
SUPERVISED MULTIPLE KERNEL LEARNING APPROACHES FOR MULTI-OMICS DATA INTEGRATION

A PREPRINT

 **Mitja Briscik***

Institut de Mathématiques
Université Paul Sabatier
Toulouse, FR 31062
mitja.briscik@math.univ-toulouse.fr

 **Gabriele Tazza***

Department of Software Engineering
University of Szeged
Szeged, HU 6720
tazza@inf.u-szeged.hu

László Vidács

Department of Software Engineering
University of Szeged, HU 6720

Marie-Agnès Dillies

Institut Pasteur
Université Paris Cité, Bioinformatics and Biostatistics Hub, F-75015, Paris

Sébastien Déjean

Institut de Mathématiques
Université Paul Sabatier, Toulouse, FR

March 28, 2024

ABSTRACT

Advances in high-throughput technologies have originated an ever-increasing availability of omics datasets. The integration of multiple heterogeneous data sources is currently an issue for biology and bioinformatics. Multiple kernel learning (MKL) has shown to be a flexible and valid approach to consider the diverse nature of multi-omics inputs, despite being an underused tool in genomic data mining. We provide novel MKL approaches based on different kernel fusion strategies. To learn from the meta-kernel of input kernels, we adapted unsupervised integration algorithms for supervised tasks with support vector machines. We also tested deep learning architectures for kernel fusion and classification. The results show that MKL-based models can compete with more complex, state-of-the-art, supervised multi-omics integrative approaches. Multiple kernel learning offers a natural framework for predictive models in multi-omics genomic data. Our results offer a direction for bio-data mining research and further development of methods for heterogeneous data integration.

Keywords Multi-omics data integration · Kernel methods · Deep learning · Data mining

1 Introduction

Data integration has recently attracted substantial attention in the research literature, both for the statistical challenges and promising potential applications in fields such as biology and medicine. Multi-omics data have become increasingly available following the significant growth of high-throughput technologies. The availability of such rich while complex data has expanded the number of available algorithms and methodologies to properly conduct analyses, with the possible need to create novel research profiles [Gomez-Cabrero et al., 2014].

*These authors contributed equally to this work.

In this context, Kernel methods have proven to be a very promising technique for integrating and analyzing high-throughput technologies-generated data. Kernel methods benefit from the possibility of providing a nonlinear version of any linear algorithm that relies solely on dot products. For instance, unsupervised methods such as Kernel Principal Component Analysis [Schölkopf et al., 1997], Kernel Canonical Correlation Analysis [Bach and Jordan, 2003], Kernel Discriminant Analysis [Roth and Steinhage, 1999a] and Kernel Clustering [Girolami, 2002] are all examples of nonlinear algorithms enabled by the so-called kernel trick.

Kernel-based methods also include supervised classification algorithms. Support vector machine is the most popular one, along with Kernel partial least squared regression [Rosipal and Trejo, 2001] or Kernel discriminant analysis [Roth and Steinhage, 1999b].

Several methodologies are also available to integrate multiple high throughput data sources through the so-called Multiple kernel learning (MKL) approach. These methods combine modern optimization techniques' power with kernel methods' framework, providing a new multi-source genomic data learning tool.

In this work, we review classical MKL algorithms, while also exploring alternative MKL approaches. Specifically, we propose adapting unsupervised algorithms for multiple kernel integration to a supervised context i.e. fitting an SVM classification model on a fused kernel obtained through an unsupervised algorithm for convex linear combination of input kernels.

More recently, Deep learning has emerged as a valid alternative to dealing with data integration challenges. A key strength of deep learning lies in its ability to learn homogeneous representations from heterogeneous data sources (images, text, tabular data), making it a perfect candidate for multi-omics integration problems.

Different deep learning methods have already been applied in this domain with promising results. Architectures such as Autoencoders [Wu and Fang, 2022], [Yu, 2022], Graph Neural Networks [Kesimoglu and Bozdag, 2023] [Wang et al., 2021] or Multi-head Attention [Gong et al., 2023] have been successfully adapted to different multi-omics integration tasks reaching the state-of-the-art.

Deep learning has also been used as an alternative approach to multiple kernel fusion [Song et al., 2017] to integrate different kernels from a single data source. This type of architecture can be easily adapted to integrate heterogeneous data sources, such as multi-omics datasets. With this intention, we introduce a deep learning framework tailored for multiple kernel learning (MKL) specifically within the context of multi-omics integration.

To sum up, while multiple kernel learning remains an under-utilized tool for genomic data mining [Wilson et al., 2019], in this work, we propose MKL methods to integrate multi-omics data based both on unsupervised convex linear optimization and deep learning. We aim to show the advantages of this setting while comparing them with the state-of-the-art methods.

2 Related work

Many machine learning (ML) methods are available to unravel biological system mechanisms and find new biomarkers. The big challenges associated with multi-omics data mining and integration are the intrinsic high dimensionality, heterogeneity and nonlinearity of the sample space. For this reason, refined methods are needed to give practitioners new direction and solutions for analyzing such complex datasets.

Numerous integration strategies are available in the literature, including early, mixed, hierarchical, intermediate and late integration. In this work, we focus on the mixed integration type, which has demonstrated to ensure great adaptability for omics data integration as reviewed in Picard et al. [2021].

Early stage integration, the easiest and fastest procedure available, nonetheless poses intrinsic drawbacks. More specifically, since early integration is based on the concatenation of the original data, it naturally increases the input dimensionality while giving more importance to omics with a bigger number of features. Moreover, while being extremely easy and fast to realize, this practice tends to mislead learning algorithms as it does not consider the specific data distribution of each input dataset.

On the contrary, mixed integration allows ML algorithms to conduct the learning phase on more refined and less dimensional datasets. As these methods produce a new version of the input dataset that is more homogeneous among original versions, it facilitates ML algorithms to operate on a unified single input for learning.

Furthermore, another very popular strategy is late integration, which consists of applying each machine learning model separately on each input dataset and then of combining their respective predictions in a later stage. However, as claimed in Picard et al. [2021], this approach may not be relevant for biological applications. Indeed, an integration based solely on the combination of different model predictions cannot be compared to a procedure that directly considers complementary information among different omics, and it can be seen as multiple single-omics analyses.

In the present work, we will investigate mixed integration techniques for multi-omics data integration in comparison to the state-of-the-art method i.e. MOGONET in Wang et al. [2021], a late integration methodology.

Mixed integration

It is generally accepted that a classification model trained with information obtained from different sources leads to a more comprehensive overview of the problem [Huang et al., 2017], [Chicco et al., 2023].

In the field of omics sciences, when different data obtained on the same individuals are available, the integrated analysis can provide richer information about the biological system compared to the results achieved using a single layer of information. New achievements have been reached in a wide area of research, for instance allowing the identification of molecular signatures of human breast tumours [Network, 2012] or for microbial communities profiling [Guidi et al., 2016].

Each omic dataset contains a different aspect of the mechanisms regulating a biological phenotype. In addition, the technologies used to collect them differ. Consequently, the nature and structure of those data are usually very diverse, generating a remarkably heterogeneous framework. Mixed integration or transformation-based strategies undertake the flaws of concatenation-based approaches applying ML algorithm to a simpler representation of each input dataset. The original omics are transformed separately to obtain a clearer, richer and lower in dimensions version. Standard transformation methods that can be used are kernel-based, graph-based, and deep learning methods.

In this work we will focus our attention on kernel-based integration and on deep learning-based methods applied on kernel learning.

Multiple kernel learning

Kernel methods have been shown to offer an elegant and natural mathematical solution to address data integration from heterogeneous sources, as using kernels enables the representation of the datasets in terms of pairwise similarities between sample points [Zhang, 2011], [Wang et al., 2014]. More specifically, given different datasets based on the same n observations x_1, \dots, x_n , kernel methods allow to represent every original dataset with \mathbf{K} , a $n \times n$ kernel matrix containing all the data pairwise similarities $\mathbf{K} = k(x_i, x_j)$. So, even if the original data types are heterogeneous (counts, factors, continuous data, networks, images), after the kernel transformation, all the M input datasets will have the form of a $n \times n$ matrix with real numbers as entries, with M equal to the number of available omic datasets.

Moreover, the meta-kernel obtained from the combination of the M input kernels is a global similarity matrix containing the sample's similarities based on the original datasets' variables. MKL assures great adaptability as many kernel functions are available, such as linear, Gaussian, polynomial, or sigmoid. In this way it is possible to choose and to apply a specific kernel function on a certain omic input, as each function may be more suitable for a specific omic.

The most common approach in multiple kernel learning is to compute a convex linear combination of kernel Gram matrices. Analytically, given M different datasets, MKL consists of the linear combination of the M kernel matrices, as in

$$K^* = \sum_{m=1}^M \beta_m K^m, \quad (1)$$

with $\beta_m \neq 0$ and $\sum_{m=1}^M \beta_m = 1$.

It directly follows that the simplest solution is to fix all the weights to be equal, i.e. to $\frac{1}{M}$. Of course, this setting does not allow us to benefit from the adaptability of the multiple kernel framework. All kernels will contribute equally to the classifier, not taking into account possible redundant or less informative sources of information. The experiment section will denote this setting as **MKL-naive**.

Contrarily, the β_m weights can be optimized more appropriately. Usually, in supervised learning, they are tuned, minimizing the prediction error. The literature offers many algorithms for supervised MKL optimization. For instance, in the work in Lanckriet et al. [2004], the weights are optimized with semidefinite programming techniques. The fused kernel is then used to train an SVM classifier, giving better performances than single omic analysis.

Another approach can be found in Rakotomamonjy et al. [2008] where the convex linear combination is obtained through a weighted 2-norm regularization constrained formulation to promote a sparse kernel combination and using a subgradient descent for weights optimization. The so-called **SimpleMKL** method is available in the R package *RMKL* developed by Wilson et al. [2019].

The *RMKL* package proposes several other algorithms such as **SEMKL**, Simple and Efficient MKL by Xu et al. [2010] where the weights computation is based on the equivalence between group-lasso and MKL. Both SimpleMKL and SEMKL belong to the class of algorithms known as wrapper methods for multiple kernel learning, thus updating kernel weights after each iteration.

A more sophisticated version of these wrapper methods specialized in the reduction of the number of SVM computations is **SpicyMKL** in Suzuki and Tomioka [2011], which is a proximal minimization method that converges super-linearly. This algorithm is also implemented in the *RMKL* package under the name of DALMKL.

A different way to find the kernel coefficients in the convex linear combination of kernels can be found in Yang et al.

[2018] with **GA-fKPLS**, where the authors propose to compute the kernel parameters and weights using genetic algorithms.

A different approach to MKL is presented in Gönen and Alpaydin [2008] and Gönen and Alpaydin [2013], where the authors question the practice of assigning the same weight to a kernel over the whole input space. In this work, they propose a localized multiple kernel learning **LMKL** based on the local selection of the appropriate kernel function, allowing to reduce the number of support vectors.

Multiple kernel learning can also be used in the unsupervised learning framework. However, selecting appropriate criteria for weight optimization is less straightforward. Thus, the solutions available in the literature are less numerous. If it is natural to optimize the weights through the minimization of the prediction error for supervised learning, the same does not apply in an unsupervised context. In Mariette and Villa-Vialaneix [2017], the authors proposed **STATIS-UMKL**, a methodology to provide an approach to reach a consensus kernel based on the resemblance of the different kernels. However, it must be pointed out that the original features are lost after the kernel transformation since the samples are addressed solely with their similarities in the new feature space. Thus, the interpretation of the unsupervised model can be more challenging. In Mariette and Villa-Vialaneix [2017], the authors also proposed a method based on kernel PCA and random permutation to evaluate the importance of the original variables. Also, in Briscik et al. [2023], the authors proposed KPCA-IG, an approach which provides a data-driven feature importance, where the influence of each original variable can be computed in the space of the kernel principal components as in the standard PCA.

Deep Learning approaches

Deep learning techniques are increasingly being employed in the context of multi-omics data analysis. One of the advantage of deep learning is its capacity to learn homogeneous representations from different input sources. In particular, multi-modal architectures allow the use of heterogeneous datasets, such as images, tabular data, time series, or graphs, to learn the underlying complex relationships among different aspects of a biological phenotype.

As reviewed in Stahlschmidt et al. [2022], this kind of architecture is gaining popularity in the biomedical field, where data are becoming increasingly multi-modal. Recently, in this context, different works introduced approaches based on multi-modal deep learning to deal with different types of omics data, these multi-modal architectures are suited for both Mixed and Late integration strategies. As introduced, we will concentrate on Mixed integration approaches compared to the Late integration methods that can be regarded as the state-of-the-art for the datasets of interest in our analysis [Wang et al., 2021] [Gong et al., 2023].

One of the most commonly used deep learning methods for Mixed integration strategies is Autoencoder. Autoencoder is an unsupervised deep learning method used to learn a latent representation of the data by minimizing the reconstruction error between the input and the reconstructed output. In the context of Mixed integration, they can be easily used to learn independent homogeneous latent representations to integrate them in a final shared layer [Xu et al., 2019]. Autoencoders can also be used to learn latent representations that depend on different omics inputs, as in Wu and Fang [2022]. In this case, the approach uses Autoencoders in two different steps, first as a pre-processing for the two different inputs and then as an integration step, part of the learning process.

Other possible approaches for Mixed integration involve the use of feed-forward neural networks. In particular, in Lin et al. [2020], the authors built an architecture based on different encoding sub-networks to learn homogeneous representations from the different types of omics data, then a fusion step to create a concatenated representation of multi-omics, and finally, a classification sub-network is used to perform the cancer subtype classification. Alternatively, in Sharifi-Noghabi et al. [2019], a similar architecture equipped with a triplet loss is used for drug response prediction.

Despite this, several state-of-the-art methods belong to the Late integration family, such as MOGONET by Wang et al. [2021] and MOADLN by Gong et al. [2023]. MOGONET transforms the input data into matrices of similarity among observations to build a graph structure and apply a Graph Convolutional Neural Network to each omic to obtain an initial prediction. After this first step, a View Correlation Discovery Network (VCDN) finally combines all the independent predictions to determine the right label.

MOADLN, instead, uses the Self-attention mechanism to build a similarity network and exploit the correlation between intra-omic observations. In this case, each input instance is an element of a set i.e. a specific observation within a single omics type, and the Self Attention mechanism learns the weights for each of these elements, meaning that it determines the significance of each instance in relation with others within the same omics type. Also, for MOADLN, the first step is the initial independent prediction for each omics type, followed by a final combination through a Multi-Omics Correlation Discovery Network (MOCDN) to explore the cross-omics relations.

3 Materials and methods

As considered in the previous section, the results presented in Wang et al. [2021] and Gong et al. [2023] can be considered the state-of-the-art in terms of predictive performance.

Dataset	Classes	Number of features mRNA, meth, miRNA	Number of features for training mRNA, meth, miRNA
ROSMAP	NC: 169, AD: 182	55,889; 23,788; 309	200; 200; 200
BRCA	Normal-like: 115, Basal-like: 131, HER2-enriched: 46, Luminal A: 436, Luminal B: 147	20,531; 20,106; 503	1000; 1000; 503

Table 1: The ROSMAP dataset contains two classes: Alzheimer’s disease (AD) patients and normal control (NC). The breast invasive carcinoma dataset (BRCA) contains PAM50 subtype classes: normal-like, basal-like, human epidermal growth factor receptor 2 (HER2)-enriched, Luminal A, and Luminal B.

In this section, we present all the experiments to test different MKL methods, architectures and combinations in order to compare possible solutions for multi-omics data integration.

3.1 Datasets

The datasets considered in this work are the publicly available ROSMAP for Alzheimer’s Disease classification and BRCA for breast invasive carcinoma (BRCA) PAM50 subtype classification. In order to be sure to conduct a fair comparison with MOGONET, we used the same datasets. Wang et al. [2021] performed an initial feature selection obtained through the sequential calculation of an ANOVA F-value on the original data to evaluate whether a feature was significantly different across different classes. Moreover, the authors kept the number of features such that the first principal component after feature pre-selection explains at least 50% of the variance.

As also Gong et al. [2023] proceed, we decided to conduct our analysis on ROSMAP and BRCA since they are the only pre-processed datasets available in Wang et al. [2021] GitHub repository, <https://github.com/txWang/MOGONET/tree/main>. For each of the 2 datasets three types of omics are considered for classification purposes: mRNA expression (mRNA), DNA methylation (meth), and miRNA expression data (miRNA). Table 1 contains all the details for the two datasets.

3.2 Methods

Both in Wang et al. [2021] and Gong et al. [2023], the authors compared the performance of their methods, MOGONET and MOADNL, respectively, with other typical classification algorithms such as K-nearest neighbours (KNN), Support vector machine (SVM) or LASSO regression.

Taking SVM as an example, the analysis is applied to the concatenation of the 3 multi-omics datasets, where its performance shows a significantly lower accuracy in both studies. However, as SVM can be viewed as a kernel-based classification algorithm, applying it to an early stage integration, i.e., to a combined dataset obtained by simple concatenation of the input datasets, can be seen as an oversimplification. Moreover, a proper parameters tuning must be carried out along with the choice of a suitable kernel function. Thus, our analysis compares MOGONET’s performance with more suitable and fair usage of multiple kernel learning with support vector machines.

Moreover, new approaches of multiple kernel learning in combination with deep learning classification models are presented in order to exploit at the same time the adaptability of kernel methods avoiding the optimization of the weights in the convex linear combination and the classification power of deep architectures.

3.2.1 Multiple kernel learning - SVM

As presented in Section 2, there are many optimization algorithms to compute the coefficients of the convex linear combination of input kernel gram matrices in the literature.

For completeness, in this work, we will present the results obtained using **MKL-naive**, **SimpleMKL** in the case of binary classification problem and **STATIS-UMKL**.

We decided to choose SimpleMKL among wrapper and more sophisticated optimization methods as all the algorithms seem to have similar performance in the case of an analysis with few kernels, as shown in Wilson et al. [2019].

On the contrary, STATIS-UMKL in Mariette and Villa-Vialaneix [2017] is an algorithm to obtain a consensus meta-kernel in an unsupervised framework. To our knowledge, STATIS-UMKL has never been used with support vector machines for classification purposes. However, the peculiarity of this procedure, which aims to take the different

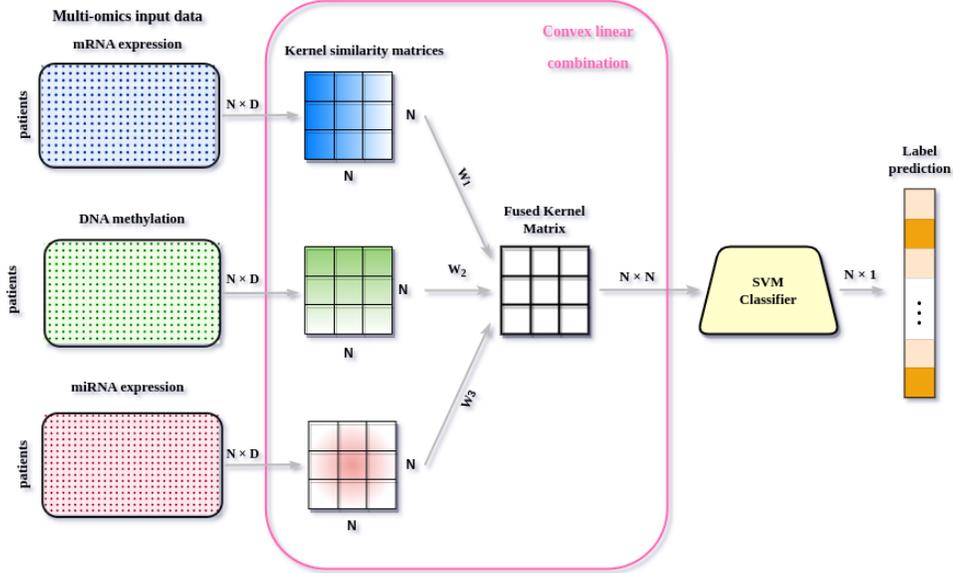


Figure 1: A kernel function is applied on each dataset separately. In MKL, a convex linear combination provides a fused Meta-kernel that summarizes the information of input omics. Then an SVM classifier is used for classification.

specificities of each dataset into account by fusing them into a single meta-kernel, may also enhance classification performance. In Figure 1, it is possible to see the network structure for all the SVM algorithms that are used for the experiments. This architecture belongs to the Mixed integration type as the integration of the input omics is preceded by a data transformation, and the SVM algorithm is applied to the convex linear combination of the datasets performed at the feature space.

For completeness, we also trained a support vector machine on the direct concatenation of original datasets (**SVM-concat**) using the same tuning procedure for the hyperparameters used for the other algorithms.

3.2.2 Deep Multiple kernel learning

As introduced previously, employing neural network architectures is another way to combine the input kernel matrices by avoiding the task of convex linear optimization. More specifically, in Song et al. [2017], a deep learning architecture that includes a dense embedding of kernels and a multi-modal neural network is used for fusing multiple kernels.

In our case, we adapted this approach to a multi-omics analysis, meaning that the kernel matrices represent different data sources and not different representations of a single data source, as in a classic multiple kernel fusion problem. As can be seen in Figure 2 and 3, we used Kernel PCA to obtain a dense embedding for each omic layer, which is then used to feed a multi-modal neural network. The NN takes as input the transformed datasets and fuses them in a mixed-stage integration fashion by concatenating the different representations right before a final classification step. We call this approach **Deep Multiple kernel learning** i.e. **Deep MKL** to highlight that it is a Multiple kernel learning method that employs deep learning to combine the different kernel information.

In the context of multi-modal architectures, cross-connections between modalities can improve the model’s performance, allowing the flow of information between modalities at different learning process levels before the fusion step [Bica et al., 2018],[Zhou, 2023]. In our case, we propose an additional architecture, called **Cross-Modal MKL** in Figure 3, which utilizes cross-connections to link the different layers of omics.

3.3 Performance evaluation

3.3.1 Experimental setup

To evaluate the classification performance of the MKL-SVM algorithms and Deep MKL, we implemented the same evaluation pipeline already used by MOGONET in Wang et al. [2021] and by MOADLN in Gong et al. [2023]. It consists of evaluating the model’s performance on 5 random train/test partitions of the dataset. To maintain the balance of class distributions among the partitions, a stratified version of the split is adopted, keeping the ratio of 30/70 % for the train/test splits.

For final evaluation, we present the mean and standard deviation of different performance metrics among the 5 randomly

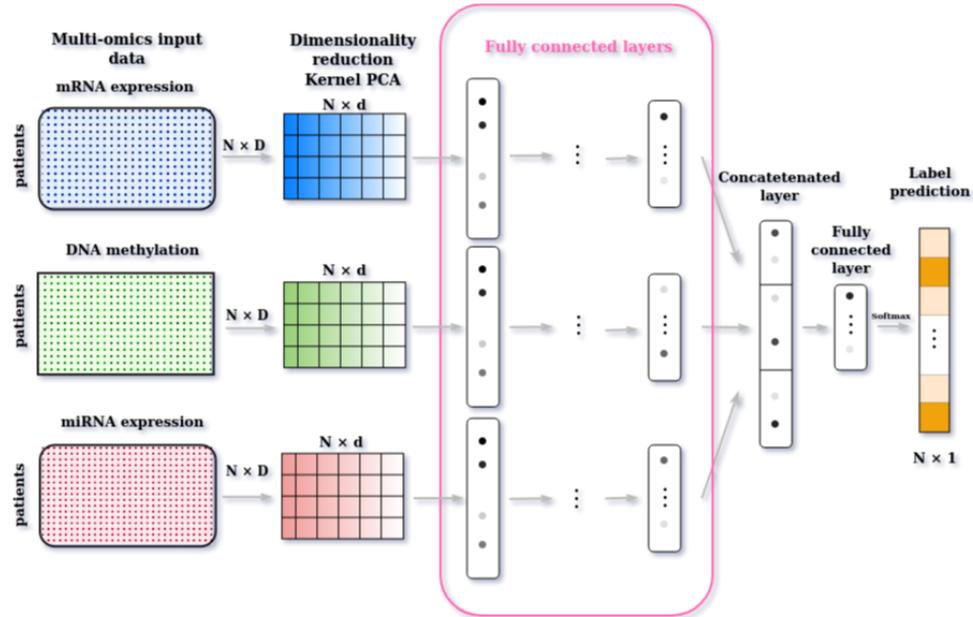


Figure 2: Deep MKL takes in input the Kernel PCA dense embeddings of different omics datasets. It extracts the features using different feed-forward sub-networks and then fuses the learnt representations by concatenating them for the final classification.

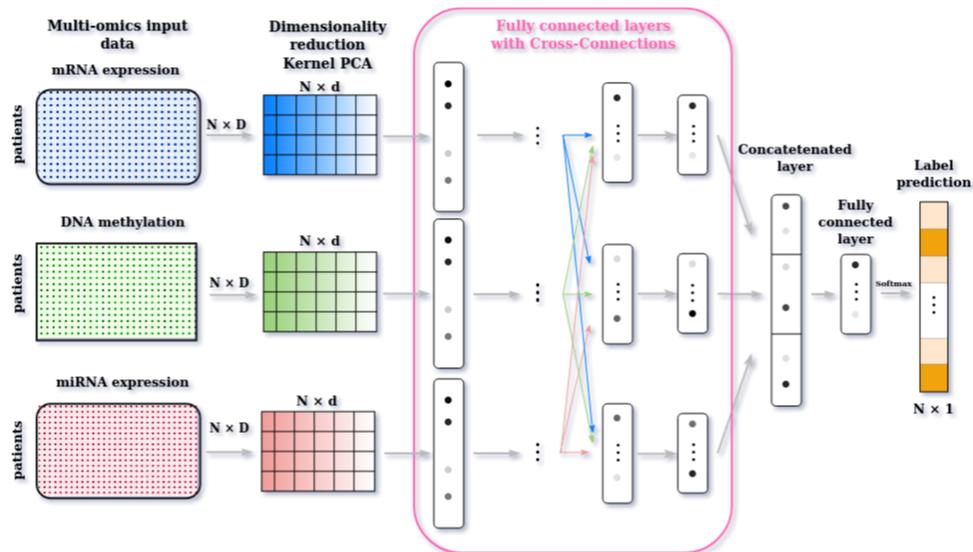


Figure 3: Cross-Modal MKL takes in input the Kernel PCA dense embeddings of different omics datasets. It extracts the features using different feed-forward sub-networks that are linked by cross-connections, then fuses the learnt representations by concatenating them for the final classification.

generated training/test splits, with a seed set of [0, 1, 2, 3, 4] for reproducibility purposes. The seeds used in MOGONET are not publicly available, meaning that the results are not completely reproducible. For this reason, we recomputed all the metrics using their publicly available code and the same seeds of our experiments in order to have a fair comparison. On the contrary, we have not recomputed the metrics for MOADLN as the code is not publicly available.

Methods	Integration	Optimized Parameters	Description
SVM concat	Early	C, σ	Direct concatenation of the datasets
SVM naive	Mixed	C, σ	Sum of the kernel
SEMKL-SVM	Mixed	C, σ	Weighted sum of kernels
STATIS-UMKL + SVM	Mixed	C, σ	Weighted sum of kernels
Deep MKL	Mixed	σ , epochs, principal components, dropout value	Deep Learning kernel fusion
Cross-Modal Deep MKL	Mixed	σ , epochs, principal components, dropout value	Deep Learning kernel fusion
MOGONET	Late	Optimized k	Graph convolutional network

Table 2: Summary and description for all the tested methods with all the tuned hyperparameters

3.3.2 Hyperparameters tuning

In the context of MKL-SVM, a Grid Search 5-folds cross-validation has been computed on the training sets employing a Gaussian radial basis kernel.

Cross-validation has been used to tune the following parameters:

- C parameter: the cost of constraints violation, the so-called C-constant of the regularization term in the Lagrange formulation of the support vector machine algorithm.
- The sigma parameter: the inverse kernel width for the radial basis kernel function.

For the experiments, the C parameter has been set in the range [1, 25], while the sigma in the range of [0.005, 0.00005] for both datasets.

In the context of Deep MKL, we employed a Random Search 5-folds cross-validation for the hyperparameters tuning. Also in this case, all the experiments have been carried out employing a Gaussian radial basis kernel for the Kernel PCA step. We fixed the number of layers and the number of neurons as in MOGONET i.e. [200, 200, 100] for ROSMAP and [400, 400, 200] for BRCA. We also fixed the learning rate at 5×10^{-5} for ROSMAP and 10^{-4} for BRCA, and the dropout intensity at 0.5 and 0.3 respectively. Adam classifier [Kingma and Ba, 2014] and a batch size at 32 are adopted for both datasets. We defined a different search space for the other hyperparameters for each dataset. In the case of ROSMAP, the sigma value for the Kernel PCA is defined in the set of {0.0005, 0.0007, 0.001}. Meanwhile, for BRCA, the set is {0.00005, 0.0005, 0.005}. Regarding the number of principal components in ROSMAP, we fixed it to 120. While in BRCA, we defined a search space in the range of [2, 20], to choose the optimal combination together with the sigma parameter.

Since the variability among the different folds made the results unreliable for an early stopping strategy, we chose the number of epochs by defining a range from 100 to 200 with an interval of 10, letting the hyperparameter tuning optimization select the best value in combination with all the other parameters.

For the reproduction of the results of MOGONET, we used the optimized parameter k as suggested by the authors, namely equal to 2 for ROSMAP and 10 for BRCA. This parameter controls the average number of edges per node of the Adjacency matrix used for the training of graph convolutional neural network.

3.3.3 Metrics

In order to have a fair comparison, we employed the same metrics of the state-of-the-art methods. For binary classification, we used accuracy (ACC), F1 score (F1) and area under the curve (AUC):

$$\text{ACC} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \quad (2)$$

with TP = True Positive, TN = True Negative, FP = False Positive and FN = False Negative.

$$\text{F1} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \quad (3)$$

where Precision = $\frac{\text{TP}}{\text{TP} + \text{FP}}$ and Recall = $\frac{\text{TP}}{\text{TP} + \text{FN}}$.

In the context of binary classification, the F1 score reflects the harmonic mean between Precision and Recall, meaning that it measures how balanced these other two metrics are for one classifier. The Precision score represents how accurate the positive predictions are. Meanwhile, the Recall metric measures how many True Positives are predicted out of the total number of positive observations.

The AUC score, or area under the ROC curve, represents the classifier’s ability to distinguish positive from negative with regard to the classification thresholds. It measures the classifier’s performance and its independence from the threshold.

For the multi-class classification task, we employed the accuracy (ACC), the macro-averaged F1 score (F1-macro) and the F1 score weighted by its support i.e. the number of instances in that class (F1-weighted).

In multi-class classification, the F1 score is calculated for each class in a one-vs-all manner. In the case of F1-macro, the F1 scores are then averaged, considering each class equally, regardless of the imbalance of the class distribution in the data.

$$\text{F1-macro} = \frac{1}{C} \sum_{i=1}^C F1_i \quad (4)$$

C is the number of classes and $F1_i$ is the F1 score for the class i .

The F1-weighted, instead, takes into account the imbalance of the class distribution in the data, and it is calculated by a weighted average where the weights are the percentage of the instances in one class.

$$\text{F1-weighted} = \frac{1}{C} \sum_{i=1}^C \left(\frac{\text{support}_i}{\text{total support}} \right) \cdot F1_i \quad (5)$$

where support_i is the number of instances of class i and total support is the total number of instances in the data.

4 Results

We compared the classification performance of different MKL algorithms with MOGONET, as MOADLN’s code is not publicly available. As anticipated, the MOGONET code seed is unavailable; therefore, we could not replicate the results exactly. Thus, we proceeded with the computation of the metrics for MOGONET based on the publicly available code and using the same environment and seed selection of our experiments.

For BRCA, as show in Table 3, all the MKL algorithms achieved high performances for all the metrics. While the results for ROSMAP, Table 4, show a similar trend for SVM-based approaches, deep learning performs worse than all the other methods.

As shown in Tables 3-4, kernel-based methods are comparable and even outperformed MOGONET on both datasets in all the computed performance metrics. These results again show the kernel framework’s advantages in genomics data mining. Even the results obtained with an SVM trained on the direct concatenation of the input datasets, SVM-concat, show a relatively good performance, especially on ROSMAP, the smallest dataset. In Wang et al. [2021], the performances obtained with SVM-concat are lower, suggesting that even a simple procedure such as early integration followed by proper parameter tuning and an appropriate kernel choice of the SVM may already give a good model

Algorithm	BRCA		
	ACC	F1_weighted	F1_macro
SVM concat	0.793 ± 0.018	0.800 ± 0.016	0.776 ± 0.017
SVM naive	0.838 ± 0.008	0.849 ± 0.008	0.828 ± 0.011
STATIS-UMKL + SVM	0.846 ± 0.011	0.858 ± 0.010	0.837 ± 0.018
Deep MKL	0.835 ± 0.016	0.840 ± 0.019	0.801 ± 0.021
Cross-Modal Deep MKL	0.827 ± 0.027	0.829 ± 0.029	0.794 ± 0.039
MOGONET	0.736 ± 0.038	0.726 ± 0.041	0.65 ± 0.053

Table 3: Metrics average and standard deviation over 5 random test splits for the performance evaluation on BRCA dataset.

Algorithm	ROSMAP		
	ACC	AUC	F1
SVM concat	0.765 ± 0.019	0.863 ± 0.044	0.763 ± 0.015
SVM naive	0.790 ± 0.006	0.881 ± 0.010	0.778 ± 0.018
SEMKL-SVM	0.775 ± 0.039	0.869 ± 0.035	0.763 ± 0.037
STATIS-UMKL + SVM	0.784 ± 0.038	0.878 ± 0.019	0.772 ± 0.039
Deep MKL	0.747 ± 0.018	0.810 ± 0.021	0.762 ± 0.014
Cross-Modal Deep MKL	0.732 ± 0.012	0.808 ± 0.018	0.752 ± 0.025
MOGONET	0.787 ± 0.027	0.878 ± 0.021	0.791 ± 0.045

Table 4: Metrics average and standard deviation over 5 random test splits for the performance evaluation on ROSMAP dataset.

alternative. However, for datasets with a bigger feature space, such as BRCA, mixed integration strategies have to be preferred not to increase the input dimensionality and to consider the specific data distribution of each input dataset. Methods such as SEMKL and STATIS-UMKL, which aim to optimise the input kernel matrices’ convex linear combination, showed high performances in most of the different metrics. It should be noted that the MKL with equal weights in SVM-naive showed the best performance in the ROSMAP dataset, indicating that the datasets are probably similarly informative in this context. For this dataset, the second best was STATIS-UMKL + SVM, where the mean over 5 runs of the 3 weights of the convex linear combination of kernel matrices of 0.361, 0.308, 0.331 suggests that the 3 omics are equally important.

The Deep MKL approach to integrate multiple kernels shows results comparable with the STATIS-UMKL + SVM method for the BRCA dataset. In the case of the ROSMAP dataset, it performs worse than all the methods based on SVM. The difference in performance between ROSMAP and BRCA can be largely attributed to the dataset sizes. This phenomenon is consistent with established understanding that deep learning models tend to underperform in scenarios involving smaller datasets [Borisov et al., 2021].

Cross-connections, which were expected to improve the predictions as they ensure more layers of integration between different omics, show worse performances than the simpler Deep MKL architecture.

5 Conclusion

Multiple kernel learning is a well-established algorithm in the machine learning community, while its use has yet to be widespread among practitioners for bio-data mining.

This work presents two novel different approaches for multiple kernel learning in the context of multi-omics data integration. One employs unsupervised learning techniques along with Support Vector Machines (SVM). The other

utilizes deep learning as a substitute for convex linear optimization to integrate kernels. The proposed methodologies are tested and compared with state-of-the-art methods performances. The experimental results on two publicly available biomedical datasets show that approaches based on kernel mixed integration exhibit comparable or even improved performance compared with state-of-the-art methods. Deep learning-based procedures used to integrate input kernels and for classification demonstrate to be a valid alternative to the more classical multiple kernel learning optimizations in the case of datasets with large enough sample size. However, for a small dataset such as ROSMAP, classical MKL with SVM should be preferred.

Future work could investigate other types of data kernel embedding and different deep architectures to exploit the kernel framework in the context of Deep multiple kernel learning. For classical multiple kernel learning, different types of kernel functions can be tested, as each omic dataset could benefit from ad-hoc kernel function choices.

MKL showed that despite being under-utilized in multi-omics data analysis, it provides a fast and reliable solution that can compete with more complex and convoluted ML architectures.

Acknowledgements

Not applicable.

Funding

This work was funded by E-MUSE MSCA-ITN-2020 European Training Network under the Marie Skłodowska-Curie grant agreement No 956126. The funding did not influence the study's design, the interpretation of data, or the writing of the manuscript.

Authors' contributions

MB and GT wrote the main manuscript and performed the analysis under the supervision of SD, MAD and LV. All authors read and approved the final manuscript.

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Availability of data and materials

BRCA and ROSMAP reduced dataset available in the GitHub repository <https://github.com/txWang/MOGONET/tree/main>.

The source code of this work can be downloaded from GitHub: https://github.com/gabrieletaz/MKL_MO

References

- David Gomez-Cabrero, Imad Abugessaisa, Dieter Maier, Andrew Teschendorff, Matthias Merklenschlager, Andreas Gisel, Esteban Ballestar, Erik Bongcam-Rudloff, Ana Conesa, and Jesper Tegnér. Data integration in the era of omics: current and future challenges. *BMC Systems Biology*, 8(Suppl 2):11, 2014. doi:10.1186/1752-0509-8-s2-i1. URL <https://doi.org/10.1186/1752-0509-8-s2-i1>.
- Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. Kernel principal component analysis. In *Artificial Neural Networks — ICANN'97*, pages 583–588, Berlin, Heidelberg, 1997. Springer Berlin Heidelberg.
- Francis Bach and Michael Jordan. Kernel independent component analysis. *Journal of Machine Learning Research*, 3: 1–48, 03 2003. doi:10.1162/153244303768966085.
- Volker Roth and Volker Steinhage. Nonlinear discriminant analysis using kernel functions. In *NIPS*, pages 568–574, 1999a.

- M. Girolami. Mercer kernel-based clustering in feature space. *IEEE Transactions on Neural Networks*, 13(3):780–784, May 2002. doi:10.1109/tnn.2002.1000150. URL <https://doi.org/10.1109/tnn.2002.1000150>.
- Roman Rosipal and Leonard J Trejo. Kernel partial least squares regression in reproducing kernel hilbert space. *Journal of machine learning research*, 2(Dec):97–123, 2001.
- Volker Roth and Volker Steinhage. Nonlinear discriminant analysis using kernel functions. *Advances in neural information processing systems*, 12, 1999b.
- Xing Wu and Qiulian Fang. Stacked autoencoder based multi-omics data integration for cancer survival prediction, 2022. URL <https://arxiv.org/abs/2207.04878>.
- Tianwei Yu. Aime: Autoencoder-based integrative multi-omics data embedding that allows for confounder adjustments. *PLOS Computational Biology*, 18(1):e1009826, January 2022. ISSN 1553-7358. doi:10.1371/journal.pcbi.1009826. URL <http://dx.doi.org/10.1371/journal.pcbi.1009826>.
- Ziyinet Nesibe Kesimoglu and Serdar Bozdog. Supreme: multiomics data integration using graph convolutional networks. *NAR Genomics and Bioinformatics*, 5(2), March 2023. ISSN 2631-9268. doi:10.1093/nargab/lqad063. URL <http://dx.doi.org/10.1093/nargab/lqad063>.
- Tongxin Wang, Wei Shao, Zhi Huang, Haixu Tang, Jie Zhang, Zhengming Ding, and Kun Huang. MOGONET integrates multi-omics data using graph convolutional networks allowing patient classification and biomarker identification. *Nature Communications*, 12(1), June 2021. doi:10.1038/s41467-021-23774-w. URL <https://doi.org/10.1038/s41467-021-23774-w>.
- Ping Gong, Lei Cheng, Zhiyuan Zhang, Ao Meng, Enshuo Li, Jie Chen, and Longzhen Zhang. Multi-omics integration method based on attention deep learning network for biomedical data classification. *Computer Methods and Programs in Biomedicine*, 231:107377, April 2023. doi:10.1016/j.cmpb.2023.107377. URL <https://doi.org/10.1016/j.cmpb.2023.107377>.
- Huan Song, Jayaraman J. Thiagarajan, Prasanna Sattigeri, Karthikeyan Natesan Ramamurthy, and Andreas Spanias. A deep learning approach to multiple kernel fusion. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2292–2296. IEEE, 2017. doi:10.1109/icassp.2017.7952565. URL <http://dx.doi.org/10.1109/ICASSP.2017.7952565>.
- Christopher M. Wilson, Kaiqiao Li, Xiaoqing Yu, Pei-Fen Kuan, and Xuefeng Wang. Multiple-kernel learning for genomic data mining and prediction. *BMC Bioinformatics*, 20(1), August 2019. doi:10.1186/s12859-019-2992-1. URL <https://doi.org/10.1186/s12859-019-2992-1>.
- Milan Picard, Marie-Pier Scott-Boyer, Antoine Bodein, Olivier Périn, and Arnaud Droit. Integration strategies of multi-omics data for machine learning analysis. *Computational and Structural Biotechnology Journal*, 19:3735–3746, 2021. ISSN 2001-0370. doi:<https://doi.org/10.1016/j.csbj.2021.06.030>. URL <https://www.sciencedirect.com/science/article/pii/S2001037021002683>.
- Sijia Huang, Kumardeep Chaudhary, and Lana X. Garmire. More is better: Recent progress in multi-omics data integration methods. *Frontiers in Genetics*, 8, June 2017. ISSN 1664-8021. doi:10.3389/fgene.2017.00084. URL <http://dx.doi.org/10.3389/fgene.2017.00084>.
- Davide Chicco, Fabio Cumbo, and Claudio Angione. Ten quick tips for avoiding pitfalls in multi-omics data integration analyses. *PLOS Computational Biology*, 19(7):e1011224, July 2023. ISSN 1553-7358. doi:10.1371/journal.pcbi.1011224. URL <http://dx.doi.org/10.1371/journal.pcbi.1011224>.
- The Cancer Genome Atlas Network. Comprehensive molecular portraits of human breast tumours. *Nature*, 490(7418):61–70, September 2012. ISSN 1476-4687. doi:10.1038/nature11412. URL <http://dx.doi.org/10.1038/nature11412>.
- Lionel Guidi, Samuel Chaffron, Lucie Bittner, Damien Eveillard, Abdelhalim Larhlimi, Simon Roux, Youssef Darzi, Stephane Audic, Léo Berline, Jennifer R. Brum, Luis Pedro Coelho, Julio Cesar Ignacio Espinoza, Shruti Malviya, Shinichi Sunagawa, Céline Dimier, Stefanie Kandels-Lewis, Marc Picheral, Julie Poulain, Sarah Searson, Lars Stemmann, Fabrice Not, Pascal Hingamp, Sabrina Speich, Mick Follows, Lee Karp-Boss, Emmanuel Boss, Hiroyuki Ogata, Stephane Pesant, Jean Weissenbach, Patrick Wincker, Silvia G. Acinas, Peer Bork, Colomban de Vargas, Daniele Iudicone, Matthew B. Sullivan, Jeroen Raes, Eric Karsenti, Chris Bowler, and Gabriel Gorsky. Plankton networks driving carbon export in the oligotrophic ocean. *Nature*, 532(7600):465–470, February 2016. ISSN 1476-4687. doi:10.1038/nature16942. URL <http://dx.doi.org/10.1038/nature16942>.
- Xinhua Zhang. *Kernel Methods*, chapter 1, page 566–570. Springer US, 2011. ISBN 9780387301648. doi:10.1007/978-0-387-30164-8_430. URL http://dx.doi.org/10.1007/978-0-387-30164-8_430.

- X. Wang, E. P. Xing, and D. J. Schaid. Kernel methods for large-scale genomic data analysis. *Briefings in Bioinformatics*, 16(2):183–192, July 2014. ISSN 1477-4054. doi:10.1093/bib/bbu024. URL <http://dx.doi.org/10.1093/bib/bbu024>.
- Gert R. G. Lanckriet, Nello Cristianini, Michael I. Jordan, and William Stafford Noble. *Kernel-Based Integration of Genomic Data Using Semidefinite Programming*, chapter 1, page 231–260. The MIT Press, July 2004. ISBN 9780262256926. doi:10.7551/mitpress/4057.003.0015. URL <http://dx.doi.org/10.7551/mitpress/4057.003.0015>.
- Alain Rakotomamonjy, Francis Bach, Stephane Canu, and Yves Grandvalet. SimpleMKL. *Journal of Machine Learning Research*, 9:2491–2521, 2008. URL <https://hal.science/hal-00218338>.
- Zenglin Xu, Rong Jin, Haiqin Yang, Irwin King, and Michael R. Lyu. Simple and efficient multiple kernel learning by group lasso. In *Proceedings of the 27th International Conference on International Conference on Machine Learning, ICML'10*, page 1175–1182, Madison, WI, USA, 2010. Omnipress. ISBN 9781605589077.
- c Taiji Suzuki and Ryota Tomioka. Spicymkl: a fast algorithm for multiple kernel learning with thousands of kernels. *Machine Learning*, 85(1–2):77–108, June 2011. ISSN 1573-0565. doi:10.1007/s10994-011-5252-9. URL <http://dx.doi.org/10.1007/s10994-011-5252-9>.
- Haitao Yang, Hongyan Cao, Tao He, Tong Wang, and Yuehua Cui. Multilevel heterogeneous omics data integration with kernel fusion. *Briefings in Bioinformatics*, November 2018. doi:10.1093/bib/bby115. URL <https://doi.org/10.1093/bib/bby115>.
- Mehmet Gönen and Ethem Alpaydin. Localized multiple kernel learning. In *Proceedings of the 25th international conference on Machine learning - ICML '08*, page 352–359. ACM Press, 2008. doi:10.1145/1390156.1390201. URL <https://doi.org/10.1145/1390156.1390201>.
- Mehmet Gönen and Ethem Alpaydin. Localized algorithms for multiple kernel learning. *Pattern Recognition*, 46(3):795–807, March 2013. ISSN 0031-3203. doi:10.1016/j.patcog.2012.09.002. URL <http://dx.doi.org/10.1016/j.patcog.2012.09.002>.
- Jérôme Mariette and Nathalie Villa-Vialaneix. Unsupervised multiple kernel learning for heterogeneous data integration. *Bioinformatics*, 34(6):1009–1015, October 2017. doi:10.1093/bioinformatics/btx682. URL <https://doi.org/10.1093/bioinformatics/btx682>.
- Mitja Briscik, Marie-Agnès Dillies, and Sébastien Déjean. Improvement of variables interpretability in kernel PCA. *BMC Bioinformatics*, 24(1), July 2023. doi:10.1186/s12859-023-05404-y. URL <https://doi.org/10.1186/s12859-023-05404-y>.
- Sören Richard Stahlschmidt, Benjamin Ulfenborg, and Jane Synnergren. Multimodal deep learning for biomedical data fusion: a review. *Briefings in Bioinformatics*, 23(2), January 2022. ISSN 1477-4054. doi:10.1093/bib/bbab569. URL <http://dx.doi.org/10.1093/bib/bbab569>.
- Jing Xu, Peng Wu, Yuehui Chen, Qingfang Meng, Hussain Dawood, and Hassan Dawood. A hierarchical integration deep flexible neural forest framework for cancer subtype classification by integrating multi-omics data. *BMC Bioinformatics*, 20(1), October 2019. ISSN 1471-2105. doi:10.1186/s12859-019-3116-7. URL <http://dx.doi.org/10.1186/s12859-019-3116-7>.
- Yuqi Lin, Wen Zhang, Huanshen Cao, Gaoyang Li, and Wei Du. Classifying breast cancer subtypes using deep neural networks based on multi-omics data. *Genes*, 11(8):888, August 2020. ISSN 2073-4425. doi:10.3390/genes11080888. URL <http://dx.doi.org/10.3390/genes11080888>.
- Hossein Sharifi-Noghabi, Olga Zolotareva, Colin C Collins, and Martin Ester. Moli: multi-omics late integration with deep neural networks for drug response prediction. *Bioinformatics*, 35(14):i501–i509, July 2019. ISSN 1367-4811. doi:10.1093/bioinformatics/btz318. URL <http://dx.doi.org/10.1093/bioinformatics/btz318>.
- Ioana Bica, Petar Velickovic, and Hui Xiao. Multi-omics data integration using cross-modal neural networks. In *The European Symposium on Artificial Neural Networks*, pages 385,390, 2018. URL <https://api.semanticscholar.org/CorpusID:53239415>.
- Tongxue Zhou. Modality-level cross-connection and attentional feature fusion based deep neural network for multi-modal brain tumor segmentation. *Biomedical Signal Processing and Control*, 81:104524, March 2023. ISSN 1746-8094. doi:10.1016/j.bspc.2022.104524. URL <http://dx.doi.org/10.1016/j.bspc.2022.104524>.
- Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2014. URL <https://arxiv.org/abs/1412.6980>.
- Vadim Borisov, Tobias Leemann, Kathrin Seßler, Johannes Haug, Martin Pawelczyk, and Gjergji Kasneci. Deep neural networks and tabular data: A survey. *CoRR*, abs/2110.01889, 2021. URL <https://arxiv.org/abs/2110.01889>.