Comparison of Machine Learning approaches for Stress Detection from Wearable Sensors Data

Michela Quadrini[1,*,†], Denise Falcone[1] and Gianluca Gerard[2]

[1]School of Science and Technology, University of Camerino, Via Madonna delle Carceri, 9, Camerino, 62032, Italy
[2]Sorint.Tek, 17 Zanica Grassobbio, BG, 24050 Italy

Abstract

Stress is a prevalent and growing phenomenon in the modern world potentially leading to significant repercussions on both physical and mental health. The analysis of physiological signals, collected from wearable sensors, has emerged as a promising approach to predicting and managing stress. Methods based on machine learning techniques have been defined in the literature and achieved promising results by using handcrafted features extracted from the signal. However, there is no consensus on the list of features, while deep learning approaches that overcomes the problem require significant computational power and a large amount of data. In this paper, we present a comprehensive view of the most common representative machine learning algorithms applied to the stress detection domain by giving a reference point for both academia and industry professionals in this application field. This study considers fragments of signals without extracting any features and uses a public dataset, WESAD, that contains high-resolution physiological, including blood volume pulse, electrocardiogram and electromyogram. The data collected from 15 subjects during a lab study are heterogeneous and characterized by different frequencies and noises due to some devices. After preprocessing, we assess the performance of ten machine learning algorithms belonging to four models (tree, ensemble, linear and neighbours) on the WESAD by facing the problem as binary (stress/no-stress) and multiclass (baseline, stress, and amusement) classifications. Our results, evaluated in terms of classical metrics, show that Random Forest outperforms the others in binary and multi-class approaches.

Keywords

Physiological Signals, Binary and multi-class classification, Wearable Sensor Data, time series

1. Introduction

Stress is a non-specific body reaction to any demand upon it. Its effects influence overall behaviour, well-being, and potential personal and professional successes [1]. Chronic stress may give rise to significant physical and mental health issues, such as cancer, cardiovascular disease, depression, and diabetes. It is an increasingly prevalent and pervasive phenomenon in the modern world: more than 50% of all work-related ill health cases in 2020/21 are due to stress [2]. Assessments based on psychologically designed questions, such as the Perceived Stress Scale (PSS) [3], are frequently used to detect stress. However, these methods may be time-consuming, psychologically invasive and lack reliability. Therefore, the definition of non-invasive approaches for rapid and accurate stress detection influences the quality and wellness of people's lives: managing stress before it causes health issues is fundamental. In the literature, it has been demonstrated that physiological signals, a response to the Autonomic Nervous System, allow us to detect and monitor stress. Hovsepian et al. [4] pioneered the stress detection by using physiological signals. Both faced the problem as a binary classification problem, whereas Gjoreski et al. [5] aimed at distinguishing different levels of stress (no stress versus low stress versus high stress). Such bioignals can be captured non-invasively by wearable devices, such as smartphones and smartwatches, commonly used among people. Such devices can monitor some physiological parameters, such as Blood Volume Pulse (BVP), Electrodermal Activity (EDA), temperature (TEMP), and heart rate (HR) etc. In the scenario of stress detection, machine learning and deep learning methodologies achieve promising results by analyzing these data. These approaches include support vector machines, random forest and k-nearest neighbours and use handcrafted features extracted from the pre-processed signal in order to reduce the data noises [6]. Moreover, no consensus on the list of features to extract from physiological data has been reached [7]. To solve the problem, advanced deep learning approaches have been applied since they have the ability to automatically comprehend patterns and, thus extract features. Nevertheless, these require significant computational power and a large amount of data. The appropriate machine learning algorithm choice for a particular problem task is not trivial: no single classifier works best across all possible scenarios, as stated by no free lunch theorem states [8]. To the best of our knowledge, no scientific work compares machine learn-

Ital-IA 2024: 4th National Conference on Artificial Intelligence, organized by CINI, May 29-30, 2024, Naples, Italy
*Corresponding author.
†These authors contributed equally.
michela.quadrini@unicam.it (M. Quadrini);
denise.facone@studenti.unicam.it (D. Falcone);
michela.quadrini@unicam.it (G. Gerard)
0000-0003-0539-0290 (M. Quadrini); 0000-0003-0539-0290 (G. Gerard)

ing methods for stress detection on the same datasets without feature extraction or dimensionality reduction.

In this paper, we present a comprehensive view of the most common representative machine learning algorithms applied to the stress detection domain by giving a reference point for both academia and industry professionals in this application field. In the analysis, we consider fragments of signals without extracting any features due to the nature of the problem: stress determines nonspecific human responses and the feature selection depends on the subject and do not can be generalized. Such signal fragments contain samples of all the physiological parameters measured. After appropriate resampling and noise reduction, these values are linearized and constitute the input of the considered ML model by following the neural network approach. This study uses the WESAD [9] dataset that is public and stores 12 physiological signals, such as blood volume pulse and electrocardiogram, collected from 15 subjects during a lab study. After preprocessing (consisting of resampling, outlier removal, and normalization), we determine a dataset of samples that are signal fragments obtained using the sliding window approach. Over these entries, we evaluate the most common and popular methods widely in various application areas. We consider eight machine learning algorithms, i.e, Decision Tree (DT), Random Forest (RF), Adaboost (AB), Extratree (ExT), Passive Aggressive Classifier (PA), Logistic Regression (LR), K-kneighbors (NKE) and Nearest Centrod (NC). We face the binary (stress/no-stress) and multi-class (baseline, stress, and amusement) problem classifications. The results, evaluated in terms of classical metrics, show that RF outperforms the others in binary and multi-class approach. We also compare the results obtained with the ones in the literature [9].

The paper is organized as follows. Section 2 describes the materials and the methods used in this study. The pipeline of the approach used in the study with the main results are described in Section 3. The paper ends with some conclusion and future work, Section 4.

2. MATERIALS AND METHODS

This work proposes a comparative evaluation of ML approaches to understand the best approach for real-time analytics. For this study, we consider the WESAD dataset.

2.1. Dataset

WESAD is a public dataset designed for stress and affective detection. It is a high-quality multimodal dataset storing physiological and movement data of 15 subjects (12 male and 3 female) during a controlled lab experiment [9]. All the participants were not heavy smokers and did not suffer from chronic mental or cardiovascular disorders. Furthermore, the females subjects were not pregnant. The dataset includes blood volume pulse (BVP), electrocardiogram (ECG), electrodermal activity (EDA), electromyogram (EMG), respiration (RESP), body temperature (TEMP), and three-axis acceleration (ACC). ECG, EDA, EMG, RESP, TEMP and ACC were recorded by a chest-worn device (RespiBan) and sampled at 700 Hz, whereas a wrist-worn device (Empatica E4) recorded BVP (sampled at 64 Hz), EDA (at 4 Hz), TEMP (at 4 Hz), and ACC (at 32 Hz). The dataset comprises 14 time series, each spanning approximately 2 hours, total experimental duration. The experiments were conducted to capture three distinct affective states: baseline, stress, and amusement with durations of 20 minutes, 392 seconds and 7 minutes, respectively. They also included two meditation periods. To capture the data during the experiment, a particular protocol, depicted in Figure 1, has been used. It consists of two different versions, where amusement and stressful conditions are interchanged between different subjects to avoid the effects of order.

Version	Block 1		Block 2		Block 3		Block 4		Block 5		Block 6	
A	Baseline	RELAX	Amusement	RELAX	Medi I	RELAX	Stress	RELAX	Rest	RELAX	Medi II	RELAX
B	Baseline	RELAX	Stress	RELAX	Rest	RELAX	Medi I	RELAX	Amusement	RELAX	Medi II	RELAX

Figure 1: The two protocol versions used to collect data

2.2. Preprocessing

The varied sampling frequencies in WESAD, as detailed in Section 2.1, necessitated a harmonization step. We resampled all data to match the 700Hz frequency of the RespiBAN. Therefore, the resampling is applied only to the time series detected by Empatica E4 using Fourier method as an *unsampling technique*.

After the resampling, we remove the outliers due to occasional anomalous peaks in some signals, which may be attributed to instrumental errors or measurement noise. We removed the anomalies from each time series by using a Hampel filter, discussed in [10]. Such a filter uses 1-minute sliding windows as input and calculates the mean (μ) and standard deviation (σ) of the values within the corresponding interval. Observations higher than the threshold of 3σ from the mean within the respective window are classified as outliers (following Pearson's rule) and are substituted with the nearest chronological value. This strategy ensures that outlier substitution doesn't introduce significant high-frequency variations.

After outliers removal, we normalize all signals in the interval $[-1, 1]$ to treat all inputs equally.Let $X = \{x_1, x_2, \ldots, x_n\}$ be the considered time series with n components, where each component corresponds to a biophysical signal. Each of them are rescaled to the in-

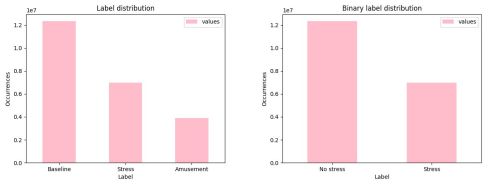

Figure 2: Label distributions of datasets created for multiclass and binary classification.

terval $[-1, 1]$ by applying the mean normalization:

$$\tilde{x}_j = \frac{(x_i - max(X)) + ((x_i - min(X))}{max(X) - min(X)}$$

where $max(X)$ and $min(X)$ is the maximum and minimum value among each component of X, respectively. Therefore, the input is a the scaled time series, $\tilde{X} = \{\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_n\}$.

2.3. Dataset Entry

After the data preprocessing phase, we create two datasets: one for binary classification and the other for multiclass. All entries are obtained by applying the sliding window technique to preprocessed signals. Specifically, the entries consist of time series fragments characterized by only an emotional state (or label) obtained by a slide of 60 seconds and a stride of 30 seconds, according to the study in [11]. To create the multiclass dataset, we consider parts of the time series associated with stress, Baseline and Amusement, as described in Section 2.1. For the binary classification, both the Baseline and Amusement states were aggregated under a single 'non-stress' label. The labels distribution of the two datasets are shown in Fig. 2.

2.4. Machine Learning Algorithms

In this section, we describe some machine learning classification techniques. Interested readers can refer to [12] for a complete treatment of machine learning approaches.

2.4.1. Decision Tree

A DT is a non-parametric supervised learning algorithm for classification and regression in the form of a tree structure [13]. It predicts the value of a target variable by learning simple decision rules inferred from the data features. The method exploits the "divide et impera" approach to learning: it learns from data with a set of if-then-else decision rules. The depth directly correlates with the complexity of these decision rules. The output is a tree comprising decision nodes and leaf nodes: a decision node has two or more branches, and a leaf node represents a classification or decision. The root of the tree corresponds to the best predictor. Usually, a DT is pruned by combining the adjacent nodes to avoid overfitting.

2.4.2. Ensemble models

Ensemble learning is a kind of model that makes predictions considering and combining a number of different models. By such a combination, an ensemble learning tends to be more flexible and less data sensitive.

Random Forest Random Forest is an ensemble model by Breiman [14] for both classification and regression. It constructs a set of decision trees during training and determines the prediction by selecting the most common class in the classification problem or calculating the mean/average prediction in the regression problem of the classes output by individual trees. This model combines the bagging approach with the random selection of features to ensure the uncorrelation among the decision trees of the forest. Feature randomness generates a random subset of features by ensuring low correlation among decision trees. In bagging, the decision trees depend on trees created from a different bootstrap sample, i.e., samples that may appear more than once in the entries of the training dataset. Differently from decision trees that consider all the possible feature splits, random forests only select a subset of those features.

AdaBoost AdaBoost, Adaptive Boosting, is an ensemble models developed by Yoav Freund et al. [15]. It employs an iterative approach to improve poor classifiers by learning from their errors. Unlike the random forest that uses parallel ensembling, Adaboost uses "sequential ensembling". Therefore, it is not possible to parallelize jobs on a multiprocessor machine like Random Forest. It creates a classifier by combining many poorly performing classifiers to obtain a good classifier of high accuracy. Such resulting classifier is accomplished with sequential weight adjustments, individual voting powers and a weighted sum of the final algorithm classifiers.

Extremely Randomized Trees Extremely Randomized Trees, introduced in [16], are ensembling methods that perform regression or classification. It creates a large number of unpruned decision trees from the training dataset and uses majority voting to select the decision trees for the classification. Different from Random Forest, it uses the entire dataset to train decision trees. Moreover, it randomly selects the values at which to split a feature and create child nodes to ensure sufficient differences between individual decision trees.

2.4.3. Linear Models

Logistic Regression Logistic Regression, introduced in [17], is a supervised learning algorithm mainly used for classification tasks where the aim is to estimate the probability of an instance belonging to a specific class based on the values of the input features. The method uses the sigmoid function to map any real-valued number into a value between 0 and 1. More specifically, it calculates a weighted sum of the input features, applies the logistic function to this sum, and then classifies the input as belonging to one of the two classes based on a chosen threshold.

Passive Aggressive The passive-aggressive algorithm, introduced in [18], is one of the few "online learning algorithms": the input data comes in sequential order, and the model is updated step-by-step. It is useful in applications that receive data as a continuous flow and need to adapt to change rapidly or autonomously or if you have limited computing resources. The algorithm is based on based on Passive and Aggressive approches. If the prediction is correct, keep the model and do not make any changes (passive), while If the prediction is incorrect, make changes to the model.

2.4.4. Neighbors-based Models

Supervised neighbors-based models can be applied for classification and regression. The principle behind nearest neighbor methods is to find a predefined number of training samples closest in distance to the new point, and predict the label from these.

K-Nearest Neighbors The k-nearest neighbours algorithm, introduced by Fix and Hodges in 1951 [19] and expanded by [20], is a non-parametric supervised learning method for classification and regression. K-nearest neighbours algorithm exploits proximity to make classifications or predictions about the grouping of an individual data point. KNN searches for the k-nearest labelled training data by using the distance metric and attributes the label which appears the most to the new observation. In our study, we use the Minkowski distance as a metric. The input consists of the k closest training examples in a data set, whereas the output depends on the task, classification or regression. Such output is a class membership or the property value for the entry, respectively.

Nearest Centroid Nearest Centroids, defined in [21], is arguably the simplest classifier. It operates on an intuitive principle: it takes data samples as input and classifies them into the class of training examples whose centroid (a geometric centre of a data distribution) is closest to it. The algorithm assumes that the centroids are distinct for each class (target label). The training data is divided into clusters based on their class labels, and then the centroid is computed for each data cluster. Each centroid is simply the mean value of each of the input variables. Such a centroid represents the "model": given new examples, the algorithm assigns the label by computing the distance between a given data and each centroid.

2.5. Metrics

We evaluate the performance and effectiveness of the approaches by using Accuracy (Acc), Precision (P), Recall (R), and F-measure (F_1), defined as follows

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

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

$$R = \frac{TP}{TP + FN}$$

$$F_1 = 2 \cdot \frac{P \cdot R}{P + R}$$

where TP represents the number of true positive, FN denotes the number of false negative, FP represents the number of false positive, TN denotes the number of true negative.

3. RESULTS

The work aims to compare various machine learning algorithms to detect stress from signals captured by wearable devices. The workflow is described in Section 3.1, while the results of the experiments are described in Section 3.2.

3.1. Methodology

Our pipeline, depicted in Fig. 3, is implemented in Python using the scikit-learn package for the machine learning approaches and *SciPy* for data manipulation and analysis. In particular, some methods of the SciPy library is used in the data preprocessing phase. The method resample permits the resampling of signals. In our approach, all signals are resampled at 700 Hz. About the outlier remotion, the Hampel filter is implemented using the ‘rolling‘, ‘mean‘, ‘std‘, ‘fillna‘, ‘mask‘, and ‘interpolate‘ methods from the Pandas library. The ‘MinMaxScaler‘ class of the scikit-learn package is used to perform data normalization. The machine learning methods Decision Tree, Random Forest , K-Nearest Neighbors and Logistic Regression are implemented via the tree, ensemble, neighbors and linear model modules, respectively. The method K-Folds is used to split the dataset into k consecutive folds without shuffling and then each fold is

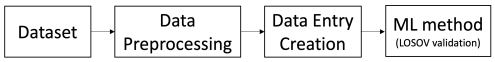

Figure 3: Pipeline used for the method comparison

then used once as a validation while the $k-1$ remaining folds form the training set. The code used in this manuscript are available from the corresponding author upon reasonable request.

3.2. Experiments

Given the small number of subjects involved in the experiment, we consider the Leave-One-Subject-Out Cross-Validation (LOSOCV), i.e., an approach that utilizes each subject as a "test" set and the remaining 14 as a "training" set. The experiments have been performed considering the decision tree, random forest, K-Nearest Neighbors and logistic regress as machine learning methods. For all experiments, we use the default parameters.

We evaluate such experiments by considering Accuracy, Precision, Recall and F1-Score as metrics. Tables 1 shows the average values with the standard deviation of the considered metrics obtained for binary and multiclass classification, respectively. Appendix A reports the values for each experiment.

	Binary Classification			
	Accuracy	**Precision**	**Recall**	**F1-Score**
DT	0.869 ± 0.150	0.924 ± 0.105	0.868 ± 0.209	0.882 ± 0.160
RF	0.920 ± 0.103	0.944 ± 0.100	0.945 ± 0.092	0.940 ± 0.076
AB	0.846 ± 0.154	$0.883 \pm 0,143$	0.907 ± 0.128	0.885 ± 0.112
ExT	0.909 ± 0.109	$0.943 \pm 0,089$	0.915 ± 0.119	0.925 ± 0.092
LR	0.822 ± 0.232	0.843 ± 0.208	0.925 ± 0.199	0.871 ± 0.186
PA	0.823 ± 0.225	0.842 ± 0.200	0.934 ± 0.187	0.874 ± 0.173
KNN	0.845 ± 0.193	0.939 ± 0.118	0.816 ± 0.251	0.851 ± 0.203
NC	0.929 ± 0.100	0.953 ± 0.097	0.949 ± 0.083	$0,945 \pm 0.075$
	Multiclass Classification			
	Accuracy	**Precision**	**Recall**	**F1-Score**
DT	0.629 ± 0.222	0.658 ± 0.195	0.629 ± 0.222	0.599 ± 0.233
RF	0.707 ± 0.171	0.663 ± 0.157	0.707 ± 0.171	0.664 ± 0.173
ExT	$0,687 \pm 0.166$	0.642 ± 0.152	0.687 ± 0.165	0.645 ± 0.165
KNNe	0.570 ± 0.239	0.615 ± 0.249	0.570 ± 0.239	0.563 ± 0.242
LR	0.623 ± 0.241	0.703 ± 0.208	0.623 ± 0.242	0.588 ± 0.249
NC	0.680 ± 0.219	0.685 ± 0.242	$0,680 \pm 0.220$	0.662 ± 0.228

Table 1
Average value with metrics with their standard deviation related to the binary and multiclass classification

The Random Forest model outpaces its counterparts in both binary and multiclass classification scenarios. For the RF model, the obtained accuracy stands at 92% (binary) and 70% (multiclass). Corresponding F1-scores are 88.2% and 60% , respectively. While multiclass classification offers insights for emotion detection via wearables, there remains room for improvement. Comparing results from Schmidt et al.'s benchmark on the WESAD dataset [9], which utilized standardized machine learning techniques and features, our study finds that the RF algorithm delivers superior performance. The accuracy and F1-score is reported in Table 2.

	Binary Classification	
	Accuracy	**F1-score**
DT	83.60 ± 1.08	80.83 ± 1.13
RF	74.97 ± 1.11	64.08 ± 1.68
KNN	74.20	69.14
	Multiclass Classification	
	Accuracy	**F1-score**
DT	63.56 ± 1.73	58.05 ± 1.61
RF	74.97 ± 1.11	64.08 ± 1.68
KNN	56.14	48.70

Table 2
Average value with metrics with their standard deviation related to the binary and multiclass classification by extraction features from signals [9]

Comparing the results, we note that the methods performs better using signal values than signal features.

4. CONCLUSIONS AND FUTURE WORK

In this work, we have compared various classical machine learning algorithms. We have used a public dataset, WESAD, to perform our study. Analyzing the results, we have noted the best results have been archived by the random forest algorithm. This evidence is in line with the results proposed in the literature [9]. We have observed that classifications based on the signal values outcome ones that consider signal features.

In future work, we intend to conduct additional experiments to discern the most relevant physiological signals. It represents another fundamental aspect of detecting stress for real-time analysis using wearable sensors and smartphones. In this case, the aim is to store the minimum information to be non-invasive and reduce the space while maintaining high model performance. We also intend to consider and employ deep learning approaches, such as graph convolution networks or recurrent neural networks, motivated by the results obtained in other scenarios [22, 23]. Moreover, we also intend to study the role of the length of the sliding windows from a theoretical perspective by taking into account various entropy-based methods that have produced evaluable outcomes in the scenario of protein-protein interaction site prediction [24]. Another crucial future investigation is to explore and define approaches to extract and describe the correlation that sliding windows represent. Other representations, like arc-annotated sequences, strings and simplicial complexes, will be explored. We will explore other representations like arc-annotated sequences for the analysis and comparison of time utilizing tools like [25] and strings or simplicial complexes, which allow applying techniques from formal methods to identify

patterns [26] or verify properties [27].

Acknowledgements. This work has been partially supported by the European Union – NextGenerationEU - National Recovery and Resilience Plan, Mission 4 Education and Research - Component 2 From research to business - Investment 1.5, ECS_00000041-VITALITY - Innovation, digitalisation and sustainability for the diffused economy in Central Italy

References

[1] B. S. McEwen, Protective and damaging effects of stress mediators, New England journal of medicine 338 (1998) 171–179.

[2] Health and Safety Executive, HSE on work-related stress. 2021, http://www.hse.gov.uk/statistics/causdis/-ffstress/index.htm, ???? Accessed on March 7, 2022.

[3] E.-H. Lee, Review of the psychometric evidence of the perceived stress scale, Asian nursing research 6 (2012) 121–127.

[4] K. Hovsepian, M. Al'Absi, E. Ertin, T. Kamarck, M. Nakajima, S. Kumar, cstress: towards a gold standard for continuous stress assessment in the mobile environment, in: Proceedings of the 2015 ACM international joint conference on pervasive and ubiquitous computing, 2015, pp. 493–504.

[5] M. Gjoreski, M. Luštrek, M. Gams, H. Gjoreski, Monitoring stress with a wrist device using context, Journal of biomedical informatics 73 (2017) 159–170.

[6] L. Shu, J. Xie, M. Yang, Z. Li, Z. Li, D. Liao, X. Xu, X. Yang, A review of emotion recognition using physiological signals, Sensors 18 (2018) 2074.

[7] R. Li, Z. Liu, Stress detection using deep neural networks, BMC Medical Informatics and Decision Making 20 (2020) 1–10.

[8] D. H. Wolpert, The lack of a priori distinctions between learning algorithms, Neural computation 8 (1996) 1341–1390.

[9] P. Schmidt, A. Reiss, R. Duerichen, C. Marberger, K. Van Laerhoven, Introducing wesad, a multimodal dataset for wearable stress and affect detection, in: Proceedings of the 20th ACM international conference on multimodal interaction, 2018, pp. 400–408.

[10] J. Astola, P. Kuosmanen, Fundamentals of nonlinear digital filtering, CRC press, 2020.

[11] M. Quadrini, S. Daberdaku, A. Blanda, A. Capuccio, L. Bellanova, G. Gerard, Stress detection from wearable sensor data using gramian angular fields and cnn, in: International Conference on Discovery Science, Springer, 2022, pp. 173–183.

[12] S. Shalev-Shwartz, S. Ben-David, Understanding machine learning: From theory to algorithms, Cambridge university press, 2014.

[13] J. R. Quinlan, Induction of decision trees, Machine learning 1 (1986) 81–106.

[14] L. Breiman, Random forests, Machine learning 45 (2001) 5–32.

[15] Y. Freund, R. E. Schapire, et al., Experiments with a new boosting algorithm, in: icml, volume 96, Citeseer, 1996, pp. 148–156.

[16] P. Geurts, D. Ernst, L. Wehenkel, Extremely randomized trees, Machine learning 63 (2006) 3–42.

[17] D. R. Cox, The regression analysis of binary sequences, Journal of the Royal Statistical Society Series B: Statistical Methodology 20 (1958) 215–232.

[18] K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, Y. Singer, Online passive aggressive algorithms (2006).

[19] E. Fix, J. L. Hodges, Discriminatory analysis. nonparametric discrimination: Consistency properties, International Statistical Review/Revue Internationale de Statistique 57 (1989) 238–247.

[20] T. Cover, P. Hart, Nearest neighbor pattern classification, IEEE transactions on information theory 13 (1967) 21–27.

[21] R. Tibshirani, T. Hastie, B. Narasimhan, G. Chu, Diagnosis of multiple cancer types by shrunken centroids of gene expression, Proceedings of the National Academy of Sciences 99 (2002) 6567–6572.

[22] M. Quadrini, S. Daberdaku, C. Ferrari, Hierarchical representation and graph convolutional networks for the prediction of protein–protein interaction sites, in: Machine Learning, Optimization, and Data Science: 6th International Conference, LOD 2020, Siena, Italy, July 19–23, 2020, Revised Selected Papers, Part II 6, Springer, 2020, pp. 409–420.

[23] M. Quadrini, S. Daberdaku, C. Ferrari, Hierarchical representation for ppi sites prediction, BMC bioinformatics 23 (2022) 96.

[24] M. Quadrini, M. Cavallin, S. Daberdaku, C. Ferrari, Prosps: protein sites prediction based on sequence fragments, in: International Conference on Machine Learning, Optimization, and Data Science, Springer, 2021, pp. 568–580.

[25] M. Quadrini, L. Tesei, E. Merelli, Aspralign: a tool for the alignment of rna secondary structures with arbitrary pseudoknots, Bioinformatics 36 (2020) 3578–3579.

[26] M. Quadrini, E. Merelli, R. Piergallini, Loop grammars to identify rna structural patterns., in: Bioinformatics, 2019, pp. 302–309.

[27] M. Loreti, M. Quadrini, A spatial logic for simplicial models, Logical Methods in Computer Science 19 (2023).