcal{E}} P(D_R = \mathcal{E}, y = i) \Bigg\}, \quad (1)$$

where $M_{ij}, i, j \in \{1, \ldots, N\}$, represents the cost of assigning label j to the example when the true label is i, while $M_{i\mathcal{E}}$ indicates the cost of forwarding the example to an external classifier when the true label is i. The first term in Eq. (1) represents the average cost of acquiring R features, while the second term captures the average cost of assigning labels using the internal classifier. The third term indicates the average cost of forwarding the example to an external classifier. Thus, our objective reduces to determining the pair (R, D_R) that minimizes the average cost in Eq. (1).

3. SEQUENTIAL APPROACH

In this section, we describe how to obtain the pair (R, D_R) that minimizes Eq. (1). We first introduce a sufficient statistic, which we then use to rewrite Eq. (1). This enables us to acquire the optimum feature acquisition and decision strategies. Finally, we describe our proposed algorithm.

3.1. Sufficient Statistic

Consider the *posterior* probability vector $\pi_k \triangleq [\pi_k^1, \ldots, \pi_k^N]$ where $\pi_k^i \triangleq P(y = i | x_1, \ldots, x_k), k = 1, \ldots, K, i = 1, \ldots, N$, denotes the probability of the current example having label i given that k features have been acquired. When $k = 0$ (i.e., no features have been acquired), $\pi_0 \triangleq [\pi_1, \ldots, \pi_N]^T$, where $P(y = i) = p_i, i = 1, \ldots, N$, represents the *prior* probability of the example under consideration having label i. From Bayes' rule, we recursively update the posterior probability as $\pi_k^i = \frac{P(x_k|y=i)\pi_{k-1}^i}{P(x_k|y=1)\pi_{k-1}^1 + \ldots + P(x_k|y=N)\pi_{k-1}^N}$, which can be compactly written in vector form as follows:

$$\pi_k = \frac{\text{diag}(\Delta_k(x_k))\pi_{k-1}}{\Delta_k^T(x_k)\pi_{k-1}}. \quad (2)$$

Note that $\Delta_k(x_k) \triangleq [P(x_k|y = 1), \ldots, P(x_k|y = N)]^T$, $P(x_k|y = i)$ represents the pmf of feature x_k given that the current example has label i, and $\text{diag}(\mathbf{z})$ represents a diagonal matrix that has the elements of vector $\mathbf{z}$. Here we assume that the features x_k are independent given the label. Even though simplistic, such an assumption results in a good trade–off between accuracy and average number of acquired features, as seen in Section 4.

3.2. Optimum Decision Strategy

Next, we discuss the *optimum decision strategy* D_R^* for any fixed number R of acquired features. First, Eq. (1) is written in terms of the posterior probability and the indicator function $\mathbb{I}_A$ (i.e., $\mathbb{I}_A \triangleq 1$ when event A occurs and 0 otherwise):

$$J(R, D_R) = \mathbb{E}\Bigg\{ \sum_{k=1}^{R} c_k + \sum_{j=1}^{N} \mathbf{M}_j^T \pi_R \mathbb{I}_{\{D_R = j\}} + \mathbf{M}_{\mathcal{E}}^T \pi_R \mathbb{I}_{\{D_R = \mathcal{E}\}} \Bigg\}, \quad (3)$$

where $\mathbf{M}_j \triangleq [M_{1j}, \ldots, M_{Nj}]^T$ and $\mathbf{M}_{\mathcal{E}} \triangleq [M_{1\mathcal{E}}, \ldots, M_{N\mathcal{E}}]^T$. To find the optimum decision strategy D_R^*, first, we consider the last two terms of Eq. (3), which depend on D_R. For any decision D_R, $\sum_{j=1}^{N} \mathbf{M}_j^T \pi_R \mathbb{I}_{\{D_R = j\}} + \mathbf{M}_{\mathcal{E}}^T \pi_R \mathbb{I}_{\{D_R = \mathcal{E}\}} \geq$

Table 1. Accuracy ("Acc") and average number of acquired features ("Feat") for the proposed approach (SDFA–SVM, SDFA–DT) and baselines.

Method	Monks Problem		Diabetes		EEG Eye State		MagicTelescope		Student		German Credit	
	Acc	Feat	Acc	Feat	Acc	Feat	Acc	Feat	Acc	Feat	Acc	Feat
SDFA–SVM	0.536	5.722	0.753	6.056	0.536	3.315	0.794	6.316	0.864	4.656	0.732	12.081
SDFA–DT	0.795	5.722	0.753	6.056	0.485	3.315	0.807	6.316	0.869	4.656	0.732	12.081
ETANA	0.529	5.188	0.749	5.935	0.500	12.261	0.775	6.302	0.864	4.617	0.714	11.846
NB	0.591	6.000	0.751	8.000	0.437	14.000	0.727	11.000	0.827	32.000	0.700	20.000
SVM	0.657	6.000	0.674	8.000	0.551	14.000	0.806	11.000	0.787	32.000	0.700	20.000
DT	0.922	6.000	0.706	8.000	0.475	14.000	0.819	11.000	0.838	32.000	0.664	20.000
Lasso	0.654	4.800	0.766	8.000	0.551	13.400	0.789	9.000	0.851	14.600	0.734	17.800

$g(\pi_R)$ where $g(\pi_R) \triangleq \min_{q\in\{1,\ldots,N\}\cup\{\mathcal{E}\}}[\mathbf{M}_q^T\pi_R]$. Thus, D_R^* for any number R of acquired features is:

$$D_R^* = \underset{q\in\{1,\ldots,N\}\cup\{\mathcal{E}\}}{\operatorname{argmin}} [\mathbf{M}_q^T\pi_R]. \quad (4)$$

Specifically, if $D_R^* \in \{1,\ldots,N\}$, the internal classifier assigns the example under consideration the label i that yields the smallest misclassification cost captured by the inner product $\mathbf{M}_i^T\pi_R$. Otherwise, if $D_R^* = \mathcal{E}$, the example is forwarded to the external classifier (e.g., SVM), which uses the R acquired features to assign a label.

3.3. Optimum Feature Acquisition Strategy

To find the *optimum feature acquisition strategy* R^*, first, the cost function in Eq. (3) is simplified using the result of Section 3.2 as $\tilde{J}(R) = \mathbb{E}\{\sum_{k=1}^{R} c_k + g(\pi_R)\}$. R^* is then obtained using dynamic programming [14] to minimize the cost function. Specifically, since the available number of features is K, there exists a maximum of $K+1$ stages for the associated dynamic programming equations, i.e.,

$$\bar{J}_k(\pi_k) = \min\Big[g(\pi_k), \bar{A}_k(\pi_k)\Big], k = 0,\ldots,K-1, \quad (5)$$

where $\bar{A}_k(\pi_k) = c_{k+1} + \sum_{x_{k+1}} \bar{J}_{k+1}(\pi_{k+1})\Big(\Delta_{k+1}^T(x_{k+1})\pi_k\Big)$, with $\bar{J}_K(\pi_K) = g(\pi_K)$. In Eq. (5), $g(\pi_k)$ represents the cost of stopping the feature acquisition process when k features have already been acquired, while $\bar{A}_k(\pi_k)$ is the cost of continuing this process. Thus, $R^* = k$ if $g(\pi_k) < \bar{A}_k(\pi_k)$ for $k < K$ or $R^* = K$ when all the features are acquired.

3.4. SDFA Algorithm

Based on the above strategies, we propose the Sequential Datum–wise Feature Acquisition (SDFA) algorithm. SDFA sequentially acquires features until it decides to *either* assign the current example the label with the smallest misclassification cost *or* forward it to the external classifier along with the acquired features. In the training phase of SDFA, the interval $[0,1]$ is appropriately quantized to generate all possible posterior probability vectors π_k such that $\pi_k\mathbf{1}^T = 1$, where $\mathbf{1}$ is a N–dimensional vector of all ones. For all such posterior probability vectors, Eqs. (4) and (5) are evaluated and numerically solved to determine the optimum feature acquisition and decision strategies. Furthermore, the conditional probabilities $P(x_k|y=i)$ and prior probabilities $P(y=i), i=1,\ldots,N$, are estimated (c.f. Section 4). In the testing phase of SDFA, the numerical solutions determined during training are used to sequentially acquire features and reach a label assignment. The process begins by initializing the posterior probability $\pi_0 \triangleq [p_1,\ldots,p_N]$, where $p_i = P(y=i), i=1,\ldots,N$. Next, features are sequentially acquired based on Eq. (5), and a final decision is reached based on Eq. (4). Assuming k features have already been acquired, if the stopping cost is greater than the continuing cost, the next feature x_{k+1} is also aquired and the posterior probability vector is updated based on Eq. (2). This continues until a subset of features is acquired, or no more features are available. In either case, at that stage, the current example is assigned a label by the internal or external classifier.

4. NUMERICAL STUDY

In this section, we present experimental results to illustrate the performance of SDFA. Specifically, we test SDFA on 6 real–world datasets, i.e., Monks Problem [15] (601 instances / 6 features / 2 classes), Diabetes [16] (768 instances / 8 features / 2 classes), EEG Eye State [17] (14,980 instances / 14 features / 2 classes), MagicTelescope [18] (19,020 instances / 11 features / 2 classes), Student performance (Student) [19] (649 instances / 32 features / 2 classes), and German Credit [20] (1,000 instances / 20 features / 2 classes). For the German Credit, we use the originally provided dataset from Kaggle, while the rest of the datasets are from OpenML. Further, we preprocess the Student dataset such that the classification variable G_3 representing the final grade is binary, i.e., we set $G_3 = 1$ if the final grade ≥ 11, else $G_3 = 0$. We compare SDFA's performance (in terms of accuracy and average number of acquired features) with the following baselines: (i) ETANA [5], an instance–wise joint feature selection and classification algorithm, (ii) L1–norm based feature selection (Lasso), an *offline* feature selection algorithm, and (iii) NB, SVM with Gaussian kernel, and Decision Tree (DT), three standard classification algorithms. As the external classifier for SDFA, we consider SVM and DT, referred to as SDFA–

SVM and SDFA–DT, respectively.

In our experiments, maximum likelihood estimator is used to estimate $P(x_k|y=i)=\frac{S_{k,i}+1}{S_i+B}, k=1,\ldots,K, i=1,\ldots,N$, during training. Here $S_{k,i}$ denotes the number of examples that have label i and x_k takes a specific value, while S_i denotes the total number of examples that have label i. B is the number of bins considered. We set the prior probability as $P(y=i)=\frac{1}{N}$, simulating the scenario that all labels are equiprobable. Features are ordered as per the increasing order of the sum of type I and II errors scaled by the cost coefficient of the kth feature to promote low–cost features. During training, the ordered feature set is used, and the external classifier (i.e., SVM, DT) is trained for $k=1,\ldots,K$, features. We consider $B=10$, $M_{ij}=1, \forall i\neq j$ and $M_{ii}=0, \forall i,j\in\{1,\ldots,N\}$, and we set $M_{i\mathcal{E}}=0.4$. We assume feature cost to be the same for all features, i.e., $c_k=c=0.0001, \forall k$. All results reported in Table 1 are five–fold cross validated.

From Table 1, we observe that in the majority of cases, using DT as the external classifier of SDFA results in same or better accuracy (1% to 48%) compared to employing SVM. We also notice that irrespective of which classifier is employed, the average number of acquired features is the same. This is expected because difficult–to–classify examples are the ones to be forwarded to the external classifier and these are selected based solely on the posterior probability vector. An interesting future direction is extending the proposed framework to enable the comparison of multiple high–performance classifiers in the hope that more informed decisions can be made. Comparing SDFA–DT with ETANA, which also sequentially acquires features, we observe that the former always achieves better accuracy (1% to 50%). However, this results in a slight increase on the number of acquired features (1% to 10%) with the exception of the EEG Eye State. In that case, SDFA can achieve an 11% improvement in accuracy using SVM as the external classifier using 76% less features. These observations confirm that using an external classifier for difficult–to–classify examples can improve accuracy. At the same time, introducing an external classifier makes the framework linger a little longer on the feature acquisition process before making a decision in favor of the internal versus the external classifier. Comparing SDFA with the three standard classifiers, we notice that in half of the datasets, SDFA achieves better accuracy (up to 23%) using 5% to 85% less features. In the cases where any of the standard classifiers outperforms SDFA in terms of accuracy, SDFA uses considerably less features (between 5% and 76%). Finally, comparing SDFA with Lasso, an offline feature selection algorithm, we observe that in half of the datasets, SDFA achieves better accuracy (between 1% and 22%) acquiring less features on average (between 24% and 75%) with the exception of the Monks Problem (19% more features). In the case where Lasso performs better in terms of accuracy, SDFA uses on average less features (between 24% and 75%) resulting in a small accuracy drop. Overall, we emphasize that the proposed algorithm achieves a good balance between accuracy and average number of acquired features, forwarding difficult–to–classify examples to the external classifier. This is critical in real–world applications where acquisition of features is either prohibitive (e.g., due to cost or unwillingness of users to provide sensitive information) or the feature space is large. Fig. 1 shows the distribution of the number of acquired features during testing for the MagicTelescope using SDFA–DT for $c=0.01$ and $M_{i\mathcal{E}}=0.35$. We observe that most examples are classified using the internal classifier with very few features (less than half of the available features). We also notice that when more than half of the features have already been acquired, SDFA tends to forward the respective examples to the external classifier suggesting that these are difficult–to–classify examples. This insight can tremendously help speed–up the training phase of SDFA by training the external classifier only on this subset of features. Finally, since SDFA acquires different number of features per example in contrast to the rest of the baselines, its decisions are interpretable [21].

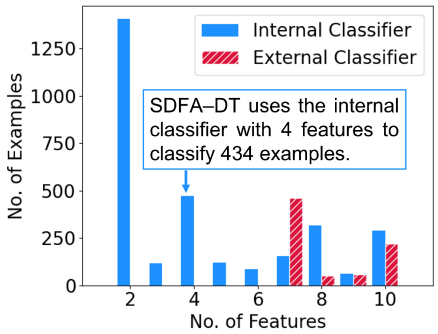

Fig. 1. Distribution of the number of acquired features during testing for the MagicTelescope dataset using SDFA–DT.

5. CONCLUSION

In this paper, we introduced and studied the problem of sequential datum–wise joint feature selection and classification in the presence of an external classifier. Our proposed approach dynamically acquires features in a sequential manner. It uses the acquired features to infer the label of the example under consideration by selecting between an internal and an external classifier. The effectiveness of the proposed approach is validated on a number of real–world datasets and compared with existing algorithms. We observe that the proposed approach achieves a good balance between accuracy and average number of acquired features. At the same time, it uses the internal classifier with easy–to–classify examples, while difficult ones are forwarded to the external classifier. In future, we plan to extend the proposed framework to accommodate multiple external classifiers and evaluate its performance on complex datasets with a large number of features.

6. REFERENCES

[1] Kevin P Murphy, *Probabilistic machine learning: an introduction*, MIT press, 2022.

[2] David P Kao, James D Lewsey, Inder S Anand, Barry M Massie, Michael R Zile, Peter E Carson, Robert S McKelvie, Michel Komajda, John JV McMurray, and JoAnn Lindenfeld, "Characterization of subgroups of heart failure patients with preserved ejection fraction with possible implications for prognosis and treatment response," *European Journal of Heart Failure*, vol. 17, no. 9, pp. 925–935, 2015.

[3] Xuegang Hu, Peng Zhou, Peipei Li, Jing Wang, and Xindong Wu, "A survey on online feature selection with streaming features," *Frontiers of Computer Science*, vol. 12, no. 3, pp. 479–493, 2018.

[4] Noura AlNuaimi, Mohammad Mehedy Masud, Mohamed Adel Serhani, and Nazar Zaki, "Streaming feature selection algorithms for big data: A survey," *Applied Computing and Informatics*, 2020.

[5] Yasitha Warahena Liyanage, Daphney-Stavroula Zois, and Charalampos Chelmis, "Dynamic instance-wise joint feature selection and classification," *IEEE Transactions on Artificial Intelligence*, vol. 2, no. 2, pp. 169–184, 2021.

[6] Yasitha Warahena Liyanage, Daphney-Stavroula Zois, and Charalampos Chelmis, "Dynamic instance-wise classification in correlated feature spaces," *IEEE Transactions on Artificial Intelligence*, vol. 2, no. 6, pp. 537–548, 2021.

[7] Yasitha Warahena Liyanage, Daphney-Stavroula Zois, and Charalampos Chelmis, "On–the–fly feature selection and classification with application to civic engagement platforms," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 3762–3766.

[8] Harsh Singh and Ognjen Arandjelović, "Data efficient support vector machine training using the minimum description length principle," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 1361–1365.

[9] Owen Queen and Scott J Emrich, "Lasso-based feature selection for improved microbial and microbiome classification," in *2021 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE, 2021, pp. 2301–2308.

[10] Noura AlNuaimi, Mohammad Mehedy Masud, Mohamed Adel Serhani, and Nazar Zaki, "Streaming feature selection algorithms for big data: A survey," *Applied Computing and Informatics*, 2019.

[11] Jing Zhou, Dean Foster, Robert Stine, and Lyle Ungar, "Streaming feature selection using alpha-investing," in *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*. ACM, 2005, pp. 384–393.

[12] Jinsung Yoon, William R Zame, and Mihaela van der Schaar, "Tops: Ensemble learning with trees of predictors," *IEEE Transactions on Signal Processing*, vol. 66, no. 8, pp. 2141–2152, 2018.

[13] Qi Xiao and Zhengdao Wang, "Mixture of deep neural networks for instancewise feature selection," in *2019 57th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. IEEE, 2019, pp. 917–921.

[14] D. P. Bertsekas, *Dynamic Programming and Optimal Control*, vol. 1, Athena Scientific, 2005.

[15] Sebastian B Thrun, Jerzy W Bala, Eric Bloedorn, Ivan Bratko, Bojan Cestnik, John Cheng, Kenneth A De Jong, Saso Dzeroski, Douglas H Fisher, Scott E Fahlman, et al., "The monk's problems: A performance comparison of different learning algorithms," Tech. Rep., 1991, [Online]. Available: "https://archive.ics.uci.edu/ml/datasets/MONK's+Problems".

[16] "diabetes," [Online]. Available: https://www.openml.org/searchtype=data&status=active&id=42608.

[17] Oliver Roesler, "EEG eye state data set," 2013, [Online]. Available: "https://archive.ics.uci.edu/ml/datasets/EEG+Eye+State".

[18] RK Bock, A Chilingarian, M Gaug, F Hakl, Th Hengstebeck, M Jiřina, J Klaschka, E Kotrč, P Savickỳ, S Towers, et al., "Methods for multidimensional event classification: a case study using images from a cherenkov gamma-ray telescope," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 516, no. 2-3, pp. 511–528, 2004.

[19] Paulo Cortez and Alice Maria Gonçalves Silva, "Using data mining to predict secondary school student performance," 2008, [Online]. Available: "https://archive.ics.uci.edu/ml/datasets/Student+Performance".

[20] Dr. Hans Hofmann, "German Credit Data," 1994, [Online]. Available: https://archive.ics.uci.edu/ml/datasets/statlog+(german+credit+data).

[21] Christoph Molnar, *Interpretable machine learning*, Lulu. com, 2020.