Universidade Federal de São Carlos– UFSCar
Centro de Ciências Exatas e de Tecnologia– CCET
Departamento de Computação– DC
Graduate Program in Computer Science– PPGCC

André Hallwas Ribeiro Alves

Hybrid and Semi-Supervised Predictive Bi-Clustering Trees for Interaction Prediction

São Carlos
2023

André Hallwas Ribeiro Alves

Hybrid and Semi-Supervised Predictive Bi-Clustering Trees for Interaction Prediction

Dissertation submitted to the Graduate Program in Computer Science from Centro de Ciências Exatas e de Tecnologia - Universidade Federal de São Carlos, as part of the requirements for obtaining the title of Master in Computer Science.

Field of Study: Artificial Intelligence

Supervisor: Ricardo Cerri

São Carlos
2023

UNIVERSIDADE FEDERAL DE SÃO CARLOS

Centro de Ciências Exatas e de Tecnologia
Programa de Pós-Graduação em Ciência da Computação

Folha de Aprovação

Defesa de Dissertação de Mestrado do candidato André Hallwas Ribeiro Alves, realizada em 05/04/2023.

Comissão Julgadora:

Prof. Dr. Ricardo Cerri (UFSCar)

Profa. Dra. Heloisa de Arruda Camargo (UFSCar)

Prof. Dr. Luiz Henrique de Campos Merschmann (UFLA)

O Relatório de Defesa assinado pelos membros da Comissão Julgadora encontra-se arquivado junto ao Programa de Pós-Graduação em Ciência da Computação.

Acknowledgements

Inicialmente agradeço a Deus e a minha família por tornar este momento possivel, pela resiliência e perseverança. Em especial aos meus pais, por me fornecer todo o apoio necessário no decorrer deste caminho. Agradeço ao meu orientador, o Ilutre Prof. Dr. Ricardo Cerri por me guiar, por nunca desanimar, sempre ser paciente, por acreditar em min, e me fornecer toda a orientação e apoio necessários durante esta longa jornada. Juntos passamos por um momento turbulento da história, mas isto não nos deixou desanimar. Agradeço aos meus colegas do DC e do BioMal, pelos momentos em que estivemos juntos, que mesmo poucos, desfrutei muito. Em especial ao Pedro, e a Elaine, dos quais me ajudaram muito no decorrer deste trabalho. Por fim, agradeço ao corpo docente do PPGCC e da UFSCar, que sempre esforçaram-se ao máximo para que este momento se tornasse realidade. Ressalto ainda, que o presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior-Brasil (CAPES) - Código de Financiamento 001.

Resumo

Dados de interação são obtidos por meio da observação e do registro das interações entre objetos. O uso de dados de interação possibilita a solução de diversos problemas complexos. Atualmente, existem várias maneiras de usar esses dados para produzir soluções, uma das quais é a previsão de novas interações a partir de interações já conhecidas. Para realizar essa tarefa, métodos de aprendizado de máquina podem ser usados. O estudo das interações entre objetos é importante em diversas áreas do conhecimento, como em sistemas de recomendação, na análise de interação em redes sociais, na indústria farmacêutica, e na bioinformática. O aprendizado de máquina é uma sub-área da inteligência artificial onde são desenvolvidos algoritmos com a capacidade de aprender e realizar tarefas automaticamente, por meio do treinamento e da exploração de um conjunto de dados previamente fornecido. Neste trabalho, desenvolvemos dois métodos baseados em Predictive Bi-Clustering Trees (PBCTs) para a predição de interações em conjuntos de dados de interações relacionadas as áreas de medicina e bioinformática. Destacamos que métodos multirrótulo baseados em uma abordagem global como PBCTs, podem prever todas as interações de um objeto em uma única tarefa de predição e explorar as relações entre os espaços de objetos. Inicialmente, construímos um modelo de aprendizado híbrido entre o PBCT e o XGBoost, onde no primeiro estágio o PBCT é usado na geração de partições na matriz de interação e, na segunda etapa, um modelo de aprendizado XGBoost é induzido em cada uma das partições, visando reduzir o desequilíbrio entre interações positivas e negativas nas predições resultantes. Dados de interações positivas indicam a ocorrência de uma interação, e dados de interações negativas indicam a não ocorrência, enquanto dados de interações desconhecidas (não rotulados) indicam casos onde não se tem informações sobre as interações. Com o objetivo de aproveitar os dados não rotulados no procedimento de indução do PBCT, propusemos uma adaptação na função split, transformando o PBCT em um método de aprendizado semi-supervisionado, podendo assim trabalhar com dados rotulados e não rotulados, com diferentes níveis de supervisão e desequilíbrio entre dados rotulados e não rotulados. Ambos os métodos introduzidos tiveram seu desempenho avaliado com base em critérios de avaliação que consideraram a eficiência preditiva e o desempenho computacional, considerando um estudo comparativo com o PBCT original. Com base nos resultados obtidos mediante procedimento experimental, ambos os métodos mostraram-se promissores, apresentando contribuições à literatura e abrindo caminho para o avanço do estado da arte.

Palavras-chave: Aprendizado de Máquina, Predição de Interações, Aprendizado Multirrótulo.

Abstract

Interaction data is obtained by observing and recording interactions between objects. The use of interaction data makes it possible to solve many complex problems. Currently, there are several ways to use this data to produce solutions. One of them is to predict new interactions based on existing interactions. Machine learning methods can be used to accomplish this task. The study of interactions between objects is essential in several areas of knowledge, such as recommendation systems, analysis of interactions in social networks, the pharmaceutical industry, and bioinformatics. Machine learning is a subarea of artificial intelligence where algorithms are developed with the ability to learn and execute automatically through training and exploration of a previously provided dataset. In this work, we developed two methods based on Predictive Bi-Clustering Trees (PBCTs) for the prediction of interactions in interactions datasets related to the areas of medicine and bioinformatics. We highlight that global approach-based multi-label methods such as PBCTs, can learn and predict all interactions of an object in a single task and explore the relationships between object spaces. Initially, we build a hybrid learning model between PBCT and Extreme Gradient Boosting (XGBoost), wherein the first stage PBCT is used to generate partitions in an interaction matrix. In the second stage, an XGBoost learning model is induced in each partition to reduce the imbalance between positive and negative interactions in outcome predictions. Data from positive interactions indicate the occurrence of an interaction, and data from negative interactions indicate that it did not occur, while data from unknown (unlabeled) interactions indicate cases where there is no information about interactions. To take advantage of the unlabeled data in the PBCT induction procedure, we propose a semi-supervised adaptation in the split function of the PBCT, thus being able to work with labeled and unlabeled data, with different levels of supervision and imbalance. Both introduced methods had their performance evaluated based on evaluation criteria that considered efficiency in predicting interactions and computational performance and through a comparative study with the original PBCT. As a result, both produced methods showed promising through an experimental procedure, presenting contributions to the literature and paving the way for the advancement of the state-of-the-art.

Keywords: Machine Learning, Interaction Prediction, Multi-label Learning.

List of Figures

List of Tables

List of algorithms

List of abbreviations and acronyms

AAC Amino Acid Composition

AUPRC Area Under the Precision-Recall Curve

AUROC Area Under the ROC Curve

BR Binary Relevance

CTF Conjoint triad feature

Clus Predictive Clustering Trees

CART Classification and Regression Trees

CC Classifier Chains

CV Cross-Validation

DTI Drug-Target interaction prediction

DDI Drug-Drug interaction prediction

ECCRF Ensemble Classifier Chains of Random Forest

ECC Ensemble Classifier Chains

GSO Global Single Output

GMO Global Multiple Output

GBDT Gradient Boost Decision Trees

HP-PPI Host-Pathogen Protein-Protein interaction

HH-PPI HIV1-Human Protein-Protein interaction

KNN K-Nearest Neighbors

LMO Local Multiple Output

MLD Multi-scale Local Descriptor

MCC Matthew's Correlation Coefficient

MLSKF Multilabel Stratified K-Fold

MLKNN Multilabel K-Nearest Neighbors

NB Naive Bayes classifier

PPI Protein-Protein interaction prediction

PCA Principal Component Analysis

PCT Predictive Clustering Tree

PWM Position Weight Matrix

PBCT Predictive Bi-Clustering Tree

PbXGB Predictive Bi-Clustering Trees with XGBoost

PbSS Semi-Supervised Predictive Bi-Clustering Trees

PbSSd Dynamical Semi-Supervised Predictive Bi-Clustering Trees

ROC Receiver Operating Characteristic

RF Random Forest

SVM Support Vector Machine

SMOTE Synthetic Minority Over-sampling

SIFT Scale-Invariant Feature Transform

UniAlign Protein structure alignment evolution

WELM Weighted Extreme Learning Machine

XGBoost Extreme Gradient Boosting

Summary

Chapter 1

Introduction

The introduction to this work is described in this chapter. The context and motivations of this research are presented in Sections 1.1 and 1.2. Section 1.3 defines our hypothesis and objectives with this work, while in Section 1.4, we show our main contributions to the literature. Finally, in Section 1.5, we present the organization of the other chapters of this document.

1.1 Context

Information about interactions is obtained by observing interactions between objects. This information is fundamental in several areas of knowledge and can enable the resolution of several complex problems. Among the several known applications for interaction data, we can highlight the Interaction Prediction that infers new interactions between objects through an inductive procedure with high confidence based on known interactions data. For example, in the field of biology, various tasks can use interaction information, such as Protein-Protein interaction prediction (PPI) and Drug-Target interaction prediction (DTI).

In interaction prediction between drugs and targets (proteins) (BAGHERIAN et al., 2020), the specific impacts caused by the interaction between drugs and targets are investigated, considering that these impacts alter the functions of targets (DING et al., 2013). Thus, the study of these interactions is fundamental in the investigation of drugs, being able to facilitate the drug discovery procedure (PLIAKOS; VENS, 2020; LEE; KEUM; NAM, 2019; EZZAT et al., 2016), assist in the prediction of side effects (ISLAM; HOSSAIN; RAY, 2021; GALEANO et al., 2020; PAUWELS; STOVEN; YAMANISHI, 2011), and the reuse of drugs (CHOI et al., 2020; SWAMIDASS, 2011; MORIAUD et al., 2011).

In the biological context, there are traditionally three ways to identify interactions between objects (RAO et al., 2014). *In Vitro*, interactions observations are carried out in a controlled environment, external to the living organism, where the experiment is guided to observe their interactions with other organisms (SILVA et al., 2020; BROWN et al., 2006); *In Vivo*, experiments are carried out inside the organism to observe their interactions (XING et al., 2016; SNIDER et al., 2015); *In Silico*, the observation of interactions is performed in a computational environment through simulations and estimates (SHASTRY; SANJAY, 2020; HAYES et al., 2016). *In Silico* interaction analysis is an area of focus of several recent studies in the literature, which is constantly growing. It has several methods to observe, identify and predict interactions, among which the machine learning methods stand out.

Interaction prediction with machine learning is an important focus today. Traditionally, machine learning methods take a dataset of previously known interactions as input and produce a model based on a learning function. This model can predict interactions on the same or another dataset in the same format through an inductive procedure. In this work, the interaction prediction problem was formulated as a multi-label machine learning problem, i.e., the prediction problem was defined by an interaction matrix that can be represented bipartite graph. In this context, we focus on a global approach-based multi-label machine learning method (global multi-label machine learning method), i.e., models that comprise the entire interaction matrix in the same learning procedure. Recently, a global multi-label machine learning method has emerged to perform interaction prediction tasks: the Predictive Bi-Clustering Trees (PBCTs) (PLIAKOS; VENS, 2020; PLIAKOS; GEURTS; VENS, 2018). This method works with the Bi-Clustering concept (DIAZ; PERES, 2019; PONTES; GIRÁLDEZ; AGUILAR-RUIZ, 2015; MADEIRA; OLIVEIRA, 2004) using datasets that can be defined by bipartite graphs for learning and have been shown to be efficient in complex prediction tasks.

1.2 Motivation

The study on interaction prediction using machine learning is currently applied in several areas of knowledge. Examples are medicine and bioinformatics, where algorithms are used to predict protein functions (RESENDE et al., 2012; CERRI et al., 2016; WEHRMANN et al., 2017), to predict interactions between proteins (PPI) (DING; TANG; GUO, 2016; ZAMIL; RAHMAN, 2018; CHEN et al., 2019a; CHEN et al., 2019b; BELTRAN; VALDEZ; NAVAL, 2019; DEY; MUKHOPADHYAY, 2019; LI et al., 2020), in Drug-Drug interaction prediction (DDI) (JIMENEZ; MOLINA; MONTENEGRO, 2019), in Drug-Target interaction prediction (DTI) (FATTAHI; REFAHI; MINAEI-BIDGOLI, 2019; NASUTION; WIJAYA; KUSUMA, 2019) and protein structure classification (WANG et al., 2008; CHENG; TEGGE; BALDI, 2008; SHAH, 2013; MANIKANDAN; RAMYA-

CHITRA, 2016; CHRYSOSTOMOU; SEKER, 2016).

Proteins are necessary for most cellular functions, such as DNA transcription and replication, metabolic cycles, and signaling cascades. In this context, the discovery of their interactions is fundamental, as it can help to identify their biological attributions in the cell and clarify their functions since proteins rarely perform their functions in isolation (DING; TANG; GUO, 2016; YOU et al., 2013).

One of the great advantages of the *In Silico* interaction prediction in the presented contexts is that it brings economy and agility in the discovery of new confirmed interactions since this task can be used to induce the experimental procedure *In Vitro* providing new interactions with a high possibility of interaction to be tested. Not only does this significantly reduce the cost of producing new drugs that rely on these interactions, but it also helps discover new treatments for diseases and understand the general interactions between proteins in an organism.

Due to current technological advances and the frequency with which new technologies are developed, there is a great increase in the amount of information collected to be processed, which is increasingly composed of more detailed data and more complex patterns to be identified (PLIAKOS; GEURTS; VENS, 2018; FATTAHI; REFAHI; MINAEI-BIDGOLI, 2019; CHEN et al., 2019b; PLIAKOS; VENS, 2020). These advances often generate new challenges, requiring improvement or the production of new methodologies for predicting interactions.

Often, three major challenges are noted in the literature (PLIAKOS; GEURTS; VENS, 2018). The first is the increasing scale of the data about the number of stored objects (cardinality); The second is the number of characteristics or attributes that describe these objects (dimensionality); The third is the structure used to represent objects (feature Vectors). In traditional learning problems, each object is represented by a vector of attributes, but representations with more complex structured data (WANG et al., 2018) are emerging and often generate the need to modify existing methodologies.

Regarding the learning process, several problems can be observed. One is the imbalance in the data (CHEN et al., 2019a; EZZAT et al., 2016). Some datasets may have few examples of positive or negative interactions. Unknown (unlabeled) interactions in the interaction matrix also tend to cause data imbalance, significantly impacting the learning process, as unknown interactions are often represented in the same way as negative interactions.

1.3 Hypothesis and Objectives

This work is based on the hypothesis that it is possible to obtain a better performance of prediction tasks by improving the Predictive Bi-Clustering Tree with a focus on improving efficiency and predictive performance or producing applicable variations to specific contexts. Nevertheless, the same foundation can be applied to other machine learning models used during this work.

The main objective of this research is to improve Predictive Bi-Clustering Trees with a focus on obtaining models with greater predictive performance, more specific to tasks, or applicable to similar contexts. For this, consolidated methods in the literature or applicable to similar tasks are the focus of this work. The following are also objectives of this work:

- Transform (Map) parts of the learning problem *Bi-Clustering* to other learning models to build a hybrid model with the ability to take advantage of the best features of both;
- Build and model interaction prediction databases using existing datasets to be used by machine learning algorithms;
- Observe which attributes and characteristics of the databases influence the performance of the methods;
- Check which is the best methods and strategies for modeling datasets based on the efficiency and predictive performance obtained by the classifiers;
- Evaluate efficiency, predictive performance and verify characteristics and limitations of prediction models;
- Use different evaluation metrics, verify their impact on the evaluation results and establish a comparative benchmark with the traditional models defined in the literature.

1.4 Contributions

The main contributions of this master's research were:

- The improvement and adaptation of the state-of-the-art Predictive Bi-Clustering Trees, through the development of hybrid methods, compatible with each other, with different approaches, and applicable to more specific contexts (i.e., Scenarios with a high level of imbalance between interaction data or with large amounts of unlabeled data);
- The development of a hybrid method combines features of Predictive Bi-clustering Trees and XGBoost (PbXGB) for global multi-label interaction prediction in scenarios with imbalanced data;

- The development of a hybrid method that combines features of Predictive Bi-clustering Trees and Semi-Supervised Learning (PbSS) to better take advantage of a large amount of unlabeled data (i.e., frequently observed in datasets) in the learning procedure;

- The comparative study between the developed methods related to the state-of-the-art Predictive Bi-Clustering Trees, and other traditional methods considering the computational and predictive performance, and evaluation criteria referring to the imbalance present in the predictions (i.e., between positive and negative interactions, labeled or not).

Publications resulting from this research work:

- ALVES, A. H. R.; CERRI, R. A two-step model for drug-target interaction prediction with predictive bi-clustering trees and XGBoost. IEEE 2022 International Joint Conference on Neural Networks (IJCNN), Jul 2022. (ALVES; CERRI, 2022)

- ALVES, A. H. R.; SILVA, P. C. I.; CERRI, R. Semi-supervised hybrid predictive bi-clustering trees for drug-target interaction prediction. In: Proceedings of the 38th ACM/SIGAPP Symposium on Applied Computing, Mar 2023. (ALVES; ILIDIO; CERRI, 2023)

Experiments related to both methods showed promise and presented a competitive performance in a general context. It is noteworthy that PbXGB performed better in a significant part of the experiments related to the data imbalance evaluation criteria, which indicates improved predictive performance on imbalanced partitions. PbSS, on the other hand, presented statistically significant results in some cases and has the advantage of switching between supervised and unsupervised.

1.5 Document organization

The remaining of this document will be structured as follows. In Chapter 2, the foundations of the methods used in this work are discussed; Chapter 3 presents the functioning of the developed methods in this work; Chapter 4 presents the necessary steps to carry out the experimental procedure, as well as Chapter 5 presents a comparative study of the results used in the validation of the methods. Finally, Chapter 6 presents this work's characteristics, disadvantages, conclusions, and future works. Additionally Appendix A presents the extended results of this work.

Chapter 2

Theoretical Fundamentation

In this chapter, the Interaction Prediction problem will be discussed from the viewpoint of a Machine Learning Problem. Section 2.1 presents our background; In Section 2.2, information regarding interaction data will be presented; in Section 2.3, possible divisions of the interaction matrix to work with prediction tasks will be presented; in Section 2.4, the multi-label machine learning is presented, while in Section 2.5, multi-label machine learning methods applicable to interaction prediction problems will be described; Finally, Section 2.6 will present how the Predictive Bi-Clustering Trees works, and Section 2.7 presents the XGBoost classifier.

2.1 Background

The interaction prediction literature has applications in several areas where Machine Learning methods have been explored. In this work, applications related to medicine and bioinformatics were considered. Within these areas, we highlight the tasks of Interaction Prediction between proteins (PPI), Interaction Prediction between drugs (DDI), Interaction Prediction between drugs and targets (DTI), and Host-Pathogen Protein-Protein interaction (HP-PPI). Table 1 presents the most recent work on interaction prediction problems in medicine and bioinformatics.

Zhao et al. (2017) computationally predicted HIV1-Human Protein-Protein interaction (HH-PPI) between the HIV-1 virus and human proteins based on the hypothesis that proteins with similar interface architectures share similar interaction partners. The protein-protein interface was defined as the contact region between two interacting proteins (ZHANG et al., 2010; BASPINAR et al., 2014). Initially, all interfaces extracted from the complexes described in the dataset were used. To obtain the structural similarity

Table 1 – Interaction Prediction Related Works

Article	Application
(ZHAO et al., 2017)	HP-PPI
(ZAMIL; RAHMAN, 2018)	PPI
(PLIAKOS; GEURTS; VENS, 2018)	PPI
(FATTAHI; REFAHI; MINAEI-BIDGOLI, 2019)	DTI
(JIMENEZ; MOLINA; MONTENEGRO, 2019)	DDI
(CHEN et al., 2019b)	PPI
(DEY; MUKHOPADHYAY, 2019)	HP-PPI
(LI et al., 2020)	PPI

between the interface architectures, the Protein structure alignment evolution (UniAlign) method (ZHAO; SACAN, 2015) was used. The Support Vector Machine (SVM) classifier was inducted and then used to predict the HH-PPIs. During experimentation on the training data set, a 10-fold Cross-Validation (CV) procedure was used to obtain the average result on the metrics used, and promising results were obtained. Subsequently, SVM was trained and used to identify new HH-PPI interactions between HIV-1 and human proteins, a possible new interaction was discovered, and it was concluded that the approach could produce promising results in the context of HH-PPI.

Zamil e Rahman (2018) explored several computational techniques of feature extraction and classification proposed for the prediction of PPIs and observed that several methodologies used in the prediction of PPIs use different information from proteins and techniques of classification. However, regardless of the methodology, extracting features from the data set is essential to improve the performance of the classification model. Then, they applied the Multi-scale Local Descriptor (MLD) (YOU; CHAN; HU, 2015) to extract attributes from the protein sequences in the dataset and used several classifiers in the prediction task of PPIs. The methodology showed promising results during experimentation.

Pliakos, Geurts e Vens (2018) explored the concept of multiple outputs in the Interaction Prediction and proposed a multi-label classification method with a global approach, where the Interaction Prediction is formulated as a multi-label classification task. This method uses decision trees with multiple outputs for structuring and predicting interaction data. Experiments were carried out on several sets of heterogeneous data, and better efficiencies and predictive performances were obtained in the proposed method concerning the other approaches with evaluated decision trees. It was concluded that the methodology obtained promising results and can be used, especially in cases where *White Box* interpretable classification models are needed.

Fattahi, Refahi e Minaei-Bidgoli (2019) noted that several approaches introduced for recognizing drug interactions are applicable, especially in homogeneous networks and bipartite models. Then, they proposed a methodology for the prediction of DTI, where the edge2vec (GAO et al., 2019) node embedding algorithm is used to represent the heterogeneous biomedical network in a low-dimensional space without loss of information. Then the SVM classifier is applied in the interaction prediction task. It was concluded through experimentation and comparison with other methodologies that the methodology obtains significantly higher performance.

A systematic review of the literature was carried out in the work of Jimenez, Molina e Montenegro (2019), where drug interactions (DDI prediction task) models and machine learning approaches employed in extracting drug interactions were observed. The study focused on the biomedicine area, identified the most used methods, summarized the most relevant approaches, and explored the outstanding challenges in predicting DDIs.

Chen et al. (2019b) found that although SVMs are often used in PPI tasks, their use is limited to problems with small datasets and low scalability due to the high computational cost involved. With this, a more efficient solution for estimating the hyper-parameters is presented. Furthermore, the classification was performed using GPU, obtaining greater efficiency and predictive performance.

Dey e Mukhopadhyay (2019) predicted interactions between dengue proteins and human proteins (HP-PPI prediction task). Initially, they investigated literature studies and observed that methods based only on the Amino Acid Composition (AAC) (ROY et al., 2009; RASHID; RAMASAMY; RAGHAVA, 2010; HASHEMIFAR et al., 2018) or on the Conjoint triad feature (CTF) of protein sequences (SHEN et al., 2007; WANG et al., 2017; WANG; WU, 2018) are often used in PPI prediction. Then, they proposed an approach for human-dengue PPI prediction based on protein sequence that combines AAC and the CTF. For the prediction of PPIs, several classifiers were used, such as SVM, K-Nearest Neighbors (KNN), and Naive Bayes classifier (NB), where the performance of the algorithms is evaluated through 10-fold CV procedure. After experimentation, the SVM was more accurate than the other methods, and it was demonstrated that the concatenation of AAC and the CTF produces better prediction results compared to the isolated use of the AAC or the CTF in the dataset used. It was concluded that the approach is promising and can be applied to other HP-PPI prediction tasks.

In Li et al. (2020), a computational methodology is proposed based on the Scale-Invariant Feature Transform (SIFT) algorithm (LOWE, 2004), and on the Weighted Extreme Learning Machine (WELM) (ZONG; HUANG; CHEN, 2013) called SIFT-WELM, to perform the prediction of PPIs. Initially, protein sequences are selected from the database and represented as Position Weight Matrix (PWM) (WANG et al., 2017; YI et al., 2018). Then the SIFT algorithm extracts attributes from the PWM matrix. Next, the Principal Component Analysis (PCA) (LUO et al., 2010; MORI; KURODA; MAKINO, 2016) is applied to the resulting data to reduce the scalability, and then the WELM is used to predict interactions. Experimental results indicated that data extraction with SIFT is efficient. Compared to methodologies based on SVM, the methodology can increase accuracy and reduce classification time. It was concluded through experimentation on several datasets that the methodology is feasible and robust.

2.2 Interaction Prediction

Interaction networks represent interactions between distinct objects (e.g., drugs, proteins, and diseases). A network of interactions can be defined by a graph, where objects are represented as nodes (vertices), and interactions are represented as edges. Each vertex can be described by a feature vector referring to the objects (i.e., information related to the drug, characteristics of the protein, and information about its structure), and each edge can be described by a feature vector referring to the interaction (i.e., knowledge of the conditions of interaction, unique behavior characteristics of objects in that interaction, and information about that interaction in another database).

Networks represented by unipartite or bipartite graphs can be defined by adjacency or bipartite matrices (Interaction Matrix) where positions denote interactions. Figure 1 illustrates an example of an interaction network where it presents a dataset with two object sets with a bipartite relationship and their interactions are represented by the arrows connecting both object sets. In this case, the interest values are the interactions between represented objects through the binary interaction matrix.

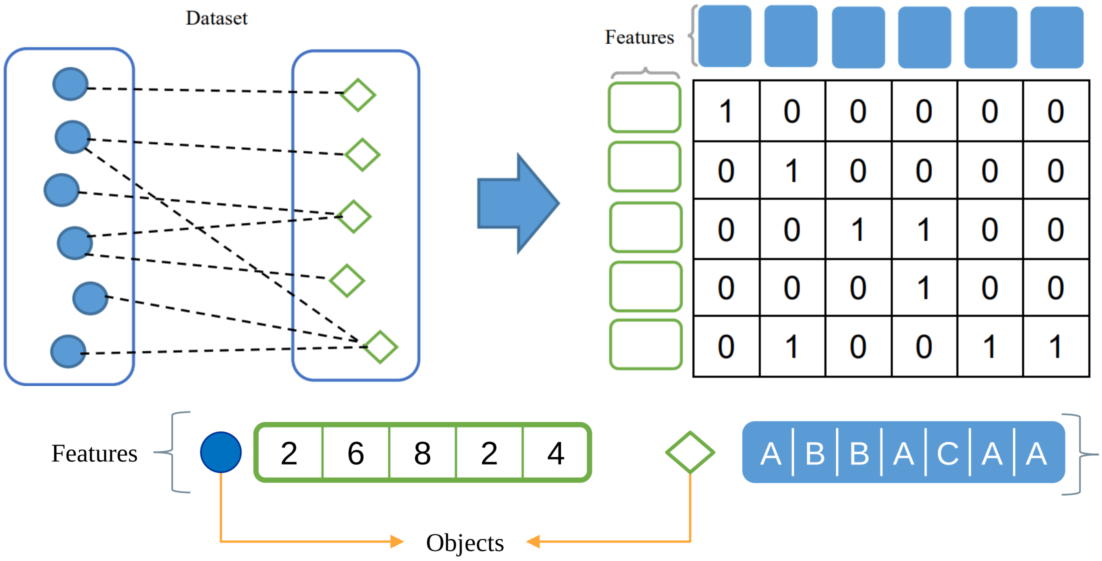


Figure 1 – Example of interaction between two object sets with bipartite relation and representation in a binary matrix.

In interaction networks, the Interaction Prediction can be defined as an edge inference problem between vertices, where through a deductive or inductive procedure, interactions (edges) between objects (vertices) present in the interaction network are inferred. Machine learning methods can be applied to interaction prediction tasks (SHASTRY; SANJAY, 2020; SARKAR; SAHA, 2019). These methods use the partial knowledge of the network based on the interaction matrix to experimentally adjust a learning model that, after obtaining the desired performance, can be used to predict unknown interactions (i.e., new interactions) in data of the same format.

In the context of machine learning, datasets may or may not be labeled. Objects in a labeled set have associated labels in addition to a group of characteristics that represent them (Features). Labels represent interactions between object pairs in a prediction task defined by a bipartite set and an interaction matrix. In unlabeled sets, only the features of the objects are known, not their interactions. Due to the ample representation space of an interaction matrix, unlabeled data is frequent and defined by object pairs whose interaction is unknown. Machine learning can be divided into several areas, including supervised and unsupervised learning (HERRERA et al., 2016a). In supervised learning, performance depends on partial knowledge of the interaction matrix (i.e., object pairs labeled as interacting or not). In unsupervised learning, pairs of labeled objects are unnecessary, as it is part of the method's operation to find the labels when needed.

Traditionally, two types of interactions are observed, positive and negative. Positive interactions refer to objects that interact with each other, while negative interactions represent objects that do not interact. In a binary interaction matrix, positive interactions are represented by the value 1, and negative or unknown (unlabeled) interactions are represented by the value 0. Ideally, there should be a balance between positive and negative interaction data samples in interaction prediction tasks. If the data are imbalanced, the generalizability of the learning model may be affected.

2.3 Prediction Tasks

Considering a supervised machine learning environment and an interaction matrix defined by two sets of objects represented by the rows r and columns c, we observe four possible prediction tasks to evaluate the predictions generated by the learning models (SCHRYNEMACKERS; KüFFNER; GEURTS, 2013; PLIAKOS; GEURTS; VENS, 2018). Taking L as the set of objects included in the learning procedure and T as the set of objects belonging to the test set, we get the following defined configurations: Predicting interactions (L_r x T_c) between row objects included in the learning set (rows from learning set L_r) and column objects belonging to the test set, invisible to the learning set (columns of the test set T_c); The Interaction Prediction (T_r x L_c) between row objects invisible to the learning set, belonging to the test set (rows from the test set T_r) and objects of columns included in the learning set (columns from the learning set L_c); The Interaction Prediction (T_r x T_c) between row and column objects available in the test set, invisible to the training set; finally, we can define the Interaction Prediction (L_r x L_c) between row and column objects present in the learning set, which represents the trivial case where all objects and their interactions are known by the learning method.

Figures 2 and 3 illustrate how these prediction tasks are organized into an interaction

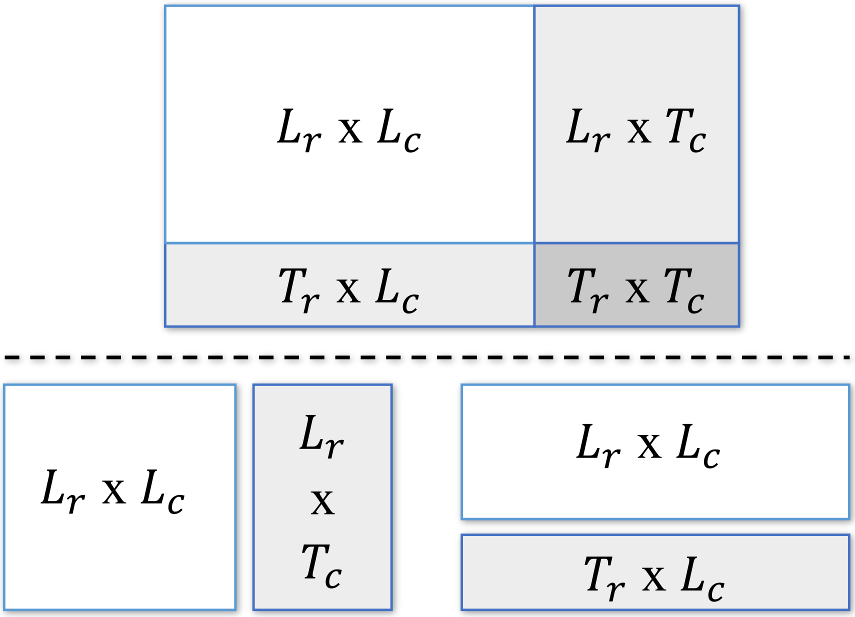


Figure 2 – Representation of visible and hidden pairs in the interactions matrix. In white are the nodes visible to the training set, and in gray are the hidden nodes of the test set to be predicted.

matrix. In particular, Figure 3 denotes the procedure referring to the prediction task T_r x T_c. Initially, the learning model is fitted on learning data and used to predict columns A) or rows B). Then, the resulting predictions are aggregated into the learning data to fit a new learning model to predict the missing interactions. e.g., in this step, in A), the model is induced on the generated sample and used to predict the row interactions and a similar procedure is carried out in B) to predict column interactions. Subsequently, the data resulting from the partial prediction T_r x T_c of A) and B) are combined, producing the final prediction for the task T_r x T_c.

The Interaction Prediction between invisible objects to the learning procedure (T_r x T_c) is a complex task, as it involves carrying out the inference procedure without the existence of observations or interactions. As a result, it tends to have a low success rate, as predictive performance depends on the availability and quality of data about interactions between available objects in the training set.

Both prediction tasks are common to the literature (SCHRYNEMACKERS; KüFFNER; GEURTS, 2013; PLIAKOS; GEURTS; VENS, 2018), but in the context of this work, the study, and gains in predictive performance for prediction tasks such as L_r x T_c, T_r x L_c and T_r x T_c may represent contributions to the literature (i.e., especially for T_r x T_c due to the difficulty of the task). In this work, disregarding the trivial case, we focus our contributions on the tasks L_r x T_c, T_r x L_c, and T_r x T_c, excluding the trivial case L_r x L_c.

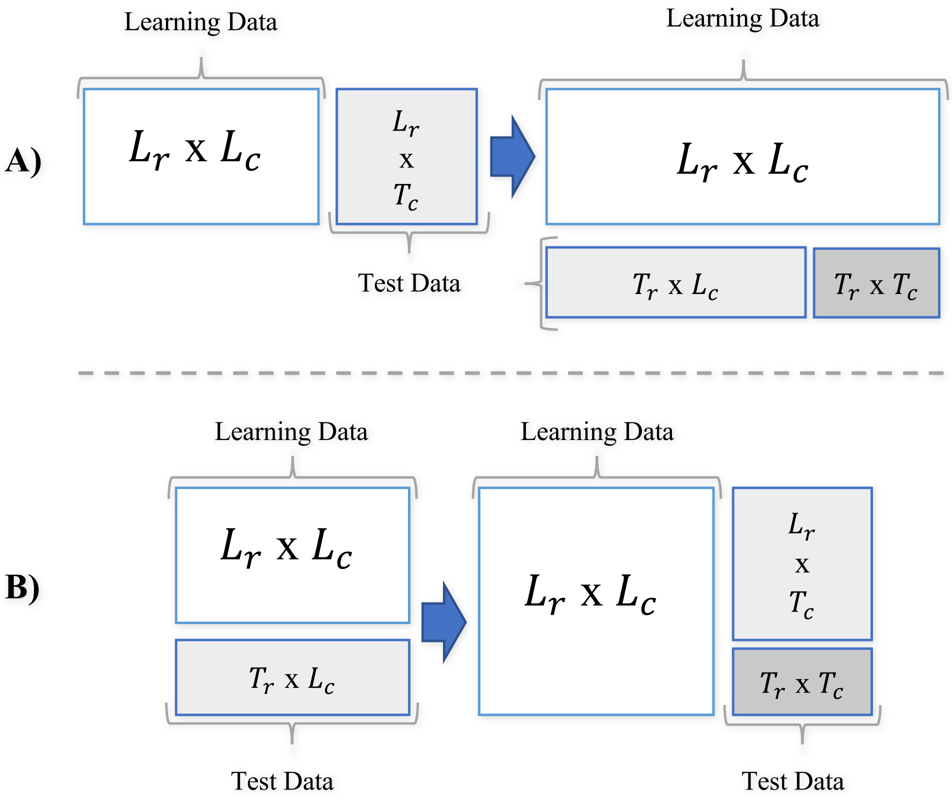


Figure 3 – Illustrative representation of how the interaction data are considered in the induction procedure of the learning algorithm. The first step of A) and B) denotes a simple L_r x T_c and T_r x L_c learning tasks, while the entire combined procedure of A) and B) denotes a T_r x T_c task.

2.4 Multi-label Machine Learning

As mentioned earlier (see Section 2.2), the machine learning area has several ramifications, including supervised and unsupervised learning (HERRERA et al., 2016b). Traditionally supervised learning defines a learning model based on a learning function fitted on data instances of objects that have been known and previously labeled. The learning model posteriorly can infer new object data instances' target values (labels), i.e., the learning model can use the previously known labeled data information to infer labels to new data that has never been seen before (HERRERA et al., 2016a).

Traditionally in single-label machine learning, each object is represented by a class value and a feature vector (Attribute Vector). In classification, the objective is to draw a hyperplane that separates the data and, through this division, infer to which class (label) the object belongs (HERRERA et al., 2016a). In regression, an estimation function is traditionally fitted to the dataset and subsequently used to estimate the values from the dataset and new data (HERRERA et al., 2016a). Despite their differences, the regression can also be used for classification, i.e., one way to use regression in this scenario is defining a threshold, where values above belong to one class and values below to another.

Different from traditional machine learning, where a single label (class) is assigned to a given data input (feature vector). In a multi-label machine learning scenario, each data instance can be simultaneously associated with several labels based on its features. In multi-label machine learning, the learning model is induced to predict multiple labels simultaneously, based on each data instance rather than a single class attribute (HERRERA et al., 2016b).

In this context, multi-label learning problems can be seen from the perspective of an extension of traditional problems (e.g., Binary or Multi-Class). However, a fundamental feature of multi-label learning is the ability to assign multiple labels simultaneously to a given data input instead of just one label.

These are some key concepts and characteristics of multi-label learning. Multi-label datasets fundamentally consist of instances of objects feature-composed with their corresponding labels belonging to the label space. Labels are often represented with a binary value (i.e., 0 or 1), indicating the existence or non-existence of the label. Several methods can be applied to multi-label learning problems, including converting multi-label problems into multiple single-label problems through problem transformation, adaptations of existing learning methods to employ the multi-label functionality on operation and ensemble learning strategies. Several algorithms have been proposed in the literature and can be used for multi-label learning, among these.

The Binary Relevance (BR) is a problem transform method that decomposes the Multi-Label problem into multiple binary classification problems to be solved individually and later aggregated into the Multi-Label solution (ZHANG et al., 2018). A learning model is induced for each class of the multi-label dataset, and finally, the Multi-Label prediction is obtained by aggregating the predictions of the generated learning models. One of the limitations of this method is that it disregards the relationships between the labels (i.e., because it works with them isolated), in addition to the high computational cost that can make the application unfeasible in some cases.

The Classifier Chains (CC) is another problem transform method. It works similarly to BR decomposing the Multi-Label problem into multiple binary classification problems (READ et al., 2011; READ et al., 2021). However, the difference is that it works with a chain of binary classifiers, where each classifier additionally considers the previous classifier chains predictions as features during the induction procedure. An advantage of this method is that it tries to capture the dependencies between the labels, thus taking advantage of their correlations. However, It is sensible to the data quality, data imbalance rates, and the order of the classifiers in the chain.

The Ensemble Classifier Chains (ECC) is an ensemble method and an extension of CC, whose objective is to take advantage of the diversity of several classifier chains to improve the performance of the learning model, considering that each chain can capture different perspectives and aspects and correlations between labels (READ et al., 2011). It generates an ensemble of chains whose order of labels is alternated. Its a highlighted advantage over CC is that it mitigates the bias that can be introduced through the initial order of labels of an isolated chain. It is also worth highlighting the increase in computational complexity as a bias and the sensitivity to imbalance in the data.

Predictive Clustering Tree (PCT) is a multi-label decision tree constructed similarly to Classification and Regression Trees (CART). In Predictive Clustering Trees, the tree structure is observed from the viewpoint of a hierarchy of clusters, where each tree node recursively partitions the data into subclusters. It stands out from the standard trees because it treats the variance and the prototype function as parameters that can be instantiated according to the learning task (VENS et al., 2008).

Multi-label machine learning is a relevant area with great challenges and constant evolution. The literature has applications in multiple areas of knowledge, especially in the biological area in PPI and DTI tasks. Among the adversities faced by multi-label learning are high dimensionality and label imbalance. They are being relevant to consider that the choice of algorithm is often linked to the specific characteristics and requirements of the learning problem. As previously mentioned in Chapter 1, In this research we investigated variations of the Predictive Bi-Clustering Tree (PBCT) to obtain models that are resilient to the adversities of multi-label learning (e.g., data imbalance). i.e., in general, the PBCT expands the concept of PCTs to Bi-Clustering interaction prediction tasks. A more in-depth description of how Predictive Bi-Clustering Trees (PBCTs) works is presented in subsequent sections.

2.4.1 Semi-Supervised Learning

Semi-supervised machine learning is an area of machine learning that combines features of supervised and unsupervised learning to discover patterns in data with scarce labeled information. One of the main goals of semi-supervised learning is to improve the learning model's performance by taking advantage of a large amount of available unlabeled data.

It is considered a case of weak supervision and is conceptually situated between supervised and unsupervised learning (ZHOU, 2021). Weak supervision is based on the idea of using auxiliary data sources (labeled or not) to produce weak labels (without human supervision) on a given data set. These weak labels are used in the induction of a supervised learning model, thus allowing the induction to occur without requiring an extensive manual data labeling activity (Human Supervision). In general terms, weak supervision is a set of learning techniques used to reduce the cost and time associated with data labeling and increase data scalability. Weak supervision methods have several positive characteristics, such as cost-effectiveness, efficiency, and the ability to use data sources not limited to labeled datasets in the learning model induction procedure.

Semi-supervised learning is generally designed to work on datasets with significant rates of imbalance between labeled and unlabeled data. Semi-supervised learning methods allow combining large amounts of unlabeled data with labeled data in the induction procedure. These methods are particularly relevant in cases where labeled data are sparse or have high rates of imbalance (i.e., between labeled and unlabeled data), as it can be difficult to induce a reliable learning model (ENGELEN; HOOS, 2019).

Semi-supervised learning is often needed in domains where labeled data is extremely difficult or costly, such as identifying protein interactions and discovering new drugs. In these cases, with assumptions about the data distribution, it is possible to take advantage of the unlabeled data to build a consistent and robust learning model (ENGELEN; HOOS, 2019; CAMARGO; BUGATTI; SAITO, 2020).

Several semi-supervised learning approaches are investigated in the literature. A salient factor is how to make assumptions about unlabeled data in the semi-supervised learning model. The way of making assumptions is the basis of a significant part of the algorithms in this area, and these models often adopt different approaches when making assumptions, explicitly or implicitly (LI; LIANG, 2019; DING; ZHU; ZHANG, 2015; ENGELEN; HOOS, 2019). Among the options observed in the literature, the smoothness and low-density hypotheses stand out (ENGELEN; HOOS, 2019).

According (ENGELEN; HOOS, 2019), the smoothness assumption states that if two samples $x, x' \in X$ are close in the input space, their labels y, y' are likely to be the same. The low-density assumption assumes that points in dense areas of the input space have the same label (i.e., due to the similarity). This way, the decision boundary must not pass through dense areas in the input space (the decision boundary must be in a low-density area). An advantage of the smoothness assumption in a semi-supervised environment is that it can be applied transitively to unlabeled data (labels can propagate transitively through related objects). When the decision threshold passes only in low-density areas, both assumptions can be satisfied (ENGELEN; HOOS, 2019).

Semi-supervised learning models are often conditional (i.e., they have specific application conditions). Concerningly the data, a necessary condition is that the marginal data distribution $p(x)$ over the input space contains information about the posterior distribution $p(y|x)$. In this case, unlabeled data can be used to obtain information about $p(x)$ and, consequently, about $p(y|x)$ (ZHOU, 2021). Otherwise, it may not be possible to improve the prediction accuracy by adding unlabeled data (ENGELEN; HOOS, 2019). Fortunately, a wide range of learning problems fit or can be adapted to suit this condition.

As mentioned before, one of the main advantages of semi-supervised learning is combining characteristics of supervised and unsupervised learning in a learning model capable of working simultaneously with labeled and unlabeled data and producing predictions with reasonable performance. A semi-supervised learning method can also favor model generalization, reducing overfitting and making it closer to real-world data (i.e., because can have access to a large amount of unlabeled data). In addition to making it possible to apply learning models in scenarios where labeling datasets is costly and inefficient.

However, despite this, several issues must always be considered during the construction of the learning model, such as how to make the aforementioned assumptions and the data quality. Some of the main disadvantages of semi-supervised learning are the time complexity (i.e., due to the need to model labeled and unlabeled data and incorporate supervised, unsupervised, and semi-supervised techniques into a single learning model); The learning model's performance depends on the balance between the quality and quantity of labeled data. (i.e., when labeled data is not comprehensive or representative, the model produced may have affected predictive performance); Another point refers to the difficulty in determining the level of supervision and the ideal amount of labeled data to be used to balance the performance and computational cost of the learning model. Semi-supervised models have the potential for overfitting in the absence of enough labeled data to drive unlabeled data; As mentioned earlier, semi-supervised algorithms often rely on assumptions made about data distribution, and these may not hold in real-world scenarios; The phenomenon of performance degeneration is another issue that makes it a great challenge to apply semi-supervised learning methods in real environments (LI; LIANG, 2019).

In summary, it is essential to note that semi-supervised approaches are not always applicable in context, and it is necessary to consider that adding unlabeled data may not result in performance improvement compared to other methods. Regardless, when applied in a proper context, a semi-supervised approach can result in a relevant or competitive performance gain with the advantages mentioned above (i.e., even if can become specific to the worked context).

2.5 Interaction Prediction as Multi-label Machine Learning Problem

In the context of Machine Learning, classification and regression can be used for prediction tasks. Considering that when new data are provided, both can infer whether or not an interaction occurs based on the learning model. As previously mentioned in Section 2.4, single-label prediction tasks assume that an object is associated with only one of two classes (Binary Classification) or more (Multiclass Classification) (SARKAR; SAHA, 2019). More complex problems closer to real life adopt the idea that an object can belong to several classes (Labels) simultaneously, i.e., the multi-label (PLIAKOS; GEURTS; VENS, 2018; HERRERA et al., 2016b; TSOUMAKAS; KATAKIS; VLAHAVAS, 2011; TSOUMAKAS; ZHANG; ZHOU, 2012) models. As aforementioned in Chapter 1, the interaction prediction problem is formulated as a Multi-label machine learning global approach-based problem in this work. In this context, two learning approaches can be applied, the local approach (SCHRYNEMACKERS; KüFFNER; GEURTS, 2013), and the global approach (PLIAKOS; GEURTS; VENS, 2018): In summary, the local approach consists of dividing the prediction problem into several minor prediction problems, where each predictor is responsible for deciding the interactions concerning a single object; and the global approach consists of using or adapting a machine learning algorithm so that the learning model applies to the entire dataset without the need for divisions. Both learning approaches are discussed in more detail in sequence, i.e., these approaches mainly relate to how learning models receive and use the input interaction data in the induction procedure.

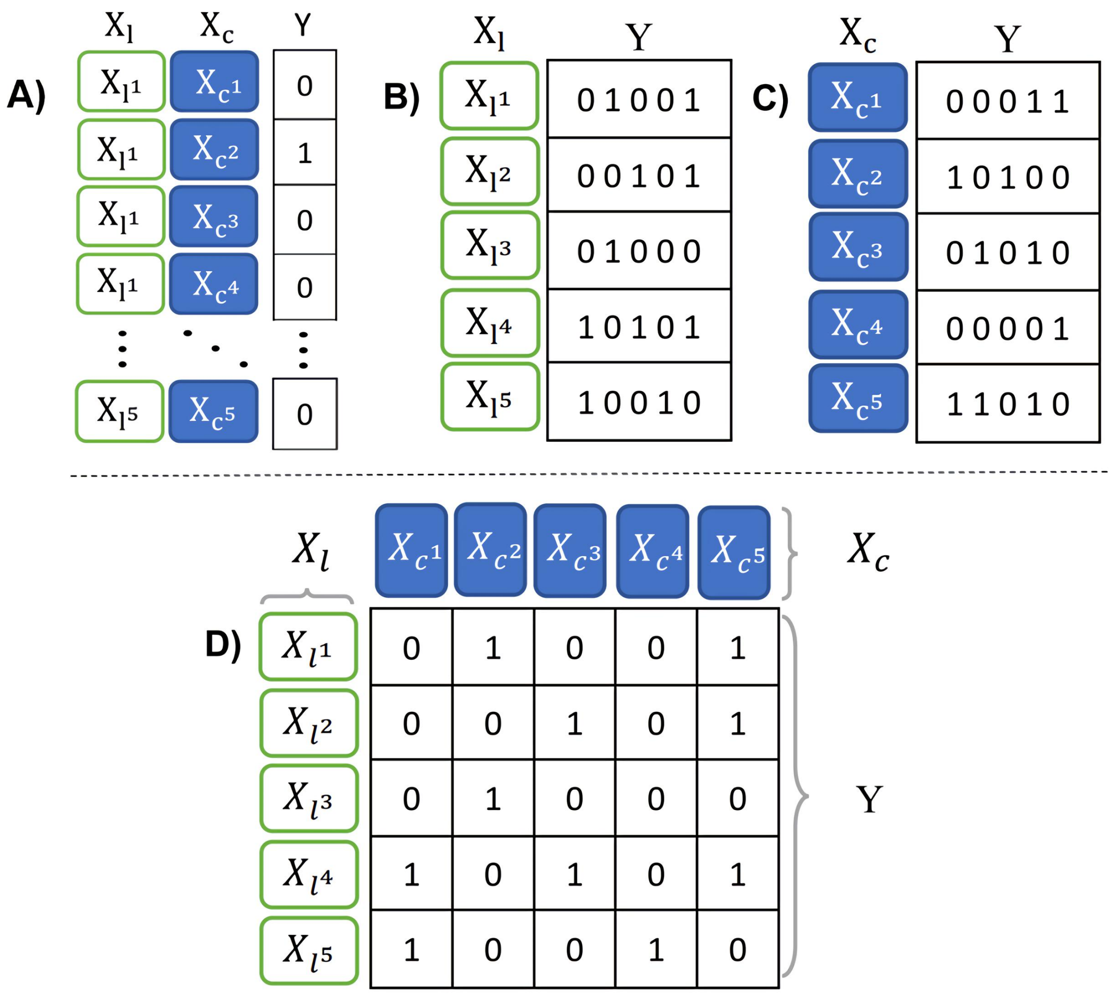
The Local Multiple Output (LMO) approach divides the classification problem into minor problems corresponding to all or just the interest objects, each defined by a sample of the learning dataset containing the correlated objects and interactions. After finishing the prediction task, the results accumulate, producing multiple outputs.

One way to apply this approach is to split the classification problem into two individual models. The first model is built on a sample of the training set referring to rows X_r to predict invisible row objects, and the second model is built on a sample referring to columns X_c, aiming to predict objects of the hidden column (SCHRYNEMACKERS; KüFFNER; GEURTS, 2013). Figure 4 (B, C) illustrates this approach, where in B, the sample of the training set referring to the rows of the interaction matrix is shown, and in C, the sample referring to the columns is shown. Each object is assigned a Y vector (labels vector) of size corresponding to all possible interactions in this way. The position Y_i of this vector receives the value one if the object interacts with another object corresponding to the position Y_i, and zero otherwise.

There are two global-based approaches, each suited to a different context: the Global Single Output (GSO) approach; and the Global Multiple Output (GMO) approach. The GSO approach applies a single classification algorithm to the learning sample. For this, the two feature vectors of the objects X_r and X_c of each interaction are concatenated (Cartesian product), and a Y binary value is added that indicates whether or not there is an interaction. Then a learning algorithm is inducted considering the entire dataset and later used to predict new interactions between visible or invisible objects in the learning process (Figure 4-A illustrates this approach).

The GMO approach consists of adapting or building a new classifier to produce a multi-output (i.e., multi-label) classification model applicable across the entire learning sample. The GMO approach has the advantage of not needing any modification to the data (e.g., like the Cartesian product) and of considering the correlations between rows and columns of the dataset's interaction matrix (i.e., we can see correlated relationships that do not exist when the learning problem is subdivided). About the LMO approach, it has the advantage of producing a single learning model, thus improving the interpretability of the learning procedure (Figure 4-D illustrates this approach).

In the context of this work, both approaches have their own characteristics and can be adequate to specific scenarios, but the GMO approach can be more flexible as it considers



X_l	X_c	Y
X_{l^1}	X_{c^1}	0
X_{l^1}	X_{c^2}	1
X_{l^1}	X_{c^3}	0
X_{l^1}	X_{c^4}	0
⋮	⋱	⋮
X_{l^5}	X_{c^5}	0

X_l	Y
X_{l^1}	01001
X_{l^2}	00101
X_{l^3}	01000
X_{l^4}	10101
X_{l^5}	10010

X_c	Y
X_{c^1}	00011
X_{c^2}	10100
X_{c^3}	01010
X_{c^4}	00001
X_{c^5}	11010

	X_{c^1}	X_{c^2}	X_{c^3}	X_{c^4}	X_{c^5}
X_{l^1}	0	1	0	0	1
X_{l^2}	0	0	1	0	1
X_{l^3}	0	1	0	0	0
X_{l^4}	1	0	1	0	1
X_{l^5}	1	0	0	1	0

Figure 4 – Approaches for supervised Interaction Prediction. A) Global Single Output Approach; B), C) Local Multiple Output Approach; D) Global Multiple Output Approach. Adapted from (ALVES; CERRI, 2022). X_l and X_c respectively represent the object feature spaces of the rows and columns, and Y the interactions.

all relationships between objects in the dataset. However, it often requires more time and computational resources as it performs the entire classification procedure in a single step.

2.6 Predictive Bi-Clustering Trees

Predictive Bi-Clustering Tree (PBCT) (PLIAKOS; GEURTS; VENS, 2018; PLIAKOS; VENS, 2020) is a global approach-based multi-label machine learning algorithm. It expands the concept of Predictive Clustering Trees (PCTs) presented by Vens et al. (2008) for the application and resolution of *Bi-Clustering* interaction prediction tasks. It is based on the structure of a Classification and Regression Trees (CART) constructed by simultaneously incorporating both label spaces (rows and columns) in the learning procedure. Each tree node contains objects that belong to both label spaces, i.e., partitioning the interactions matrix horizontally and vertically.

Considering a dataset S composed of an interaction matrix Y and two object feature spaces represented by Rows X_r and columns X_c. Where a feature vector represents each object belonging to the space of rows or columns so that i and j represent indices, and X_{ri} and X_{cj} features, a PBCT tree can be induced as defined below. The detailed functioning of the PBCT induction algorithm is defined in Algorithm 1, illustrated in Figure 5, and expanded in sequence (PLIAKOS; GEURTS; VENS, 2018).

Algorithm 1: Predictive Bi-Clustering Tree Induction.

```
Data: A dataset $S$ that consists of $Xr$, $Xc$, and $Y$;
Result: A global multi-output tree;
Function GMOT($S$):
1     $(t*, P*) \leftarrow BestTest(S)$
2     if $t* \neq none$ then
3         for $node\ S_k \in P*$ do
4             $tree_k \leftarrow GMOT(S_k)$
          end
5         return $node(t*, \cup_k\{tree_k\})$
6     else
7         return $leaf(Prototype(S))$
      end
8 return tree
Function BestTest($S$):
9     $(t*, h*, P*) = (none, 0, \varnothing)$
10    for possible test $t = t_r \cup t_c$ do
11        if $t \in t_r$ then
12            $P$ = horizontal partitioning of $S$ by $t$
          else
13            $P$ = vertical partitioning of $S$ by $t$
          end
14        $h = \left[Var(S) - \sum_{S_k \in p} \frac{|S_k|}{|S|} Var(S_k)\right] \frac{|S|}{S_{root}}$
15        if $h > h*$ then
16            $(t*, h*, P*) = (t, h, P)$
          end
      end
17 return $(t*, P*)$
Function Prototype($S$):
18    pt1 = columnwise average vector of leaf partition $S$
19    pt2 = rowwise average vector of leaf partition $S$
20    pt3 = setwise average of leaf partition $S$
21 return $(pt1, pt2, pt3)$
```

As defined in the function *GMOT* (Algorithm 1), at each tree node, the decision function *BestTest* is used in the selection of the label space X_r or X_c where the division of the interaction matrix Y will occur (split).

This function is based on the impurity reduction gain calculation (i.e., defined in line 14 of the Algorithm 1), which is performed on both label spaces, and the division with the best evaluation (highest value) will be selected (i.e., as defined in lines 15 and 16 of Algorithm 1). The tree splits occur top-down to the leaves, where the function indicates no more impurity reduction gain. In line 14 of Algorithm 1, the $Var(S)$ function (defined by Equation 1) represents the variances sum of the target variables of the objects in set (i.e., calculated for all possible tree splits) (PLIAKOS; GEURTS; VENS, 2018; VENS et al., 2008).

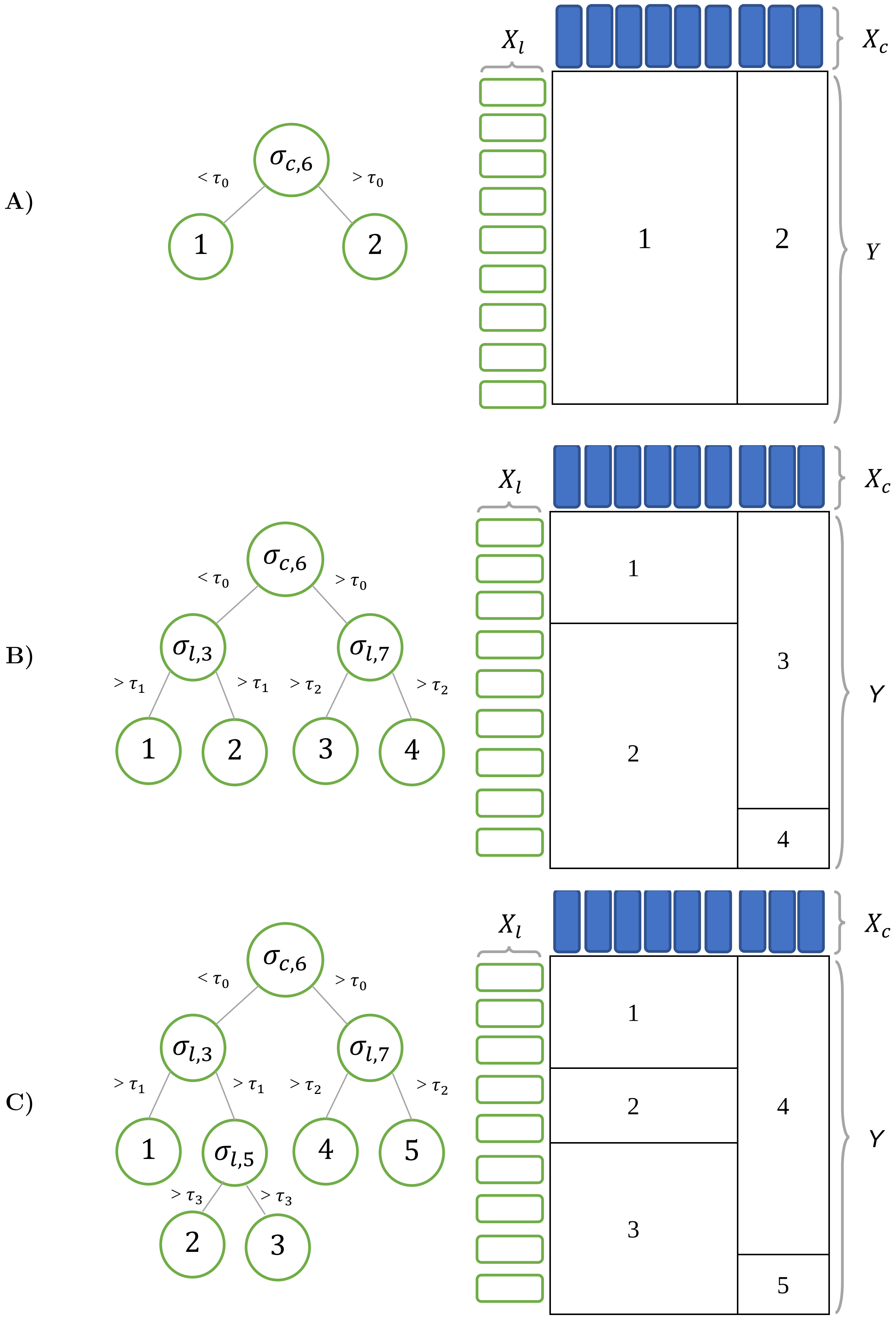


Figure 5 – An illustrative example of how Predictive Bi-Clustering Tree works. Each leaf represents a partition, and each tree node denotes a division in the interactions matrix, σ_n represent space divisions, c defines column divisions, l row divisions, and τ_n the thresholds. Adapted from Alves e Cerri (2022).

$$Var(S) = \sum_{i=1}^{T} Impurity(Y_i) \tag{1}$$

In particular, given input data (set of objects) S, in Equation 1 the $Impurity(Y_i)$ in $Var(S)$ function can denote any reasonable impurity measure. As here we are working with binary labels, the Gini impurity (see Equation 2) was chosen according to the procedure described by Pliakos, Geurts e Vens (2018), Vens et al. (2008).

$$Gini(E, Y) = 1 - \textstyle\sum_{i=1}^{C} {p_i}^2 \tag{2}$$

The Gini impurity can be calculated for input data E and a specific target variable Y as defined in Equation 2 where C are the possible values for the class Y (e.g., In the binary classification, C=2, and p_i is the prior probability of the class c_i). In the impurity reduction gain function (line 14 of Algorithm 1), considering the number $|S|$ of samples in the current node and the total number of samples $|S_{root}|$, the factor $\frac{|S|}{|S_{root}|}$ is used as split quality score in order to avoid cardinality bias (i.e., Otherwise, partitioning would tend to always occur in the same direction) (PLIAKOS; GEURTS; VENS, 2018).

In particular, Figure 5 illustrates the tree-growing of PBCT; in step A), we can see the first split of the tree on interaction matrix, defined by $\sigma_{c,6}$ when σ is a tree split, and $c, 6$ defines a split at the sixth index of the object space of columns, and τ_0 defines the threshold on the split-selected object features. This split results in two partitions, i.e., defined by 1 and 2 on the interaction matrix. In step B) we can see two more splits defined by $\sigma_{l,3}$ and $\sigma_{l,7}$ this time in the object space of rows, producing four partitions (i.e., 1, 2, 3, and 4) in the interaction matrix. Finally, in step C), we can see one more split on the rows object space defined by $\sigma_{1,5}$ in the place where partition 2 was in step B), i.e., in this way, dividing the partition space 2 of step B) into two new partitions, defined in this step by 2 and 3. The fact that partitions 1, 3, and 4 of Step B) did not produce new partitions in Step C) indicates that there was no additional gain in impurity reduction. Therefore these partitions are now represented as PBCT leaves.

The partitions (i.e., final samples of the interaction matrix present in the leaves of a function tree) contain the atomic divisions resulting from the induction procedure necessary for the inference procedure (i.e., prediction of new objects). The interpretation of partition data will differ depending on the prediction task. However, in both cases, it is based on the output pt (i.e., a single value or a vector of values) produced by a function f_p, considering p as the partition data, which will be used to guide the prediction (i.e. the final prediction value is based on the tree leaf partition data). This procedure can be seen in lines 18-21 of Algorithm 1 in the *Prototype* function, when $pt1$, $pt2$, and $pt3$ are the possible outputs to be used to guide the prediction task.

As defined in Section 2.3, excluding the trivial case (L_r x L_c), there are three prediction tasks, these being, T_r x T_c, L_r x T_c, and T_r x L_c. Considering Algorithms 1, and 2, the approach used in tasks T_r x T_c is to calculate the average of the values of the interaction matrix (i.e., partition) present in the leaf (i.e., setwise average of partition). In tasks, T_r x L_c, the function f_p produces the columnwise average vector of partition (i.e., a vector $v = \{avg(x_1), avg(x_2), ..., avg(x_n)\}$ containing the average value $avg(x_i) = \frac{1}{n}\sum_{j=1}^{n} x_{ij}$ of the j values of each partition column x_i, considering n as the number of column values), which is later used to find the prediction values. Tasks L_r x T_c are performed in the same way but with a rowwise average vector of partition (i.e., a similar procedure to that of columns, but with rows). Notably, Rows and Columns of an interaction matrix define objects belonging to different spaces but interacting with each other. In this way, a vector containing the average of the interaction values of one of the objects space defines for each index the general probability of interaction of this object with the others present in the partition. Thus, during the prediction, the final value of the prediction resulting from the selection of the value (i.e., referring to the index of the object defined as the partition cutoff point) of the vector of means represents the average probability of interaction between this object and the others in the referred partition.

Algorithm 2: Predictions with a PBCT Global Multi-Output Tree

```
Data: A global multi-output $Tree$ and an unseen pair of learning set $TestPair$.
Result: A prediction for an unseen pair $TestPair$
Function Predict($Tree$, $TestPair$):
1     $L$ = leaf node associated with $TestPair$
2     $(pt1, pt2, pt3) = Prototype(L)$
3     if $TestPair \in L_r \times T_c$ then
4         $j$ = row index of $TestPair$ in $pt1$
5         return $pt1[j]$
6     else if $TestPair \in T_r \times L_c$ then
7         $j$ = column index of $TestPair$ in $pt2$
8         return $pt2[j]$
      else
9         return $pt3$
      end
end
```

This procedure is demonstrated in Algorithm 2, where for tasks L_r x T_c, or T_r x L_c the row indices or columns of *TestPairs* are used together with the average vectors $pt1$ or $pt2$ in the search for the final prediction task value. The resulting value is assigned as a prediction result, i.e., the prediction is based on a $TestPair$, composed of two feature vectors of both objects whose interaction is being predicted. During the prediction step, a top-down search is performed for the leaf node in the tree corresponding to $TestPair$ (line 1 of the Algorithm 2). The resulting data (line 2 of the Algorithm 2), in turn, corresponds to the previously defined partition variance and is used to guide the procedure for obtaining the final result of the prediction task (lines 3-9 of the Algorithm 2).

2.7 Extreme Gradient Boosting

Extreme Gradient Boosting (XGBoost) (CHEN; GUESTRIN, 2016) is an ensemble, scalable, supervised machine learning method used for classification and regression tasks.

Ensemble learning strategies combine weak predictors (i.e., learning models) to build a strong predictor that performs the prediction together (e.g., XGBoost uses a technique called gradient boosting, which combines several weak decision tree models, creating an ensemble learning model).

XGBoost works by dividing the data into several subgroups and building a separate decision tree for each subgroup. Then, the algorithm sequentially builds decision trees (predictors) to predict residuals or errors of previous predictors (i.e., in order to fix the errors committed by previous trees). Finally, these predictors are added to the learning model and considered in the ensemble prediction procedure.

XGBoost applies the concept of boosting with the difference that it uses gradient descent to minimize the loss function when adding new predictors (CHRISTINELLI et al., 2021); Uses a set of Gradient Boost Decision Trees (GBDT) to improve performance and speed (BELTRAN; VALDEZ; NAVAL, 2019); and it is a gradient boosting method that employs regularization techniques to prevent overfitting and improve generalization, thus enhancing the performance of the learning model (CHEN; GUESTRIN, 2016); Moreover, finally, it uses an efficient tree pruning algorithm that reduces the complexity of the model while maintaining the level of accuracy (CHEN; GUESTRIN, 2016).

It has already been used in several prediction contexts in the literature (CHEN et al., 2019a; BELTRAN; VALDEZ; NAVAL, 2019) and has been shown to perform well in classification and regression tasks. In addition, it has been applied to a variety of learning problems. The works below use XGBoost for interaction prediction tasks in contexts similar to those studied in this research.

Chen et al. (2019a) investigated in the literature and observed that in the task of predicting HP-PPIs, there are several challenges related to representation algorithms and imbalanced datasets, so they proposed a two-layer structured model. In the first layer, the data imbalance rate is reduced using the XGBoost algorithm (CHEN; GUESTRIN, 2016) and the Synthetic Minority Over-sampling (SMOTE) (CHAWLA et al., 2002) technique. In the second layer, the interaction prediction is performed on the balanced produced dataset in the first layer using the SVM classifier. During the experimental phase, several rates of data imbalance and several classification methodologies were tested, and the results indicated that the methodology performs better compared to other similar methodologies described in the literature and traditional machine learning models in the task of HP-PPIs prediction.

Beltran, Valdez e Naval (2019) investigated the classification algorithms frequently used in the prediction of PPIs, and observed that recent approaches use *ensemble* to perform the classification task. Several classifiers are aggregated in *Ensemble learning* to produce a high-performance classifier. So, they proposed an approach using XGBoost for the interaction prediction. In this approach, interaction data were obtained from several databases and combined with several feature extraction methodologies frequently observed in the literature to generate sets of attributes representing interactions between proteins. Then, considering the data produced by the feature extraction methodologies, XGBoost is used for the interaction prediction task. Other classification methods were used during the experiment, demonstrating that the XGBoost approach produced better results than the other classifiers analyzed in predicting PPIs.

The main advantages of XGBoost are its speed, scalability, and flexibility: it is efficient and scalable, making it suitable for large-scale machine learning tasks with large real-world datasets and sparse data (CHEN; GUESTRIN, 2016); Fast, making it possible to train and test models quickly (CHEN; GUESTRIN, 2016); Furthermore, finally, it demonstrated significant performance in learning tasks on imbalanced datasets (BELTRAN; VALDEZ; NAVAL, 2019; CHEN et al., 2019a).

XGBoost has become a popular choice for machine learning applications because of its scalability and performance. It is an ensemble machine learning method that combines multiple decision tree models to create a more robust model. In summary, XGBoost is a promising algorithm that has been shown to be effective in predicting and classifying data.

Chapter 3

Proposed Methods

This chapter presents the methods proposed in this research. First, in Section 3.1 we presented a background of these methods. Section 3.2 presents a hybrid learning method, which combines Predictive Bi-Clustering Trees (PBCTs) and Extreme Gradient Boosting (XGBoost) called PbXGB to solve prediction problems with data imbalance. Then, in Section 3.3 we present the Semi-Supervised Predictive Bi-Clustering Tree (PbSS), a PBCT variation that combines supervised and unsupervised learning characteristics to make predictions considering labeled data and a large amount of unlabeled data simultaneously in the learning procedure. With the development of these methods, we aim to advance the state-of-the-art in their respective application areas.

3.1 Background

As previously defined in Section 2.6, PBCT is a machine learning algorithm that applies a method based on CART to generate partitions in the interaction matrix and, through them, performs predictive tasks. This algorithm has two fundamental pillars: the function responsible for performing divisions in the tree based on the sample of the interactions matrix; and how to use partitions to make predictions (i.e., what directly influences the learning model's performance). However, PBCT is sensitive to factors such as data imbalance, the amount of labeled and unlabeled data, and issues such as high dimensionality and cardinality. Thus, our work brings contributions on two fronts (i.e., in both concepts) through modifications in the tree division function and how partitions are used during the prediction procedure.

3.2 A hybrid model using Predictive Bi-Clustering Trees and XGBoost

Originally partitions produced by the induction procedure of a PBCT were directly used as the base for predictions. Despite this, other strategies can be applied. Here we propose to divide the learning process into two stages. As illustrated in Figure 6, in the first step, the original PBCT induction model procedure is used to make partitions in the interactions matrix. In the second step, an XGBoost classifier is fitted on each partition (i.e., based on the partition's data) (ALVES; CERRI, 2022).

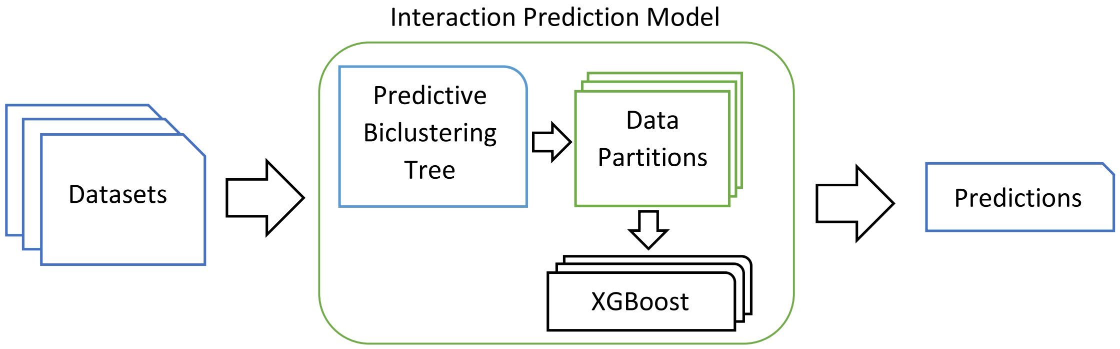
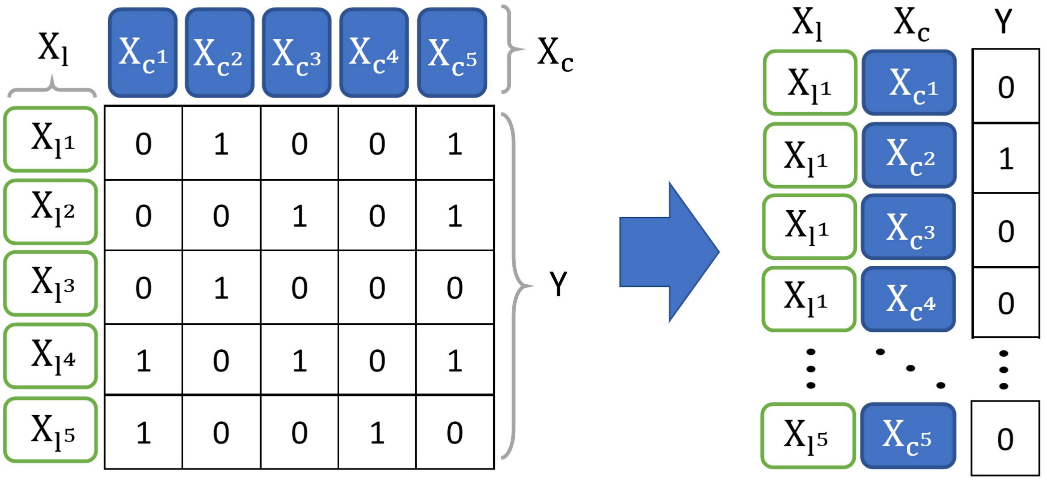


Figure 6 – The proposed interaction prediction model (ALVES; CERRI, 2022).

Partitions produced through the PBCT induction procedure are presented in multi-label interaction matrices format (i.e., the GMO input data format). However, each partition must be mapped to a GSO input data format (PLIAKOS; GEURTS; VENS, 2018) to be compatible with the XGBoost classifier induction procedure. In this conversion, all possible rows and columns pairs feature vectors are concatenated, and a binary value is added as a class indicating whether there is an interaction or not (see Figure 7).



	X_{c^1}	X_{c^2}	X_{c^3}	X_{c^4}	X_{c^5}
X_{l^1}	0	1	0	0	1
X_{l^2}	0	0	1	0	1
X_{l^3}	0	1	0	0	0
X_{l^4}	1	0	1	0	1
X_{l^5}	1	0	0	1	0

X_l	X_c	Y
X_{l^1}	X_{c^1}	0
X_{l^1}	X_{c^2}	1
X_{l^1}	X_{c^3}	0
X_{l^1}	X_{c^4}	0
⋮	⋱	⋮
X_{l^5}	X_{c^5}	0

Figure 7 – Mapping a partition to a Global Single Output (GSO) data format. Adapted from Alves e Cerri (2022).

In our proposal, we trained XGBoost classifiers using only the training data from partitions with a high rate of leaf imbalance, cases where the original PBCT can obtain low predictive performance (using prototype vectors). Density thresholds have been added to the model as a decision criterion for choosing in which imbalanced partitions XGBoost are inducted.

As shown in Figure 8, the density D represents the defined average value of a partition and P a range between two density values $\{D_{min} <= P <= D_{max}\}$. Density values range from 0 to 1. When the density value is close to 0, there are likely to be more negative interactions on the partition. When the density value is close to 1, we will likely have more positive interactions in the partition. Thus, when the density value is close to the limits (0 or 1), we use the original PBCT prediction strategy. However, in this work, we assume that the farther the density value is from the density limit values (thresholds), the greater the uncertainty about positive and negative interactions. In this scenario, our proposal improves predictive performance on leaf partitions with more impure interaction data that are more difficult to predict (ALVES; CERRI, 2022).

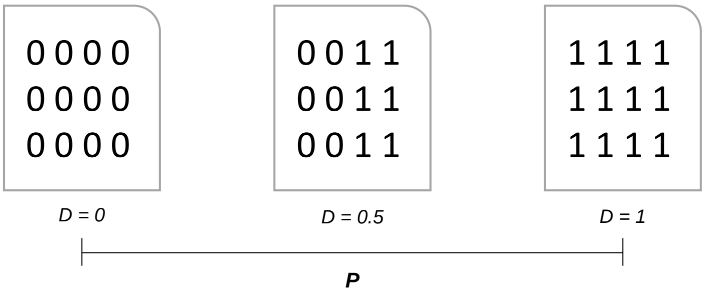


Figure 8 – Illustrations of the density interval (ALVES; CERRI, 2022).

More specifically with the density limits approach, we evaluated the relationship between uncertainty and unbalance in the partitions, so that the closer a partition is to the density limits, the greater the certainty about its label, and the further away, the greater the uncertainty regarding its label. Thus, density limits become a delicate balancing factor between predictive performance and computational performance. i.e., inducing an XGBoost learning model on partitions with higher certainty can degrade computational performance. It is worth noting that this approach is designed to work especially on unbalanced partitions, so as mentioned earlier there is a delicate balance between density limits (i.e., thresholds), e.g., if the density limits (D_{min} or D_{max}) are close to the limits (like 0 or 1, see Figure 8), they may consider partitions with high certainty about their label, and with little or no imbalance thus impacting the computational performance (due the XGBoost induction procedure), if the density limits are close they may disregard imbalanced partitions, impacting the predictive performance.

Originally, the PBCT induction algorithm could be divided into three main functions, as seen in Algorithm 1 (see in Section 2.6). The *GMOT* function is responsible for

Algorithm 3: Proposed algorithm to build leaves with a PBCT Global Multi-Output Tree

```
Data: A partition data $S$ that consists of $Xr$, $Xc$, and $Y$
Result: An adjusted learning model that consists in ($pt1$, $pt2$, $D$) or ($clf$, $D$)
Function Prototype($S$):
    $P$ = partition data
    $D$ = setwise average of leaf partition $S$
    if $I_{lim} <= D <= S_{lim}$ then
        $(X, y) \leftarrow$ partition data $P$ mapped to a single-label global format
        $clf$ = learn a XGBoost classifier
        return ($clf$, $D$)
    else
        pt1 = columnwise average vector of leaf partition $S$
        pt2 = rowwise average vector of leaf partition $S$
        return (pt1, pt2, $D$)
    end
end
```

Algorithm 4: Proposed algorithm for predictions with a PBCT Global Multi-Output Tree and XGBoost

```
Data: A global multi-output $Tree$ and an unseen pair of learning set $TestPair$.
Result: A prediction for an unseen pair $TestPair$
Function Predict($Tree, TestPair$):
    $L$ = leaf node associated with $TestPair$
    $D$ = density associated with $L$
    if $I_{lim} <= D <= S_{lim}$ then
        $clf$ = learned XGBoost model associated with $L$
        return prediction of $TestPair$ with $clf$ model
    else
        $(pt1, pt2, pt3) = L$
        if $TestPair \in L_r \times T_c$ then
            $j$ = row index of $TestPair$ in $pt1$
            return $pt1[j]$
        else if $TestPair \in T_r \times L_c$ then
            $j$ = column index of $TestPair$ in $pt2$
            return $pt2[j]$
        else
            return $pt3$
        end
    end
end
```

building the tree; The *BestTest* function is responsible for choosing the best tree split; and the *Prototype* function defines the values later used in the prediction.

Algorithm 3 introduces our approach. We mainly modified the original prototype function of Algorithm 1 (see in Section 2.6). Initially, the partition density D is obtained (i.e., as defined in step 2 of Algorithm 3). Then, if the density is within the specified limits, lower bound I_{lim} and upper bound S_{lim}, an XGBoost classifier is fitted to the leaf data (i.e., steps 4 to 6 of Algorithm 3). Otherwise (i.e., as seen in Algorithm 3 steps 7 to 9), the original PBCT strategy is used.

The original PBCT function (see Algorithm 2 in Section 2.6) is also modified for prediction. If the partition density is in the range defined by $\{I_{lim} <= D <= S_{lim}\}$ (Steps 2 and 3 of Algorithm 4), the XGBoost classifier clf associated with the leaf will be used to make predictions (steps 4 and 5 of Algorithm 4); otherwise, the original PBCT procedure is used (steps 6 to 13 of Algorithm 4).

3.3 Semi-Supervised Predictive Bi-Clustering Tree

In this method, we explore characteristics of semi-supervised learning to take better advantage of a large amount of unlabeled data present in the datasets to improve the PBCT divisions (splits) through a method based on the works of Levatić et al. (2017), and Alves, Ilidio e Cerri (2023), more specifically for scenarios with unlabeled data. Thus, we investigate the interaction prediction problem additionally from another viewpoint, from the perspective of an unlabeled data problem, specifically Positive-Unlabeled data (BEKKER; DAVIS, 2020; HAMMOUDEH; LOWD, 2020). The presented method simultaneously incorporates labeled and unlabeled data into the learning procedure. So, more specifically, we propose a variation of the semi-supervised impurity reduction function (i.e., adapted to Bi-clustering datasets and the PBCT) to improve the way impurity is evaluated in tree divisions (Splits) (step 6 from Algorithm 5).

Traditionally, in an interaction matrix, unknown or unlabeled interaction data are considered negative and represented by the value zero. However, depending on the case and the data quality, this can add a significant bias, which worsens due to the imbalance between the existing and the produced interactions. In this method, we added an unsupervised part to the impurity reduction function, similar to the procedure described by Levatić et al. (2017).

$$Impurity_{SSL}(E) = \underbrace{\frac{w}{T}.\sum_{i=1}^{T} Impurity(E_l, Y_i)}_{Supervised} + \underbrace{\frac{1-w}{D}.\sum_{i=1}^{D} Impurity(E, X_i)}_{Not\ Supervised} \quad (3)$$

As noted, Equation 3 references the *var* function (see Equation 1 in Section 2.6) of the PBCT tree splits function (see the step 6 from Algorithm 5), where the impurity

Algorithm 5: Predictive Bi-Clustering Tree Splits.

```
Data: A partition $S$ that consists of $Xr$, $Xc$, and $Y$;
Result: The best split for tree node;
Function BestTest($S$):
    $(t*, h*, P*) = (none, 0, \varnothing)$
    for possible test $t = t_r \cup t_c$ do
        if $t \in t_r$ then
            $P$ = horizontal partitioning of $S$ by $t$
        else
            $P$ = vertical partitioning of $S$ by $t$
        end
        $h = \left[Var(S) - \sum_{S_k \in p} \frac{|S_k|}{|S|} Var(S_k)\right] \frac{|S|}{S_{root}}$
        if $h > h*$ then
            $(t*, h*, P*) = (t, h, P)$
        end
    end
return $(t*, P*)$
```

function has two parts, a supervised part based on the Gini impurity criterion (Equation 4) (i.e., calculated from similar to the original PBCT algorithm) and an unsupervised part based on Equation 6. Each part of the equation (e.g., supervised and not supervised part) has weights to define its relevance (i.e., defined by w value).

In the Equation 3, E represents the split input data (i.e., containing labeled or unlabeled examples), E_l is a sample containing only the labeled examples, Y_i represents each T target attribute, X_i represents each of the D descriptive attributes, and w is a weighting criterion ranging from 0 to 1 to define the relevance of each part (supervised or unsupervised). When the value is 1, only the supervised part is considered; When the value is 0, only the unsupervised part is considered; and when the value is between 0 and 1, both parts will be considered according to their relevance.

Considering a labeled set E_l and the target attributes Y_i, the labeled impurity is calculated as defined in Equation 4, where E_l^{train} represents the labeled dataset from the root of the tree.

$$Impurity(E_l, Y_i) = \frac{Gini(E_l, Y_i)}{Gini(E_l^{Train}, Y_i)} \tag{4}$$

The unlabeled impurity only applies to the descriptive characteristics of the unlabeled examples. In this work, the unsupervised term of the semi-supervised impurity function is calculated once for each tree split (i.e., which brings more computational performance, since it is used only as a decision option for selecting the split-axis, Vertical or Horizontal). The unlabeled impurity for a partially labeled dataset E and the numerical descriptive attributes X_i are calculated as defined in Equation 5.

$$\underbrace{Impurity(E, X_i) = \frac{Var(E, X_i)}{Var(E^{Train}, X_i)}}_{Numeric\ Attributes} \tag{5}$$

Where E_{Train} represents the dataset at the tree's root. The impurity Var of the attribute i-th in the set E over the descriptive attributes of the feature X_i is calculated as defined in Equation 6. When N represents the number of values of a given feature described by X_i.

$$Var(E, X_i) = \frac{\sum_{j=1}^{N}(x_{ij})^2 - \frac{1}{N}.(\sum_{j=1}^{N} x_{ij})^2}{N} \tag{6}$$

A major advantage of this approach is the supervision control of the impurity function because both unlabeled data and unsupervised impurity can negatively affect the learning model's performance. Therefore, the proposed algorithm can obtain a performance equal to or better than the state-of-the-art, depending on the value of w. Thus, in scenarios where the semi-supervised impurity negatively affects the learning performance, we can set $w = 1$ and work only with the supervised part (i.e., similar to the original PBCT procedure). On the other hand, by setting $w = 0$, we can only work with the unsupervised impurity, making this a hybrid model between supervised and unsupervised learning (LEVATIĆ et al., 2017).

3.3.1 Dynamic Weights

Traditionally, we can work with the w weight in two ways, inferring values in the range of 0 and 1 or using a heuristic function. In this work, we also aim at contributions by determining the level of supervision. In this case, we work in both directions, globally defining the level of supervision, and through a heuristic function to determine the value of w as defined in Equation 7.

$$Weight(Y) = 0.1 + 0.9(\frac{\sum_{i=1}^{n}\sum_{j=1}^{m} Y_{ij}}{n.m}) \tag{7}$$

Equation 7 considers an interaction matrix Y, the number of row objects n, and the number of column objects m. Consider that each partition can represent a different learning problem, i.e., with different imbalance rates between positive, negative, labeled, and unlabeled interactions. When determining the w value dynamically, we assume a w value that automatically adjusts to each training partition. With this, in addition to not needing to statically assume the value of w in each learning problem, we also open an alternative path for solving learning problems where the static definition of the supervision criterion w tends to present low performance. Note that Equation 7 does not comprise the value 0. This occurs because labeled examples (i.e., in this case, represented by the value 1) are necessary for the induction of the proposed semi-supervised model.

Chapter 4

Methodology

This chapter presents the experimental validation of the proposed methods and the materials, procedures, and methods necessary to carry out the evaluation. Section 4.1 presents the datasets used and their characteristics, while Section 4.2 details the computational environment and the tools used in developing the methods and executing the experimental procedure. Section 4.3 presents the evaluation measures used and their characteristics. Finally, Section 4.4 presents the experimental parameters used during the experimental procedure.

4.1 Datasets

This work focused on interactions in the field of medicine and bioinformatics, and related datasets were used in the accomplished experiments (PLIAKOS; GEURTS; VENS, 2018; SCHRYNEMACKERS et al., 2015). Table 2 presents the general details and characteristics of the used datasets (i.e., the dataset names, the number of rows and columns of the interactions matrix, and the percentage of positive and validated interactions). Below, the datasets are described in detail.

Table 2 – Dataset characteristics

Dataset	Rows x Columns	Ratio of positive/negative interactions
DPI-N	26 x 54	90/1404 (6.4%)
DPI-G	95 x 223	635/21185 (3%)
DPI-I	204 x 210	1476/42840 (3.4%)
DPI-E	664 x 445	2926/295480 (1%)

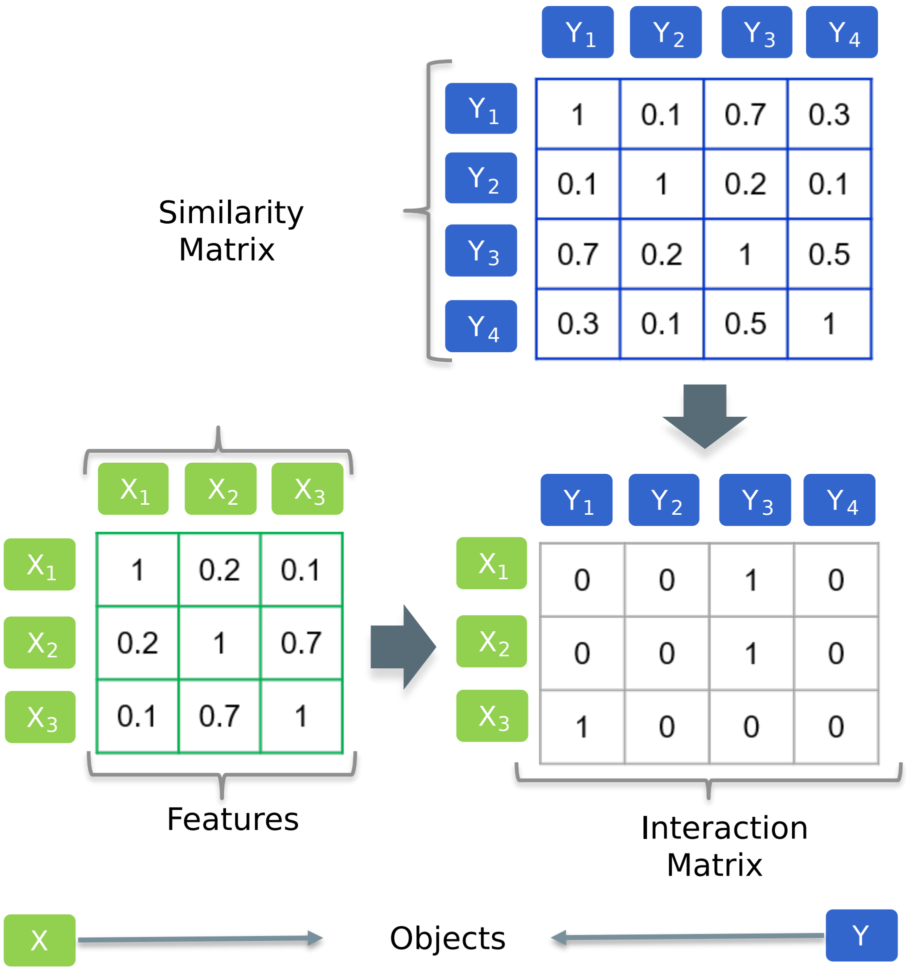
The Gold-Standard *Drug–Protein interaction networks* defined by Yamanishi et al. (2008) consists of 4 heterogeneous bipartite drug-protein interaction networks (DPI)[1]: Enzymes (DPI-E), ion channels (DPI-I), GPCR (DPI-G), and nuclear receptors (DPI-N).

As noted by Alves e Cerri (2022), enzymes are groups of organic substances, mainly proteins, that act as biocatalysts, accelerating metabolic reactions in organisms (ROBINSON, 2015). Ion channels are protein molecules that allow the passage of ions between the extracellular and intracellular environments through membranes (BARKER et al., 2017). G protein-coupled receptors (GPCRs) are a group of proteins used by cells to detect extracellular signals and molecules and activate intracellular responses. They mediate much of our physiological responses to neurotransmitter hormones and, consequently, responses to sight, smell, and taste signals (ZHAO et al., 2016). Nuclear receptors are ligand-activated transcription factors involved in many human biological aspects (ZHAO; ZHOU; GUSTAFSSON, 2019). They regulate vital functions, serving as stimulus sensors and regulators of molecular events, and often, when deregulated, they are associated with various diseases (FRIGO; BONDESSON; WILLIAMS, 2021).

For the construction of these datasets, Yamanishi et al. (2008) used several data sources, among them KEGG BRITE, BRENDA, SUPER TARGET, and DrugBank (THAFAR et al., 2021; PAHIKKALA et al., 2014). Both feature vectors of the datasets are composed of similarity matrices in both spaces, e.g., we can build similarity matrices in different ways (LIU et al., 2015; DING et al., 2013), between proteins, one can use the score produced by amino acid sequence alignment algorithms (LIU et al., 2015), and between drugs, it is possible to assess the similarity between their chemical composition (LIU et al., 2015; YAMANISHI et al., 2008). In this case, the similarity of the compounds chemical structure was calculated using the SIMCOMP algorithm (PAHIKKALA et al., 2014; HATTORI et al., 2010; HATTORI et al., 2003), and the similarity between the target protein sequences was produced using the normalized version of the Smith-Waterman Score (SSW) (LIU et al., 2015; DING et al., 2013) amino acids sequence alignment algorithm. Chemical structure data were obtained from the KEGG LIGAND and KEGG DRUG datasets, and target amino acid sequences were obtained from the KEGG Genes (THAFAR et al., 2021) database. According to Pahikkala et al. (2014), SIMCOMP represents two-dimensional chemical structures as graphs and calculates the similarity between compounds based on the size of common substructures between two graphs using the Jaccard coefficient. Figure 9 illustrates the data composition in the datasets.

These datasets are comprehensive, have been extensively explored in the literature (LIU et al., 2015; DING et al., 2013; PAHIKKALA et al., 2014), and represent a large space of different characteristics and scenarios, including different amounts of

[1] Yamanishi et al. (2008) datasets are available at: URL <http://web.kuicr.kyoto-u.ac.jp/supp/yoshi/drugtarget/> [Accessed 28 04. 2023])



	Y_1	Y_2	Y_3	Y_4
Y_1	1	0.1	0.7	0.3
Y_2	0.1	1	0.2	0.1
Y_3	0.7	0.2	1	0.5
Y_4	0.3	0.1	0.5	1

	X_1	X_2	X_3
X_1	1	0.2	0.1
X_2	0.2	1	0.7
X_3	0.1	0.7	1

	Y_1	Y_2	Y_3	Y_4
X_1	0	0	1	0
X_2	0	0	1	0
X_3	1	0	0	0

Figure 9 – An illustrative example of the organization of similarity data. The Feature vectors are each composed of a similarity matrix between the objects of each domain. Adapted from (PLIAKOS; GEURTS; VENS, 2018; ALVES; ILIDIO; CERRI, 2023).

interactions, rates of imbalance between interaction data, noise, and data quality (YAMANISHI et al., 2008; PAHIKKALA et al., 2014) in addition to having a large amount of unlabeled data, and can represent a Positive-Unlabeled scenario (BEKKER; DAVIS, 2020; HAMMOUDEH; LOWD, 2020) (i.e., where only positive interactions data have confirmed labels). The study of DTI in these scenarios can mean significant advances in discovering new drugs and treating severe diseases.

4.2 Development Environment

Figure 10 presents the development environment designed for this work. As can be seen, this environment incorporates several factors, such as programming languages, programs, frameworks, operating systems, and the hardware used. The primary program-

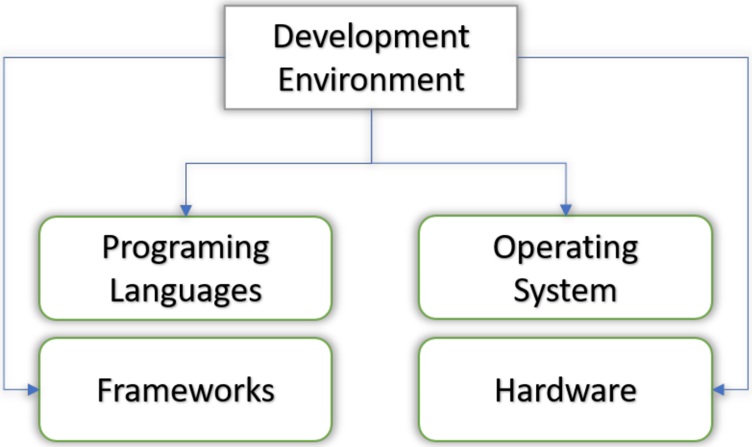


Figure 10 – The ilustration of our development environment

ming language used throughout this work was Python[2] (ROSSUM; DRAKE, 2009), and all implementations focused on this language. With some exceptions where external frameworks implemented in Java[3] (ARNOLD; GOSLING; HOLMES, 2000) were incorporated. Python is a powerful programming language frequently used by the machine learning community, having an arsenal of related libraries implemented and consolidated in the literature. On the other hand, Java has been a reference programming language for several years and has some of the most popular tools in the machine learning area. Therefore, it was used in the context of this work in some cases where it was necessary to use external frameworks implemented in that language (e.g., Scikit-Learn[4] and Clus[5]). The code was implemented in a desktop environment and is available on all platforms compatible with Python and Java. The experiments were carried out mainly on the servers of the Bioinformatics and Machine Learning Group (BioMal) and on the Cluster provided by the Universidade Federal de São Carlos (UFSCar) (special care was taken with the benchmark data, produced in the same environment, in the BioMal servers). The hardware used in this work comprises Intel(R) Xeon(R) CPU E5-2630 v3 @ 2.40GHz, 94 GB of RAM Memory, with Ubuntu Linux.

4.3 Evaluation Criteria

To evaluate the performance of the predictive models in this work, the K-Fold CV (GOMEZ; NOBLE; RZHETSKY, 2003; SACCÀ et al., 2014; BELTRAN;

[2] Python programming language (Python Software Foundation, Available at: URL <https://www.python.org/>[Accessed 11 12. 2022]).

[3] Java programming language (Oracle, Available at: URL <https://www.java.com/> [Accessed 11 12. 2022]).

[4] Scikit-Learn (Python Framework, Available at: URL <https://scikit-learn.org/> [Accessed 28 04. 2023])

[5] Clus: A Predictive Clustering System (Java Framework, Available at: URL <https://dtai.cs.kuleuven.be/clus/> [Accessed 28 04. 2023])

VALDEZ; NAVAL, 2019), and the Multilabel Stratified K-Fold (MLSKF) CV procedure (SECHIDIS; TSOUMAKAS; VLAHAVAS, 2011) were used. Metrics were used as evaluation criteria, such as Matthew's Correlation Coefficient (MCC) (CHICCO; TöTSCH; JURMAN, 2021; CHICCO; JURMAN, 2020; ZHU, 2020; SARKAR; SAHA, 2019; BOUGHORBEL; JARRAY; EL-ANBARI, 2017) (Equation 8), the Receiver Operating Characteristic (ROC) curve, and Area Under the Precision-Recall Curve (AUPRC) (BOYD; ENG; PAGE, 2013). These metrics are widely explored in the literature and were used in several works, such as (RESENDE et al., 2012; SACCÀ et al., 2014; HUANG et al., 2015; DEY; MUKHOPADHYAY, 2019; BELTRAN; VALDEZ; NAVAL, 2019; LI et al., 2020). The MCC is an efficient criterion to evaluate the balanced performance prediction of binary classifiers (i.e., when both classes have the same weight). It considers the balanced proportions of all prediction results (e.g., TN, FN, TP, and FP) and can resist and correctly measure imbalanced results (CHICCO; TöTSCH; JURMAN, 2021; SARKAR; SAHA, 2019). While for a threshold-independent predictive performance representation, the Area Under the ROC Curve (AUROC) and AUPRC curves can be performed (SARKAR; SAHA, 2019).

$$MCC = \frac{TP*TN-FP*FN}{\sqrt{(TP+FN)*(TN+FP)*(TP+FP)*(TN+FN)}} \tag{8}$$

In Equation 8, True Positive (TP) is the number of objects that belong to a class C_i and were predicted to belong to the class C_i; True Negative (TN) is the number of objects that do not belong to a class C_i, and were predicted as not belonging to the class C_i; False Positive (FP) is the number of objects that do not belong to a class C_i, and were predicted as belonging to the class C_i; False Negative (FN) corresponds to the number of objects that belong to a class C_i, and were predicted as not belonging to the class C_i; *Recall* is the true positive rate (i.e., the terms in Equation 8 are linked to class, e.g., the value of TP is linked to the reference class C_i, so it can also be interpreted as TP_i); Finally, the corresponding AUROC and AUPRC curves will be generated to evaluate the methodology's performance.

4.4 Experimental Parameters

As discussed in the Section 2.3, the interaction matrix was divided into three prediction tasks during the experimental procedure: Lr x Tc, Tr x Lc, and Tr x Tc. These divisions were performed within a K-Fold Cross-Validation (CV) procedure for both tasks. In the tasks L_r x T_c, the CV is applied in the columns, while in the tasks T_r x L_c, it is applied in the rows of the interaction matrix. For the Tr x Tc task, CV is applied to rows and columns by excluding a row fold and a column fold from the learning set and using their combined interactions as a test set (PLIAKOS; GEURTS; VENS, 2018).

For the experiments, ten folds were considered for the Lr x Tc and Tr x Lc tasks and five folds for the Tr x Tc task due to the low rate of positive samples in the folds and the method complexity. Fixed density limits were used by PbXGB in this work, were $\{0.02 <= P <= 0.98\}$.

During the XGBoost induction procedure, the same hyperparameters were used for the tasks Tr x Lc and Lr x Tc, considering *number of estimators* $= 300$, *learning rate* $= 0.1$ and *scale pos weight* $= 1000$. However, due to the complexity of the task Tr x Tc, we use a slight variation of these hyperparameters, considering *learning rate* $= 1000$ and *number of estimators* $= 10$. The hyperparameters used for the induction of PbSS were *minimum number of features* $= 4$, *minimum number of samples per leaf* $= 20$, and *maximum level* $= 10$. In the impurity function (see Equations 3 and 4 in Section 3.3), we consider $E_l = E$, i.e., in this case, all data are considered to be the labeled sample E_l and unlabeled sample E with the main difference that in the unlabeled part the labels are disregarded. This is mainly attributed to the worked Positive-Unlabeled scenario since the labeled data is scarce and imbalanced. However, this work also considered a scenario with a MLSKF CV procedure for both methods (i.e., PbXGB, and PbSS).

A Grid-Search procedure was also used to estimate the hyperparameters in some scenario. For the Grid-Search procedure, we considered the *number of splits* $= 5$ for the inner CV procedure. The hyperparameters considered for the XGBoost in PbXGB with Grid-Search procedure for L_r x T_c and T_r x L_c predicion tasks are *number of estimators* $= \{10, 300, 600\}$, *learning rate* $= \{0.02, 0.1\}$, and *scale pos weight* $= \{10, 100, 1000, rate\}$. For T_r x T_c prediction task a sight variation of this hyperparameters was used, considering *number of estimators* $= \{5, 10, 20\}$, and *learning rate* $= \{0.02, 0.1, 0.5, 100, 1000\}$, and *scale pos weight* $= \{100, 1000, rate\}$. The rate (α) is defined by $\alpha = \frac{\max\{x,y\}}{\min\{x,y\}}$, considering x as the number of negative interactions and y the number of positive interactions in a partition, and is a heuristic imbalance measure of the partition of the interaction matrix.

Data referring to the efficiency of the model were recorded during the tests for later comparison. In particular, the times in seconds related to induction, prediction, and the entire learning procedure were recorded. Finally, after applying our proposal, the predictions were evaluated using the AUPRC, AUROC, and MCC evaluation measures.

Chapter 5

Experimental Validation and Discussion

The present study and proposed methods were evaluated through a comparative analysis with the state-of-the-art PBCT (i.e., considering that gains in computational performance and predictive performance imply advances in the state-of-the-art). Nevertheless, Appendix A contains more in-depth results (i.e., where the comparative study considers adjacent methods and methodologies). To evaluate the results, we considered comparative analyses, predictive performance (i.e., based on the previously mentioned in Section 4.3 evaluation criteria), computational efficiency analysis (i.e., concerning induction and prediction times), and statistical analysis based on the Wilcoxon signed rank test (paired samples) with Bonferroni Correction. Thus, in Section 5.1, we present the synthesis of the obtained results from the experimental procedure; In Section 5.2, we present the predictive performance results; in Section 5.3, we present the comparison between computational efficiency; and finally, in Section 5.4, we present the statistical analysis.

5.1 Development Evolution

As mentioned above, this section presents a synthesis of how the experimental procedure was conducted, as well as the way tables and figures were built. The figures and tables mentioned in this section are presented in the respective sections. As presented and discussed by Alves e Cerri (2022) and arranged in the tables below, the experimental results from the comparative study between the Original PBCT and the PbXGB are promising. The performance of PbXGB in this scenario was evaluated according to the previously established criteria (sections 4.3 and 4.4), these being the predictive performance (i.e., in terms of AUPRC, AUROC, and MCC), computational performance (i.e., concerning induction, prediction, and total procedure times of the learning model), and statistical analysis (i.e., performed using the Wilcoxon signed-rank Test (Paired Samples) (ROSNER; GLYNN; LEE, 2005; WOOLSON, 2008; REY; NEUHäUSER, 2011; TAHERI; HESAMIAN, 2012) with Bonferroni correction (NAHLER, 2009), considering all evaluation criteria, and previously defined prediction tasks).

In this context, Tables 3, 9, and Figure 11 A) present the results obtained by the method presented by Alves e Cerri (2022). Tables 3 and 9 are presented and discussed respectively in Sections 5.2 and 5.3, while details regarding Figure 11 A) are presented in Section 5.4. Nevertheless, a more in-depth study is presented and discussed in this work. Furthermore, this study considered two additional aspects, considering the application of a MLSKF CV procedure (SECHIDIS; TSOUMAKAS; VLAHAVAS, 2011) and the Grid-Search to estimate the XGBoost Hyperparameters in the second stage of the PbXGB learning procedure. The results produced by these experiments were displayed in Table 4 and Figure 11 B), both discussed in more detail respectively in Section 5.2, and 5.4. An extended study is presented in Appendix A.2, considering a MLSKF CV procedure with static XGBoost parameters (see Table 11).

With regard to PbSS, Tables 5 and 6 (discussed in Section 5.2), and Figure 12 (discussed in the Section 5.4) provide the results that were obtained by (ALVES; ILIDIO; CERRI, 2023). This study contemplated the application of PbSS in two different scenarios, defining the w parameter dynamically or assigning the value zero (i.e., evaluating the impact of the unsupervised part in the learning procedure). The experimental procedure was constructed similarly to PbXGB, considering a comparative study between the Original PBCT and both variations of PbSS, considering a K-Fold CV evaluation procedure for all learning tasks and the previously presented evaluation criteria. This work presents a more comprehensive study, including a MLSKF CV procedure and an analysis of the model's computational performance. In this way, Tables 7 and 8 discussed in the Section 5.2 define the results obtained through an experimental procedure for dynamic w (PbSSd) and $w = 0$, Table 10 discussed in Section 5.3 presents the computational performance of the models, while Figure 13 discussed in Section 5.4 displays the statistical analysis for both methods. Nevertheless, an extended study is presented in Appendix A.3, considering variations of the value of $w = \{0.25, 0.5, 0.75\}$, and computational performance, e.g., being references to Tables 12, 13, and 14 respectively.

In Appendix A, a comprehensive comparative study regarding the Global Single Output (GSO) (Appendix A.4) and Local Multiple Output (LMO) (Appendix A.5) approaches was also presented. Considering previously established evaluation criteria, the same scenario was considered for both approaches, with a MLSKF CV procedure for all learning tasks. In addition, the Grid-Search procedure was performed to establish the Hyperparameters of the models when valid. Considering GSO, the learning models Clus

(Table 15), XGBoost (Table 16), Random Forest RF (NANDI; AHMED, 2019; ZAMIL; RAHMAN, 2018) (Table 17), and the KNN (Table 18). Considering LMO, the learning models Clus (Table 19), Multilabel K-Nearest Neighbors (MLKNN) (ZHANG; ZHOU, 2005; ZHANG; ZHOU, 2007) (Table 20), and Classifier Chains Ensemble (Random Forest) (ECCRF) (READ et al., 2011; ROCHA; VAREJÃO; SEGATTO, 2022) (Table 21).

5.2 Predictive Performance

In this section, the predictive performance of the learning methods defined in the course of this work will be presented. In the presented tables, the fields marked in bold indicate the best results obtained, the areas indicated with ± represent the standard deviation, and the fields signalized with $(*)$ indicate statistical difference gains to the proposed methods concerning the original PBCT (p <= 0.05).

As mentioned, the predictive performance was evaluated according to previously established evaluation criteria (AUPRC, AUROC, and MCC) through a comparative study against the Original PBCT and initially through a K-Fold CV procedure for all prediction tasks.

Table 3 – Results for Evaluation Measures obtained for compared methods (ALVES; CERRI, 2022), considering PBCT and PbXGB.

Measure	Data	L_r x T_c		T_r x L_c		T_r x T_c	
		PBCT	PbXGB (*)	PBCT	PbXGB (*)	PBCT	PbXGB
AUPRC	DPI-N	0.203 (± 0.102)	**0.398 (± 0.136)***	0.236 (± 0.222)	**0.291 (± 0.203)**	0.208 (± 0.137)	**0.224 (± 0.176)**
	DPI-I	0.197 (± 0.074)	**0.220 (± 0.075)**	0.458 (± 0.061)	**0.582 (± 0.080)***	0.047 (± 0.022)	**0.056 (± 0.022)***
	DPI-G	0.184 (± 0.048)	**0.237 (± 0.054)***	0.273 (± 0.078)	**0.373 (± 0.138)***	0.048 (± 0.021)	**0.081 (± 0.040)***
	DPI-E	0.169 (± 0.054)	**0.196 (± 0.062)**	**0.614 (± 0.069)**	0.587 (± 0.083)	**0.018 (± 0.012)**	0.011 (± 0.004)
		PBCT	PbXGB	PBCT	PbXGB	PBCT	PbXGB
AUROC	DPI-N	0.575 (± 0.090)	**0.661 (± 0.071)***	**0.631 (± 0.155)**	0.593 (± 0.113)	**0.543 (± 0.145)**	0.504 (± 0.124)
	DPI-I	**0.684 (± 0.061)**	0.606 (± 0.055)	**0.805 (± 0.042)**	0.794 (± 0.054)	**0.546 (± 0.038)**	0.545 (± 0.054)
	DPI-G	**0.658 (± 0.065)**	0.623 (± 0.039)	**0.703 (± 0.058)**	0.698 (± 0.076)	0.576 (± 0.065)	**0.579 (± 0.059)**
	DPI-E	**0.693 (± 0.051)**	0.627 (± 0.045)	**0.828 (± 0.042)**	0.817 (± 0.051)	**0.548 (± 0.052)**	0.508 (± 0.013)
		PBCT	PbXGB (*)	PBCT	PbXGB (*)	PBCT	PbXGB
MCC	DPI-N	0.117 (± 0.144)	**0.318 (± 0.156)***	**0.196 (± 0.237)**	0.193 (± 0.222)	**0.044 (± 0.138)**	0.000 (± 0.000)
	DPI-I	**0.216 (± 0.059)**	0.195 (± 0.092)	0.441 (± 0.061)	**0.600 (± 0.077)***	**0.036 (± 0.029)**	0.035 (± 0.041)
	DPI-G	0.196 (± 0.068)	**0.249 (± 0.053)***	0.243 (± 0.079)	**0.426 (± 0.134)***	0.052 (± 0.040)	**0.056 (± 0.039)**
	DPI-E	0.182 (± 0.055)	**0.234 (± 0.071)**	0.433 (± 0.068)	**0.578 (± 0.089)***	**0.025 (± 0.026)**	0.009 (± 0.013)

Initially, Table 3 presents the results obtained for PbXGB in the study by Alves e Cerri (2022). The experiment that resulted in this table considered all evaluation criteria and datasets and showed the results compared to the Original PBCT. As can be noticed, promising results were obtained by AUPRC and MCC in the tasks L_r x T_c and T_r x L_c, i.e., in the context of this work, the AUPRC and MCC evaluation criteria can be considered as a focus, considering that the objective was to obtain balanced results. Competitive results were obtained for AUROC, scenarios where PBCT stood out with slight differences for most tasks. It is noteworthy that for MCC, it is visible that PbXGB obtained a relevant performance gain for tasks L_r x T_c and T_r x L_c. Significant gains for MCC show that XGBoost improved performance on imbalanced partitions produced by the PBCT induction procedure in this case (ALVES; CERRI, 2022). The robust performance gains for AUPRC achieved by PbXGB indicate a better balance between accuracy and recall in predictions (ALVES; CERRI, 2022). Considering tasks T_r x T_c, it is possible to observe that competitive results were obtained for most of the evaluation criteria, scenarios where PBCT stood out more for AUROC and MCC, and PbXGB for AUPRC. Regarding the standard deviation (i.e., fields marked with ±), in a general context, it is possible to observe that both methods are in equilibrium (i.e., they present a similar situation, with slight variation when related in the general context). In a more specific context, it is possible to highlight that both methods have a high standard deviation rate (i.e., which can be attributed to several variables, such as the imbalance between positive and negative interactions in the evaluated cases or data noise), e.g., in this case, the results obtained by DPI-N, considering all the prediction tasks and the AUPRC and MCC evaluation criteria, stand out.

Following the same context, this work presents a more comprehensive study. The Grid-Search is considered to estimate the XGBoost hyperparameters (i.e., applied in the second stage of PbXGB) and a MLSKF CV procedure for evaluating learning models. Considering that each partition of the interactions matrix represents a new learning problem (i.e., with features such as imbalance rate and different data quality), estimating the XGBoost hyperparameters on each leaf during the learning procedure avoids the need to use arbitrary and global hyperparameters (i.e., so that all aspects of XGBoost are partition-adjusted during the induction procedure). The use of the MLSKF CV procedure occurs with a focus on providing a more robust validation method and, at the same time, providing more balanced folds (i.e., trying to avoid folds with only samples of negative interactions, which may imply a reduction of the high rate of standard deviation observed in the results).

Table 4 – Results for Evaluation Measures obtained for compared methods, considering PBCT and PbXGB with Grid Search and MLSKF CV Procedure.

Measure	Data	*Lr* x *Tc*		*Tr* x *Lc*		*Tr* x *Tc*	
		PBCT	PbXGB (*)	PBCT	PbXGB (*)	PBCT	PbXGB
AUPRC	DPI-N	0.247 (± 0.083)	**0.433 (± 0.132)***	0.213 (± 0.105)	**0.409 (± 0.227)***	**0.252 (± 0.179)**	0.122 (± 0.127)
	DPI-I	0.210 (± 0.071)	**0.282 (± 0.076)***	0.474 (± 0.065)	**0.580 (± 0.089)***	**0.056 (± 0.021)**	0.048 (± 0.011)
	DPI-G	0.246 (± 0.039)	**0.273 (± 0.057)**	0.270 (± 0.046)	**0.327 (± 0.077)**	**0.050 (± 0.025)**	0.040 (± 0.017)
	DPI-E	**0.163 (± 0.050)**	0.135 (± 0.049)	**0.604 (± 0.058)**	0.587 (± 0.069)	**0.026 (± 0.010)**	0.015 (± 0.005)
		PBCT	PbXGB	PBCT	PbXGB	PBCT	PbXGB
AUROC	DPI-N	0.634 (± 0.067)	**0.679 (± 0.070)**	0.649 (± 0.103)	**0.655 (± 0.121)**	**0.570 (± 0.140)**	0.518 (± 0.106)
	DPI-I	**0.709 (± 0.081)**	0.644 (± 0.058)	**0.807 (± 0.033)**	0.788 (± 0.042)	**0.549 (± 0.040)**	0.535 (± 0.028)
	DPI-G	**0.714 (± 0.022)**	0.637 (± 0.034)	**0.713 (± 0.046)**	0.661 (± 0.043)	**0.561 (± 0.077)**	0.538 (± 0.046)
	DPI-E	**0.694 (± 0.065)**	0.585 (± 0.034)	**0.830 (± 0.036)**	0.816 (± 0.038)	**0.566 (± 0.036)**	0.527 (± 0.025)
		PBCT	PbXGB (*)	PBCT	PbXGB (*)	PBCT	PbXGB
MCC	DPI-N	0.217 (± 0.100)	**0.364 (± 0.149)***	0.237 (± 0.146)	**0.336 (± 0.236)**	**0.061 (± 0.131)**	0.033 (± 0.132)
	DPI-I	0.247 (± 0.080)	**0.276 (± 0.086)**	0.441 (± 0.051)	**0.608 (± 0.089)***	0.037 (± 0.031)	**0.040 (± 0.031)**
	DPI-G	0.267 (± 0.026)	**0.304 (± 0.070)**	0.269 (± 0.066)	**0.419 (± 0.089)***	**0.045 (± 0.055)**	0.040 (± 0.049)
	DPI-E	**0.185 (± 0.069)**	0.168 (± 0.055)	0.452 (± 0.029)	**0.595 (± 0.059)***	**0.029 (± 0.016)**	0.021 (± 0.020)

Thus, the results of this procedure can be observed in Table 4 (i.e., in these experiments, the MLSKF CV procedure was also considered for the PBCT). Initially, it is possible to observe that in a general context, the results arranged in Table 4, with some caveats, do not present a very different scenario from that observed in Table 3. However, it is worth mentioning that we obtained significant gains for DPI-N T_r x L_c for all evaluation criteria and that the use of MLSKF CV procedure significantly improved not only the performance of the PbXGB but also made PBCT performance more consistent. On the other hand, it is also possible to observe that in some cases, the procedure also reduced the performance of PbXGB, e.g., for almost all T_r x T_c tasks (i.e., in this case, performance gains and losses are also affected by adjusting XGBoost hyperparameters with Grid-Search). However, it is worth noting that in a general context, the results remain competitive (i.e., in balance with gains and losses in the learning model performance). The same is reflected in the standard deviation of the results, which, except for some cases (e.g., PBCT DPI-N considering all evaluation criteria for the task T_r x L_c), in general, showed no significant increase or decrease.

Regarding PbSS, Tables 5 and 6 present the results obtained in the study by Alves, Ilidio e Cerri (2023). The experimental scenario considered in this study considered all previously mentioned evaluation criteria and a K-Fold CV procedure was performed for each learning task. Furthermore, this study took into account two variations of the supervision criterion w (i.e., w = 0, and Dynamic w), evaluating not only the influence of the unsupervised part of the semi-supervised impurity function (Table 6) but also the impact of a w heuristic supervision criterion, dynamically defined on each partition (Table 5).

Table 5 – Results for Evaluation Measures obtained for compared methods (ALVES; ILIDIO; CERRI, 2023), considering PBCT and PbSSd with dynamic w.

Measure	Data	*Lr* x *Tc*		*Tr* x *Lc*		*Tr* x *Tc*	
		PBCT	PbSS	PBCT	PbSS	PBCT	PbSS
AUPRC	DPI-N	0.216 (± 0.115)	**0.302 (± 0.169)**	0.259 (± 0.255)	**0.360 (± 0.315)**	**0.162 (± 0.145)**	0.139 (± 0.186)
	DPI-I	**0.202 (± 0.086)**	0.176 (± 0.088)	0.456 (± 0.089)	**0.555 (± 0.094)***	0.046 (± 0.015)	**0.064 (± 0.036)***
	DPI-G	**0.192 (± 0.051)**	0.184 (± 0.047)	0.281 (± 0.074)	**0.458 (± 0.186)***	**0.069 (± 0.069)**	0.055 (± 0.032)
	DPI-E	**0.166 (± 0.053)**	0.153 (± 0.102)	**0.603 (± 0.069)**	0.314 (± 0.078)	0.016 (± 0.009)	**0.022 (± 0.018)**
		PBCT	PbSS	PBCT	PbSS	PBCT	PbSS
AUROC	DPI-N	0.589 (± 0.101)	**0.655 (± 0.112)**	**0.659 (± 0.179)**	0.640 (± 0.165)	0.428 (± 0.204)	**0.494 (± 0.240)**
	DPI-I	**0.688 (± 0.065)**	0.584 (± 0.056)	**0.796 (± 0.058)**	0.772 (± 0.053)	0.546 (± 0.046)	**0.573 (± 0.087)**
	DPI-G	**0.673 (± 0.077)**	0.613 (± 0.031)	0.715 (± 0.049)	**0.743 (± 0.082)**	**0.571 (± 0.067)**	0.571 (± 0.089)
	DPI-E	**0.687 (± 0.052)**	0.623 (± 0.042)	**0.826 (± 0.042)**	0.689 (± 0.029)	0.539 (± 0.050)	**0.560 (± 0.080)**
		PBCT	PbSS	PBCT	PbSS	PBCT	PbSS
MCC	DPI-N	0.132 (± 0.153)	**0.289 (± 0.178)***	0.221 (± 0.254)	**0.243 (± 0.279)**	-0.011 (± 0.121)	**0.048 (± 0.137)**
	DPI-I	**0.221 (± 0.063)**	0.210 (± 0.122)	0.436 (± 0.090)	**0.542 (± 0.096)***	0.035 (± 0.036)	**0.060 (± 0.064)***
	DPI-G	0.210 (± 0.076)	**0.263 (± 0.060)***	0.257 (± 0.067)	**0.425 (± 0.154)***	0.046 (± 0.041)	**0.052 (± 0.060)**
	DPI-E	**0.174 (± 0.052)**	0.162 (± 0.088)	**0.423 (± 0.065)**	0.251 (± 0.073)	0.019 (± 0.024)	**0.033 (± 0.041)**

Regarding the results arranged in Table 5 (i.e., where the Dynamic w value was considered), we can observe competitive results regarding the Original PBCT and PbSS. There are cases where PbSS stands out, e.g., as for T_r x L_c considering AUPRC and MCC, and cases where it is possible to observe a slight variation between the results of Original PBCT and PbSSd. It is noteworthy that PbSSd stood out mainly in the tasks T_r x L_c and T_r x T_c, scenarios where it excelled for almost all datasets for AUPRC and MCC. Also noteworthy are the results obtained for DPI-E T_r x L_c, scenario where PbSS had the worst performance, i.e., the low performance in this scenario is credited mainly to the size of the data, and the low number of positive interactions (see Table 2 in Section 4.1). For T_r x T_c, the results obtained for MCC stand out, a scenario where PbSSd stood out for all datasets. Finally, it is worth mentioning that although both methods have a high standard deviation rate, PbSSd stood out in most cases.

Table 6 – Results for Evaluation Measures obtained for compared methods (ALVES; ILIDIO; CERRI, 2023), considering PBCT and PbSS with $w = 0$.

Measure	Data	*Lr* x *Tc*		*Tr* x *Lc*		*Tr* x *Tc*	
		PBCT	PbSS	PBCT	PbSS	PBCT	PbSS
AUPRC	DPI-N	0.216 (± 0.115)	**0.288** (± **0.159**)	0.259 (± 0.255)	**0.371** (± **0.310**)	**0.162** (± **0.145**)	0.101 (± 0.132)
	DPI-I	0.202 (± 0.086)	**0.203** (± **0.104**)	0.456 (± 0.089)	**0.581** (± **0.106**)*	0.046 (± 0.015)	**0.062** (± **0.036**)*
	DPI-G	**0.192** (± **0.051**)	0.184 (± 0.045)	0.281 (± 0.074)	**0.458** (± **0.186**)*	**0.069** (± **0.069**)	0.056 (± 0.029)
	DPI-E	0.166 (± 0.053)	**0.167** (± **0.112**)	**0.603** (± **0.069**)	0.315 (± 0.063)	0.016 (± 0.009)	**0.018** (± **0.012**)
		PBCT	PbSS	PBCT	PbSS	PBCT	PbSS
AUROC	DPI-N	0.589 (± 0.101)	**0.657** (± **0.108**)	**0.659** (± **0.179**)	0.639 (± 0.165)	0.428 (± 0.204)	**0.489** (± **0.244**)
	DPI-I	**0.688** (± **0.065**)	0.606 (± 0.077)	**0.796** (± **0.058**)	0.792 (± 0.057)	0.546 (± 0.046)	**0.552** (± **0.061**)
	DPI-G	**0.673** (± **0.077**)	0.612 (± 0.032)	0.715 (± 0.049)	**0.744** (± **0.081**)	0.571 (± 0.067)	**0.576** (± **0.087**)
	DPI-E	**0.687** (± **0.052**)	0.636 (± 0.049)	**0.826** (± **0.042**)	0.687 (± 0.031)	**0.539** (± **0.050**)	0.523 (± 0.049)
		PBCT	PbSS	PBCT	PbSS	PBCT	PbSS
MCC	DPI-N	0.132 (± 0.153)	**0.289** (± **0.178**)*	0.221 (± 0.254)	**0.235** (± **0.273**)	-0.011 (± 0.121)	**0.045** (± **0.160**)
	DPI-I	**0.221** (± **0.063**)	0.171 (± 0.116)	0.436 (± 0.090)	**0.544** (± **0.100**)*	0.035 (± 0.036)	**0.048** (± **0.055**)
	DPI-G	0.210 (± 0.076)	**0.262** (± **0.060**)*	0.257 (± 0.067)	**0.423** (± **0.155**)*	0.046 (± 0.041)	**0.057** (± **0.061**)
	DPI-E	**0.174** (± **0.052**)	0.169 (± 0.098)	**0.423** (± **0.065**)	0.248 (± 0.056)	**0.019** (± **0.024**)	0.013 (± 0.028)

In Table 6, the results represent a case with the value of $w = 0$, where it is possible to observe a similar scenario (i.e., with slight variation in results) concerning that observed in Table 5, but with some differences (e.g., for L_r x T_c it is observed that PbSS also obtained gains in DPI-I and DPI-E datasets, considering AUPRC). Despite the subtle variations observed in the results, the observed competitive performance evidences the relevance of the unsupervised impurity in calculating the semi-supervised impurity of PbSS.

The results arranged in Tables 7 and 8 present an experimental scenario similar to the previously presented, considering all the evaluation criteria, but with a MLSKF CV procedure. This scenario represents a further study regarding Dynamic and Static PbSS with $w = 0$, where the CV procedure tries to guarantee that there are samples of both classes (i.e., positive and negative interactions) in all folds, as well as evaluating through more robust validation procedure and also adding to the results the impact of the inexistence or the low amount of labeled interactions in the learning samples.

Table 7 presents the result of this experimental procedure referring to PbSSd. Initially, we can highlight that in the general context, competitive results were obtained. Regarding Table 5, both the PBCT and the PbXGB presented in a general context a slight variation in the results, and the main variations were observed for T_r x T_c, task where it is possible to

Table 7 – Results for Evaluation Measures obtained for compared methods, considering PBCT and PbSSd with dynamic w and MLSKF CV Procedure.

Measure	Data	*Lr* x *Tc*		*Tr* x *Lc*		*Tr* x *Tc*	
		PBCT	PbSS	PBCT	PbSS	PBCT	PbSS
AUPRC	DPI-N	0.247 (± 0.083)	**0.299 (± 0.093)**	0.213 (± 0.105)	**0.356 (± 0.285)**	**0.252 (± 0.179)**	0.114 (± 0.113)
	DPI-I	**0.210 (± 0.071)**	0.195 (± 0.067)	0.474 (± 0.065)	**0.576 (± 0.101)***	0.056 (± 0.021)	**0.066 (± 0.021)**
	DPI-G	**0.246 (± 0.039)**	0.231 (± 0.051)	0.270 (± 0.046)	**0.474 (± 0.151)***	0.050 (± 0.025)	**0.060 (± 0.030)**
	DPI-E	**0.163 (± 0.050)**	0.130 (± 0.100)	**0.604 (± 0.058)**	0.327 (± 0.047)	**0.026 (± 0.010)**	0.020 (± 0.010)
		PBCT	PbSS	PBCT	PbSS	PBCT	PbSS
AUROC	DPI-N	0.634 (± 0.067)	**0.685 (± 0.065)**	**0.649 (± 0.103)**	0.638 (± 0.145)	**0.570 (± 0.140)**	0.517 (± 0.151)
	DPI-I	**0.709 (± 0.081)**	0.586 (± 0.038)	**0.807 (± 0.033)**	0.781 (± 0.043)	0.549 (± 0.040)	**0.578 (± 0.051)***
	DPI-G	**0.714 (± 0.022)**	0.626 (± 0.030)	0.713 (± 0.046)	**0.743 (± 0.072)**	0.561 (± 0.077)	**0.585 (± 0.064)**
	DPI-E	**0.694 (± 0.065)**	0.615 (± 0.050)	**0.830 (± 0.036)**	0.696 (± 0.031)	0.566 (± 0.036)	**0.568 (± 0.092)**
		PBCT	PbSS	PBCT	PbSS	PBCT	PbSS
MCC	DPI-N	0.217 (± 0.100)	**0.315 (± 0.105)**	0.237 (± 0.146)	**0.256 (± 0.254)**	**0.061 (± 0.131)**	0.017 (± 0.140)
	DPI-I	**0.247 (± 0.080)**	0.210 (± 0.095)	0.441 (± 0.051)	**0.570 (± 0.087)***	0.037 (± 0.031)	**0.067 (± 0.046)***
	DPI-G	0.267 (± 0.026)	**0.295 (± 0.080)**	0.269 (± 0.066)	**0.443 (± 0.113)***	0.045 (± 0.055)	**0.064 (± 0.048)**
	DPI-E	**0.185 (± 0.069)**	0.146 (± 0.073)	**0.452 (± 0.029)**	0.253 (± 0.026)	0.029 (± 0.016)	**0.033 (± 0.047)**

observe cases where PbSSd excelled (e.g., for AUPRC and AUROC in the DPI-G dataset), and cases where it did not (e.g., for DPI-N AUROC and MCC, and DPI-E AUPRC). In a general context, a slight reduction in the standard deviation can also be noted, with emphasis on L_r x T_c considering AUPRC, and T_r x T_c for all evaluation criteria).

A similar scenario can be observed in Table 8 when compared to Table 6 (i.e., with subtle variations in the results obtained). With highlights for the DPI-E dataset considering the task L_r x T_c and all the evaluation criteria, a scenario where the PbSS had a relative loss of performance, and for the task T_r x T_c in the DPI-I dataset considering all evaluation criteria, a scenario where a relatively expressive gain can be observed. There is also a slight reduction in the standard deviation in general, especially in tasks L_r x T_c and T_r x L_c for MCC. The smooth performance gains added to the standard deviation reduction indicate more consistent results, i.e., that the MLSKF CV procedure positively impacted the experimental procedure, allowing the learning model to better adjust itself

Table 8 – Results for Evaluation Measures obtained for compared methods, considering PBCT and PbSS with $w = 0$ and MLSKF CV Procedure.

Measure	Data	*Lr* x *Tc*		*Tr* x *Lc*		*Tr* x *Tc*	
		PBCT	PbSS	PBCT	PbSS	PBCT	PbSS
AUPRC	DPI-N	0.247 (± 0.083)	**0.292 (± 0.100)**	0.213 (± 0.105)	**0.370 (± 0.277)**	**0.252 (± 0.179)**	0.100 (± 0.081)
	DPI-I	**0.210 (± 0.071)**	0.197 (± 0.124)	0.474 (± 0.065)	**0.582 (± 0.090)***	0.056 (± 0.021)	**0.076 (± 0.035)***
	DPI-G	**0.246 (± 0.039)**	0.230 (± 0.049)	0.270 (± 0.046)	**0.474 (± 0.151)***	0.050 (± 0.025)	**0.056 (± 0.028)**
	DPI-E	**0.163 (± 0.050)**	0.128 (± 0.089)	**0.604 (± 0.058)**	0.330 (± 0.033)	**0.026 (± 0.010)**	0.018 (± 0.015)
		PBCT	PbSS	PBCT	PbSS	PBCT	PbSS
AUROC	DPI-N	0.634 (± 0.067)	**0.650 (± 0.040)**	**0.649 (± 0.103)**	0.645 (± 0.142)	**0.570 (± 0.140)**	0.529 (± 0.124)
	DPI-I	**0.709 (± 0.081)**	0.603 (± 0.070)	**0.807 (± 0.033)**	0.787 (± 0.044)	0.549 (± 0.040)	**0.577 (± 0.035)***
	DPI-G	**0.714 (± 0.022)**	0.625 (± 0.029)	0.713 (± 0.046)	**0.743 (± 0.072)**	0.561 (± 0.077)	**0.580 (± 0.066)**
	DPI-E	**0.694 (± 0.065)**	0.625 (± 0.042)	**0.830 (± 0.036)**	0.698 (± 0.031)	**0.566 (± 0.036)**	0.515 (± 0.061)
		PBCT	PbSS	PBCT	PbSS	PBCT	PbSS
MCC	DPI-N	0.217 (± 0.100)	**0.291 (± 0.087)**	0.237 (± 0.146)	**0.263 (± 0.241)**	**0.061 (± 0.131)**	0.028 (± 0.119)
	DPI-I	**0.247 (± 0.080)**	0.182 (± 0.123)	0.441 (± 0.051)	**0.552 (± 0.071)***	0.037 (± 0.031)	**0.076 (± 0.033)***
	DPI-G	0.267 (± 0.026)	**0.293 (± 0.077)**	0.269 (± 0.066)	**0.442 (± 0.112)***	0.045 (± 0.055)	**0.062 (± 0.048)**
	DPI-E	**0.185 (± 0.069)**	0.150 (± 0.057)	**0.452 (± 0.029)**	0.261 (± 0.031)	**0.029 (± 0.016)**	0.010 (± 0.035)

to the data, reducing the imbalance present in folds (i.e., this indicates that balanced data, with a higher rate of labeled interactions, can affect the performance of the learning model in this context).

5.3 Computational Performance

This section will present the results obtained by evaluating the proposed models' computational performance. The evaluation of the computational performance is carried out considering a comparative study between the Original PBCT and the proposed methods, taking into account the Induction time, Test time, and Total execution of the learning model's time. It is noteworthy that the experimental procedure was carried out in the development environment (Section 4.2) previously defined and established (i.e., a controlled environment, which replicates the same experimental conditions for both methods).

Table 9 – Run Time Comparison in Seconds for compared methods (ALVES; CERRI, 2022), considering PBCT and PbXGB.

	Data	*Lr* x *Tc*		*Tr* x *Lc*		*Tr* x *Tc*	
		PBCT	PbXGB	PBCT	PbXGB	PBCT	PbXGB
All	DPI-N	**0.038**	7.642	**0.014**	9.124	**0.089**	2.227
	DPI-I	**1.118**	134.392	**1.176**	131.230	**3.232**	42.008
	DPI-G	**0.520**	65.899	**0.672**	65.171	**1.312**	18.674
	DPI-E	**24.111**	409.514	**24.277**	397.146	**73.531**	121.091
Train	DPI-N	**0.029**	6.554	**0.011**	7.690	**0.030**	0.099
	DPI-I	**1.048**	122.282	**1.108**	119.295	**1.344**	3.641
	DPI-G	**0.486**	60.375	**0.643**	59.354	**0.550**	1.582
	DPI-E	**23.384**	387.188	**23.493**	373.323	**30.385**	30.880
Test	DPI-N	**0.009**	1.088	**0.003**	1.434	**0.009**	1.018
	DPI-I	**0.070**	12.110	**0.068**	11.935	**0.206**	15.965
	DPI-G	**0.034**	5.524	**0.029**	5.817	**0.087**	7.259
	DPI-E	**0.727**	22.326	**0.784**	23.823	**2.497**	25.144

The comparative study of the computational performance referring to PBCT and PbXGB is presented in Table 9. As mentioned earlier, in this context, three scenarios were considered to estimate the execution times. In Table 9 *All* represents the total execution time (i.e., Induction time + Prediction time); *Train* refers to the training time of the models, while *Test* represents the time needed to make predictions. It is possible to observe that in all the scenarios displayed in Table 9 the execution time of PbXGB was superior to that of PBCT. This is expected since PbXGB has a second phase (see Section 3.2) where an XGBoost learning model (see in Section 2.7) is induced on each leaf, and during the prediction phase, performs the prediction for each sample to be predicted.

Table 10 presents the results of the comparative study concerning PBCT and both Dynamic (PbSSd) and Statically Defined (PbSS) methods where a scenario similar to that of Table 9 is observed (i.e., PbSS presented a longer execution time in all evaluated cases). This is mainly attributed to the semi-supervised impurity function, which significantly increases the time complexity of the method by introducing the unsupervised term of the equation (see Section 3.3).

Table 10 – Run Time Comparison in Seconds for compared methods, considering PBCT, PbSSd, and PbSS.

	Data	*Lr* x *Tc*			*Tr* x *Lc*			*Tr* x *Tc*		
		PBCT	PbSSd	PbSS	PBCT	PbSSd	PbSS	PBCT	PbSSd	PbSS
All	DPI-N	**0.011**	0.934	0.929	**0.010**	3.002	3.079	**0.035**	6.107	6.384
	DPI-I	**0.478**	147.642	302.582	**0.472**	202.155	152.881	**1.647**	803.391	536.731
	DPI-G	**0.189**	46.621	46.983	**0.205**	183.214	183.779	**0.686**	436.675	430.707
	DPI-E	**11.866**	57230.648	41293.040	**11.145**	4822.878	4646.302	**40.186**	118825.243	93198.253
Train	DPI-N	**0.008**	0.932	0.926	**0.008**	2.999	3.075	**0.012**	2.716	2.740
	DPI-I	**0.455**	147.612	302.536	**0.448**	202.124	152.854	**0.666**	263.370	245.811
	DPI-G	**0.178**	46.605	46.969	**0.194**	183.196	183.761	**0.287**	186.256	194.489
	DPI-E	**11.607**	57230.217	41292.760	**10.881**	4822.668	4646.107	**14.951**	44401.755	34748.164
Test	DPI-N	0.003	**0.002**	0.002	**0.002**	0.003	0.003	**0.003**	0.008	0.007
	DPI-I	**0.024**	0.029	0.046	**0.023**	0.031	0.028	**0.078**	0.124	0.118
	DPI-G	**0.011**	0.016	0.014	**0.011**	0.018	0.017	**0.037**	0.063	0.064
	DPI-E	**0.259**	0.431	0.280	0.263	0.211	**0.196**	**0.943**	1.409	1.080

It is noteworthy that even though the induction time (*Train*) increased significantly, the test time did not increase in similar proportions (i.e., this occurs because no modification was made in the test procedure concerning the original PBCT). It is noteworthy that the slight increase in the execution time of the test procedure indicates that the trees were built with greater depth, fitting better to the data. This is expected, since the semi-supervised impurity simultaneously considers the labeled data and unlabeled, thus using unlabeled data as a means to find impurity reduction paths in cases where only the labeled impurity fails. Notably, the unlabeled part of the semi-supervised function is sensitive to cases where the labeled data have low quality or noise and may present low performance.

Another observation is that PbSSd presented a better computational performance for all cases *All* and *Train* except for the DPI-E dataset for tasks L_r x T_c and T_r x T_c against PbSS with $w = 0$ in Table 10. This may indicate that dynamically estimating the supervision level reduces the size of the generated trees in terms of depth. It is also worth noting that PbSS obtained a better computational performance for T_r x L_c considering the DPI-E dataset and *Test* time.

5.4 Statistical Analysis

This section presents the data from the statistical analysis referring to the defined models (i.e., PbXGB and PbSS). Statistical analysis of the data was performed using Wilcoxon Signed Rank Test (Paired Samples) with Bonferroni Correction. The statistical analysis was conducted on the results obtained by each fold of the CV procedure. The evaluation is done for each prediction task (e.g., T_r x L_c) and evaluation criterion (e.g., AUPRC). Results are statistically significant when the value of $p < 0.05$ (i.e., when $p \leq y$ the results demonstrate statistically significant differences, while $p = y$ shows cases where no statistically significant differences were observed).

Figure 11 A) was constructed considering the experimental results of PbXGB used in the generation of Table 3, and the previously described scenario, where a statistical analysis was conducted with the Wilcoxon Signed Rank Test (Paired Samples) with Bonferroni Correction considering each of the folds of the CV procedure. Where it is possible to observe that the PbXGB presented statistically significant differences for the tasks T_r x L_c and L_r x T_c, considering the evaluation criteria AUPRC and MCC, where in some cases the results obtained for p were much smaller than 0.05. Note that a similar scenario is observed in Figure 11 B), referring to the statistical analysis of the results referring to Table 4.

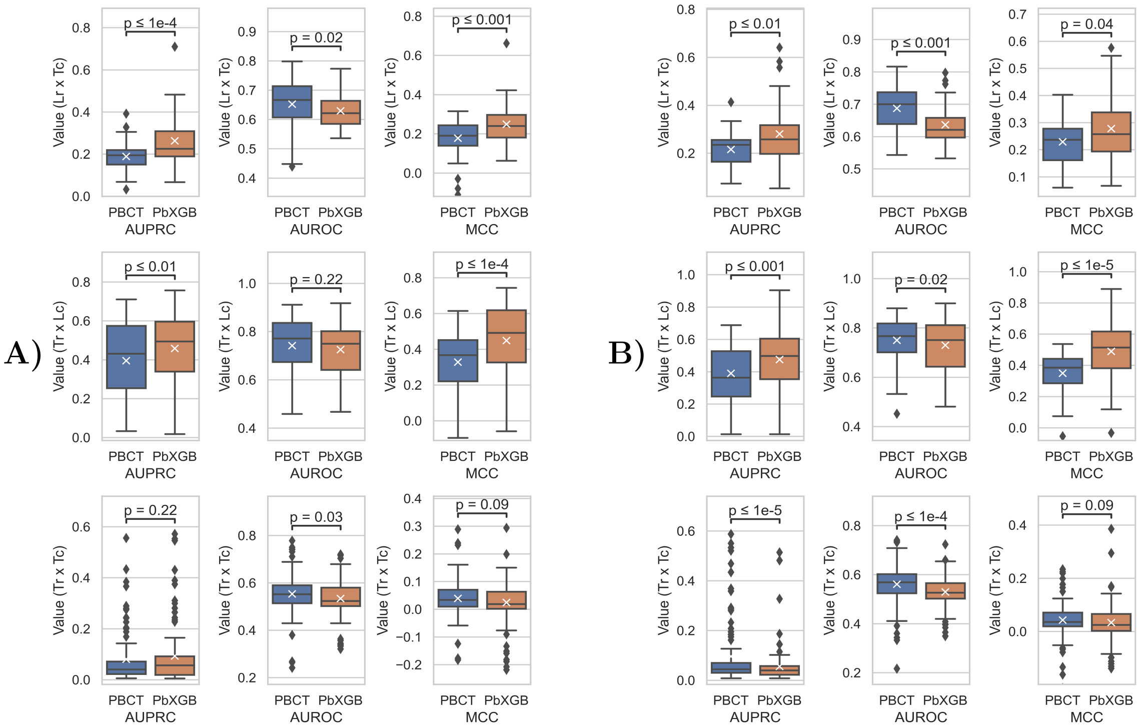


Figure 11 – Comparative statistical difference analysis of PBCT and PbXGB performances considering the established scenario. A) is a PbXGB (ALVES; CERRI, 2022), B) is a PbXGB with Grid-Search and MLSKF CV Procedure.

Figure 12 was built considering a similar scenario to the previous one, however with experimental data referring to Tables 5 and 6, for the DPI-I, DPI-G, and DPI-N datasets. Where, in particular, for both definitions of w, PbSS demonstrated statistically significant gains for MCC and AUPRC, considering the tasks T_r x L_c and the performance obtained by all folds in almost all datasets (i.e., the DPI-N, DPI-G, and DPI-I datasets), scenarios where the value of p was much less than 0.05. It is possible to observe that the statistical performance obtained is in harmony with that observed in Tables 5 and 6 (i.e., both results are in sync with the predictive performance obtained).

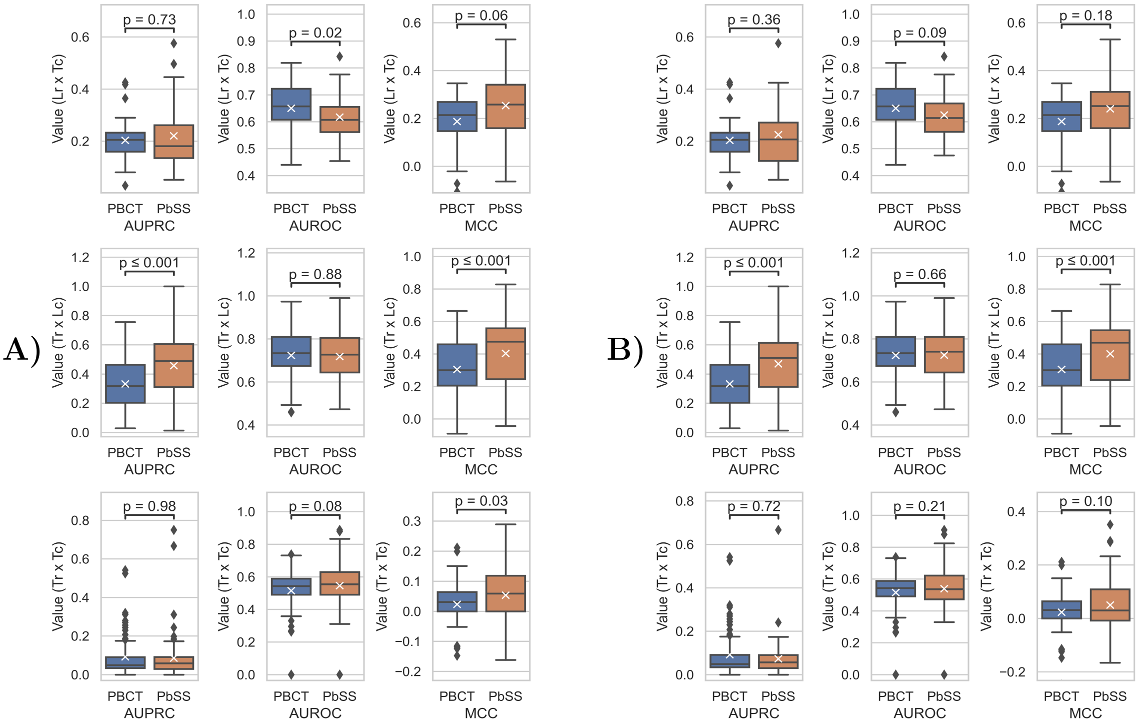


Figure 12 – Comparative statistical difference analysis of PBCT and PbSS performances considering the established scenario (ALVES; ILIDIO; CERRI, 2023), and dynamic w (A) and $w = 0$ (B).

A similar scenario can be seen in Figure 13, which refers to the statistical analysis of the results obtained considering the DPI-I, DPI-G, and DPI-N datasets, and the MLSKF CV procedure, referring to Tables 7 and 8. Furthermore, it is worth noting that in the statistical analysis procedure referring to Figures 12, and 13, the DPI-E was disregarded because it was considered an outlier (see Tables 5, 6, 7, and 8) concerning the other datasets (i.e., the Appendix A.1 presents the result of the statistical analysis procedure considering the DPI-E dataset in both cases). Finally, it is highlighted that the statistically significant results obtained indicate consolidated gains in predictive performance (i.e., concerning the Original PBCT in the evaluated cases).

5.5 Discussion

This section aims to raise a discussion of the relevant points referring to the previously presented results, thus pointing out the positive and negative aspects of each method and serving as a finalization of this chapter regarding the results obtained in the course of this work. As a fundamental principle of this work, we work with the hypothesis that it

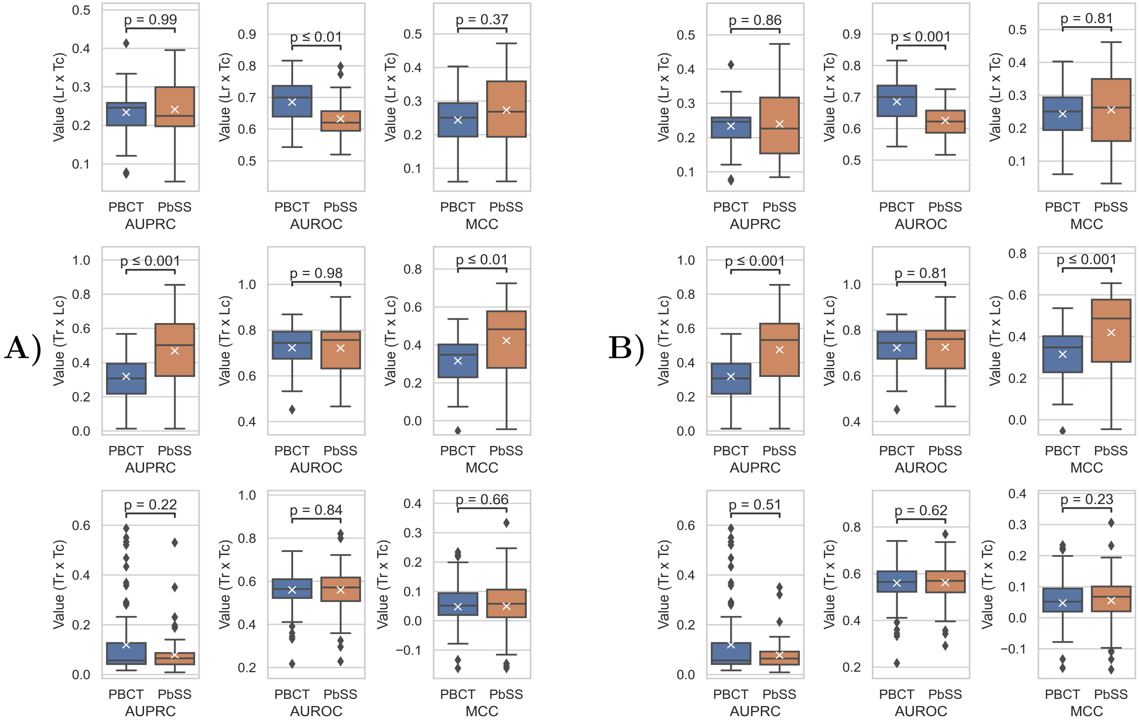


Figure 13 – Comparative statistical difference analysis of PBCT and PbSS performances considering the established scenario for DPI-I, DPI-G, and DPI-N datasets, with a MLSKF CV Procedure. A) is PbSSd with dynamic w, B) is PbSS with $w = 0$.

is possible to improve Predictive Bi-clustering Trees in terms of predictive performance and computational efficiency through more specific adaptations and modifications to the context in the method's operation (see Section 1.3). More specifically, in this work, we focus on two specific interaction prediction problems, namely real scenarios of imbalance between positive and negative interactions and unlabeled data (e.g., Positive-Unlabeled data scenarios), i.e., this work focuses on scenarios with real data (not directly considering artificial data). In this way, advances represent contributions and are directly applicable in their respective areas of the literature. In the context of this work, we can consider the vast amount of unlabeled data as a case of imbalance, not only between positive and negative interactions but also between known and unknown interactions. In this context, both proposed methods aim to treat data imbalance, and each one applies to a specific problem.

In this context, we developed two approaches: the PbXGB (i.e., suitable for scenarios with imbalanced data) and the PbSS (i.e., suitable for scenarios with unlabeled data). It was verified through an experimental procedure that the main positive factor of PbXGB is its ability to reduce the imbalance in the leaves of the PBCT (i.e., the XGBoost model induced in the leaf partitions has the potential to produce less imbalanced predictions during the prediction) making more balanced predictions. In this way PbXGB has the potential to obtain more consistent predictions even in scenarios with extremely imbalanced data. On the other hand, the main negative factor of PbXGB is related to its computational performance (i.e., since in the second stage, a learning model is induced in each tree leaf). This establishes an ideal scenario for applying specific cases with extremely imbalanced data between positive and negative interactions, scenarios where computational performance is not the focus but the quality of predictions.

Strengths of PbSS are its ability to work with labeled and unlabeled data and dynamically or statically determine the level of supervision. As negative factors, we have low computational performance since the time complexity of the model increases significantly according to the size of the datasets, thus establishing an ideal scenario for applying specific cases, such as Positive-Unlabeled data when we have a high amount of unlabeled data. It is also worth mentioning that PbSS is hybrid and can work only with labeled data when necessary.

Thus, considering the performance of both methods, we can experimentally validate their contributions in their respective areas of knowledge and literature, advancing the state-of-the-art in this area and paving the way for new research in the respective areas.

Chapter 6

Final Considerations and Future Works

6.1 Work Syntesis

In this work, the prediction of *In Silico* interactions was approached from the viewpoint of PBCTs. Two main topics were studied in this context: 1) Extreme Gradient Boosting XGBoost; and 2) Semi-Supervised Machine Learning. The objective was to improve PBCTs reagarding issues such as Imbalance between positive and negative interactions and Unlabeled data.

The first proposed algorithm, PbXGB, is a hybrid method between PBCTs and XGBoost, where PBCT produce partitions in the interactions matrix, and XGBoost is used to learn from the partition data and make predictions. The method's performance was promising through an experimental procedure, indicating that XGBoost can learn from data from imbalanced partitions and produce balanced predictions, despite being more computationally expensive.

The second proposed algorithm, PbSS, transforms PBCTs into a semi-supervised learning method by changing the impurity function for the semi-supervised impurity and adapting the tree-splitting method to accept labeled and unlabeled data in the learning procedure. Despite the high computational cost, the experiments demonstrated gains on the baseline, with the advantage of considering labeled and unlabeled data in the learning procedure (i.e., which usually represent a significant part of the datasets).

6.2 Future Works

The algorithms presented in this work (PbXGB and PbSS) were developed considering context-specific scenarios (i.e., with significant levels of imbalance between positive and negative interactions and between labeled and unlabeled interactions) and applied to DTI datasets that replicate a valid real-world scenario. However, the literature defines an immensity of real-world problems, and a vast way is visible to those who aim at contributions applicable simultaneously to several real-world problems.

Considering this question, it is possible to consolidate the proposed methods through extensive application and adaptation to other scenarios with different real-world problems, e.g., Recommendation Systems (i.e., in recommendation systems, an interaction in the format defined in this work can be interpreted as a link or recommendation), and Triclustering Problems (NARMADHA; RATHIPRIYA, 2016; HENRIQUES; MADEIRA, 2019). The way splits are structured in PBCTs indicates that it is possible to induce Triclustering problems through adaptations, thus building trees that simultaneously consider three Feature Spaces and a Tri-Dimensional Interaction Matrix, including the possibility of paving the way for the study of N-Clustering problems through the construction and induction of N-Dimensional trees.

One factor that impacts the computational performance of PbSS is the fact that splits only occur in the Interaction Matrix (i.e., which means that the Unsupervised Term of the Semi-Supervised Impurity Function applied to all Features regardless of the depth of the tree, and even on the leaves). For the solution to this predicament, it is possible to indicate two solutions. Applying and adapting the Feature Importance approach in the PbSS induction and prediction procedure (i.e., Feature Importances can be used to select and propagate groups of features through the tree levels). A similar approach considers applying and adapting Dimensionality Reduction methods in feature space. When combined with the appropriate heuristics, both approaches indicate that they have the potential to provide a way to improve PbSS.

In the context of PbSS, an alternative way can be taken in cases where the data have a high rate of imbalance (i.e., between Positive and Negative Interactions, Labeled or Unlabeled), employing anomaly detection methods (e.g., as the Isolation Forest (LIU; TING; ZHOU, 2008; LIU; TING; ZHOU, 2012; TOKOVAROV; KARCZMAREK, 2022)), i.e., in this case, we can consider scarce data as anomalies, seeking to filter and heuristically select other anomalies to balance the data.

In short, after considering the limitations and considering the way for the evolution of the proposed algorithms based on what is indicated as promising, it is possible to suggest future works:

- Consolidate the proposed learning models through application and adaptation in broader literature scenarios;

- Improve the computational performance of PbSS by adapting Feature Importance's and Dimensionality Reduction methods;
- Adapt, or produce a variation of PbSS that works based on anomaly detection in cases where there is a serious imbalance in the data;
- The construction of N-Dimensional PBCT (i.e., Predictive N-Clustering Trees (PNCTs)) for the induction, investigation, and study of Multi-dimensional N-Clustering problems (e.g., Tri-clustering), and other Multi-Dimensional problems.

Bibliography

ALVES, A.; ILIDIO, P.; CERRI, R. Semi-supervised hybrid predictive bi-clustering trees for drug-target interaction prediction. ACM, mar 2023.

ALVES, A. H. R.; CERRI, R. A two-step model for drug-target interaction prediction with predictive bi-clustering trees and XGBoost. IEEE, jul 2022.

ARNOLD, K.; GOSLING, J.; HOLMES, D. **The Java Programming Language**. 3rd. ed. USA: Addison-Wesley Longman Publishing Co., Inc., 2000. ISBN 0201704331.

BAGHERIAN, M. et al. Machine learning approaches and databases for prediction of drug–target interaction: a survey paper. **Briefings in Bioinformatics**, Oxford University Press (OUP), v. 22, n. 1, p. 247–269, jan 2020.

BARKER, B. et al. Ion channels. In: _____. **Conn's Translational Neuroscience**. [S.l.]: Elsevier, 2017. p. 11–43.

BASPINAR, A.; CUKUROGLU, E.; NUSSINOV, R.; KESKIN, O.; GURSOY, A. PRISM: a web server and repository for prediction of protein–protein interactions and modeling their 3d complexes. **Nucleic Acids Research**, Oxford University Press (OUP), v. 42, n. W1, p. W285–W289, may 2014.

BEKKER, J.; DAVIS, J. Learning from positive and unlabeled data: a survey. **Machine Learning**, Springer Science and Business Media LLC, v. 109, n. 4, p. 719–760, apr 2020.

BELTRAN, J. C.; VALDEZ, P.; NAVAL, P. Predicting protein-protein interactions based on biological information using extreme gradient boosting. In: **2019 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)**. [S.l.]: IEEE, 2019.

BOUGHORBEL, S.; JARRAY, F.; EL-ANBARI, M. Optimal classifier for imbalanced data using matthews correlation coefficient metric. **PLOS ONE**, Public Library of Science (PLoS), v. 12, n. 6, p. e0177678, jun 2017.

BOYD, K.; ENG, K. H.; PAGE, C. D. Area under the precision-recall curve: Point estimates and confidence intervals. In: **Advanced Information Systems Engineering**. [S.l.]: Springer Berlin Heidelberg, 2013. p. 451–466.

BROWN, H. S.; GALETIN, A.; HALLIFAX, D.; HOUSTON, J. B. Prediction of in vivo drug-drug interactions from in vitro data. **Clinical Pharmacokinetics**, Springer Science and Business Media LLC, v. 45, n. 10, p. 1035–1050, 2006.

CAMARGO, G.; BUGATTI, P. H.; SAITO, P. T. M. Active semi-supervised learning for biological data classification. **PLOS ONE**, Public Library of Science (PLoS), v. 15, n. 8, p. e0237428, aug 2020.

CERRI, R.; BARROS, R. C.; CARVALHO, A. C. P. L. F. de; JIN, Y. Reduction strategies for hierarchical multi-label classification in protein function prediction. **BMC Bioinformatics**, Springer Science and Business Media LLC, v. 17, n. 1, sep 2016.

CHAWLA, N. V.; BOWYER, K. W.; HALL, L. O.; KEGELMEYER, W. P. SMOTE: Synthetic minority over-sampling technique. **Journal of Artificial Intelligence Research**, AI Access Foundation, v. 16, p. 321–357, jun 2002.

CHEN, H.; WANG, L.; CHI, C.-H.; SHEN, J. Leveraging SMOTE in a two-layer model for prediction of protein-protein interactions. In: **2019 Seventh International Conference on Advanced Cloud and Big Data (CBD)**. [S.l.]: IEEE, 2019.

CHEN, H. et al. Hyperparameter estimation in SVM with GPU acceleration for prediction of protein-protein interactions. In: **2019 IEEE International Conference on Big Data (Big Data)**. [S.l.]: IEEE, 2019.

CHEN, T.; GUESTRIN, C. XGBoost. In: **Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining**. [S.l.]: ACM, 2016.

CHENG, J.; TEGGE, A.; BALDI, P. Machine learning methods for protein structure prediction. **IEEE Reviews in Biomedical Engineering**, Institute of Electrical and Electronics Engineers (IEEE), v. 1, p. 41–49, 2008.

CHICCO, D.; JURMAN, G. The advantages of the matthews correlation coefficient (MCC) over f1 score and accuracy in binary classification evaluation. **BMC Genomics**, Springer Science and Business Media LLC, v. 21, n. 1, jan 2020.

CHICCO, D.; TöTSCH, N.; JURMAN, G. The matthews correlation coefficient (MCC) is more reliable than balanced accuracy, bookmaker informedness, and markedness in two-class confusion matrix evaluation. **BioData Mining**, Springer Science and Business Media LLC, v. 14, n. 1, feb 2021.

CHOI, Y.; SHIN, B.; KANG, K.; PARK, S.; BECK, B. R. Target-centered drug repurposing predictions of human angiotensin-converting enzyme 2 (ACE2) and transmembrane protease serine subtype 2 (TMPRSS2) interacting approved drugs for coronavirus disease 2019 (COVID-19) treatment through a drug-target interaction deep learning model. **Viruses**, MDPI AG, v. 12, n. 11, p. 1325, nov 2020.

CHRISTINELLI, W. A. et al. Two-dimensional MoS2-based impedimetric electronic tongue for the discrimination of endocrine disrupting chemicals using machine learning. **Sensors and Actuators B: Chemical**, Elsevier BV, v. 336, p. 129696, jun 2021.

CHRYSOSTOMOU, C.; SEKER, H. Structural classification of protein sequences based on signal processing and support vector machines. In: **2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)**. [S.l.]: IEEE, 2016.

DEY, L.; MUKHOPADHYAY, A. A classification-based approach to prediction of dengue virus and human protein-protein interactions using amino acid composition and conjoint triad features. In: **2019 IEEE Region 10 Symposium (TENSYMP)**. [S.l.]: IEEE, 2019.

DIAZ, A. K. R.; PERES, S. M. Biclustering and coclustering: concepts, algorithms and viability for text mining. **Revista de Informática Teórica e Aplicada**, Universidade Federal do Rio Grande do Sul, v. 26, n. 2, p. 81–117, aug 2019.

DING, H.; TAKIGAWA, I.; MAMITSUKA, H.; ZHU, S. Similarity-based machine learning methods for predicting drug–target interactions: a brief review. **Briefings in Bioinformatics**, Oxford University Press (OUP), v. 15, n. 5, p. 734–747, aug 2013.

DING, S.; ZHU, Z.; ZHANG, X. An overview on semi-supervised support vector machine. **Neural Computing and Applications**, Springer Science and Business Media LLC, v. 28, n. 5, p. 969–978, nov 2015.

DING, Y.; TANG, J.; GUO, F. Predicting protein-protein interactions via multivariate mutual information of protein sequences. **BMC Bioinformatics**, Springer Science and Business Media LLC, v. 17, n. 1, sep 2016.

ENGELEN, J. E. van; HOOS, H. H. A survey on semi-supervised learning. **Machine Learning**, Springer Science and Business Media LLC, v. 109, n. 2, p. 373–440, nov 2019.

EZZAT, A.; WU, M.; LI, X.-L.; KWOH, C.-K. Drug-target interaction prediction via class imbalance-aware ensemble learning. **BMC Bioinformatics**, Springer Science and Business Media LLC, v. 17, n. S19, dec 2016.

FATTAHI, F.; REFAHI, M. S.; MINAEI-BIDGOLI, B. Drug-target interaction prediction using edge2vec algorithm on the heterogeneous network via SVM. In: **2019 5th Iranian Conference on Signal Processing and Intelligent Systems (ICSPIS)**. [S.l.]: IEEE, 2019.

FRIGO, D. E.; BONDESSON, M.; WILLIAMS, C. Nuclear receptors: from molecular mechanisms to therapeutics. **Essays in Biochemistry**, Portland Press Ltd., v. 65, n. 6, p. 847–856, nov 2021.

GALEANO, D.; LI, S.; GERSTEIN, M.; PACCANARO, A. Predicting the frequencies of drug side effects. **Nature Communications**, Springer Science and Business Media LLC, v. 11, n. 1, sep 2020.

GAO, Z. et al. edge2vec: Representation learning using edge semantics for biomedical knowledge discovery. **BMC Bioinformatics**, Springer Science and Business Media LLC, v. 20, n. 1, jun 2019.

GOMEZ, S. M.; NOBLE, W. S.; RZHETSKY, A. Learning to predict protein-protein interactions from protein sequences. **Bioinformatics**, Oxford University Press (OUP), v. 19, n. 15, p. 1875–1881, oct 2003.

HAMMOUDEH, Z.; LOWD, D. Learning from positive and unlabeled data with arbitrary positive shift. In: LAROCHELLE, H.; RANZATO, M.; HADSELL, R.; BALCAN, M.; LIN, H. (Ed.). **Advances in Neural Information Processing Systems**. [S.l.]: Curran Associates, Inc., 2020. v. 33, p. 13088–13099.

HASHEMIFAR, S.; NEYSHABUR, B.; KHAN, A. A.; XU, J. Predicting protein–protein interactions through sequence-based deep learning. **Bioinformatics**, Oxford University Press (OUP), v. 34, n. 17, p. i802–i810, sep 2018.

HATTORI, M.; OKUNO, Y.; GOTO, S.; KANEHISA, M. Development of a chemical structure comparison method for integrated analysis of chemical and genomic information in the metabolic pathways. **Journal of the American Chemical Society**, American Chemical Society (ACS), v. 125, n. 39, p. 11853–11865, sep 2003.

HATTORI, M.; TANAKA, N.; KANEHISA, M.; GOTO, S. SIMCOMP/SUBCOMP: chemical structure search servers for network analyses. **Nucleic Acids Research**, Oxford University Press (OUP), v. 38, n. Web Server, p. W652–W656, may 2010.

HAYES, S.; MALACRIDA, B.; KIELY, M.; KIELY, P. A. Studying protein–protein interactions: progress, pitfalls and solutions. **Biochemical Society Transactions**, Portland Press Ltd., v. 44, n. 4, p. 994–1004, aug 2016.

HENRIQUES, R.; MADEIRA, S. C. Triclustering algorithms for three-dimensional data analysis. **ACM Computing Surveys**, Association for Computing Machinery (ACM), v. 51, n. 5, p. 1–43, jan 2019.

HERRERA, F.; CHARTE, F.; RIVERA, A. J.; JESUS, M. J. del. Introduction. In: **Multilabel Classification**. [S.l.]: Springer International Publishing, 2016. p. 1–16.

______. Multilabel classification. In: **Multilabel Classification**. [S.l.]: Springer International Publishing, 2016. p. 17–31.

HUANG, Q.; YOU, Z.; ZHANG, X.; ZHOU, Y. Prediction of protein–protein interactions with clustered amino acids and weighted sparse representation. **International Journal of Molecular Sciences**, MDPI AG, v. 16, n. 12, p. 10855–10869, may 2015.

ISLAM, S. M.; HOSSAIN, S. M. M.; RAY, S. DTI-SNNFRA: Drug-target interaction prediction by shared nearest neighbors and fuzzy-rough approximation. **PLOS ONE**, Public Library of Science (PLoS), v. 16, n. 2, p. e0246920, feb 2021.

JIMENEZ, C.; MOLINA, M.; MONTENEGRO, C. Deep learning – based models for drug-drug interactions extraction in the current biomedical literature. In: **2019 International Conference on Information Systems and Software Technologies (ICI2ST)**. [S.l.]: IEEE, 2019.

LEE, I.; KEUM, J.; NAM, H. DeepConv-DTI: Prediction of drug-target interactions via deep learning with convolution on protein sequences. **PLOS Computational Biology**, Public Library of Science (PLoS), v. 15, n. 6, p. e1007129, jun 2019.

LEVATIĆ, J.; CECI, M.; KOCEV, D.; DŽEROSKI, S. Semi-supervised classification trees. **Journal of Intelligent Information Systems**, Springer Science and Business Media LLC, v. 49, n. 3, p. 461–486, mar 2017.

LI, J. et al. Using weighted extreme learning machine combined with scale-invariant feature transform to predict protein-protein interactions from protein evolutionary information. **IEEE/ACM Transactions on Computational Biology and Bioinformatics**, Institute of Electrical and Electronics Engineers (IEEE), p. 1–1, 2020.

LI, Y.-F.; LIANG, D.-M. Safe semi-supervised learning: a brief introduction. **Frontiers of Computer Science**, Springer Science and Business Media LLC, v. 13, n. 4, p. 669–676, jun 2019.

LIU, F. T.; TING, K. M.; ZHOU, Z.-H. Isolation forest. IEEE, dec 2008.

_____. Isolation-based anomaly detection. **ACM Transactions on Knowledge Discovery from Data**, Association for Computing Machinery (ACM), v. 6, n. 1, p. 1–39, mar 2012.

LIU, H.; SUN, J.; GUAN, J.; ZHENG, J.; ZHOU, S. Improving compound–protein interaction prediction by building up highly credible negative samples. **Bioinformatics**, Oxford University Press (OUP), v. 31, n. 12, p. i221–i229, jun 2015.

LOWE, D. G. Distinctive image features from scale-invariant keypoints. **International Journal of Computer Vision**, Springer Science and Business Media LLC, v. 60, n. 2, p. 91–110, nov 2004.

LUO, Y.; ZHAO, Y.; CHENG, L.; JIANG, P.; WANG, J. Protein-protein interaction network comparison based on wavelet and principal component analysis. In: **2010 IEEE International Conference on Bioinformatics and Biomedicine Workshops (BIBMW)**. [S.l.]: IEEE, 2010.

MADEIRA, S.; OLIVEIRA, A. Biclustering algorithms for biological data analysis: a survey. **IEEE/ACM Transactions on Computational Biology and Bioinformatics**, Institute of Electrical and Electronics Engineers (IEEE), v. 1, n. 1, p. 24–45, jan 2004.

MANIKANDAN, P.; RAMYACHITRA, D. Prediction of protein structural classes based on secondary structure sequence using improved support vector machine (ISVM). In: **2016 IEEE Uttar Pradesh Section International Conference on Electrical, Computer and Electronics Engineering (UPCON)**. [S.l.]: IEEE, 2016.

MORI, Y.; KURODA, M.; MAKINO, N. Nonlinear principal component analysis. In: **Nonlinear Principal Component Analysis and Its Applications**. [S.l.]: Springer Singapore, 2016. p. 7–10.

MORIAUD, F. et al. Identify drug repurposing candidates by mining the protein data bank. **Briefings in Bioinformatics**, Oxford University Press (OUP), v. 12, n. 4, p. 336–340, apr 2011.

NAHLER, G. Bonferroni correction. Springer Vienna, p. 18–18, 2009.

NANDI, A. K.; AHMED, H. Decision trees and random forests. In: PRESS, W.-I. (Ed.). **Condition Monitoring with Vibration Signals: Compressive Sampling and Learning Algorithms for Rotating Machines**. [S.l.]: Wiley, 2019. p. 199–224.

NARMADHA, N.; RATHIPRIYA, R. Triclustering: An evolution of clustering. IEEE, nov 2016.

NASUTION, A. K.; WIJAYA, S. H.; KUSUMA, W. A. Prediction of drug-target interaction on jamu formulas using machine learning approaches. In: **2019 International Conference on Advanced Computer Science and information Systems (ICACSIS)**. [S.l.]: IEEE, 2019.

PAHIKKALA, T. et al. Toward more realistic drug-target interaction predictions. **Briefings in Bioinformatics**, Oxford University Press (OUP), v. 16, n. 2, p. 325–337, apr 2014.

PAUWELS, E.; STOVEN, V.; YAMANISHI, Y. Predicting drug side-effect profiles: a chemical fragment-based approach. **BMC Bioinformatics**, Springer Science and Business Media LLC, v. 12, n. 1, p. 169, 2011.

PLIAKOS, K.; GEURTS, P.; VENS, C. Global multi-output decision trees for interaction prediction. **Machine Learning**, Springer Science and Business Media LLC, v. 107, n. 8-10, p. 1257–1281, may 2018.

PLIAKOS, K.; VENS, C. Drug-target interaction prediction with tree-ensemble learning and output space reconstruction. **BMC Bioinformatics**, Springer Science and Business Media LLC, v. 21, n. 1, feb 2020.

PONTES, B.; GIRÁLDEZ, R.; AGUILAR-RUIZ, J. S. Biclustering on expression data: A review. **Journal of Biomedical Informatics**, Elsevier BV, v. 57, p. 163–180, oct 2015.

RAO, V. S.; SRINIVAS, K.; SUJINI, G. N.; KUMAR, G. N. S. Protein-protein interaction detection: Methods and analysis. **International Journal of Proteomics**, Hindawi Limited, v. 2014, p. 1–12, 2014.

RASHID, M.; RAMASAMY, S.; RAGHAVA, G. P. A simple approach for predicting protein-protein interactions. **Current Protein & Peptide Science**, Bentham Science Publishers Ltd., v. 11, n. 7, p. 589–600, nov 2010.

READ, J.; PFAHRINGER, B.; HOLMES, G.; FRANK, E. Classifier chains for multi-label classification. **Machine Learning**, Springer Science and Business Media LLC, v. 85, n. 3, p. 333–359, jun 2011.

_____. Classifier chains: A review and perspectives. **Journal of Artificial Intelligence Research**, AI Access Foundation, v. 70, p. 683–718, feb 2021.

RESENDE, W. K.; NASCIMENTO, R. A.; XAVIER, C. R.; LOPES, I. F.; NOBRE, C. N. The use of support vector machine and genetic algorithms to predict protein function. In: **2012 IEEE International Conference on Systems, Man, and Cybernetics (SMC)**. [S.l.]: IEEE, 2012.

REY, D.; NEUHäUSER, M. Wilcoxon-signed-rank test. Springer Berlin Heidelberg, p. 1658–1659, 2011.

ROBINSON, P. K. Enzymes: principles and biotechnological applications. **Essays in Biochemistry**, Portland Press Ltd., v. 59, p. 1–41, oct 2015.

ROCHA, V. F.; VAREJÃO, F. M.; SEGATTO, M. E. V. Ensemble of classifier chains and decision templates for multi-label classification. **Knowledge and Information Systems**, Springer Science and Business Media LLC, v. 64, n. 3, p. 643–663, jan 2022.

ROSNER, B.; GLYNN, R. J.; LEE, M.-L. T. The wilcoxon signed rank test for paired comparisons of clustered data. **Biometrics**, Wiley, v. 62, n. 1, p. 185–192, jul 2005.

ROSSUM, G. V.; DRAKE, F. L. **Python 3 Reference Manual**. Scotts Valley, CA: CreateSpace, 2009. ISBN 1441412697.

ROY, S.; MARTINEZ, D.; PLATERO, H.; LANE, T.; WERNER-WASHBURNE, M. Exploiting amino acid composition for predicting protein-protein interactions. **PLoS ONE**, Public Library of Science (PLoS), v. 4, n. 11, p. e7813, nov 2009.

SACCÀ, C.; TESO, S.; DILIGENTI, M.; PASSERINI, A. Improved multi-level protein–protein interaction prediction with semantic-based regularization. **BMC Bioinformatics**, Springer Science and Business Media LLC, v. 15, n. 1, p. 103, 2014.

SARKAR, D.; SAHA, S. Machine-learning techniques for the prediction of protein–protein interactions. **Journal of Biosciences**, Springer Science and Business Media LLC, v. 44, n. 4, aug 2019.

SCHRYNEMACKERS, M.; KüFFNER, R.; GEURTS, P. On protocols and measures for the validation of supervised methods for the inference of biological networks. **Frontiers in Genetics**, Frontiers Media SA, v. 4, 2013.

SCHRYNEMACKERS, M.; WEHENKEL, L.; BABU, M. M.; GEURTS, P. Classifying pairs with trees for supervised biological network inference. **Molecular BioSystems**, Royal Society of Chemistry (RSC), v. 11, n. 8, p. 2116–2125, 2015.

SECHIDIS, K.; TSOUMAKAS, G.; VLAHAVAS, I. On the stratification of multi-label data. Springer Berlin Heidelberg, p. 145–158, 2011.

SHAH, H. Protein secondary structure prediction using support vector machines (SVMs). In: **2013 International Conference on Machine Intelligence and Research Advancement**. [S.l.]: IEEE, 2013.

SHASTRY, K. A.; SANJAY, H. A. Machine learning for bioinformatics. In: **Statistical Modelling and Machine Learning Principles for Bioinformatics Techniques, Tools, and Applications**. [S.l.]: Springer Singapore, 2020. p. 25–39.

SHEN, J. et al. Predicting protein-protein interactions based only on sequences information. **Proceedings of the National Academy of Sciences**, Proceedings of the National Academy of Sciences, v. 104, n. 11, p. 4337–4341, mar 2007.

SILVA, R. M. da et al. Prediction of seriniquinone-drug interactions by in vitro inhibition of human cytochrome p450 enzymes. **Toxicology in Vitro**, Elsevier BV, v. 65, p. 104820, jun 2020.

SNIDER, J. et al. Fundamentals of protein interaction network mapping. **Molecular Systems Biology**, EMBO, v. 11, n. 12, p. 848, dec 2015.

SWAMIDASS, S. J. Mining small-molecule screens to repurpose drugs. **Briefings in Bioinformatics**, Oxford University Press (OUP), v. 12, n. 4, p. 327–335, jun 2011.

TAHERI, S. M.; HESAMIAN, G. A generalization of the wilcoxon signed-rank test and its applications. **Statistical Papers**, Springer Science and Business Media LLC, v. 54, n. 2, p. 457–470, mar 2012.

THAFAR, M. A. et al. DTi2vec: Drug–target interaction prediction using network embedding and ensemble learning. **Journal of Cheminformatics**, Springer Science and Business Media LLC, v. 13, n. 1, sep 2021.

TOKOVAROV, M.; KARCZMAREK, P. A probabilistic generalization of isolation forest. **Information Sciences**, Elsevier BV, v. 584, p. 433–449, jan 2022.

TSOUMAKAS, G.; KATAKIS, I.; VLAHAVAS, I. Random k-labelsets for multilabel classification. **IEEE Transactions on Knowledge and Data Engineering**, Institute of Electrical and Electronics Engineers (IEEE), v. 23, n. 7, p. 1079–1089, jul 2011.

TSOUMAKAS, G.; ZHANG, M.-L.; ZHOU, Z.-H. Introduction to the special issue on learning from multi-label data. **Machine Learning**, Springer Science and Business Media LLC, v. 88, n. 1-2, p. 1–4, apr 2012.

VENS, C.; STRUYF, J.; SCHIETGAT, L.; DžEROSKI, S.; BLOCKEEL, H. Decision trees for hierarchical multi-label classification. **Machine Learning**, Kluwer Academic Publishers, USA, v. 73, n. 2, p. 185–214, nov. 2008. ISSN 0885-6125.

WANG, B.; LIU, Y.; YUN, J.; LIU, S. Application research of protein structure prediction based support vector machine. In: **2008 International Symposium on Knowledge Acquisition and Modeling**. [S.l.]: IEEE, 2008.

WANG, H.; WU, P. Prediction of RNA-protein interactions using conjoint triad feature and chaos game representation. **Bioengineered**, Informa UK Limited, v. 9, n. 1, p. 242–251, jan 2018.

WANG, J.; ZHANG, L.; JIA, L.; REN, Y.; YU, G. Protein-protein interactions prediction using a novel local conjoint triad descriptor of amino acid sequences. **International Journal of Molecular Sciences**, MDPI AG, v. 18, n. 11, p. 2373, nov 2017.

WANG, R.; LI, S.; WONG, M. H.; LEUNG, K. S. Drug-protein-disease association prediction and drug repositioning based on tensor decomposition. In: **2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)**. [S.l.]: IEEE, 2018.

WANG, Y.-B.; YOU, Z.-H.; LI, L.-P.; HUANG, Y.-A.; YI, H.-C. Detection of interactions between proteins by using legendre moments descriptor to extract discriminatory information embedded in PSSM. **Molecules**, MDPI AG, v. 22, n. 8, p. 1366, aug 2017.

WEHRMANN, J.; BARROS, R. C.; DÔRES, S. N. das; CERRI, R. Hierarchical multi-label classification with chained neural networks. In: **Proceedings of the Symposium on Applied Computing - SAC '17**. [S.l.]: ACM Press, 2017.

WOOLSON, R. F. Wilcoxon signed-rank test. John Wiley & Sons, Inc., sep 2008.

XING, S.; WALLMEROTH, N.; BERENDZEN, K. W.; GREFEN, C. Techniques for the analysis of protein-protein interactions in vivo. **Plant Physiology**, Oxford University Press (OUP), p. 004702016, apr 2016.

YAMANISHI, Y.; ARAKI, M.; GUTTERIDGE, A.; HONDA, W.; KANEHISA, M. Prediction of drug-target interaction networks from the integration of chemical and genomic spaces. **Bioinformatics**, Oxford University Press (OUP), v. 24, n. 13, p. i232–i240, jun 2008.

YI, H.-C. et al. A deep learning framework for robust and accurate prediction of ncRNA-protein interactions using evolutionary information. **Molecular Therapy - Nucleic Acids**, Elsevier BV, v. 11, p. 337–344, jun 2018.

YOU, Z.-H.; CHAN, K. C. C.; HU, P. Predicting protein-protein interactions from primary protein sequences using a novel multi-scale local feature representation scheme and the random forest. **PLOS ONE**, Public Library of Science (PLoS), v. 10, n. 5, p. e0125811, may 2015.

YOU, Z.-H.; LEI, Y.-K.; ZHU, L.; XIA, J.; WANG, B. Prediction of protein-protein interactions from amino acid sequences with ensemble extreme learning machines and principal component analysis. **BMC Bioinformatics**, Springer Science and Business Media LLC, v. 14, n. S8, may 2013.

ZAMIL, K. M. S.; RAHMAN, J. Prediction of protein-protein interaction from amino acid sequence using ensemble classifier. In: **2018 International Conference on Computer, Communication, Chemical, Material and Electronic Engineering (IC4ME2)**. [S.l.]: IEEE, 2018.

ZHANG, M.-L.; LI, Y.-K.; LIU, X.-Y.; GENG, X. Binary relevance for multi-label learning: an overview. **Frontiers of Computer Science**, Springer Science and Business Media LLC, v. 12, n. 2, p. 191–202, mar 2018.

ZHANG, M.-L.; ZHOU, Z.-H. A k-nearest neighbor based algorithm for multi-label classification. IEEE, 2005.

_____. ML-KNN: A lazy learning approach to multi-label learning. **Pattern Recognition**, Elsevier BV, v. 40, n. 7, p. 2038–2048, jul 2007.

ZHANG, Q. C.; PETREY, D.; NOREL, R.; HONIG, B. H. Protein interface conservation across structure space. **Proceedings of the National Academy of Sciences**, Proceedings of the National Academy of Sciences, v. 107, n. 24, p. 10896–10901, jun 2010.

ZHAO, C.; SACAN, A. UniAlign: protein structure alignment meets evolution. **Bioinformatics**, Oxford University Press (OUP), v. 31, n. 19, p. 3139–3146, jun 2015.

ZHAO, C.; ZANG, Y.; QUAN, W.; HU, X.; SACAN, A. HIV1-human protein-protein interaction prediction based on interface architecture similarity. In: **2017 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)**. [S.l.]: IEEE, 2017.

ZHAO, J.; DENG, Y.; JIANG, Z.; QING, H. G protein-coupled receptors (GPCRs) in alzheimer's disease: A focus on BACE1 related GPCRs. **Frontiers in Aging Neuroscience**, Frontiers Media SA, v. 8, mar 2016.

ZHAO, L.; ZHOU, S.; GUSTAFSSON, J.-Å. Nuclear receptors: recent drug discovery for cancer therapies. **Endocrine Reviews**, The Endocrine Society, mar 2019.

ZHOU, Z.-H. Semi-supervised learning. In: **Machine Learning**. [S.l.]: Springer Singapore, 2021. p. 315–341.

ZHU, Q. On the performance of matthews correlation coefficient (MCC) for imbalanced dataset. **Pattern Recognition Letters**, Elsevier BV, v. 136, p. 71–80, aug 2020.

ZONG, W.; HUANG, G.-B.; CHEN, Y. Weighted extreme learning machine for imbalance learning. **Neurocomputing**, Elsevier BV, v. 101, p. 229–242, feb 2013.

Appendix

APPENDIX A

Expanded Results

A.1 The Statistical Difference of PbSS methods considering all datasets

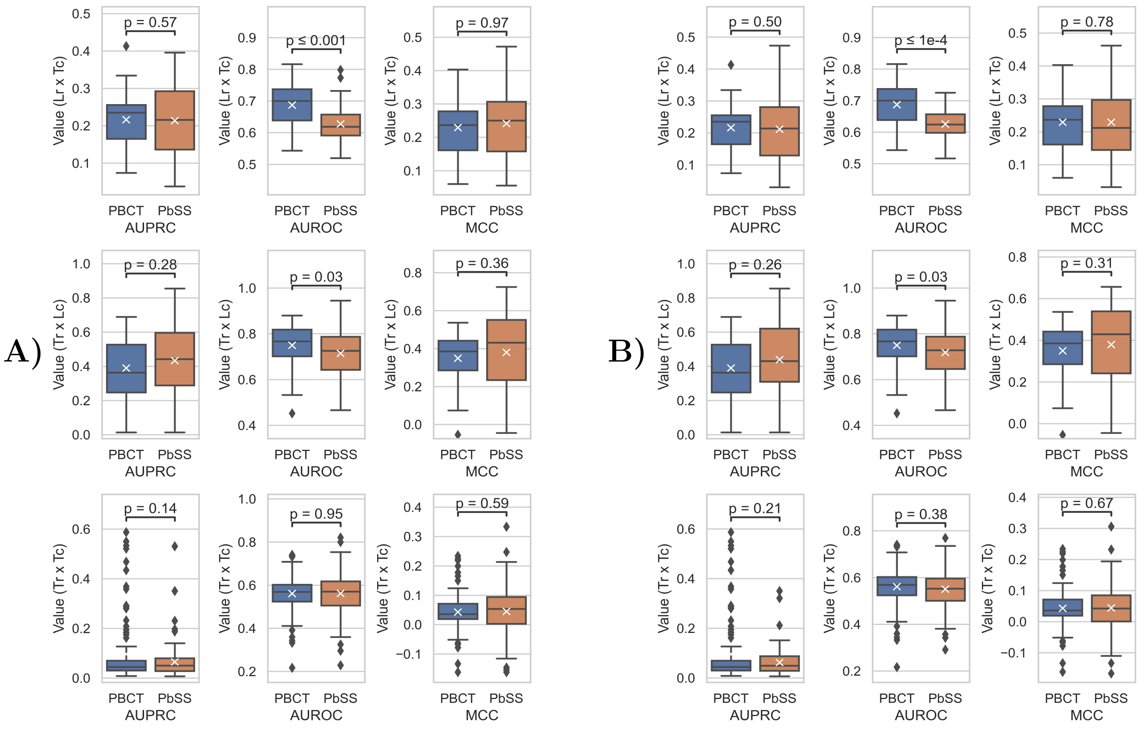


Figure 14 – Comparative statistical difference analysis of PBCT and PbSS performances considering the established scenario for all datasets, with a MLSKF CV Procedure. A) is PbSSd with dynamic w, B) is PbSS with $w = 0$.

A.2 The PbXGB with MLSKF

Table 11 – Results for Evaluation Measures obtained for compared methods, considering PBCT and PbXGB with MLSKF CV Procedure.

Measure	Data	Lr x Tc		Tr x Lc		Tr x Tc	
		PBCT	PbXGB	PBCT	PbXGB	PBCT	PbXGB
AUPRC	DPI-N	0.247 (± 0.083)	**0.419 (± 0.139)***	0.213 (± 0.105)	**0.348 (± 0.158)***	**0.252 (± 0.179)**	0.241 (± 0.154)
	DPI-I	0.210 (± 0.071)	**0.276 (± 0.066)***	0.474 (± 0.065)	**0.586 (± 0.086)***	0.056 (± 0.021)	**0.060 (± 0.025)**
	DPI-G	0.246 (± 0.039)	**0.282 (± 0.052)**	0.270 (± 0.046)	**0.335 (± 0.094)**	0.050 (± 0.025)	**0.080 (± 0.053)***
	DPI-E	0.163 (± 0.050)	**0.168 (± 0.073)**	**0.604 (± 0.058)**	0.596 (± 0.053)	**0.026 (± 0.010)**	0.015 (± 0.005)
AUROC	DPI-N	0.634 (± 0.067)	**0.708 (± 0.117)**	0.649 (± 0.103)	**0.655 (± 0.103)**	**0.570 (± 0.140)**	0.487 (± 0.122)
	DPI-I	**0.709 (± 0.081)**	0.641 (± 0.050)	**0.807 (± 0.033)**	0.797 (± 0.043)	**0.549 (± 0.040)**	0.533 (± 0.046)
	DPI-G	**0.714 (± 0.022)**	0.650 (± 0.032)	**0.713 (± 0.046)**	0.679 (± 0.059)	0.561 (± 0.077)	**0.572 (± 0.069)**
	DPI-E	**0.694 (± 0.065)**	0.608 (± 0.054)	**0.830 (± 0.036)**	0.821 (± 0.033)	**0.566 (± 0.036)**	0.530 (± 0.027)
MCC	DPI-N	0.217 (± 0.100)	**0.333 (± 0.152)**	0.237 (± 0.146)	**0.273 (± 0.151)**	**0.061 (± 0.131)**	-0.022 (± 0.114)
	DPI-I	0.247 (± 0.080)	**0.267 (± 0.082)**	0.441 (± 0.051)	**0.610 (± 0.083)***	**0.037 (± 0.031)**	0.025 (± 0.035)
	DPI-G	0.267 (± 0.026)	**0.305 (± 0.063)**	0.269 (± 0.066)	**0.397 (± 0.090)***	0.045 (± 0.055)	**0.052 (± 0.049)**
	DPI-E	0.185 (± 0.069)	**0.202 (± 0.066)**	0.452 (± 0.029)	**0.604 (± 0.044)***	0.029 (± 0.016)	**0.031 (± 0.027)**

A.3 Variations of Static PbSS values of w

As previously mentioned, here are presented the results for predictive performance referring to the behavior of the learning model through variations of $w = 0.25$, 0.5, 0.75 for PbSS presented. This study considered a MLSKF CV procedure for all prediction tasks (e.g., T_r x L_c, L_r x T_c, and T_r x T_c), evaluation criteria 4.3, and datasets (e.g., DPI-I, DPI-N, DPI-G, and DPI-E).

Table 12 – Results for Evaluation Measures obtained for compared methods, considering PBCT and PbSS with $w = 0.25$ and MLSKF CV Procedure.

Measure	Data	Lr x Tc		Tr x Lc		Tr x Tc	
		PBCT	PbSS	PBCT	PbSS	PBCT	PbSS
AUPRC	DPI-N	0.247 (± 0.083)	**0.300 (± 0.090)**	0.213 (± 0.105)	**0.356 (± 0.285)**	**0.252 (± 0.179)**	0.114 (± 0.113)
	DPI-I	**0.210 (± 0.071)**	0.195 (± 0.067)	0.474 (± 0.065)	**0.576 (± 0.101)**	0.056 (± 0.021)	**0.066 (± 0.021)**
	DPI-G	**0.246 (± 0.039)**	0.231 (± 0.051)	0.270 (± 0.046)	**0.474 (± 0.151)**	0.050 (± 0.025)	**0.060 (± 0.030)**
	DPI-E	**0.163 (± 0.050)**	0.130 (± 0.100)	**0.604 (± 0.058)**	0.327 (± 0.047)	**0.026 (± 0.010)**	0.020 (± 0.010)
AUROC	DPI-N	0.634 (± 0.067)	**0.685 (± 0.065)**	**0.649 (± 0.103)**	0.638 (± 0.145)	**0.570 (± 0.140)**	0.517 (± 0.152)
	DPI-I	**0.709 (± 0.081)**	0.586 (± 0.038)	**0.807 (± 0.033)**	0.781 (± 0.043)	0.549 (± 0.040)	**0.578 (± 0.051)**
	DPI-G	**0.714 (± 0.022)**	0.626 (± 0.030)	0.713 (± 0.046)	**0.743 (± 0.072)**	0.561 (± 0.077)	**0.585 (± 0.064)**
	DPI-E	**0.694 (± 0.065)**	0.615 (± 0.050)	**0.830 (± 0.036)**	0.696 (± 0.031)	0.566 (± 0.036)	**0.568 (± 0.092)**
MCC	DPI-N	0.217 (± 0.100)	**0.315 (± 0.105)**	0.237 (± 0.146)	**0.256 (± 0.254)**	**0.061 (± 0.131)**	0.017 (± 0.141)
	DPI-I	**0.247 (± 0.080)**	0.210 (± 0.095)	0.441 (± 0.051)	**0.570 (± 0.087)**	0.037 (± 0.031)	**0.067 (± 0.046)**
	DPI-G	0.267 (± 0.026)	**0.295 (± 0.080)**	0.269 (± 0.066)	**0.443 (± 0.113)**	0.045 (± 0.055)	**0.064 (± 0.048)**
	DPI-E	**0.185 (± 0.069)**	0.146 (± 0.073)	**0.452 (± 0.029)**	0.253 (± 0.026)	0.029 (± 0.016)	**0.033 (± 0.047)**

Table 13 – Results for Evaluation Measures obtained for compared methods, considering PBCT and PbSS with w = 0.5 and MLSKF CV Procedure.

Measure	Data	*Lr* x *Tc*		*Tr* x *Lc*		*Tr* x *Tc*	
		PBCT	PbSS	PBCT	PbSS	PBCT	PbSS
AUPRC	DPI-N	0.247 (± 0.083)	**0.300 (± 0.090)**	0.213 (± 0.105)	**0.356 (± 0.285)**	**0.252 (± 0.179)**	0.115 (± 0.114)
	DPI-I	**0.210 (± 0.071)**	0.195 (± 0.067)	0.474 (± 0.065)	**0.576 (± 0.101)**	0.056 (± 0.021)	**0.066 (± 0.021)**
	DPI-G	**0.246 (± 0.039)**	0.231 (± 0.051)	0.270 (± 0.046)	**0.474 (± 0.151)**	0.050 (± 0.025)	**0.060 (± 0.030)**
	DPI-E	**0.163 (± 0.050)**	0.130 (± 0.100)	**0.604 (± 0.058)**	0.327 (± 0.047)	**0.026 (± 0.010)**	0.020 (± 0.010)
AUROC	DPI-N	0.634 (± 0.067)	**0.685 (± 0.065)**	**0.649 (± 0.103)**	0.638 (± 0.145)	**0.570 (± 0.140)**	0.517 (± 0.151)
	DPI-I	**0.709 (± 0.081)**	0.586 (± 0.038)	**0.807 (± 0.033)**	0.781 (± 0.043)	0.549 (± 0.040)	**0.578 (± 0.051)**
	DPI-G	**0.714 (± 0.022)**	0.626 (± 0.030)	0.713 (± 0.046)	**0.743 (± 0.072)**	0.561 (± 0.077)	**0.585 (± 0.065)**
	DPI-E	**0.694 (± 0.065)**	0.615 (± 0.050)	**0.830 (± 0.036)**	0.696 (± 0.031)	0.566 (± 0.036)	**0.568 (± 0.092)**
MCC	DPI-N	0.217 (± 0.100)	**0.315 (± 0.105)**	0.237 (± 0.146)	**0.256 (± 0.254)**	**0.061 (± 0.131)**	0.016 (± 0.138)
	DPI-I	**0.247 (± 0.080)**	0.210 (± 0.095)	0.441 (± 0.051)	**0.570 (± 0.087)**	0.037 (± 0.031)	**0.067 (± 0.046)**
	DPI-G	0.267 (± 0.026)	**0.295 (± 0.080)**	0.269 (± 0.066)	**0.443 (± 0.113)**	0.045 (± 0.055)	**0.063 (± 0.049)**
	DPI-E	**0.185 (± 0.069)**	0.146 (± 0.073)	**0.452 (± 0.029)**	0.253 (± 0.026)	0.029 (± 0.016)	**0.033 (± 0.047)**

Table 14 – Results for Evaluation Measures obtained for compared methods, considering PBCT and PbSS with w = 0.75 and MLSKF CV Procedure.

Measure	Data	*Lr* x *Tc*		*Tr* x *Lc*		*Tr* x *Tc*	
		PBCT	PbSS	PBCT	PbSS	PBCT	PbSS
AUPRC	DPI-N	0.247 (± 0.083)	**0.298 (± 0.090)**	0.213 (± 0.105)	**0.356 (± 0.285)**	**0.252 (± 0.179)**	0.115 (± 0.114)
	DPI-I	**0.210 (± 0.071)**	0.195 (± 0.067)	0.474 (± 0.065)	**0.576 (± 0.101)**	0.056 (± 0.021)	**0.066 (± 0.021)**
	DPI-G	**0.246 (± 0.039)**	0.231 (± 0.051)	0.270 (± 0.046)	**0.474 (± 0.151)**	0.050 (± 0.025)	**0.060 (± 0.031)**
	DPI-E	**0.163 (± 0.050)**	0.130 (± 0.100)	**0.604 (± 0.058)**	0.327 (± 0.047)	**0.026 (± 0.010)**	0.020 (± 0.010)
AUROC	DPI-N	0.634 (± 0.067)	**0.683 (± 0.068)**	**0.649 (± 0.103)**	0.638 (± 0.145)	**0.570 (± 0.140)**	0.521 (± 0.145)
	DPI-I	**0.709 (± 0.081)**	0.586 (± 0.038)	**0.807 (± 0.033)**	0.781 (± 0.043)	0.549 (± 0.040)	**0.578 (± 0.051)**
	DPI-G	**0.714 (± 0.022)**	0.626 (± 0.030)	0.713 (± 0.046)	**0.743 (± 0.072)**	0.561 (± 0.077)	**0.585 (± 0.065)**
	DPI-E	**0.694 (± 0.065)**	0.615 (± 0.050)	**0.830 (± 0.036)**	0.696 (± 0.031)	0.566 (± 0.036)	**0.568 (± 0.092)**
MCC	DPI-N	0.217 (± 0.100)	**0.310 (± 0.112)**	0.237 (± 0.146)	**0.256 (± 0.254)**	**0.061 (± 0.131)**	0.019 (± 0.134)
	DPI-I	**0.247 (± 0.080)**	0.210 (± 0.095)	0.441 (± 0.051)	**0.570 (± 0.087)**	0.037 (± 0.031)	**0.067 (± 0.046)**
	DPI-G	0.267 (± 0.026)	**0.295 (± 0.080)**	0.269 (± 0.066)	**0.443 (± 0.113)**	0.045 (± 0.055)	**0.063 (± 0.048)**
	DPI-E	**0.185 (± 0.069)**	0.146 (± 0.073)	**0.452 (± 0.029)**	0.253 (± 0.026)	0.029 (± 0.016)	**0.033 (± 0.047)**

A.4 The Global Single Output obtained Results

This section presents the predictive performance results of the experimental procedure of GSO correlated approaches in a comparative study with PBCT. Among these are the Clus, the XGBoost, the RF, and the KNN. This experimental procedure considered all prediction tasks (e.g., T_r x L_c) for the DPI-N, DPI-I, and DPI-G datasets (i.e., and the DPI-E for Clus), evaluation measures 4.3, as well as a MLSKF CV procedure, and the results in predictive performance is defined in the tables below. i.e., the DPI-E dataset was disregarded for XGBoost, RF, and KNN since it exceeded the runtime limit (i.e., 24 hours), or available resources (i.e., in terms of memory and execution time). Finally, despite the computational cost, the GSO approach methods showed gains in predictive performance relevant to PBCT in some cases of T_r x L_c and L_r x T_c, these gains are even reflected in the results obtained for PbXGB and PbSS. However, it should be noted that these gains were not reflected in predictive performance gains for the T_r x T_c task.

These scenarios consider the previously determined experimental settings (see Section 4.4), with additional details referring to the Hyperparameters used in the Grid-Search procedure. For XGBoost we considered the *number of estimators* = $\{10, 300, 600\}$,

Table 15 – Results for Evaluation Measures obtained for compared methods, considering PBCT and GSO Clus and MLSKF CV Procedure.

Measure	Data	*Lr* x *Tc*		*Tr* x *Lc*		*Tr* x *Tc*	
		PBCT	CLUS	PBCT	CLUS	PBCT	CLUS
AUPRC	DPI-N	0.247 (± 0.083)	**0.407 (± 0.121)**	0.213 (± 0.105)	**0.339 (± 0.270)**	**0.252 (± 0.179)**	0.077 (± 0.060)
	DPI-I	**0.210 (± 0.071)**	0.207 (± 0.105)	**0.474 (± 0.065)**	0.462 (± 0.072)	**0.056 (± 0.021)**	0.036 (± 0.009)
	DPI-G	**0.246 (± 0.039)**	0.152 (± 0.046)	**0.270 (± 0.046)**	0.206 (± 0.095)	**0.050 (± 0.025)**	0.031 (± 0.007)
	DPI-E	**0.163 (± 0.050)**	0.127 (± 0.033)	**0.604 (± 0.058)**	0.538 (± 0.049)	**0.026 (± 0.010)**	0.011 (± 0.002)
AUROC	DPI-N	0.634 (± 0.067)	**0.650 (± 0.060)**	**0.649 (± 0.103)**	0.587 (± 0.108)	**0.570 (± 0.140)**	0.480 (± 0.132)
	DPI-I	**0.709 (± 0.081)**	0.606 (± 0.071)	**0.807 (± 0.033)**	0.741 (± 0.035)	**0.549 (± 0.040)**	0.494 (± 0.020)
	DPI-G	**0.714 (± 0.022)**	0.574 (± 0.028)	**0.713 (± 0.046)**	0.598 (± 0.045)	**0.561 (± 0.077)**	0.496 (± 0.032)
	DPI-E	**0.694 (± 0.065)**	0.586 (± 0.025)	**0.830 (± 0.036)**	0.800 (± 0.025)	**0.566 (± 0.036)**	0.503 (± 0.016)
MCC	DPI-N	0.217 (± 0.100)	**0.360 (± 0.133)**	**0.237 (± 0.146)**	0.211 (± 0.249)	**0.061 (± 0.131)**	-0.023 (± 0.126)
	DPI-I	**0.247 (± 0.080)**	0.226 (± 0.124)	0.441 (± 0.051)	**0.536 (± 0.067)**	**0.037 (± 0.031)**	-0.007 (± 0.022)
	DPI-G	**0.267 (± 0.026)**	0.170 (± 0.056)	**0.269 (± 0.066)**	0.242 (± 0.131)	**0.045 (± 0.055)**	-0.005 (± 0.033)
	DPI-E	**0.185 (± 0.069)**	0.181 (± 0.048)	0.452 (± 0.029)	**0.624 (± 0.043)**	**0.029 (± 0.016)**	0.002 (± 0.013)

Table 16 – Results for Evaluation Measures obtained for compared methods, considering PBCT and GSO XGBoost with Grid Search and MLSKF CV Procedure.

Measure	Data	*Lr* x *Tc*		*Tr* x *Lc*		*Tr* x *Tc*	
		PBCT	XGBoost	PBCT	XGBoost	PBCT	XGBoost
AUPRC	DPI-N	0.247 (± 0.083)	**0.487 (± 0.161)**	0.213 (± 0.105)	**0.336 (± 0.202)**	**0.252 (± 0.179)**	0.070 (± 0.056)
	DPI-I	0.210 (± 0.071)	**0.312 (± 0.109)**	0.474 (± 0.065)	**0.774 (± 0.070)**	**0.056 (± 0.021)**	0.037 (± 0.011)
	DPI-G	0.246 (± 0.039)	**0.335 (± 0.081)**	0.270 (± 0.046)	**0.443 (± 0.104)**	**0.050 (± 0.025)**	0.031 (± 0.009)
AUROC	DPI-N	0.634 (± 0.067)	**0.667 (± 0.080)**	**0.649 (± 0.103)**	0.638 (± 0.120)	**0.570 (± 0.140)**	0.467 (± 0.114)
	DPI-I	**0.709 (± 0.081)**	0.633 (± 0.049)	0.807 (± 0.033)	**0.862 (± 0.028)**	**0.549 (± 0.040)**	0.504 (± 0.030)
	DPI-G	**0.714 (± 0.022)**	0.618 (± 0.025)	**0.713 (± 0.046)**	0.669 (± 0.058)	**0.561 (± 0.077)**	0.498 (± 0.035)
MCC	DPI-N	0.217 (± 0.100)	**0.356 (± 0.141)**	0.237 (± 0.146)	**0.293 (± 0.249)**	**0.061 (± 0.131)**	-0.022 (± 0.144)
	DPI-I	0.247 (± 0.080)	**0.358 (± 0.102)**	0.441 (± 0.051)	**0.757 (± 0.056)**	**0.037 (± 0.031)**	0.004 (± 0.027)
	DPI-G	0.267 (± 0.026)	**0.348 (± 0.067)**	0.269 (± 0.066)	**0.449 (± 0.097)**	**0.045 (± 0.055)**	-0.003 (± 0.035)

learning rate = $\{0.02, 0.1\}$, and the *scale pos weight* = $\{10, 100, 1000\}$ for all prediction tasks. For RF, we considered *number of estimators* = $\{5, 10, 25, 50, 100, 200, 500\}$, the *max depth* = 5, and bootstrap = $\{True, False\}$. Finally, for KNN, we considered *number of neighbors* = $\{3, 5, 10, 20\}$.

Table 17 – Results for Evaluation Measures obtained for compared methods, considering PBCT and GSO RF with Grid Search and MLSKF CV Procedure.

Measure	Data	*Lr* x *Tc*		*Tr* x *Lc*		*Tr* x *Tc*	
		PBCT	RF	PBCT	RF	PBCT	RF
AUPRC	DPI-N	0.247 (± 0.083)	**0.377 (± 0.181)**	0.213 (± 0.105)	**0.360 (± 0.219)**	**0.252 (± 0.179)**	0.085 (± 0.067)
	DPI-I	**0.210 (± 0.071)**	0.165 (± 0.088)	**0.474 (± 0.065)**	0.437 (± 0.079)	**0.056 (± 0.021)**	0.038 (± 0.010)
	DPI-G	**0.246 (± 0.039)**	0.180 (± 0.071)	**0.270 (± 0.046)**	0.252 (± 0.042)	**0.050 (± 0.025)**	0.031 (± 0.009)
AUROC	DPI-N	**0.634 (± 0.067)**	0.567 (± 0.044)	**0.649 (± 0.103)**	0.530 (± 0.064)	**0.570 (± 0.140)**	0.480 (± 0.131)
	DPI-I	**0.709 (± 0.081)**	0.518 (± 0.020)	**0.807 (± 0.033)**	0.584 (± 0.023)	**0.549 (± 0.040)**	0.511 (± 0.030)
	DPI-G	**0.714 (± 0.022)**	0.502 (± 0.006)	**0.713 (± 0.046)**	0.533 (± 0.011)	**0.561 (± 0.077)**	0.487 (± 0.042)
MCC	DPI-N	0.217 (± 0.100)	**0.286 (± 0.168)**	**0.237 (± 0.146)**	0.098 (± 0.205)	**0.061 (± 0.131)**	-0.012 (± 0.139)
	DPI-I	**0.247 (± 0.080)**	0.115 (± 0.115)	**0.441 (± 0.051)**	0.384 (± 0.063)	**0.037 (± 0.031)**	0.009 (± 0.023)
	DPI-G	**0.267 (± 0.026)**	0.006 (± 0.018)	**0.269 (± 0.066)**	0.220 (± 0.046)	**0.045 (± 0.055)**	-0.008 (± 0.030)

We can highlight that in these GSO scenarios the increase in memory consumption and computational resources of our computational environment is remarkable (i.e., sometimes presenting a consumption 10 times higher or greater than the original PBCT). This memory increase is credited to the complexity of the models involved and the significant

Table 18 – Results for Evaluation Measures obtained for compared methods, considering PBCT and GSO KNN with Grid Search and MLSKF CV Procedure.

Measure	Data	*Lr* x *Tc*		*Tr* x *Lc*		*Tr* x *Tc*	
		PBCT	KNN	PBCT	KNN	PBCT	KNN
AUPRC	DPI-N	0.247 (± 0.083)	**0.477 (± 0.142)**	0.213 (± 0.105)	**0.314 (± 0.204)**	**0.252 (± 0.179)**	0.071 (± 0.058)
	DPI-I	0.210 (± 0.071)	**0.323 (± 0.085)**	0.474 (± 0.065)	**0.801 (± 0.067)**	**0.056 (± 0.021)**	0.037 (± 0.011)
	DPI-G	0.246 (± 0.039)	**0.325 (± 0.076)**	0.270 (± 0.046)	**0.600 (± 0.104)**	**0.050 (± 0.025)**	0.031 (± 0.011)
AUROC	DPI-N	0.634 (± 0.067)	**0.660 (± 0.059)**	**0.649 (± 0.103)**	0.587 (± 0.071)	**0.570 (± 0.140)**	0.522 (± 0.129)
	DPI-I	**0.709 (± 0.081)**	0.643 (± 0.063)	0.807 (± 0.033)	**0.847 (± 0.030)**	**0.549 (± 0.040)**	0.502 (± 0.023)
	DPI-G	**0.714 (± 0.022)**	0.628 (± 0.027)	0.713 (± 0.046)	**0.741 (± 0.084)**	**0.561 (± 0.077)**	0.496 (± 0.046)
MCC	DPI-N	0.217 (± 0.100)	**0.431 (± 0.157)**	0.237 (± 0.146)	**0.281 (± 0.205)**	**0.061 (± 0.131)**	0.014 (± 0.117)
	DPI-I	0.247 (± 0.080)	**0.327 (± 0.104)**	0.441 (± 0.051)	**0.737 (± 0.059)**	**0.037 (± 0.031)**	0.002 (± 0.021)
	DPI-G	0.267 (± 0.026)	**0.330 (± 0.059)**	0.269 (± 0.066)	**0.566 (± 0.102)**	**0.045 (± 0.055)**	-0.003 (± 0.038)

increase in data size in the GSO representation due to the concatenation of features in object space.

A.5 The Local Multiple Output obtained Results

The predictive performance resulting from the experimental procedure refers to the comparative study between PBCT and LMO correlated learning methods. The experimental procedure defined here considered all datasets, a MLSKF CV procedure for all prediction tasks (e.g., L_r x T_c), considering all evaluation measures 4.3, and predictive performance results are presented in the tables below. i.e., following the previously determined Scenario (see Section 4.4) for MLKNN we considered $k = \{2, 3, 5\}$ and $s = \{0.5, 0.7, 1\}$ for the Grid-Search Procedure, and for ECCRF we consider *number of chains* = 10, and *number of estimators* = 20 for RF.

Table 19 – Results for Evaluation Measures obtained for compared methods, considering PBCT and LMO Clus with MLSKF CV Procedure.

Measure	Data	*Lr* x *Tc*		*Tr* x *Lc*		*Tr* x *Tc*	
		PBCT	CLUS	PBCT	CLUS	PBCT	CLUS
AUPRC	DPI-N	0.247 (± 0.083)	**0.335 (± 0.176)**	0.213 (± 0.105)	**0.411 (± 0.277)**	**0.252 (± 0.179)**	0.116 (± 0.120)
	DPI-I	**0.210 (± 0.071)**	0.137 (± 0.040)	**0.474 (± 0.065)**	0.461 (± 0.081)	**0.056 (± 0.021)**	0.044 (± 0.014)
	DPI-G	**0.246 (± 0.039)**	0.167 (± 0.076)	**0.270 (± 0.046)**	0.254 (± 0.106)	**0.050 (± 0.025)**	0.047 (± 0.027)
	DPI-E	**0.163 (± 0.050)**	0.118 (± 0.054)	**0.604 (± 0.058)**	0.321 (± 0.054)	**0.026 (± 0.010)**	0.017 (± 0.006)
AUROC	DPI-N	0.634 (± 0.067)	**0.662 (± 0.116)**	**0.649 (± 0.103)**	0.648 (± 0.266)	**0.570 (± 0.140)**	0.561 (± 0.143)
	DPI-I	**0.709 (± 0.081)**	0.571 (± 0.040)	**0.807 (± 0.033)**	0.771 (± 0.051)	**0.549 (± 0.040)**	0.519 (± 0.024)
	DPI-G	**0.714 (± 0.022)**	0.594 (± 0.037)	**0.713 (± 0.046)**	0.632 (± 0.068)	**0.561 (± 0.077)**	0.530 (± 0.049)
	DPI-E	**0.694 (± 0.065)**	0.589 (± 0.053)	**0.830 (± 0.036)**	0.775 (± 0.054)	**0.566 (± 0.036)**	0.524 (± 0.019)
MCC	DPI-N	0.217 (± 0.100)	**0.334 (± 0.200)**	0.237 (± 0.146)	**0.481 (± 0.323)**	**0.061 (± 0.131)**	0.047 (± 0.138)
	DPI-I	**0.247 (± 0.080)**	0.173 (± 0.085)	0.441 (± 0.051)	**0.590 (± 0.093)**	**0.037 (± 0.031)**	0.023 (± 0.027)
	DPI-G	**0.267 (± 0.026)**	0.233 (± 0.100)	0.269 (± 0.066)	**0.321 (± 0.139)**	**0.045 (± 0.055)**	0.035 (± 0.058)
	DPI-E	0.185 (± 0.069)	**0.211 (± 0.142)**	0.452 (± 0.029)	**0.581 (± 0.095)**	**0.029 (± 0.016)**	0.025 (± 0.019)

We can highlight that the approach methods LMO considered, in a general context, showed gains in all predictive tasks. While Clus (19) and MLKNN (20) showed gains mainly for the prediction tasks T_r x L_c and L_r x T_c, ECCRF (21) also showed gains (i.e., concerning the PBCT, PbXGB, and PbSS) on T_r x T_c tasks.

Table 20 – Results for Evaluation Measures obtained for compared methods, considering PBCT and LMO MLKNN with Grid Search and MLSKF CV Procedure.

Measure	Data	*Lr* x *Tc*		*Tr* x *Lc*		*Tr* x *Tc*	
		PBCT	MLKNN	PBCT	MLKNN	PBCT	MLKNN
AUPRC	DPI-N	0.247 (± 0.083)	**0.486 (± 0.119)**	0.213 (± 0.105)	**0.639 (± 0.298)**	**0.252 (± 0.179)**	0.142 (± 0.160)
	DPI-I	**0.210 (± 0.071)**	0.208 (± 0.057)	0.474 (± 0.065)	**0.808 (± 0.067)**	**0.056 (± 0.021)**	0.047 (± 0.016)
	DPI-G	**0.246 (± 0.039)**	0.235 (± 0.069)	0.270 (± 0.046)	**0.650 (± 0.070)**	0.050 (± 0.025)	**0.054 (± 0.024)**
	DPI-E	**0.163 (± 0.050)**	0.153 (± 0.043)	0.604 (± 0.058)	**0.784 (± 0.065)**	**0.026 (± 0.010)**	0.017 (± 0.007)
AUROC	DPI-N	0.634 (± 0.067)	**0.714 (± 0.077)**	0.649 (± 0.103)	**0.701 (± 0.274)**	**0.570 (± 0.140)**	0.547 (± 0.138)
	DPI-I	**0.709 (± 0.081)**	0.625 (± 0.076)	0.807 (± 0.033)	**0.875 (± 0.043)**	**0.549 (± 0.040)**	0.544 (± 0.024)
	DPI-G	**0.714 (± 0.022)**	0.631 (± 0.038)	0.713 (± 0.046)	**0.788 (± 0.083)**	**0.561 (± 0.077)**	0.547 (± 0.050)
	DPI-E	**0.694 (± 0.065)**	0.632 (± 0.067)	0.830 (± 0.036)	**0.877 (± 0.038)**	**0.566 (± 0.036)**	0.531 (± 0.043)
MCC	DPI-N	0.217 (± 0.100)	**0.480 (± 0.134)**	0.237 (± 0.146)	**0.566 (± 0.319)**	**0.061 (± 0.131)**	0.052 (± 0.146)
	DPI-I	0.247 (± 0.080)	**0.346 (± 0.154)**	0.441 (± 0.051)	**0.793 (± 0.069)**	0.037 (± 0.031)	**0.050 (± 0.027)**
	DPI-G	0.267 (± 0.026)	**0.349 (± 0.083)**	0.269 (± 0.066)	**0.616 (± 0.123)**	0.045 (± 0.055)	**0.045 (± 0.048)**
	DPI-E	0.185 (± 0.069)	**0.324 (± 0.167)**	0.452 (± 0.029)	**0.809 (± 0.052)**	**0.029 (± 0.016)**	0.018 (± 0.024)

Table 21 – Results for Evaluation Measures obtained for compared methods, considering PBCT and LMO ECCRF with MLSKF CV Procedure.

Measure	Data	*Lr* x *Tc*		*Tr* x *Lc*		*Tr* x *Tc*	
		PBCT	ECCRF	PBCT	ECCRF	PBCT	ECCRF
AUPRC	DPI-N	0.247 (± 0.083)	**0.518 (± 0.173)**	0.213 (± 0.105)	**0.706 (± 0.282)**	**0.252 (± 0.179)**	0.123 (± 0.161)
	DPI-I	0.210 (± 0.071)	**0.463 (± 0.128)**	0.474 (± 0.065)	**0.809 (± 0.061)**	0.056 (± 0.021)	**0.085 (± 0.024)**
	DPI-G	0.246 (± 0.039)	**0.392 (± 0.104)**	0.270 (± 0.046)	**0.528 (± 0.079)**	0.050 (± 0.025)	**0.064 (± 0.049)**
	DPI-E	0.163 (± 0.050)	**0.507 (± 0.064)**	0.604 (± 0.058)	**0.788 (± 0.055)**	0.026 (± 0.010)	**0.036 (± 0.030)**
AUROC	DPI-N	0.634 (± 0.067)	**0.667 (± 0.067)**	**0.649 (± 0.103)**	0.619 (± 0.256)	**0.570 (± 0.140)**	0.508 (± 0.063)
	DPI-I	**0.709 (± 0.081)**	0.618 (± 0.049)	**0.807 (± 0.033)**	0.801 (± 0.041)	**0.549 (± 0.040)**	0.536 (± 0.019)
	DPI-G	**0.714 (± 0.022)**	0.602 (± 0.023)	**0.713 (± 0.046)**	0.663 (± 0.054)	**0.561 (± 0.077)**	0.515 (± 0.024)
	DPI-E	**0.694 (± 0.065)**	0.579 (± 0.028)	**0.830 (± 0.036)**	0.817 (± 0.039)	**0.566 (± 0.036)**	0.512 (± 0.015)
MCC	DPI-N	0.217 (± 0.100)	**0.451 (± 0.137)**	0.237 (± 0.146)	**0.361 (± 0.333)**	**0.061 (± 0.131)**	0.019 (± 0.110)
	DPI-I	0.247 (± 0.080)	**0.378 (± 0.134)**	0.441 (± 0.051)	**0.703 (± 0.086)**	0.037 (± 0.031)	**0.068 (± 0.033)**
	DPI-G	0.267 (± 0.026)	**0.332 (± 0.078)**	0.269 (± 0.066)	**0.463 (± 0.095)**	**0.045 (± 0.055)**	0.036 (± 0.057)
	DPI-E	0.185 (± 0.069)	**0.361 (± 0.073)**	0.452 (± 0.029)	**0.763 (± 0.072)**	**0.029 (± 0.016)**	0.027 (± 0.032)