María Isabel Pérez Ribera

Computational prediction of molecular tandem mass spectra using deep learning algorithms

Degree Final Project

Supervised by Dr Marta Sales Pardo and Dr Roger Guimerà Manrique

Bachelor's degree of Biomedical Engineering

UNIVERSITAT ROVIRA i VIRGILI

Tarragona
2022

Abstract

Disease diagnosis and personalized medicine based on metabolomics using changes in metabolite concentrations, are attracting the attention of more and more researchers. Nevertheless, compound identification remains a problem in most metabolomics studies based on mass spectrometry (MS), as the percentage of known MS molecular spectra is very low. Software tools for metabolite annotation and identification can solve this problem by predicting the mass spectra of molecules through Deep Learning.

We have proposed different methodologies to obtain the best prediction of the tandem mass spectra of molecules from their molecular formula, comparing different types of Neural Networks. Also, we have obtained a very good prediction ability, achieving better results than the best in silico tool for the prediction of MS/MS spectra up to date.

Keywords: machine learning; neural networks; metabolomics; mass spectrometry.

Acknowledgments

I would like too express my most sincere gratitude to Marta and Roger for giving me this opportunity and for their help and guidance during this beautiful project, from which I have learned a lot.

I would also like to thank the whole team of the SEES:lab research group, who have not hesitated for a second to solve all the problems and questions I have encountered.

Finally, to my family, friends and especially to my roommates, who have been a great support during this intense year.

Contents

1 Introduction

Currently, medicine is undergoing numerous changes as a consequence of the advances in technology and the interest of society in health, which has recently increased as a result of the global pandemic of COVID-19. Areas of medicine such as personalized medicine are gaining widespread interest, due to people's concern for improving their quality of life. Personalized medicine (PM), also known as Precision medicine, consist in therapies taylored to the individual needs of patients. PM takes into account genetics, biomarkers, phenotypes, or psychosocial characteristics that distinguish one patient from another with similar clinical manifestations [1]. In this way, PM is now becoming an integral personalization strategy in healthcare, which will evolve to ultimately provide patients with better monitoring of their health, together with individualized prevention, early and proactive diagnosis, and personalized treatment [2].

In this way, thanks to improvements, evolutions, and joint work in sectors such as genetics, informatics, image processing, cell sorting, proteomics, and metabolomics, the scope of Precisionin personalized medicine is widening, giving rise to important applications in the prognosis and treatment of diseases [1]. The onset and progression of a disease can lead to alterations in the internal balance of the biological system, its components, and the interactions between them. To study such alterations from a systemic approach, omics sciences such as genomics, transcriptomics, proteomics, and metabolomics are crucial [3].

Metabolomics is an emerging field that aims to exhaustively measure metabolites concentrates and low molecular weight molecules in a given organism or biological sample [4], [5]. Disease diagnosis and personalized medicine based on metabolomics using changes in metabolite concentrations are attracting the attention of many researchers.

One of the most common examples is the use of metabolomics to diagnose cancer states directly in body fluids, thus facilitating rapid screening for this type of disease, or to classify histological samples of different tumours by extraction of metabolites followed by mass spectrometric analysis [6]. In a study by Chan et al. [7], metabolites from biopsied colorectal tumours and from normal mucosa obtained from 31 patients with colorectal cancer (CRC) were analysed. Using discriminant analysis it was possible to discern between benign and malignant mucosa, and to identify 31 biomarkers associated with metabolomic perturbations expected in CRC disease (elevated tissue hypoxia, glycolysis, nucleotide biosynthesis, lipid metabolism, inflammation and steroid metabolism) [7]. Thus, this area of study and metabolomic profiling can be used in a variety of ways, however, two approaches can be established depending on whether it is based on all the measured concentrations or on a selection of biomarkers [6].

On the one hand, it is possible to base a diagnostic method purely on metabolic fingerprinting by applying pattern recognition techniques to determine whether a suspected patient has a certain disease [6]. This unique fingerprint is the composition of the metabolome, which represents all relevant primary and secondary metabolites, typically with molecular masses of less than 1.5-2 kDa, in a given organism. This fingerprint changes dynamically in response to external effects and may show an evolution in the phenotype of the organism [8]. So, this strategy can potentially be useful for rapid screening of large patient populations if a sufficiently specific method is developed. On the other hand, the second approach is to develop metabolomics strategies to find a single biomarker that is specific enough to reliably detect a specific disease. Nevertheless, often more than one metabolite is needed for diagnosis and the optimal approach is somewhere between the two methodologies, depending on the

objectives sought [6].

In recent years, the simultaneous measurement of a large number of metabolites in a single biological sample has been improved by the development of high-performance analytical techniques such as nuclear magnetic resonance spectroscopy (NMR) and mass spectrometry (MS) [6]. NMR and MS are the two main analytical approaches for metabolomic studies, which complement each other. While MS-based approaches have a higher sensitivity, NMR spectrometry is highly reproducible and requires minimal sample preparation [9]. Moreover, optical spectrometric techniques such as Raman spectroscopy and infrared (IR) spectroscopy are also used [6]. In this work, we will focus on mass spectrometry, whose main advantages are specificity, sensitivity, and the ability to analyze complex samples [10].

Mass spectrometry is an analytical technique that accurately measures the molecular masses of compounds and individual atoms by converting them into charged ions. This requires a mass spectrometer consisting of an ion source, a mass analyser, a detector, a data system, a vacuum system, and electronic control units. The data is represented as a mass spectrum, where the x-axis is the M/Z-values and the y-axis the respective intensities. By means of this graph, the mass of the analyzed speciality and its structure can be determined [10].

As mentioned above, the first step is ionisation, which converts the analyte molecules or atoms into ionic species in the gas phase by the removal or addition of an electron or proton. This is followed by separation and mass analysis of the molecular ions and their charged fragments based on their M/Z ratios. Finally, the ion current due to these mass separated ions is measured, amplified, and displayed in the form of a mass spectrum [10].

The simplest and most common method of ion formation in mass spectrometry is to bombard the gas-phase sample molecules with a beam of electrons. During this process, an electron is removed from the most occupied molecular orbital of the sample molecule to form a positively charged molecular ion. The data output in mass spectral format is usually characterized by a molecular ion region, where the molecular ion signals plus associated heavy isotope satellite ions are located, and a fragment ion region is related to the sample molecule. The general practice is to designate the most abundant ion in the spectrum as the base peak with an arbitrarily assigned relative height of 100. Otherwise, all other ions are presented as percent abundances relative to this base peak [10].

For non-targeted MS-based metabolomics, high-performance or ultra-high performance liquid chromatography (HPLC or UHPLC) is used to separate the compounds from the sample, and then electrospray ionization (ESI) mass spectrometry (MS and MS/MS) is used to collect the mass spectra of each chromatographic peak [11]. The combination of liquid chromatography and mass spectrometry (LC-MS), based metabolic fingerprinting is proven to be a powerful tool to identify significant diferentiating compounds and related metabolic changes [8]. The efficient physical separation of chemical substances dissolved in a mobile phase, performed by liquid chromatography, is combined with the mass spectrometer being able to sort and identify the components (gaseous ions) in electric and magnetic fields according to their mass-to-charge ratios [12]. To identify individual compounds, the resulting MS/MS spectra, together with the chromatographic retention time and masses of the parent ions of the compound of interest, are compared with the MS/MS spectra and retention time of authentic standards to confirm the identity of the compound. Putative identification (MSI level 2) is

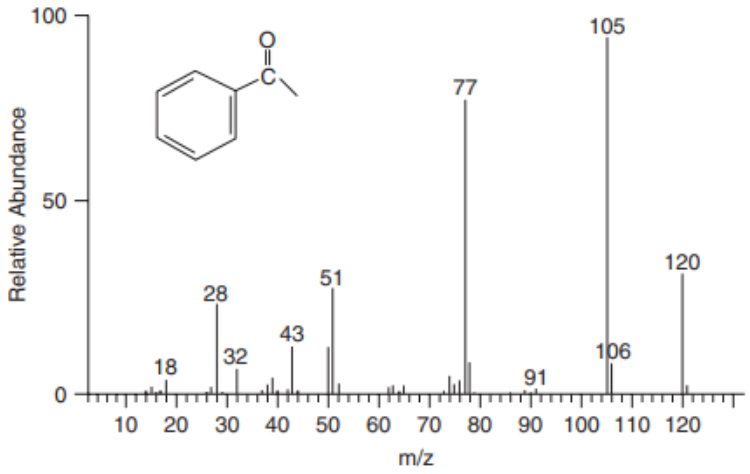

Figure 1. Basic concept of mass spectrometry analysis. Adapted from [10].

achieved by comparing the MS/MS spectra with reference spectra collected experimentally from various MS/MS spectral databases [11].

In contrast to the genome or the proteome, the composition of the human metabolome is not fully defined. Few public databases, such as the Human Metabolome Database – HMDB [13], METLIN [14], or KEGG [15], provide information on the metabolites present in human fluids [3].

As a result, the percentage of MS spectral features that can be confidently assigned to known compounds is usually less than 2%. Thus, the compound identification step remains a central bottleneck in almost all MS-based metabolomics studies [11]. To overcome this difficulty, researchers have developed software tools for the annotation and identification of metabolites, ranging from simple accurate mass searches using library comparisons to in silico tools based on the MS spectrum of the experimentally acquired compound [11], [16], [17].

One way to tackle this problem, therefore, is to use Machine Learning, since the relationship between sequence and fragment abundance can be learned based on a large training dataset without explicit knowledge of the physical mechanisms behind it. However, there may also be predictive models that are not only black boxes but identify the most significant sequence features and properties to make the prediction [18].

Despite the fact that mass spectral prediction has been attempted in the past, its success has been limited. Currently, the best method for predicting EI-MS and ESI-MS/MS spectra of a given compound is the machine learning-based fragmentation modeling method, called CFM-ID. The CFM-ID method first generates all theoretically possible fragments of a molecular structure in a combinatorial way, structuring a fragmentation graph. Each node of the graph represents a theoretically possible ionic fragment, and each edge between nodes encodes the possibility of one ionic fragment directly producing another through a single fragmentation event. CFM-ID then estimates the probabilities of each transition using parameters obtained from training data of known molecules and their respective MS spectra. Finally, CFM-ID constructs the MS spectrum of the introduced molecule from the fragmentation plot and the

associated probability estimates [17].

At present, CFM-ID version 4.0 can predict ESI-MS/MS spectra of a large number of chemical compounds with higher accuracy, performing better than any in silico tool published to date [17].

1.1 Aims of the project

The main objective of this project is to predict the MS/MS spectra of molecules from their molecular formula, using Deep Learning models.

In order to develop a predictive model, we have specified a series of objectives to be met, which are the following:

- Obtain suitable mathematical representations for each molecule, in order to use them as input in our models.
- Implement Artificial Neural Networks (ANN) models that allow us to predict the existence of a peak for each position of the mass spectrum.
- Implement Graph Neural Networks (GNN) models that allow us to predict the existence of a peak for each position in the mass spectrum.
- Compare the predictive performance between the two Deep Learning models (ANN and GNN).
- Compare the results obtained with the results of the best performing in silico tools for the prediction of MS/MS spectra to date.

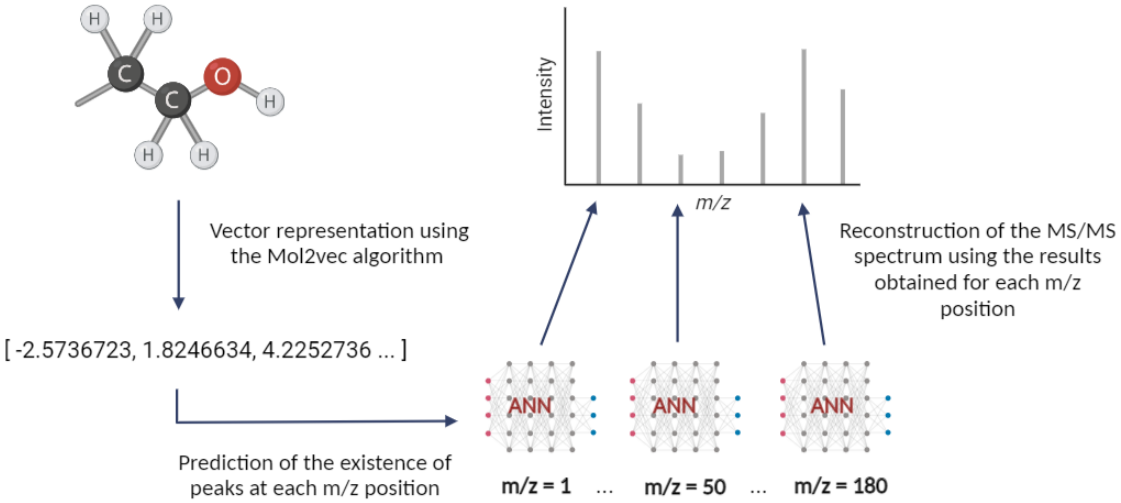

Figure 2. Made-up example of the milestones to be followed to meet the objectives using the ANNs models. Created with BioRender.com.

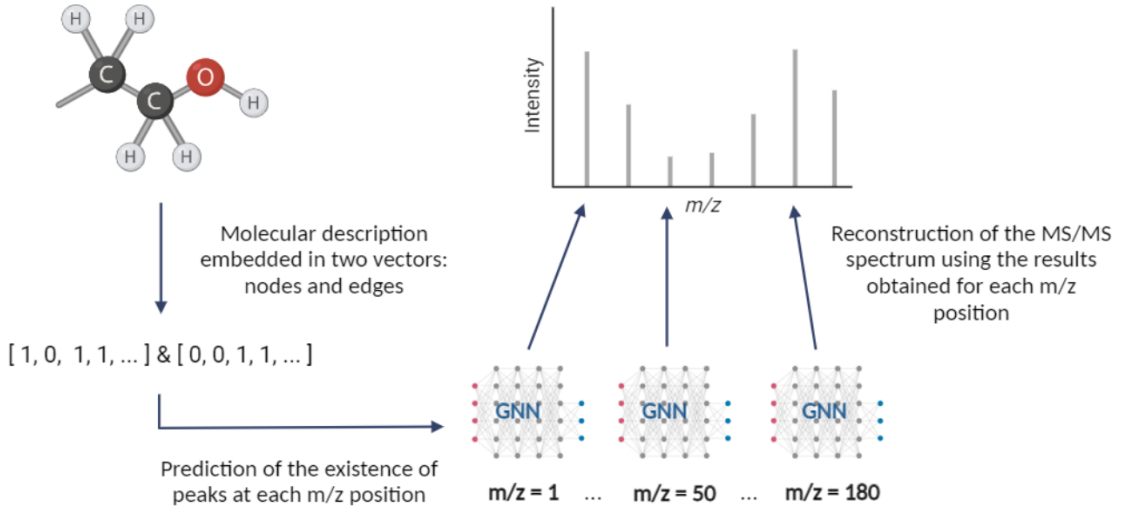

Figure 3. Made-up example of the milestones to be followed to meet the objectives using the GNNs models. Created with BioRender.com.

2 Data

2.1 Data Origin

We used data obtained from MassBank [19], an ecosystem of databases and tools for mass spectrometry reference spectra, provided as an open-source. MassBank is supported by the German Network for Bioinformatics Infrastructure [20], and it is developed in different research groups at the Leibniz Institute of Plant Biochemistry [21], the Helmholtz Centre for Environmental Research [22] and the University of Luxembourg [23]. The data comprised 37500 unique molecules, for which we had their molecular formula in SMILES format, their identifier, and the values of their tandem mass spectra for the [M+H]+ adduct.

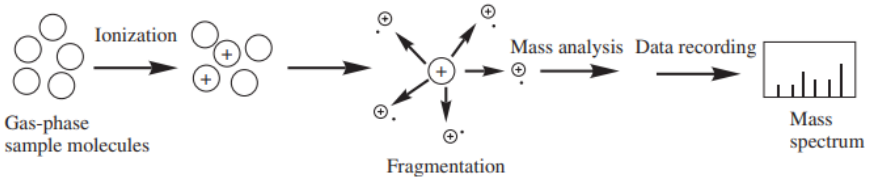

Figure 4. Example of the electron ionization mass spectrum of acetophenone. Adapted from [10].

- **Ionization**: This process is needed to help the separation and detection of ionic species, by applying an electric field and magnetic forces to control their energy and velocity. For the ionization of the samples used, most of them were carried out by Electrospray Ionisation (ESI), a technique applicable to a wide range of liquid phase samples. The ESI process produces highly charged droplets under the influence of an intense electric field. Evaporation of the solvent converts these charged droplets into gas-phase ions. Nevertheless, a smaller number of samples were ionized by Electron Ionization (EI). In this process, the vaporized sample molecules are bombarded with an energetic electron beam at low pressure, where an electron of the target molecule is ejected during this collision process to convert the molecule into a positive ion with an odd number of electrons [10]. It should be noted that more adducts than [M+H]+ can be produced, although this is the most common. On the other hand, the instrument mostly used to detect the mass spectrum was a Quadrupole-Time-Of-Flight (QTOF-MS). This instrument is characterized by its good specificity and high resolution due to the mass Accuracy provided by the Time Of Flight (TOF) detectors, combined with the structural information obtained in the MS/MS mode [24]. A TOF detector measures the time difference from when a particle starts from the initial position until it hits the detector. Such an instrument can provide particle identification by measuring the time-of-flight of particles with known momenta and determining the coincidence of multiple particles detected from a single interaction [25].
- **Spectrum type**: Tandem mass spectrometry (MS/MS) refers to the coupling of two mass analysis steps, either in time or space. Of all ionisation techniques, only electron ionisation (EI) provides a wealth of structural information. To obtain additional structure information by other ionisation techniques, MS/MS is required. Thus, in the first step (MS-1) a desired ion is selected (e.g. [M+H]+) from a stream of ions produced in the

ion source. This mass-selected undergoes a unimolecular fragmentation or chemical intermediate reaction. The second step (MS-2) then performs mass analysis of the product ions formed in the intermediate step [10].

2.2 Data Structure

As mentioned above, obtained data from MassBank consisted of a set of molecules, for which we had their SMILES, their identifier and the peaks of their MS/MS:

	smiles	id	peaks
0	CC(=C)C(=O)/C(=C/C(=O)O)/OC	AC000668	[(97.0658, 111.0441, 112.0519, 125.0597, 146.9...
1	CC1=C[C@@H]2[C@](C[C@@H]1O)([C@]3([C@@H]([C@H]...	AC000631	[(64.7497, 71.0768, 91.0542, 93.0699, 95.0491,...
2	CC12CC3=C(C4=C(C(=O)C=C(C4=O)OC)C(=C3C(O1)C[C@...	AC000164	[(167.0339, 169.0648, 183.0441, 185.0597, 189....
3	CC(C)(C=C)C12CC3C(=O)NC(C(=O)N3C1NC4=CC=CC=C24...	AC000807	[(195.0876, 198.1277, 324.1455, 336.1455, 392....
4	CC1=C[C@@H]2[C@]([C@@H](C1=O)O)([C@]3(C[C@H]([...	AC000384	[(69.0342, 79.0542, 81.0707, 93.0699, 95.0491,...
...	...	...	...
37495	N#CN=C(S2)N(CC2)Cc(c1)cnc(Cl)c1	WA000084	[(63.0, 64.0, 72.0, 90.0, 91.0, 98.0, 100.0, 1...
37496	CCCCCCCCCCCCCN(C1)CC(C)OC(C)1	WA000108	[(281.0, 283.0, 298.0, 300.0), (106.0, 8.0, 99...
37497	COP(=O)(OC)SCN(C(=O)1)c(n2)c(cc(Cl)c2)O1	WA000002	[(139.0, 155.0, 183.0, 184.0, 325.0), (55.0, 1...
37498	COP(=O)(OC)SCN(C(=O)1)c(n2)c(cc(Cl)c2)O1	WA000003	[(111.0, 124.0, 138.0, 139.0, 140.0, 183.0, 18...
37499	Cc(c3)cc(cc(C)3)C(=O)N(NC(=O)c(c1)c(C)c(C2)c(O...	WA000054	[(104.0, 119.0, 133.0, 147.0, 175.0, 176.0), (...

37500 rows × 3 columns

Figure 5. Original dataset example. Each row corresponds to a molecule, and the columns refer to the attributes explained below.

- **SMILES (smiles)**: SMILES (Simplified Input Entry Line Entry System) is a chemical notation system designed for chemical information processing. This system allows the molecular structure to be rigorously specified by means of a linear string of symbols, similar to natural language. As a result, it allows for increased speed and improved computer performance in chemical information processing methods [26].

  SMILES denotes a molecular structure as a graph that is essentially the two-dimensional valence-oriented picture chemists draw to describe a molecule. SMILES notation is a series of characters ending with a space, where atoms are represented by the letters of their atomic symbols. In addition, hydrogen atoms may be omitted or included.

  Each non-hydrogen atom is specified by its atomic symbol in square brackets. The second letter of two-character symbols must be entered in lower case. Organic subset elements (B, C, N, O, P, S, F, Cl, Br or I) may be written without brackets if the number of attached hydrogens conforms to the lowest normal valence consistent with explicit bonds. Finally, aromatic ring atoms are written in lower case and branching is specified by enclosures in parentheses [26].

- **Identifier (id)**: The molecule identifier, given by MassBank.

- **Spectrum values (peaks)**: List containing the result of the mass spectrum for each molecule. This solution is presented in the form of a list, which itself presents two tuples

where the first refers to the x-axis of the spectrum, that is, the M/Z values of the ions arriving at the detector, and the y-axis their respective abundances [10]. This is the column used to compare our predicted results.

2.3 Data Pre-processing

First, we checked that we had all the information (**smiles**, **id** and **peaks**) for all the molecules. We found 237 out of 37500 that did not contain a SMILES chain.

Then, from each SMILES string, we obtained a Mol object belonging to the RDKit module that contains the functionality to work with molecular formats [27]. This allowed us to introduce it into our Neural Network. Each Mol object corresponding to each SMILE was stored in a new **mol** column, which we added to our original dataframe.

We discarded three molecules for which Mol object could not be obtained. Thus, we were left with a final dataset of 37260 molecules (Fig. 6).

	smiles	id	peaks	mol
0	CC(=C)C(=O)/C(=C/C(=O)O)/OC	AC000668	[(97.0658, 111.0441, 112.0519, 125.0597, 146.9...	<img src="...
1	CC1=C[C@@H]2[C@](C[C@@H]1O)([C@]3([C@@H]([C@H]...	AC000631	[(64.7497, 71.0768, 91.0542, 93.0699, 95.0491,...	<img src="...
2	CC12CC3=C(C4=C(C(=O)C=C(C4=O)OC)C(=C3C(O1)C[C@...	AC000164	[(167.0339, 169.0648, 183.0441, 185.0597, 189....	<img src="...
3	CC(C)(C=C)C12CC3C(=O)NC(C(=O)N3C1NC4=CC=CC=C24...	AC000807	[(195.0876, 198.1277, 324.1455, 336.1455, 392....	<img src="...
4	CC1=C[C@@H]2[C@]([C@@H](C1=O)O)([C@]3(C[C@H]([...	AC000384	[(69.0342, 79.0542, 81.0707, 93.0699, 95.0491,...	<img src="...
...	...	...	...	...
37255	N#CN=C(S2)N(CC2)Cc(c1)cnc(Cl)c1	WA000084	[(63.0, 64.0, 72.0, 90.0, 91.0, 98.0, 100.0, 1...	<img src="...
37256	CCCCCCCCCCCCCN(C1)CC(C)OC(C)1	WA000108	[(281.0, 283.0, 298.0, 300.0), (106.0, 8.0, 99...	<img src="...
37257	COP(=O)(OC)SCN(C(=O)1)c(n2)c(cc(Cl)c2)O1	WA000002	[(139.0, 155.0, 183.0, 184.0, 325.0), (55.0, 1...	<img src="...
37258	COP(=O)(OC)SCN(C(=O)1)c(n2)c(cc(Cl)c2)O1	WA000003	[(111.0, 124.0, 138.0, 139.0, 140.0, 183.0, 18...	<img src="...
37259	Cc(c3)cc(cc(C)3)C(=O)N(NC(=O)c(c1)c(C)c(C2)c(O...	WA000054	[(104.0, 119.0, 133.0, 147.0, 175.0, 176.0), (...	<img src="...

37260 rows × 4 columns

Figure 6. Final dataset example. Each row corresponds to a molecule, and the columns refer to the attributes explained.

Lastly, in order to run our models, we must say that we were interested in predicting the location of the peaks that refer to the molecular masses of compounds, so the intensity was secondary. Thus, from the spectrum obtained for each molecule in the database, which indicated the M/Z position where there was a peak and its respective intensity, we used this information to create a binary vector simulating the original spectrum, having ones at the positions where there was a peak.

First, we calculated the highest M/Z position of all molecules. In this way, we created a vector of zeros from zero to the maximum M/Z value. Then, for each molecule, we calculated the M/Z positions where there was intensity and placed a one at that position in the vector created. Note, however, that each vector was discretised so that each position of the vector included bins of size one.

2.4 Data Splitting

To carry out the predictions of the spectra, we divided the data into training, validation and test set. The training dataset was used to fit the model, the validation set was used to provide an unbiased evaluation of the model as the hyperparameters of the model were being tuned, and the test set was finally used to provide an unbiased evaluation of the performance of the final model. In our case, we split the original dataset in a test set of 30.000 molecules, a validation set of 5.000 and a test set of 2.600. Therefore, we trained our models with the same dataset in two different ways: firstly, by inputting the data in the same order as they were in the original dataset, and secondly, in a randomised way but keeping the same order for the two Neural Network cases (ANN and GNN).

3 Machine Learning algorithms to predict Mass Spectra

As we discussed, predicting a mass spectrum of a molecule is a complex task that can be tackled by Machine Learning (ML). Our hypothesis, therefore, was that Machine Learning could learn the relationships between formulas and fragment abundances from large datasets of training examples, without explicit knowledge of the physical and chemical mechanisms behind them.

Thus, in the following section, we theoretically define the Machine Learning methods that we implemented to predict the spectrum of molecules. Likewise, we focus on a specific type called Deep Learning (DL), explaining the algorithms that we finally carried out to achieve the goals of our project.

3.1 General concepts of Machine Learning and Deep Learning

Machine Learning is a branch of computer algorithms designed to simulate human intelligence by learning from the surrounding environment. Based on existing data, ML algorithms can be used to create models for classification, pattern recognition, and predictions [28]. Machine Learning works by solving questions to which it was never explicitly programmed. In this sense, it uses a previously identified mathematical function to calculate the output from a given input. In order to calculate this function, the model needs to be trained with a large example dataset, achieving a mathematical relationship between the input data and the output data.

This branch of artificial intelligence is divided into two areas: supervised and unsupervised learning. In both cases, the input is a data matrix that hopefully represents the relevant part of the data that the model will learn on. Supervised learning algorithms also need a matrix with the desired outcome for each input element. In this case, the algorithm learns the best way to relate the inputs to outputs. One successful example of supervised learning was to use gene expression data to classify patients into different clinical groups and identify new disease clusters. On the other hand, if only the input matrix is available, unsupervised learning takes place. A well-known unsupervised example was the use of genetic coding to predict the secondary structure of proteins. These algorithms mostly work with statistical methods capable of identifying the areas within the input data with the highest density, returning clusters made from these. In other words, unsupervised learning tries to guess the possible hidden structure of the data [29], [30].

Deep learning, or the Deep Neural Network, can learn complex non-linear functions that relate the input x to its prediction y, by adding multiple hidden layers between x and y. In a sense, Deep Learning is conceptually no different from traditional Machine Learning, but the distinction is in the layers [31].

A Neural Network is a network of computational units that is intended to simulate the functioning of the human brain. These computational units or artificial neurons are organised in homogeneous layers, which are connected to each other in such a way that information flows from layer to layer, always in the form of real numbers. This connection is intended to mimic the behaviour of synapses in the human brain [29]. However, not all Neural Networks are equal, so the same results cannot be obtained with all of them. For this reason, in our

project we considered two types of Neural Networks, Artificial Neural Networks and Graph Neural Networks, in order to find out which one worked better for the prediction of mass spectra.

3.2 Artificial Neural Network

3.2.1 Input data

First, before specifying the design of the ANN, we discuss the input data that we introduced in our models. Our input data were vector representations of molecules. In other words, for each molecule, which we had in SMILES format, we obtained a vector that represented it. To do this, we used *Mol2vec*.

Mol2vec produces representations of molecular substructures by using information about their proximity in molecular graphs to compute lower dimensional vectors. It is based on a popular current approach in the field of text mining, called *Word2vec*. This assumes that each word has several meanings depending on its context, so it considers molecular substructures as ”words” in the context of neighbouring fragments [32].

Mol2vec is an unsupervised method inspired by natural language processing, which considers compound substructures derived from *Morgan's algorithm*. By applying this algorithm to a corpus of compounds, high-dimensional embeddings of the substructures are obtained, where the vectors of the chemically related substructures occupy the same part of the vector space [33].

Morgan's algorithm is an iterative process that assigns numerical identifiers to each atom, initially using a rule that encodes the invariant numerical information of the atom into a unicial atom identifier, and subsequently using the identifiers from the previous iteration [34]. In this way, Morgan fingerprints monitor the presence or absence of circular fragments that include all atoms within a given radius, expressed in number of bonds, around a central atom. When the radius is set to one, Morgan's algorithm considers all fragments with atoms that have at most one bond from the central atom.

In general, Morgan's fingerprints are converted into a bit string, in order to facilitate a quick similarity score between bits, for which a vector length is set [32]. The iteration process continues until each identifier for each atom is unique. Ultimately, intermediate results are discarded and the initial identifiers provide a canonical numbering scheme for the atoms [34].

Depending on the intended application, Mol2vec can be trained using one of the two Word2vec approaches: if the task is to predict a word (a molecule in our case) from the words in the context, *Continuous bag of words (CBOW)* is used, or *Skip-gram* if the context is predicted from a word. In CBOW, the order of the words is not important due to the overlapping bag-of-words, whereas in Skip-gram adjacent words are assigned higher weights. In addition, there are two parameters *window size*, number of words that the model considers before and after the target word, and *dimensional embeddings*, length of the output vector, to find the best settings for Mol2vec. Finally, the vector for each molecule is obtained by summing all the vectors of the Morgan substructures of the molecule [33].

3.2.2 Output data

As previously specified in the project objectives, we wanted to predict whether our query molecule had a peak or not at a certain value of M/Z. To that end, we trained 171 models corresponding to the 171 M/Z positions in the spectrum where the molecules in the training set most often presented peaks. In this way, for each molecule in the test set, each model gave as an output a value between 0 and 1, which represented the probability that according to the model the input molecule had a peak in that value of M/Z. We used these values to establish a threshold from which to decide whether or not a molecule had a peak at the position for which the model was trained.

We built the predicted spectrum of a molecule by joining the results obtained with each model (for a specific M/Z value) for that molecule. We used the same output format for all the machine models we used in the study.

3.2.3 Architecture and Functioning

Artificial Neural Networks have been developed as generalizations of mathematical models of biological nervous systems. In a simplified mathematical model of the neuron, the effects of the synapses are represented by connection weights that modulate the effect of the associated input signals and the non-linear characteristic of the neurons by a transfer function. The neuron's impulse is then calculated as the weighted sum of the input signals, transformed by the transfer function. In such a way, the learning capacity of an artificial neuron is achieved by adjusting the weights according to the chosen learning algorithm [35].

The learning rule that artificial Neural Networks use is called the *Perceptron rule*. The perceptron is a single-layer Neural Network that has the goal of producing a target vector from an input vector. Each neuron weights input values differently to produce an element of the output vector (after some nonlinear transformation). The weights are trained to maximise the similarity between target vectors and outputs for each input vector. Perceptrons are particularly suitable for simple pattern classification problems. Suppose we have a set of learning samples consisting of an input vector x and a desired output $d(k)$. For a classification task, $d(k)$ is usually +1 or -1 [35]. The perceptron learning rule can be expressed as follows:

1. Start with random weights for the connections.
2. An input vector x is selected from the samples in the training set.
3. If the perceptron gives an incorrect answer, i.e. the output is $y_k \neq d(k)$, all connection weights w_i are modified by an maount δw_i according to:

$$\delta w_i = \eta(d_k - y_k)x_i \; ; (\eta\text{= learning rate}) \tag{1}$$

4. Return to step 2.

However, the simple perceptron is only capable of handling linear separation or linearly independent problems. If the interest is to learn non-linear functions, the *Multi-Layer Perceptron (MLP)* should be considered, which contains one or more hidden layers apart from

the input and output ones. By taking the partial derivative of the network error with respect to each weight, the direction in which the network error is moving can be learned. This is where the *Backpropagation algorithm* comes into play, where the partial derivatives are taken and then applied to each one of the weights, starting from the output layer to the hidden layer, and from the hidden layer to the input layer weights. This is necessary to have several layers of neurons (as in Deep Learning) since changing this set of weights requires knowledge of the partial derivatives calculated in the back layer [35].

Two different methods can be used for such training: *Online mode* and *Batch mode*. In the first method, the weight updates of the online method are calculated for each input data sample, and the weights are coded after each sample. In the second one, it is possible to calculate the update weights for each input sample but store these values for one pass through the training set, which is called *epoch*. At the end of the epoch, all contributions are suppressed, and only then the weights are updated with the composite value. This method adapts the weights with an update of the weights, so it will follow the gradient more closely. In short, batch training consists of feeding the training samples as input vectors through a Neural Network, calculating the error of the output layer, and then adjusting the weights of the network to minimise the error [35].

Back to the functioning of the perceptron, it accepts inputs, moderates them with certain weight values, and then applies the transformation function to obtain the result. To address this, first and foremost a *Sum function* (2) multiplies all the inputs of x by the weights w and then sums them [36]. The activation function then transforms the summed weighted input of the node into the node's activation or output of that input.

$$f(x) = \sum_{i=1}^{m} w_i x_i; (m = \text{number of inputs to the Perceptron}) \tag{2}$$

One of the activation functions that we considered is the *Rectified Linear Activation Function* or *ReLU* (3), which is a piece-wise linear function that will generate the input directly if it is positive, otherwise it will yield zero [37]. This function allows to eliminate negative units in an ANN [36].

$$f(x) = \max(0, x) \tag{3}$$

Thereafter, in the *Sigmoid Function* or *Logistic Function* (4) the input to the function will be transformed into a value between 0.0 and 1.0 [37]. In other words, it will lead to a probability of the value between 0 and 1 [36]. The form of the function for all possible inputs is an "S" [37].

$$f(x) = \frac{1}{1 + e^{-x}} \tag{4}$$

It should be noted that there are other types of activation functions that can be used, although we chose the ones mentioned above.

3.3 Graph Neural Network

3.3.1 Input data

We also considered a type of Neural Network in which neurons in different layers are not connected all-to-all but instead using the connections of a known graph. A graph represents the relationships (edges) between a collection of entities (nodes). To describe each node, edge, or the entire graph in more detail, we can store information in each of these parts of the graph. Graphs are convenient mathematical objects to represent molecules since their atoms can act as nodes and bonds as edges [38].

A Graph Neural Network (GNN) is an optimisable transformation in all graph attributes that preserves the symmetry of the graph. The simplest GNN architecture allows learning new embeddings for all graph attributes. This GNN uses a separate multilayer perceptron (MLP) on each component of a graph. For each node vector, a MLP can be applied to obtain a learned node vector. The same thing can be done for each edge, learning one embedding per edge, and also for the global context vector, learning a single embedding for the whole graph [38].

In our case, to obtain for each molecule the embedded vector that referred to the edges of the molecule, we created a function where we stored the information of all the bonds of the molecule in a single vector. Likewise, for each edge, we obtained the indices of the atoms found on both sides of the bond and the type of bond: whether it was aromatic, single, double, or triple.

To acquire the node embedding, we also created a function that stored information for each atom of a molecule in a single vector. In this way, for each atom, we got the sequence of its neighbouring atoms, the total number of hydrogens implicit in it, its formal charge, whether the atom was in a ring, and whether it was aromatic, as well as its mass.

3.3.2 Output data

The output obtained in this Neural Network was exactly the same as in the ANN, that is, for each model referring to an M/Z position in the spectrum, we obtained a value between 0 and 1 that later was used to establish whether there was a peak or not, depending on the threshold used.

As in the previous case, we trained 171 models corresponding to 171 peaks, in order to reconstruct the mass spectrum of each molecule.

3.3.3 Architecture and Functioning

Graph Neural Networks (GNNs) are a family of Neural Networks that unlike other models such as ANNs, which consider individual entities in isolation, can extract and utilise features of the underlying graph, making more informed predictions about the entities in these interactions [39].

In GNNs may want to be made predictions at the node level, at the edge level or even predict a global property, having only features at the level of one of those components. The way to do this is to build a GNN model that allows for binary predictions by routing information

between different parts of the graph. This clustering technique serves as a building block for constructing more sophisticated GNN models.

To make the learned embeddings aware of the connectivity of the graphs, this can be done using *Message passing*. In this technique, neighbouring nodes or edges exchange information and influence each other's updated embeddings. Message passing works in three steps:

1. For each node in the graph, all the embeddings (or messages) from neighbouring nodes are gathered.
2. All messages are pooled through an aggregate function. In our case we used three: a summation function, a function that obtains the maximum values and a function that calculates the average. In addition, we combined the results of these three functions to create an embedding with more information.
3. All the pooled messages are passed through an update function, usually a learned Neural Network.

It is worth noting that just as clustering can be applied to nodes or edges, message passing can occur between nodes or edges. In turn, these steps are key to exploiting the connectivity of graphs, so they can be used to create GNN models with increasing expressiveness and power [38]. As can be seen, message passing forms the backbone of many GNN architectures today. In this project, we used the Graph Attention Networks (GAT) architecture introduced by Petar Veličković in 2017 [40], which is one of the most popular types of graph Neural Networks because unlike Graph Convolutional Networks (GCN), all neighbours are not of equal importance as they set a weighting factor to each connection, called *Attention scores*.

To calculate attention scores (a_{i_j}), a three-step process is used:

1. First, to calculate the importance of each connection, pairs of hidden vectors created by concatenating vectors of two nodes are needed. Then, a linear transformation is carried out using a weight matrix,
$$a_{ij} = W_{att}[Wx_i \parallel Wx_j] \ . \tag{5}$$
Where W is a weight matrix applied to the nodes embedding, and W_{att} is a weight matrix applied to the concatenation of nodes i and j.
2. Next, an activation function needs to be included. In our case, we used the function *LeakyReLU Function* (6), which is based on a ReLU, but it has a small slope for negative values instead of a value of zero. In consequence, the gradient does not disappear for negative values, but the affected weights can still be adjusted and used for the backwards propagation.
$$a_{ij}^{'} = \text{max}(\alpha\, a_{ij}, a_{ij}); (\alpha\text{= slope}). \tag{6}$$
3. To compare scores, the same scale can be applied by using the *Softmax Function* (7). So, in the following formula the term on the bottom is the normalization term which

ensures that all the output values of the function will sum to 1, thus constituting a valid probability distribution [41],

$$a_{ij}^{''} = \frac{e^{a_{ij}^{'}}}{\sum_{k=1}^{m} e^{a_{ik}^{'}}}; (m = \text{number of neighbour nodes of node i}). \quad (7)$$

However, to improve the performance of self-attention, multiheaded attention can be used. This consists of replicating the same steps above several times to average or concatenate the results. In this way, we will obtain a hidden vector for each attention head [42].

3.4 Metric

Once the models were available, we could predict for each molecule whether or not it had a peak at the M/Z position for which the model was constructed, and its MS/MS spectrum by reconstructing the values obtained by each model. These predictions, both with the models independently and for the entire spectrum reconstruction, underwent a validation process to check the performance of each network used. There were different metrics to do so, but we considered *Precision*, *Recall*, *Accuracy*, *F1*, and *Cosine similarity* [43].

To define some of them, we use the following nomenclature: *True Negative (TN)*, *True Positive (TP)*, *False Positive (FP)* and *False Negative (FN)*. Where the model is *Positive* if its prediction is 1, and *Negative* if it is 0. In addition, it is *True* if the prediction is correct, or *False* if it is wrong.

- **Precision**: measures the quality of the model in classification tasks.

$$Precision = \frac{TP}{TP + FP} \quad (8)$$

- **Recall**: reports the amount that the model is able to identify.

$$Recall = \frac{TP}{TP + FN} \quad (9)$$

- **F1**: combines the Precision and Recall measures into a single value.

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \quad (10)$$

- **Accuracy**: measures the percentage of cases that the model has been correct.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (11)$$

- **Cosine Similarity**: measures the similarity between two vectors of an inner product space, determining by the cosine of the angle between them if they point in approximately the same direction [44].

4 Implementation of the algorithms and results

In this part of the project, we combined the implementation of the algorithms mentioned in the previous section with the results collected during the process. As explained above, we first focused on the methods based on Artificial Neural Networks models (Fig. 7), and then on those based on Graph Neural Networks models (Fig. 8). As a result, we obtained 8 different spectrum prediction methods, depending on the type of models used, the input data in the training set and the different parameters established for the prediction.

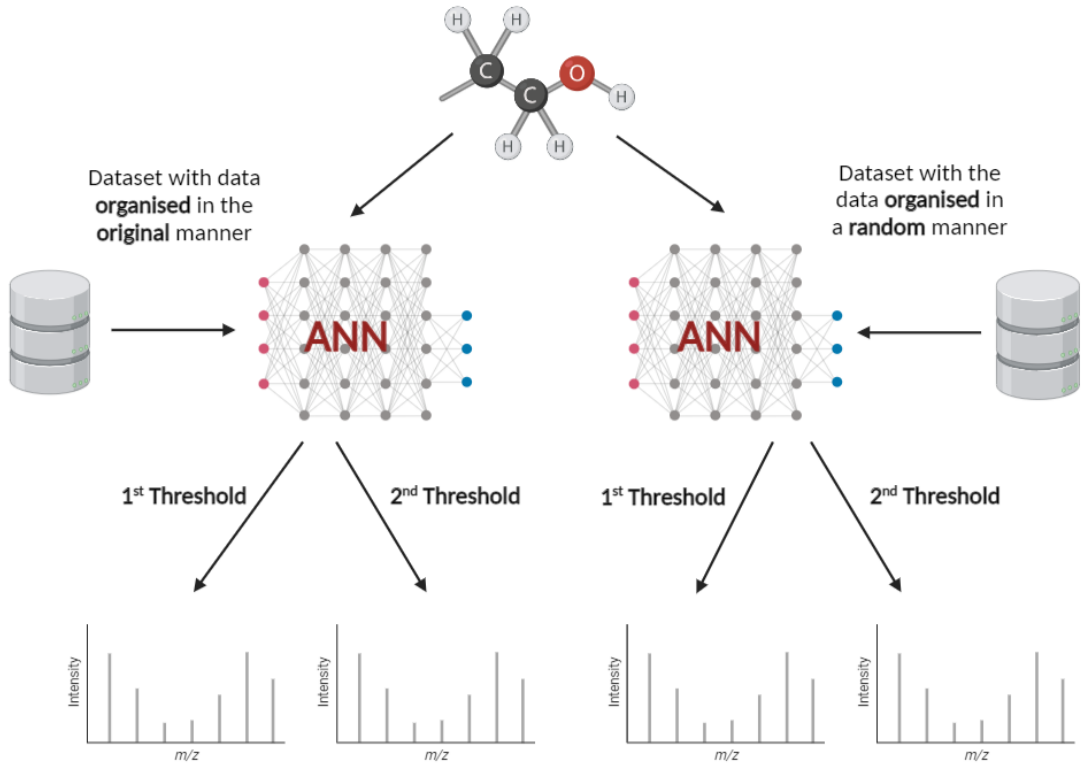

Figure 7. Made-up example of the 4 different spectrum prediction methods obtained with the ANN models. Created with BioRender.com.

It should be noted that we worked with *Python* to programme all these algorithms (see Appendix 1).

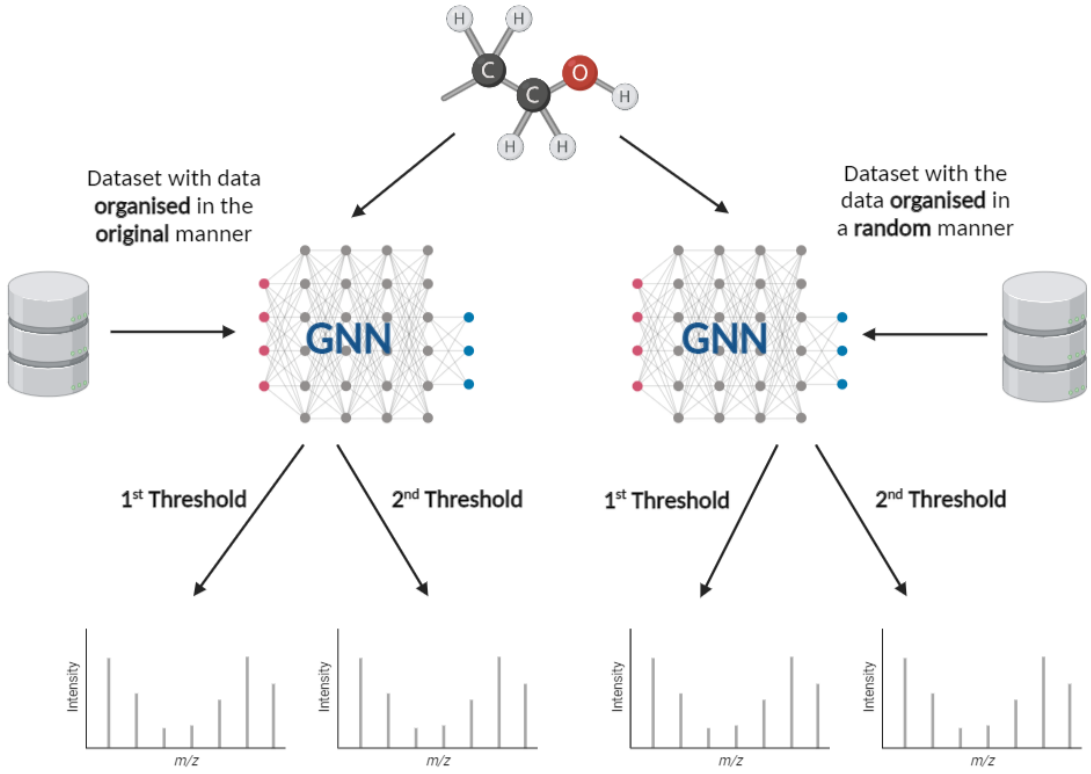

Figure 8. Made-up example of the 4 different spectrum prediction methods obtained with the GNN models. Created with BioRender.com.

4.1 Artificial Neural Network

Before we begin, we should outline the steps we followed using ANN models to obtain predictions of the MS/MS spectra of the molecules:

1. Obtain the mathematical representations of the molecules using the Mol2vec method.
2. Train a model for each of the 171 most common positions, i.e. where the molecules most frequently present a peak, of the MS/MS spectra.
3. To merge the independent performance of the models in order to reconstruct and predict 171 positions of the MS/MS spectra of molecules. Furthermore, evaluate these results with the results that would be expected from comparing our predictions at random.
4. Evaluate the results of the predictions with those of the CFM-ID.

In the following, we explain in more detail how we carried out the implementation of the above steps.

4.1.1 Mol2vec vectors

First, we obtained the vectors that refer to the molecules in the dataset, using a Mol2vec model. To do this, we trained two types of models: the Skip-Gram and the CBOW, using all the molecules in the dataset. For both of them, we declared the output vector of measurement 300 and a window of 15, i.e. the model considers 15 "words" before and after the target word. Therefore, we got molecules represented in a vector of dimension 300.

To ensure that these vectors obtained from the molecules made sense and could be sufficiently representative, we checked the Cosine similarity between the predictions carried out by the Mol2vec model with the Cosine similarity of the same molecules from the given input spectra. As a result, we obtained the following graphs (Fig. 9 and Fig. 10), which are quite similar for both the Skip-Gram and CBOW models. Observing the charts, there was a relationship between the similarity of the molecules compared by the real values of their spectra and by their vectorial representation obtained by Mol2vec. Furthermore, we could confirm this relationship as both pvalue values were highly significant, they were less than 0.01.

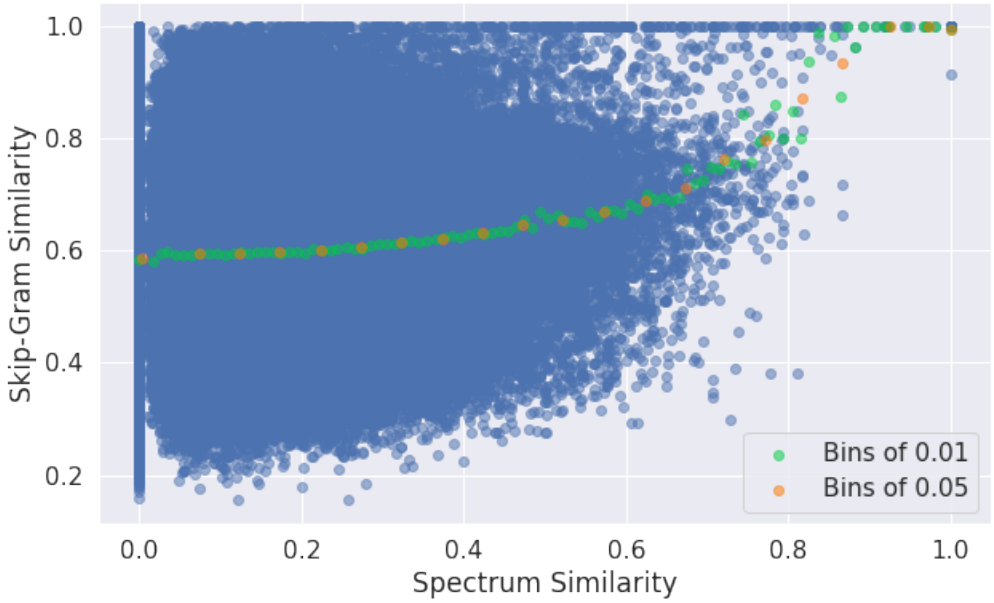

Figure 9. Comparison of the similarity between the representations of the molecules obtained with the Mol2vec Skip-Gram model and their original spectra. In blue are the points for each similarity comparison. In orange, we show the mean value of the similarities that were found by making bins of 0.05 on the spectrum similarity axis. In green the same occurs, but with bins of 0.01. In order to be able to compare the level of similarity, we obtained a rho value of **0.070**, and a pvalue of **0.0**, using *Spearman's correlation* for the unbinarised values. For the similarities using a bin of 0.05 we obtained a rho value of **0.995**, and a pvalue of **3.565e-21**, and for a bin of 0.01 we obtained a rho value of **0.993**, and a pvalue of **6.005e-90**.

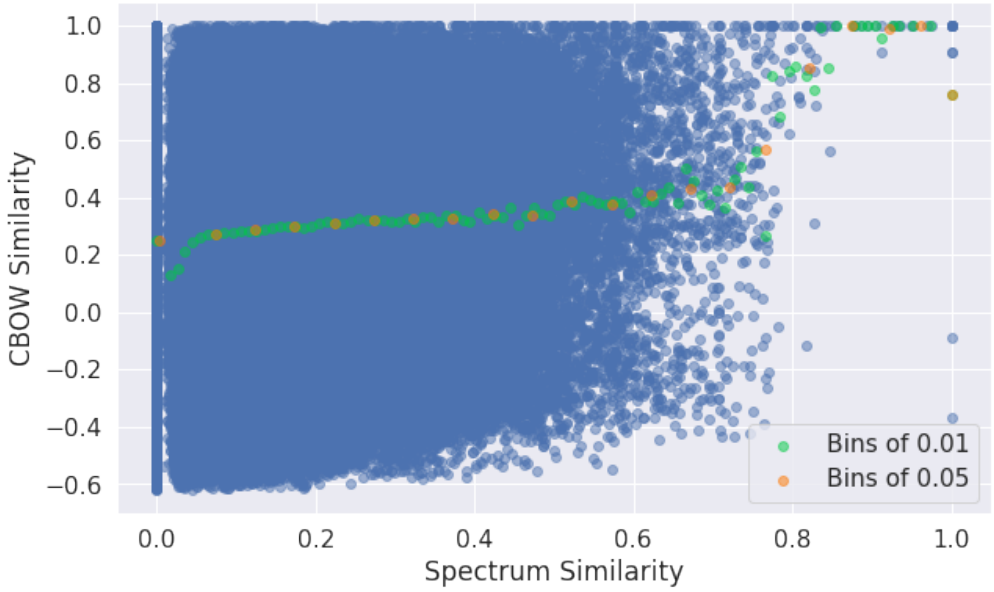

Figure 10. Comparison of the similarity between the representations of the molecules obtained with the Mol2vec CBOW model and their original spectra. In blue are the points for each similarity comparison. In orange, we show the mean value of the similarities that were found by making bins of 0.05 on the spectrum similarity axis. In green the same occurs, but with bins of 0.01. In order to be able to compare the level of similarity, we obtained a rho value of **0.071**, and a pvalue of **0.0**, using *Spearman's correlation* for the unbinarised values. For the similarities using a bin of 0.05 we obtained a rho value of **0.983**, and a pvalue of **1.774e-15**, and for a bin of 0.01 we obtained a rho value of **0.943**, and a pvalue of **8.192e-47**.

As the following steps involved large computational calculations, we decided to use from this point the representations of the molecules obtained using the Skip-Gram model as it scored lower pvalues when comparing its representations with the original spectra.

4.1.2 Peak Models

As we already specified in previous sections, we trained 171 models corresponding to the positions of the spectrum most frequented by the fragments of the molecules in the training data set. However, as in one case we used the dataset with the order of the molecules as it was originally, and in another, we randomised it, the molecules in the training, validation and test set changed in each case. Nevertheless, the most frequented positions of the spectrum and, consequently, the trained models practically coincided for both cases.

Thus, to train the ANN models we used the aforementioned Batch mode method with a batch size of 16 and a learning rate of 0.1. Furthermore, in the training, we defined a minimum of 400 epochs and a maximum of 10.000. For each case, we gave the models as trained when there started to be overfitting, that was when the model started to generalise the knowledge we wanted it to acquire, and we kept the trained model that obtained a smaller loss in the validation dataset.

Once we had the models trained, we obtained a score for each molecule in the test set and model (peak). The score we got was a value between 0 and 1 for each position. For example, for the ANN model 130 (M/Z=130) with the randomised data, we obtained for molecule ID #4 a score of **0.699478**. We then had to decide whether this score corresponded to a 1 or a 0 in our predictions. We used two types of thresholds to transform scores into binary variables:

- **Threshold 1 - TH1:** For the first type, we defined a threshold that corresponded to the fraction between the number of times a peak appeared at the given position for the molecules in the training set, and the total number of molecules in the training set (30.000). Going back to our example, for position 130 we obtained a fraction of molecules in the training set that had a peak in this position equal to **0.174**, so that TH1 = 0.174. As the score we got for molecule ID #4 was higher than the TH1 (0.174 < 0.699478), our prediction was that molecule ID #4 had peak at position 130.
- **Threshold 2 - TH2:** In this case, we considered a threshold such that test and training sets were 'calibrated' and we predicted a fraction of peaks in the test that was equal to the fraction of peaks we observed in the training for a specific position. For instance, on our example, for position 130, we predicted a number of peaks that was equal to the fraction of molecules in the training set with a peak in that position 0.174 times the number of molecules in the test set (2.260), so that we predicted 393,239, which we rounded up to **394**. Thus, for the 394 molecules with the highest score obtained by the ANN random model at position 130, we predicted a peak in position 130.

In order to first evaluate the results in this first part, we compared, using the metrics described previously (Section 3.4), the predictions obtained by our models considering the two types of thresholds, with the value of the spectrum solutions obtained in the original data.

Looking at the graphs (Fig. 11, 12, 13 and 14), we could see that there was a certain tendency towards learning as more models were trained, since the Accuracy value increased. On the other hand, the values of Recall, Precision and F1 were higher in the models with randomised input data. It should be added that in the predictions in which the second threshold

was used, these three metrics were also very similar due to the calibration of the results with respect to the test set.

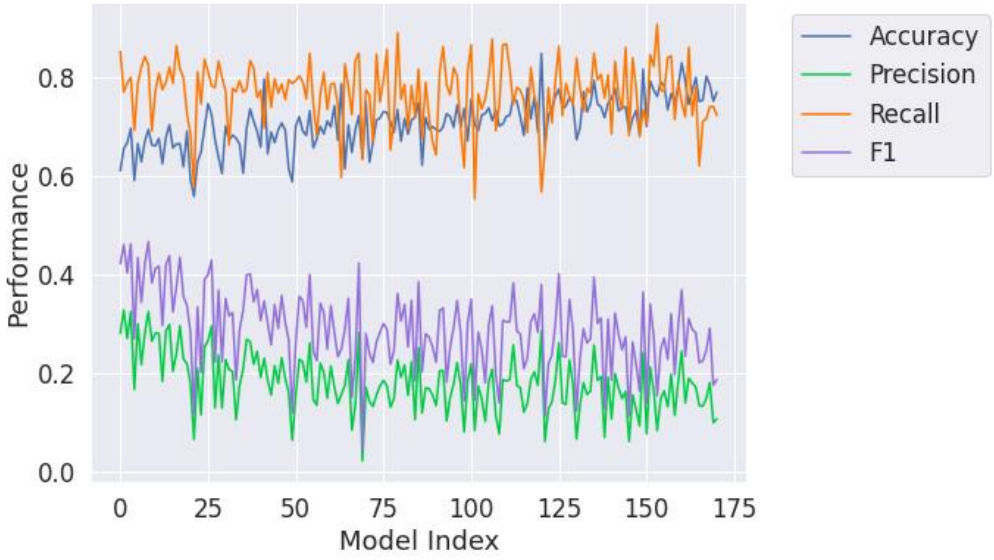

Figure 11. Prediction performance metrics for each ANN type model, with non-randomised input data and the first threshold (TH1). Each point represents the result of the metrics obtained by each model, trained with a test set of 2.600 molecules. On the x-axis, the 171 trained models were arranged from highest to lowest peak frequency, so that the model with Model Index 0 corresponded to the model that considered the position in the spectrum where the molecules in the training set most commonly presented a peak.

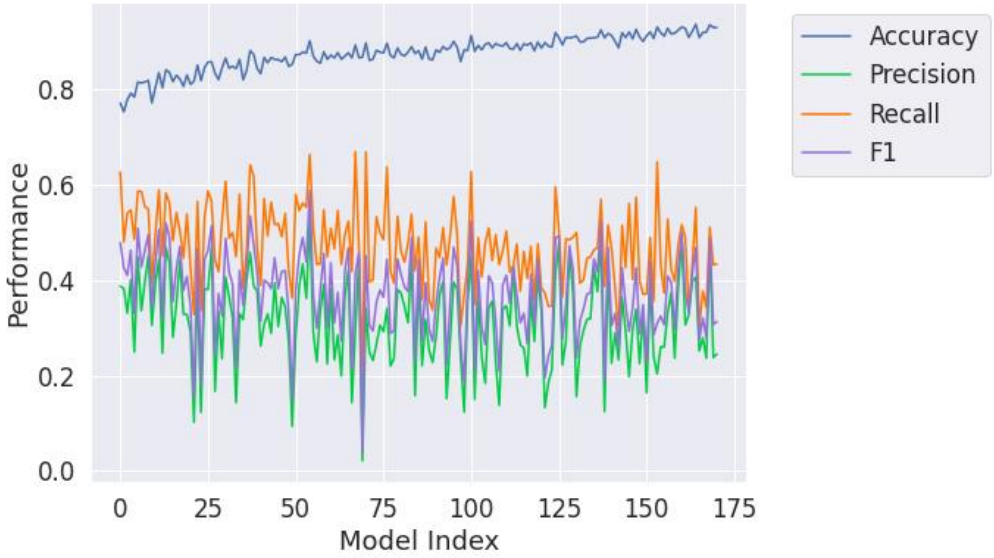

Figure 12. Results of the metrics for each ANN type model, with non-randomised input data and the second threshold (TH2). Each point represents the result of the metrics obtained by each model, trained with a test set of 2.600 molecules. On the x-axis, the 171 trained models were arranged from highest to lowest peak frequency, so that the model with Model Index 0 corresponded to the model that considered the position in the spectrum where the molecules in the training set most commonly presented a peak.

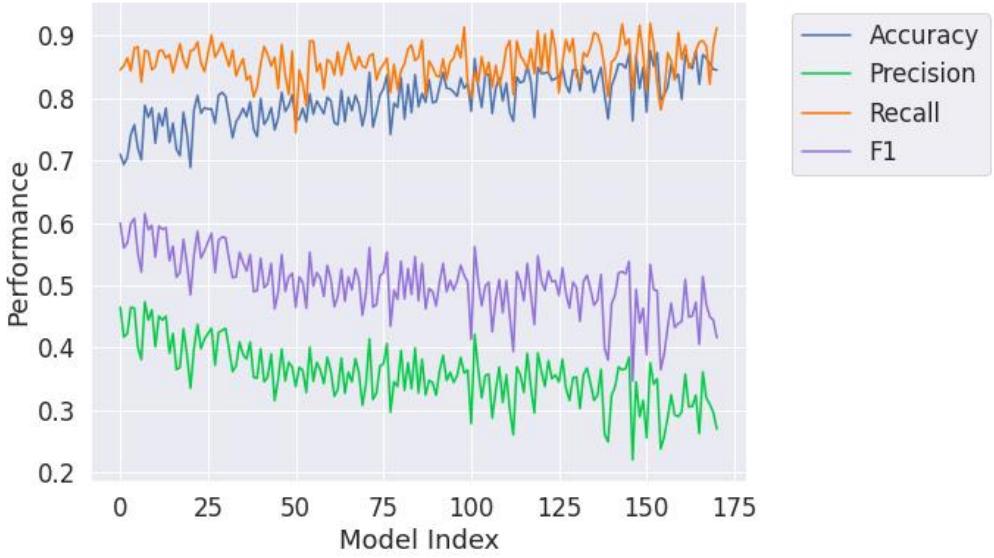

Figure 13. Results of the metrics for each ANN type model, with randomised input data and the first threshold (TH1 Random). Each point represents the result of the metrics obtained by each model, trained with a test set of 2.600 molecules. On the x-axis, the 171 trained models were arranged from highest to lowest peak frequency, so that the model with Model Index 0 corresponded to the model that considered the position in the spectrum where the molecules in the training set most commonly presented a peak.

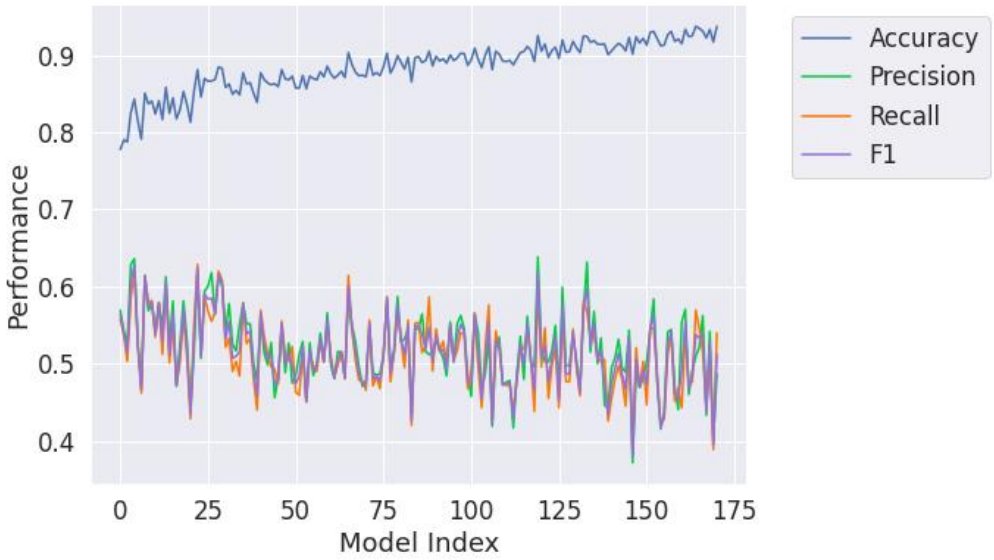

Figure 14. Results of the metrics for each ANN type model, with randomised input data and the second threshold (TH2 Random). Each point represents the result of the metrics obtained by each model, trained with a test set of 2.600 molecules. On the x-axis, the 171 trained models were arranged from highest to lowest peak frequency, so that the model with Model Index 0 corresponded to the model that considered the position in the spectrum where the molecules in the training set most commonly presented a peak.

4.1.3 Reconstruction of Spectra

To predict the spectra of the molecules, we started from the predictions calculated above, where for each molecule we got the possibility of having or not having a peak at the M/Z positions for which we trained models.

In this way, molecule by molecule we reconstructed their spectra by joining the peak predictions obtained at their respective positions. This process was carried out considering the two types of thresholds, for each ANN model exemplary, one with randomised data entry and the other without.

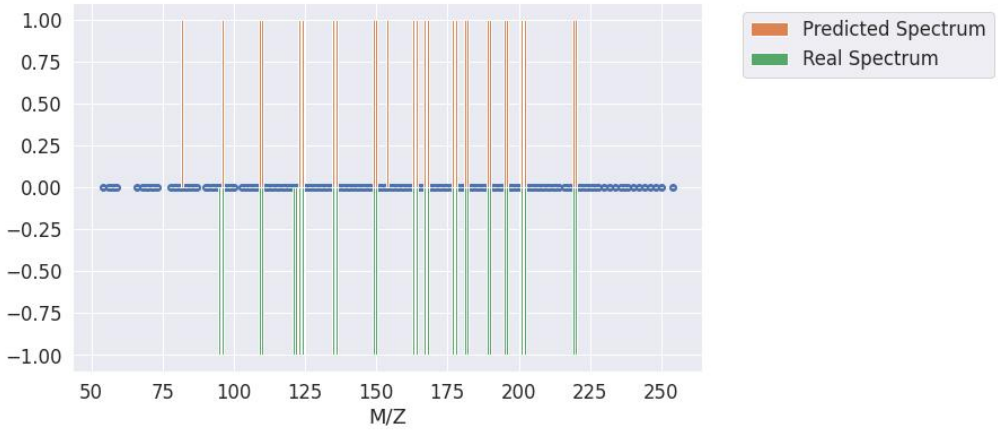

Figure 15. Reconstruction of the spectrum of molecule ID #786, using the ANN method with the second threshold and randomised input data (TH2 Random). Orange shows the peaks that we predicted, green shows the real peaks and blue shows the positions for which we had a model.

To evaluate the results, we first compared the spectra obtained by summing our models with the original spectrum of each molecule (but only considering the positions for which we had a model (Fig. 15)). Then, we calculated the model metrics for these two spectra. However, to see if our reconstruction really made sense, we decided to also compare reconstructed spectra of molecules with the true spectral value of other molecules. The purpose of this random comparison was to find out if our model really had knowledge or if its performance was similar to randomly comparing two different molecules (Fig. 16, 17, 18 and 19).

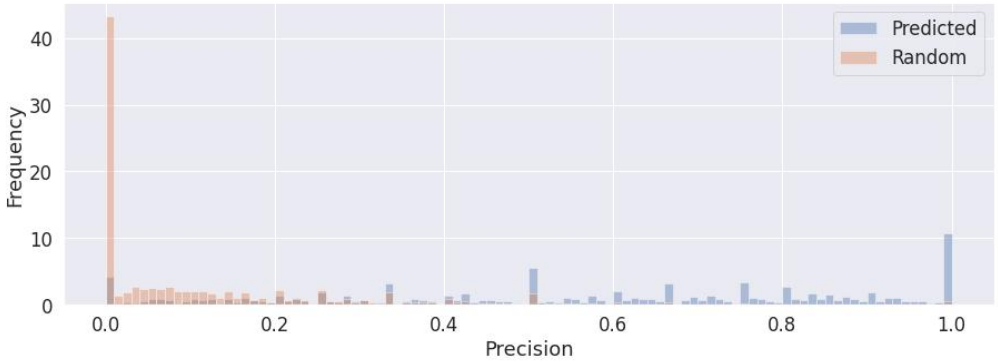

Figure 16. Distribution of the Precision obtained by the ANN models with the second threshold type and randomised input data (TH2 Random), versus the Precision of comparing our predictions with real random spectra.

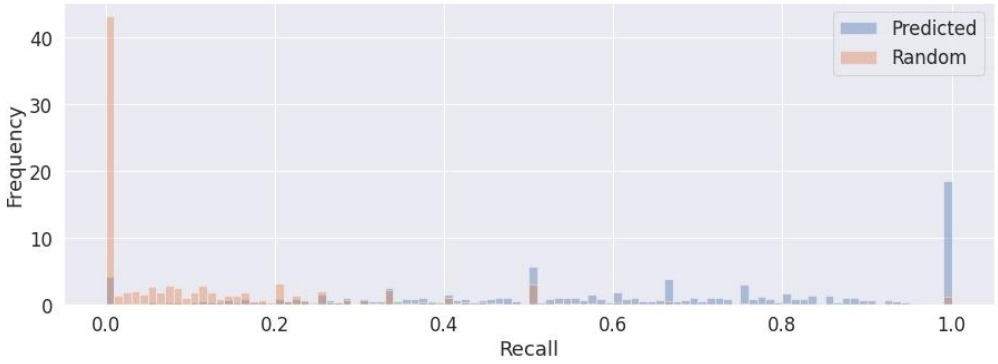

Figure 17. Distribution of the Recall obtained by the ANN models with the second threshold type and randomised input data (TH2 Random), versus the Recall of comparing our predictions with real random spectra.

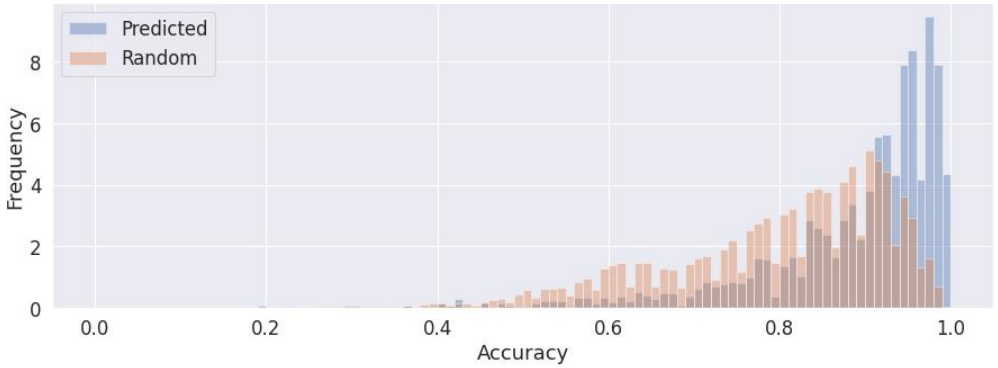

Figure 18. Distribution of the Accuracy obtained by the ANN models with the second threshold type and randomised input data (TH2 Random), versus the Accuracy of comparing our predictions with real random spectra.

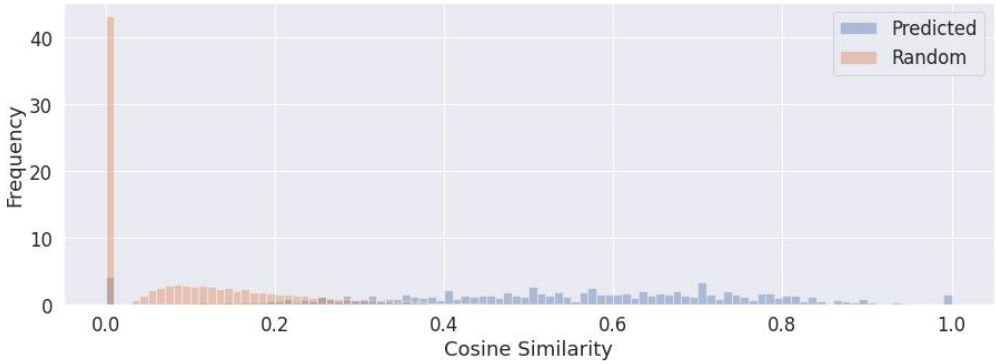

Figure 19. Distribution of the Cosine similarity obtained by the ANN models with the second threshold type and randomised input data (TH2 Random), versus the Cosine similarity of comparing our predictions with real random spectra.

4.1.4 CFM-ID Spectra Reconstruction

On the other hand, we wanted to see how our predictions performed in comparison with the predictions carried out by the *CFM-ID Spectra Predicition*, a website that predicts QToF MS/MS spectra for multiple collision energies for an input small molecule [45]. The spectra are calculated for low (10 eV), medium (20 eV) and high (40 eV) collision energy levels and are represented by a list of ”mass intensity” pairs, each corresponding to a peak in the spectrum.

In this way, we first downloaded the CFM-ID predictions of the molecules that we had in each of our test sets (one with the molecules as in the same order as they were in the original database, and another with the ones that resulted in the test set after randomisation). Then, as we did not introduce the type of collision energies as a parameter in our model, we decided that in order to generalise the predictions of the platform as well, we considered for each molecule all the peaks in its spectrum obtained as a result of all the collision energies. Once we collected all the peaks of the molecules, obtained using all available collision energies, we discretised and binarised them in the same way as we did for the solution of the original spectra of the molecules. Then, to compare the CFM-ID prediction with ours, we only considered peaks that were at the M/Z positions for which we had a trained model. Afterwards, we carried out a comparison between the CFM-ID spectra and the true spectra of the molecules (Fig. 20), both binarised and discretised.

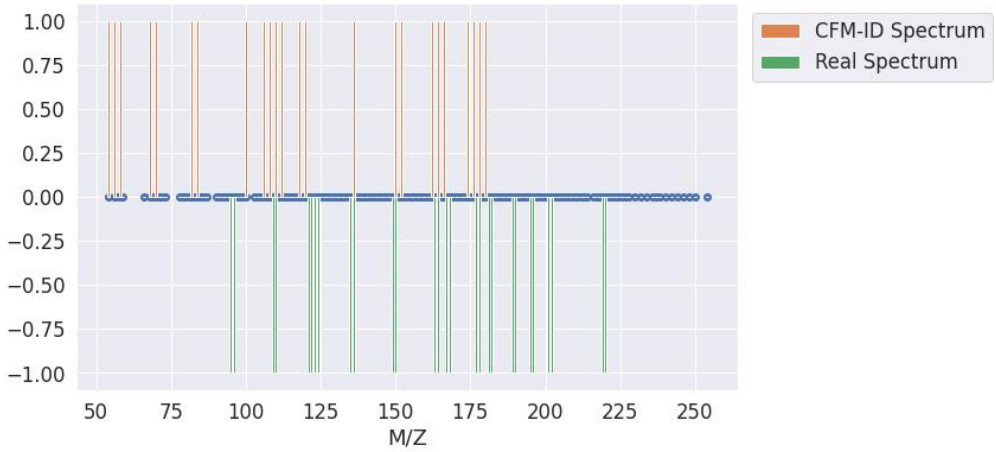

Figure 20. Reconstruction of the spectrum of the molecule ID #786 using the CFM-ID. Orange shows the peaks that CFM-ID predicted, green shows the real peaks and blue shows the positions for which we had a model.

Finally, we compared the model metrics obtained for each molecule spectrum between our predictions and those of the CFM-ID (Fig. 21, 22, 23 and 24). In doing so, we aimed to observe quantitatively which prediction algorithm performed better.

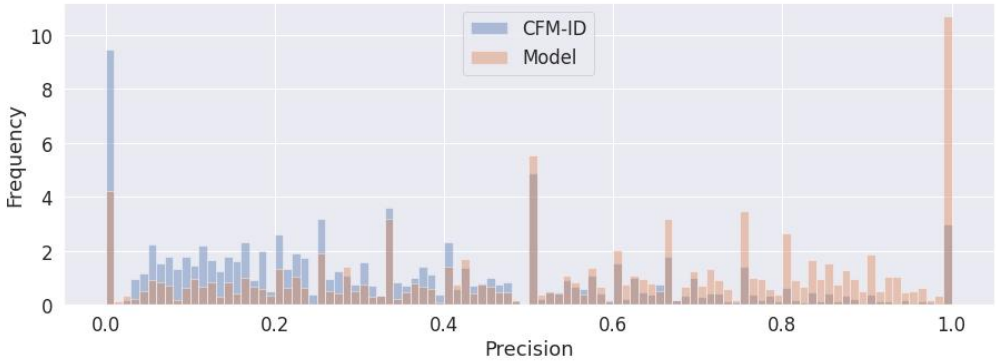

Figure 21. Distribution of the Precision obtained by the ANN models with the second threshold type and randomised input data (TH2 Random), versus the Precision obtained by the CFM-ID.

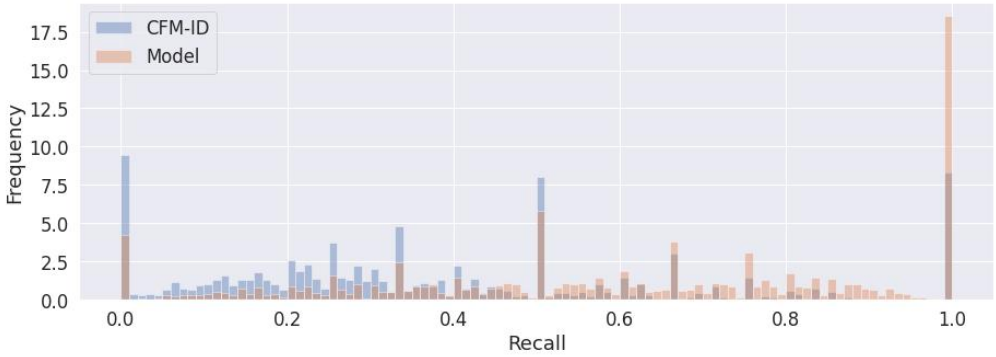

Figure 22. Distribution of the Recall obtained by the ANN models with the second threshold type and randomised input data (TH2 Random), versus the Recall obtained by the CFM-ID.

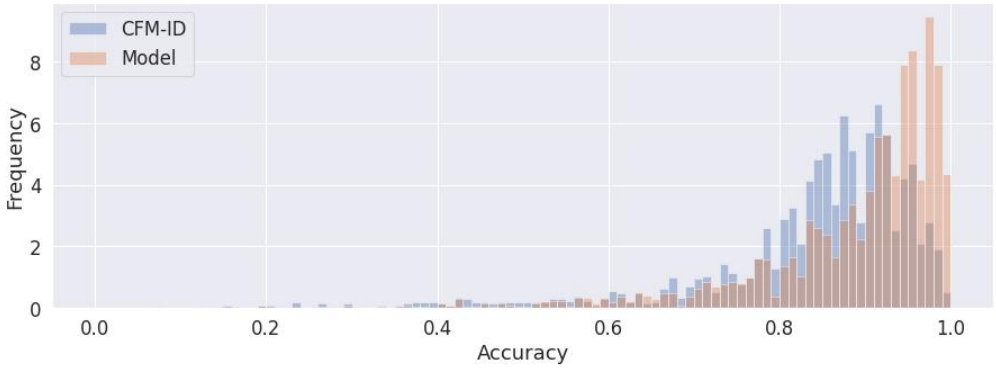

Figure 23. Distribution of the Accuracy obtained by the ANN models with the second threshold type and randomised input data (TH2 Random), versus the Accuracy obtained by the CFM-ID.

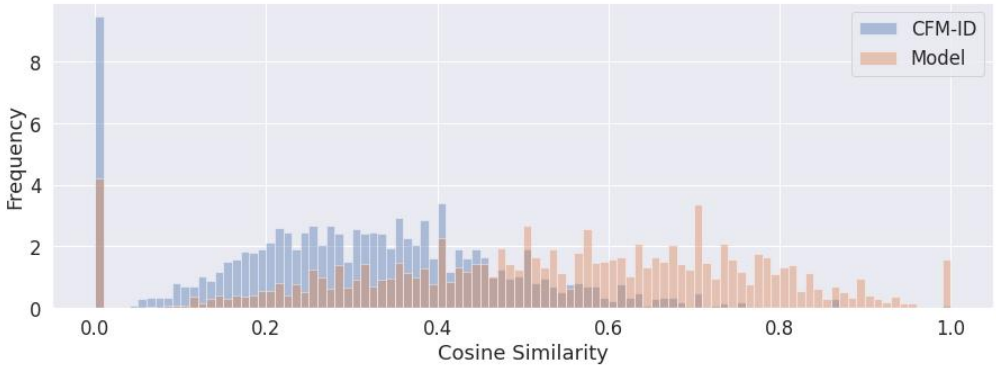

Figure 24. Distribution of the Cosine similarity obtained by the ANN models with the second threshold type and randomised input data (TH2 Random), versus the Cosine similarity obtained by the CFM-ID.

4.1.5 Summary and interpretation

To finish with the Artificial Neural Network and to be able to quantitatively compare the results obtained, we carried out dispersion plots of the Precision, Recall, Accuracy and Cosine similarity obtained for the predictions of each molecule of the CFM-ID, compared to those obtained from the prediction of our model (figures in Appendix 2). We then noted the number of molecules that obtained the best results for each type of metric for each algorithm, and the number of molecules that obtained the same results for both algorithms. In this way, it was much easier to know quantitatively who performed better. These numerical values are shown in the bar graph and noted in the table below.

In view of the results, the ANN model with the data without randomisation and considering the first type of threshold (TH1), stood out for obtaining a higher Recall and Cosine similarity compared to the CFM-ID (Fig. 25). This implies that our model predicted a higher number of peaks per molecule, and that the similarity between the real spectrum and our predicted spectrum was better than that of the CFM-ID. However, the CFM-ID's higher number of molecules in the Precision implied that the quality of its model was better with respect to the classification task.

For the model with the data without randomisation but with the second type of threshold (TH2), it can be seen in the results that our model obtained better scores for all types of metrics compared to those of the CFM-ID (Fig. 26). However, the difference between the number of molecules that obtained a better Recall with our model and with the CFM-ID was lower than that obtained in the previous case (TH1).

Regarding the case with randomised data with the first type of threshold (TH1 Random), our model obtained better scores for Precision, Recall and Cosine similarity, of which the high score and difference with respect to the CFM-ID for Recall and Cosine similarity stood out (Fig. 27).

For the last case, with randomised data and the second type of threshold (TH2 Random), our results were better for all four metrics and although the values were also very high (Fig. 28), the difference between the Recall of our model and the CFM-ID was higher in the previous case (TH1 Random) than in this one (TH2 Random).

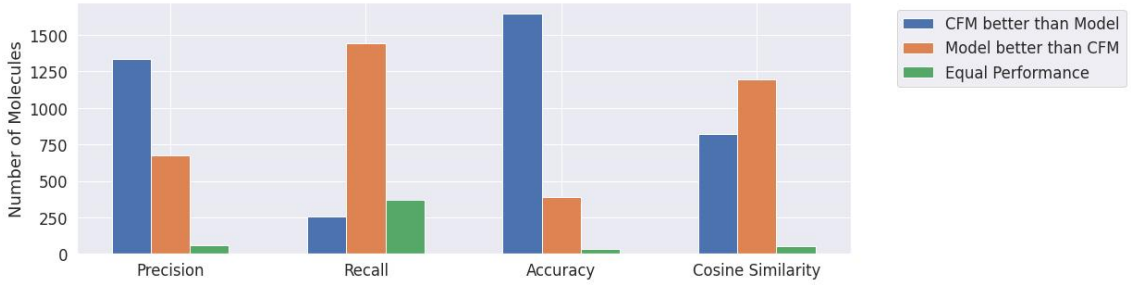

Figure 25. Count of the number of molecules for which a better metric was obtained between the ANN method with non-random data input and the first threshold (TH1), versus the CFM-ID.

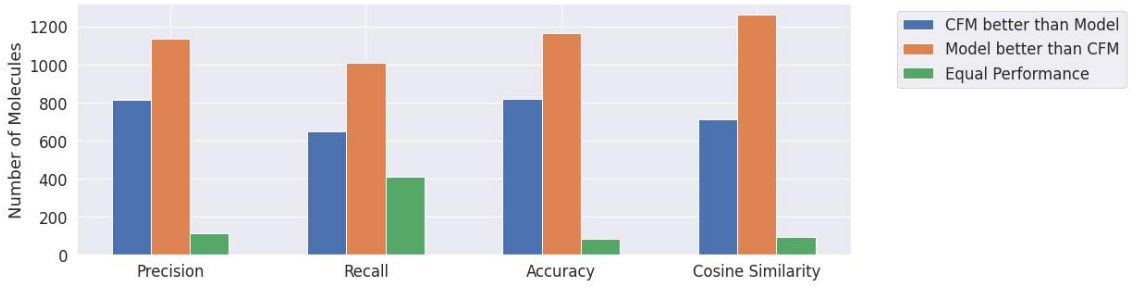

Figure 26. Count of the number of molecules for which a better metric was obtained between the ANN method with non-random data input and the second threshold (TH2), versus the CFM-ID.

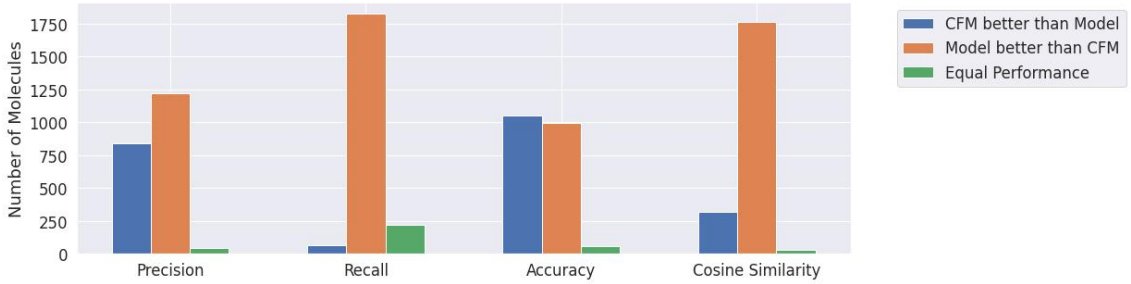

Figure 27. Count of the number of molecules for which a better metric was obtained between the ANN method with random data input and the first threshold (TH1 Random), versus the CFM-ID.

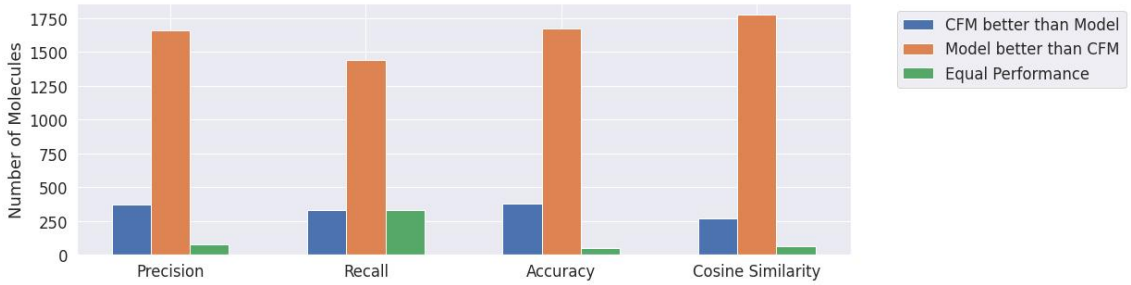

Figure 28. Count of the number of molecules for which a better metric was obtained between the ANN method with random data input and the second threshold (TH2 Random), versus the CFM-ID.

To sum up the first part, we can say that overall our prediction algorithms could reconstruct spectra better than CFM-ID, since for all the models we trained the Recall and Cosine similarity were higher. This means that our algorithms identify a larger number of peaks for each tandem mass spectra of the molecules and that the spectra they predict are more similar to the original ones than those of the CFM-ID. Add that we do not give as much importance to the metric Accuracy since it measures the percentage of cases where the model has been correct, whether or not to predict a peak, so it includes the number of predicted

Table 1. Summary of the number of molecules predicted better by ANN.

Model	Metric	CFM	Model	Equals
TH1	Precision	1336	674	56
	Recall	256	1441	369
	Accuracy	1644	387	35
	Cosine Similarity	272	1774	61
TH2	Precision	816	1137	113
	Recall	648	1007	411
	Accuracy	818	116	82
	Cosine Similarity	711	1260	95
TH1 Random	Precision	844	1222	41
	Recall	65	1824	218
	Accuracy	1051	998	58
	Cosine Similarity	316	1763	28
TH2 Random	Precision	372	1659	76
	Recall	334	1440	333
	Accuracy	382	1672	53
	Cosine Similarity	272	1774	61

zeros. Thus, a high value of this metric will not be as significant, since it will be more general and not as discriminative as one that considers only the prediction of peaks. On the other hand, better results were obtained by algorithms that have randomised data as input. This means that the data in the database from which we obtained them were ordered by some factor, so that if we divide the data set without mixing the rows into three values, in each data set the characteristics of the data are more homogeneous. For this reason, by randomising the data, they and their features are interleaved, allowing the Neural Networks to learn from a more heterogeneous data set with more information.

Finally, from this first part we highlight the performance of the algorithm with the randomised data but with the second type of threshold (TH2 Random), as it obtained very good scores for all metrics, including a good Precision value, which implies a high quality in the classification task of the models.

4.2 Graph Neural Network

In the same way as with ANN-based methods, we outline the steps we followed using GNN models to obtain predictions of the MS/MS spectra of the molecules:

1. Obtain the mathematical representations of the molecules using functions based on graph functions, where the nodes are the atoms of the molecules, and the edges are the bonds.
2. Train a model for each of the 171 most common positions.
3. To merge the independent performance of the models in order to reconstruct and predict 171 positions of the MS/MS spectra of molecules, and evaluate these results with the ones expected at random.
4. Evaluate the results of the predictions with those of the CFM-ID.

Subsequently, we explain in more detail how we carried out the implementation of the above steps.

4.2.1 Edge and nodes embeddings

In this case, we directly introduced the molecules in our Neural Network, and within it, we called the functions described in Section 3.3.1, which were responsible of obtaining vectors for each molecule, from its edges and nodes.

4.2.2 Peak Models

Afterwards, to obtain the predictive models for the 171 most frequent positions in the spectrum, we followed the same methodology as for the ANN. This means that we calculated two types of models, depending on the molecules of the input dataset, i.e. considering the randomised dataset and the non-randomised one. In addition, to train the models we used the Batch mode method, with the same batch size as before, but with a learning rate of 0.01. Regarding the criteria for training our models, it was exactly the same, defining a range of epochs from 400 to 10.000, and keeping in each case the model that obtained the lowest losses for the validation dataset.

Once we obtained the peak predictions for each molecule, as in the other Neural Network model, we considered the same two types of thresholds (**Threshold 1** and **Threshold 2**) to define whether a prediction indicated a peak or not. We also plotted the metrics of the models, by comparing the results obtained with the actual peaks present in the molecules.

Looking at the figures (Fig. 29, 30, 31 and 32), as in the previous case (ANN), we saw that Accuracy tend to increase as the number of trained models increased. The values of Precision, Recall and F1 were higher when the data inputs were randomised, as well as being very similar when the second threshold was used.

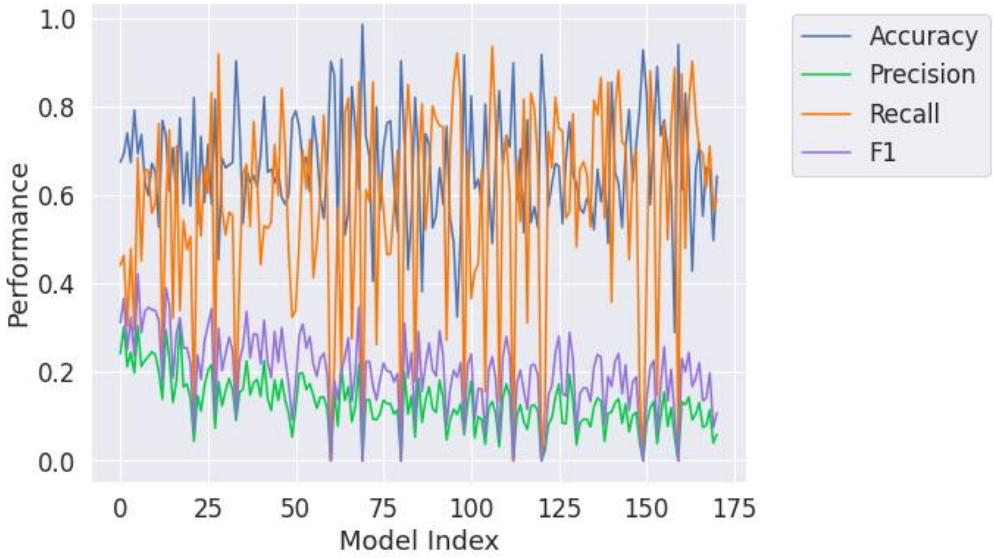

Figure 29. Results of the metrics for each GNN type model, with non-randomised input data and the first threshold (TH1). Each point represents the result of the metrics obtained by each model, trained with a test set of 2.600 molecules. On the x-axis, the 171 trained models were arranged from highest to lowest peak frequency, so that the model with Model Index 0 corresponded to the model that considered the position in the spectrum where the molecules in the training set most commonly presented a peak.

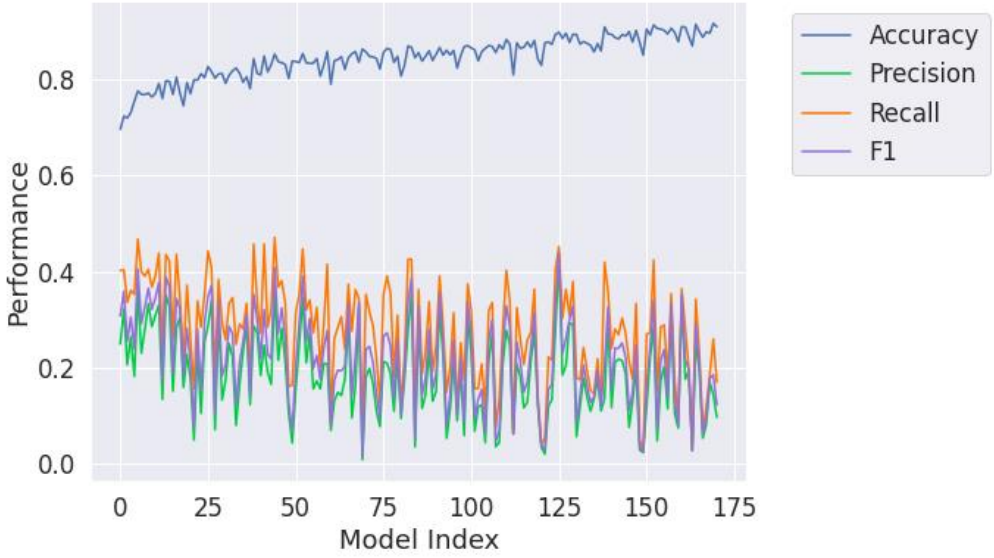

Figure 30. Results of the metrics for each GNN type model, with non-randomised input data and the second threshold (TH2). Each point represents the result of the metrics obtained by each model, trained with a test set of 2.600 molecules. On the x-axis, the 171 trained models were arranged from highest to lowest peak frequency, so that the model with Model Index 0 corresponded to the model that considered the position in the spectrum where the molecules in the training set most commonly presented a peak.

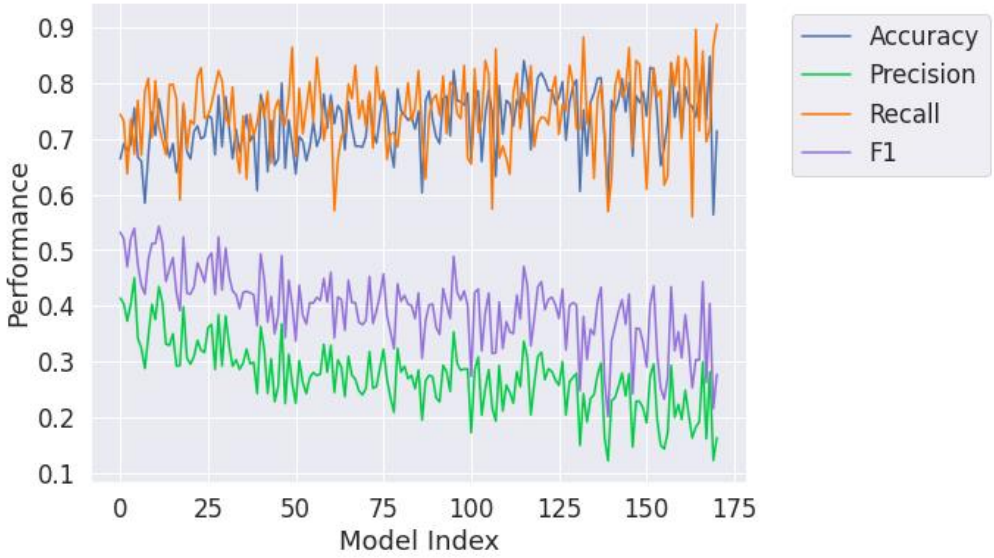

Figure 31. Results of the metrics for each GNN type model, with randomised input data and the first threshold (TH1 Random). Each point represents the result of the metrics obtained by each model, trained with a test set of 2.600 molecules. On the x-axis, the 171 trained models were arranged from highest to lowest peak frequency, so that the model with Model Index 0 corresponded to the model that considered the position in the spectrum where the molecules in the training set most commonly presented a peak.

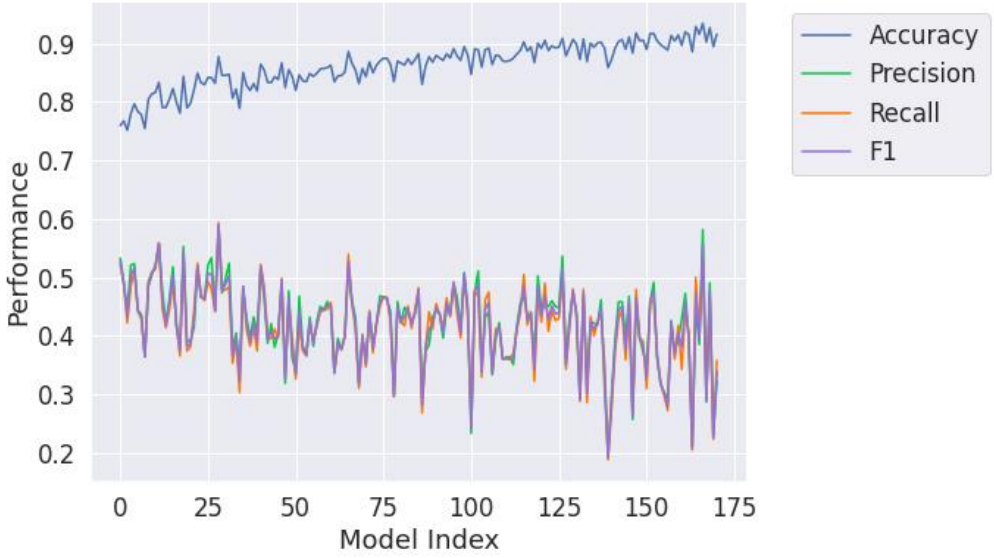

Figure 32. Results of the metrics for each GNN type model, with randomised input data and the second threshold (TH2 Random). Each point represents the result of the metrics obtained by each model, trained with a test set of 2.600 molecules. On the x-axis, the 171 trained models were arranged from highest to lowest peak frequency, so that the model with Model Index 0 corresponded to the model that considered the position in the spectrum where the molecules in the training set most commonly presented a peak.

4.2.3 Spectra Reconstruction

To reconstruct the spectra from the results obtained by the models for the different M/Z positions of the molecules, we performed the same procedure described in Section 4.1.3. Below, we show an example of a reconstruction of the spectrum compared to its real spectrum (Fig. 33). As in the previous case, and to see that the predictions obtained really showed

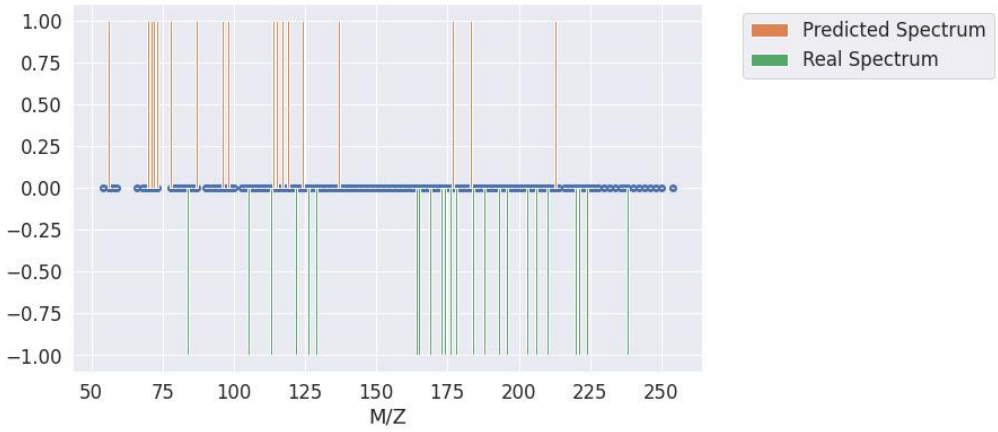

Figure 33. Reconstruction of the spectrum of molecule ID #786, using the GNN method with the second threshold and randomised input data (TH2 Random). Orange shows the peaks that we predicted, green shows the real peaks and blue shows the positions for which we had a model.

knowledge and did not work as if they were a random case, we calculated the model metrics between our results with their solutions, and our results with the spectra of other molecules (Fig. 34, 35, 36 and 37).

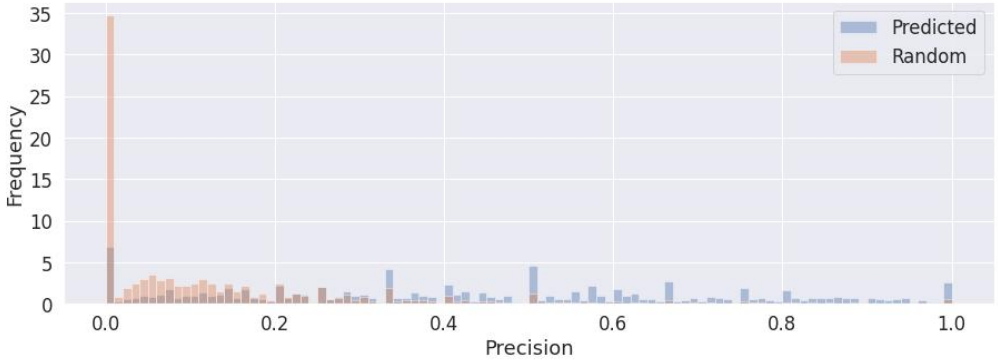

Figure 34. Distribution of the Precision obtained by the GNN models with the second threshold type and randomised input data (TH2 Random), versus the Precision of comparing our predictions with real random spectra.

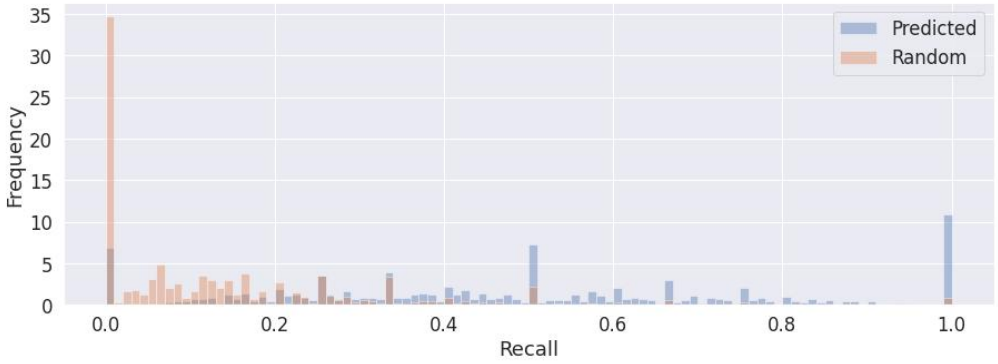

Figure 35. Distribution of the Recall obtained by the GNN models with the second threshold type and randomised input data (TH2 Random), versus the Recall of comparing our predictions with real random spectra.

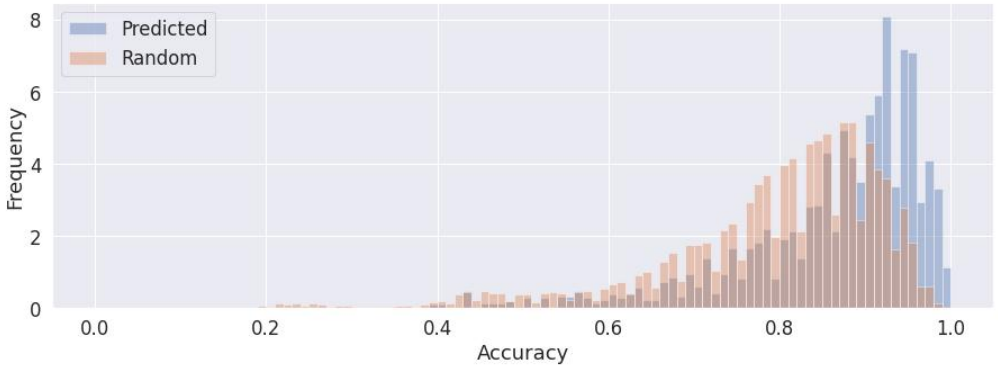

Figure 36. Distribution of the Accuracy obtained by the GNN models with the second threshold type and randomised input data (TH2 Random), versus the Accuracy of comparing our predictions with real random spectra.

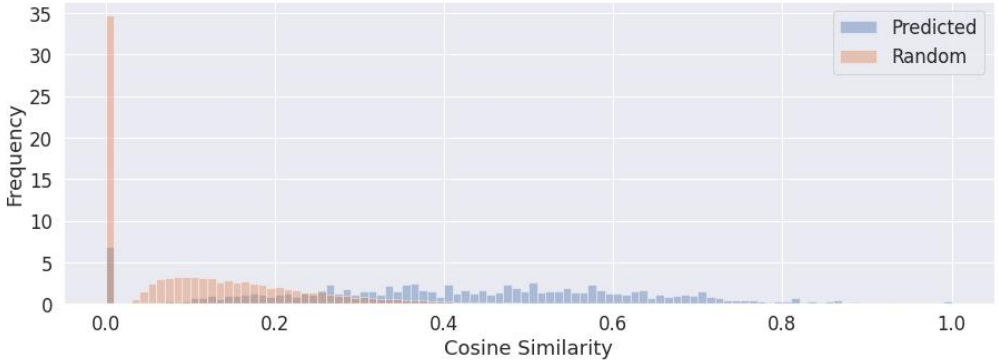

Figure 37. Distribution of the Cosine similarity obtained by the GNN models with the second threshold type and randomised input data (TH2 Random), versus the Cosine similarity of comparing our predictions with real random spectra.

4.2.4 CFM-ID Spectra Reconstruction

Finally, just as in the ANN we compared the results of our reconstructed spectra with those predicted by the *CFM-ID Spectra Prediction*, we did the same for the GNN. In this case, these were the results obtained for the method with randomly ordered data at the input, and the second threshold (Fig. 38, 39, 40 and 41).

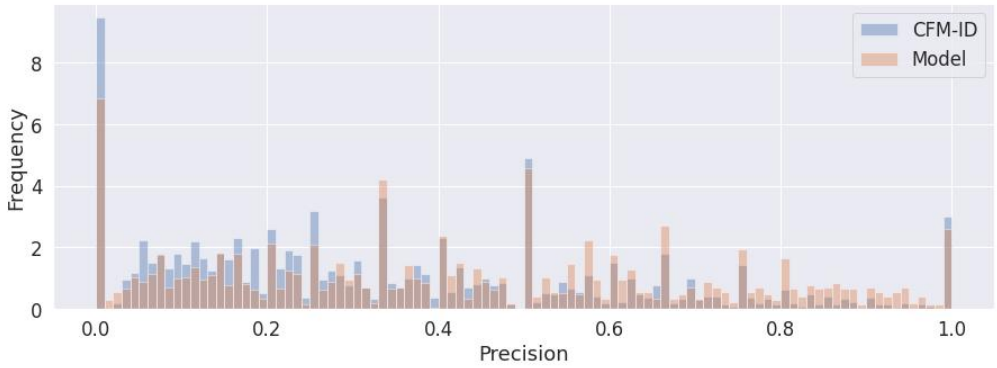

Figure 38. Distribution of the Precision obtained by the GNN models with the second threshold type and randomised input data (TH2 Random), versus the Precision obtained by the CFM-ID.

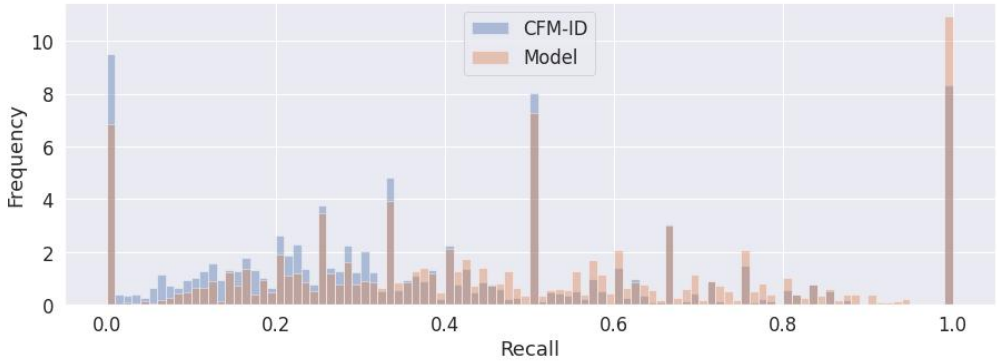

Figure 39. Distribution of the Recall obtained by the GNN models with the second threshold type and randomised input data (TH2 Random), versus the Recall obtained by the CFM-ID.

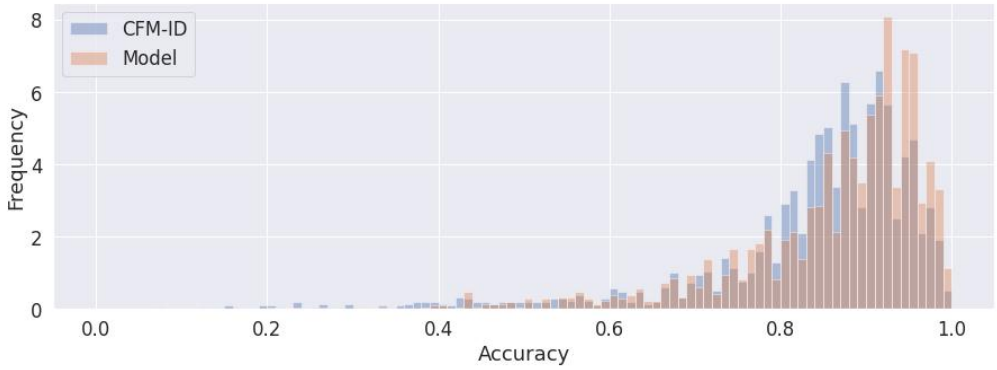

Figure 40. Distribution of the Accuracy obtained by the GNN models with the second threshold type and randomised input data (TH2 Random), versus the Accuracy obtained by the CFM-ID.

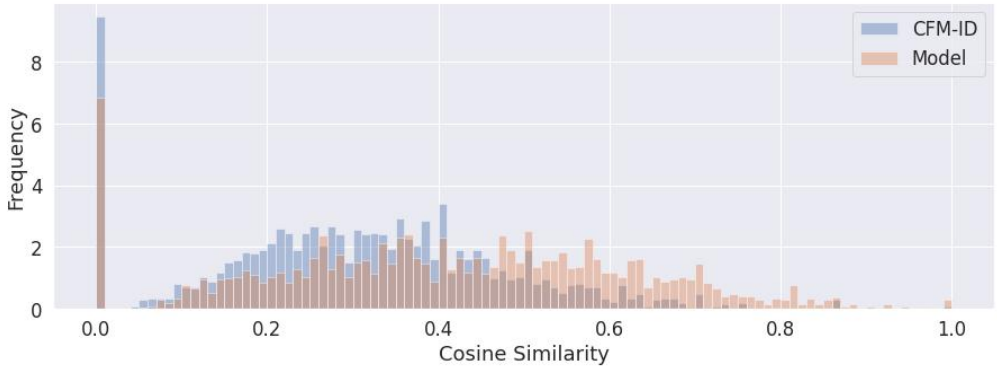

Figure 41. Distribution of the Cosine similarity obtained by the GNN models with the second threshold type and randomised input data (TH2 Random), versus the Cosine similarity obtained by the CFM-ID.

4.2.5 Summary and interpretation

To finish with the comparison between the results obtained using Graph Neural Networks, we followed the same steps as for the ANNs. That is, we first performed scatter plots to compare the Precision, Recall, Accuracy and Cosine similarity results of our algorithms with those of the CFM-ID (figures in Appendix 2), from which we noted the number of molecules that obtained better scores for each type, or if they obtained the same result for both cases. Ultimately, we put these results together in a bar chart and recorded the quantitative values in the table below.

First, for the algorithm with the non-randomised data with the first threshold type (TH1), the only score that was higher for our algorithm was the Recall score, since the number of predicted peaks per molecule was superior (Fig. 42). Nonetheless, the Precision, Accuracy and Cosine similarity scores were better for the CFM-ID algorithm.

For the method without the randomised data but with the second threshold (TH2), the

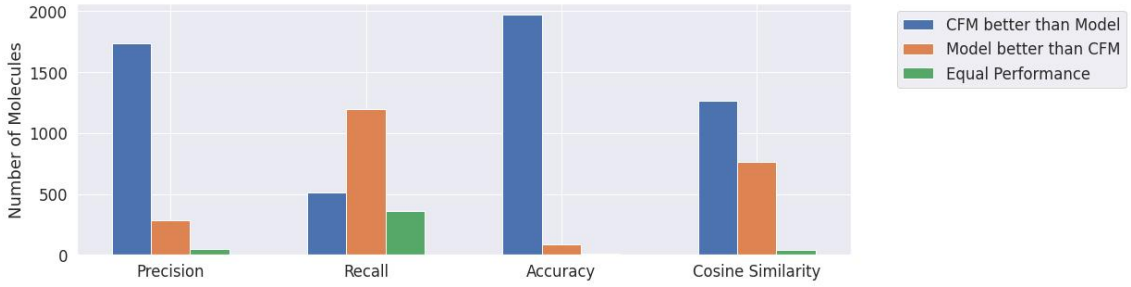

Figure 42. Count of the number of molecules for which a better metric was obtained between the GNN method with non-random data input and the first threshold (TH1), versus the CFM-ID.

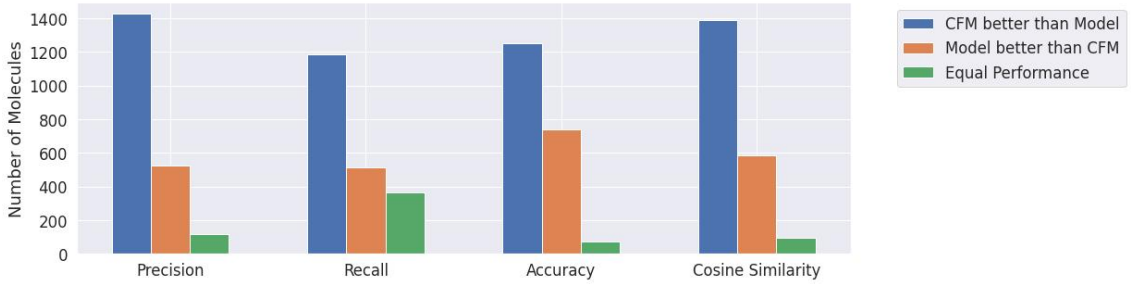

Figure 43. Count of the number of molecules for which a better metric was obtained between the GNN method with non-random data input and the second threshold (TH2), versus the CFM-ID.

prediction by the CFM-ID was better as it performed better on all model metrics (Fig. 43).

In the case of the randomised data and with the first threshold (TH1 Random), the values of our model for Recall and Cosine similarity were better than those of the CFM-ID, so that the similarity between the real spectrum of the molecules and the one predicted in our case was closer than the one predicted by the CFM-ID. Although, as our Precision result was worse, the CFM-ID model obtained a better quality in the classification tasks (Fig. 44).

In the last case, where the data were randomised and the second threshold (TH2 Random) was used, the results for all four types of metrics were performed better by our predictions (Fig. 45). Moreover, the difference between our Recall versus CFM-ID score difference was lower than the one obtained in the previous case (TH1 Random). However, in this case (TH2 Random) the results obtained for Precision were much better.

To conclude this part, we note that in this case there was a significant difference in the prediction of the molecule spectra depending on whether the order of the input data was randomised or not. In the first two types, the only better score obtained by our model compared to the CFM-ID was the Recall (for the TH1 case), when with the mixed input data, for the TH1 Random case the Recall and the Cosine similarity were better and for the TH2 Random case all the metrics were better for our model. For this reason, the predictions of the spectra carried out with the randomised data and with the second type of threshold stood out from the other types of algorithms.

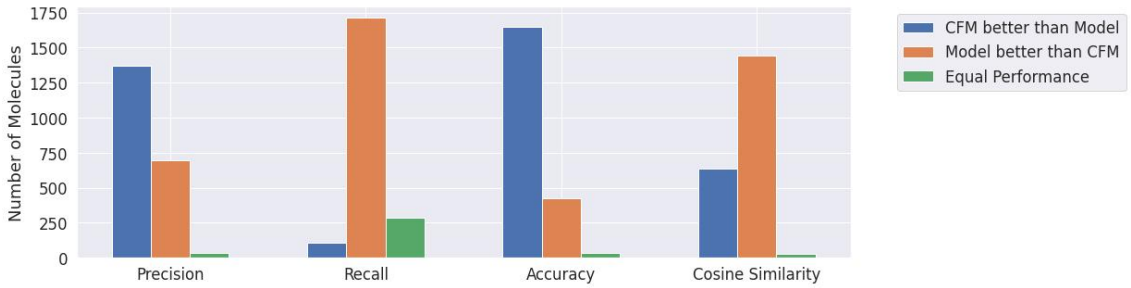

Figure 44. Count of the number of molecules for which a better metric was obtained between the GNN method with random data input and the first threshold (TH1 Random), versus the CFM-ID.

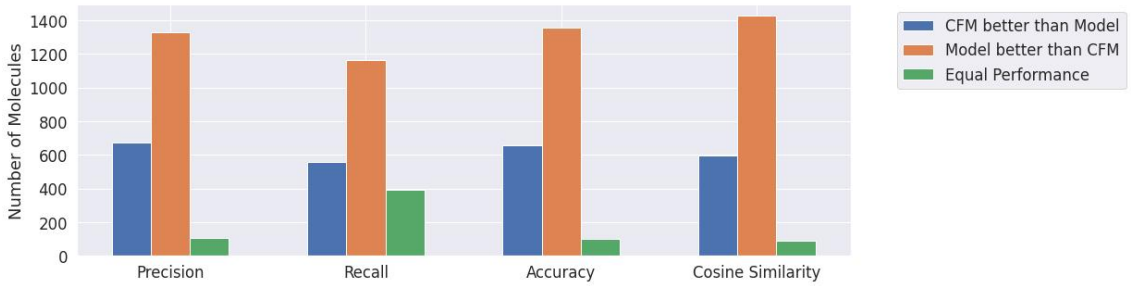

Figure 45. Count of the number of molecules for which a better metric was obtained between the GNN method with random data input and the second threshold (TH2 Random), versus the CFM-ID.

Table 2. Summary of the number of molecules predicted better by GNN.

Model	**Metric**	**CFM**	**Model**	**Equals**
TH1	Precision	1738	282	46
	Recall	511	1198	357
	Accuracy	1967	89	10
	Cosine Similarity	1263	765	38
TH2	Precision	1425	526	115
	Recall	1184	516	366
	Accuracy	1251	740	75
	Cosine Similarity	1388	584	94
TH1 Random	Precision	1371	699	37
	Recall	108	1712	287
	Accuracy	1647	426	34
	Cosine Similarity	636	1444	27
TH2 Random	Precision	675	1328	104
	Recall	555	1161	391
	Accuracy	654	1354	99
	Cosine Similarity	594	1425	88

5 Discussion

In this section we put all the results together to discuss what we found most significant in the experimental work and the most suitable model to solve the proposed problem.

Firstly, in order to carry out the predictions of the MS/MS spectra of the molecules, we created suitable mathematical representations for each of them. In the case of the Artificial Neural Network, we obtained the representations using the unsupervised Deep Learning method Mol2vec. In this way, for each molecule we had an embedded vector of 300 dimensions that related the substructures of the molecule using Morgan's algorithm. On the other hand, for the Graph Artificial Network we created two functions that calculated two embeddings for each molecule, one corresponding to the type of atoms present in the molecule, and the other to the bonds.

From the very beginning, as far as the Mol2vec representations were concerned, we could already see that they were quite closely related to the original structure of the molecules by obtaining a quite significant pvalue when relating these two vectors by means of the Cosine similarity. On the other hand, we already knew that the input data for the GNNs were representative of the molecules since our functions directly described their composition.

Subsequently, we trained 171 models for each type of Neural Network (ANN and GNN), corresponding to the 171 M/Z positions most frequented by peaks in the MS/MS spectra of the molecules in the training set. Likewise, for each type of Neural Network we trained one type of models with the input data ordered in the same way as in the original dataset, and others with the data randomised. In addition, for each case we predicted two different types of results, based on two different thresholds. The first one considered the fraction between the number of peaks appearing at the M/Z position of the model in question, and the total number of molecules in the training set. The second threshold was calculated in the same way, with the difference that it was calibrated according to the molecules in the test set. In other words, we calculated a total of eight predictive approaches to MS/MS spectra, formed from the reconstruction of the values obtained by each M/Z model. The eight types of predictions were:

1. Predictions with ANN-type models with the data not randomised according to the first threshold.
2. Predictions with ANN-type models with the data not randomised according to the second threshold.
3. Predictions with ANN-type models with the data randomised according to the first threshold.
4. Predictions with ANN-type models with the data randomised according to the second threshold.
5. Predictions with GNN-type models with the data not randomised according to the first threshold.
6. Predictions with GNN-type models with the data not randomised according to the second threshold.

7. Predictions with GNN-type models with the data randomised according to the first threshold.
8. Predictions with GNN-type models with the data randomised according to the second threshold.

For each of the above predictions, we calculated metrics to evaluate the classification of our models, and consequently, the quality of the reconstructions of the molecule spectra. From the outset, we could see that the predictions achieved by the Neural Networks made a certain amount of sense since, when representing the predicted spectrum against the real binarised spectrum, many peaks of example molecules coincided.

Therefore, in order to perform a quantitative comparison, we compared our results with those of CFM-ID. Currently, CFM-ID is the best fragmentation modelling method for predicting EI-MS and ESI-MS/MS spectra of a given compound based on Machine Learning. Its version 4.0 can predict a large number of chemical compounds more precisely, with better performance than any in silico tool published to date [17]. Thus, we calculated the prediction using CFM-ID for the same molecules for which we had results (both for those in the case with randomised data and those without). We were then able to compare the Precision, Accuracy, Recall and Cosine similarity for each MS/MS spectrum predicted by each of our methods and by CFM-ID. Thus, we noted which obtained better results for each metric.

According to the results, the best method to carry out the predictions of the MS/MS spectra of molecules was the one obtained by the ANNs trained with random data and considering the second type of threshold. The results for this method showed better performance for all types of metrics compared to the CFM-ID, which means that this method has better quality in the classification tasks, identifies a higher number of peaks, obtains a higher percentage of correct predictions and results in more similar spectra compared to the real ones.

Consequently, in relation to the comparison between the two types of Neural Networks we showed that for this case the Artificial Neural Network obtains better results. This means that the embedded vector formed from the Mol2vec method is sufficiently characteristic for each molecule, even though it is only a representation of the structure and not of specific characteristics of the atoms and bonds of the molecules as in the GNN.

On the other hand, with regard to the comparison of our models with those of the CFM-ID, we clearly observed a better performance on our side. However, it should be noted that our models only predict whether there is a peak or not at certain positions in the binarised spectrum. The CMF-ID algorithm represents the spectrum of the molecules by peaks with varying intensities. But the intensity of the signal of the ions in a mass spectrum is not such an identifying value of the molecules since it can be variable and depends on many factors. The energy of the electron beam, the location of the sample with respect to the beam, the vapour pressure of the sample, the temperature of the ion source, etc., give rise to significant variations in the relative abundance of the ions for spectra obtained in different laboratories and under different conditions. Thus, it is the x-axis in a spectrum that contains mass information, since it represents a relationship between the mass of the ion and the number of elementary charges carried by it [46].

Turning to the predictive capability of our models, we highlight two important aspects

to consider. Firstly, all the methods we implemented to carry out the predictions are able to predict whether or not there is a peak at a position in the spectrum which corresponds to a 1 M/Z bin. This means that in reality, within the 1 M/Z interval, a molecule can have several peaks, which may or may not be related. This implies that the level of difficulty is higher since molecules with very different structures may have peaks for the same M/Z interval, when in reality they correspond to very different fragments. Secondly, our models do not need to know the collision energy at which a molecule has fragmented, but directly predict all the possible fragments into which it can be broken. In the case of CFM-ID, different MS/MS spectra are obtained for different collision energies and the M/Z positions correspond to specific values frequented by molecules fragments.

6 Conclusions

The project's main objective was to find the best model that could predict the tandem mass spectra of any molecule, given its molecular formula as a SMILE format. We have followed a workflow that has allowed us to build and compare different Deep Learning algorithms that we thought to be potential for this purposed.

In doing so, we have been able to compare the performance of an Artificial Neural Network and a Graph Neural Network, as well as to compare our results with the best in silico tool for the prediction of the MS/MS spectrum of molecules to date. Accordingly, we have been able to determine that between the models used in this project and the CFM-ID algorithm, the best method for prediction is the one formed by ANN type models, with a randomly ordered data input (according to our database used) and a calibrated threshold.

To achieve this, we have reached other objectives defined at the beginning of the work, such as obtaining adequate mathematical representations of the molecules. In this way, both for one Neural Network and for the other, their input data have been sufficiently representative since the results have been shown to have a high predictive potential. In all but one case, i.e. in seven of the eight cases, the Recall values were higher than those of the CFM-ID.

Finally, we can conclude that although our project has obtained excellent results, Neural Networks and Deep Learning are the art of imagining a design, training it and adjusting it according to its performance. So they can always be improved (by changing and adding parameters and layers), refining the details to reach the final goal in the best possible way. For this motivation, we think that the results would be even better if instead of dividing the x-axis of the spectrum in bins of 1 M/Z, we focus directly on the positions where we know that the molecule fragments are located in most cases. This could be done by obtaining from large databases the exact positions for which the molecules have peaks. Furthermore, instead of analysing the prediction results of ANNs and GNNs separately, we could combine these Neural Networks by building one that includes the predictive power of both. With these possible improvements and on the basis that our project has already obtained excellent results, we believe that future research on this project could lead to a promising tool for the identification of molecules in the field of metabolomics.

7 Bibliography

[1] J. L. Jameson and D. L. Longo, "Precision medicine—personalized, problematic, and promising," *Obstetrical & gynecological survey*, vol. 70, no. 10, pp. 612–614, 2015.

[2] J. Hristova and D. Svinarov, "Enhancing precision medicine through clinical mass spectrometry platform," *Biotechnology & Biotechnological Equipment*, vol. 36, no. 1, pp. 106–116, 2022.

[3] R. Bujak, W. Struck-Lewicka, M. J. Markuszewski, and R. Kaliszan, "Metabolomics for laboratory diagnostics," *Journal of pharmaceutical and biomedical analysis*, vol. 113, pp. 108–120, 2015.

[4] P. Hernández-Alonso, N. Becerra-Tomás, C. Papandreou, *et al.*, "Plasma metabolomics profiles are associated with the amount and source of protein intake: A metabolomics approach within the predimed study," *Molecular Nutrition & Food Research*, vol. 64, no. 12, p. 2 000 178, 2020.

[5] J. R. Idle and F. J. Gonzalez, "Metabolomics," *Cell metabolism*, vol. 6, no. 5, pp. 348–351, 2007.

[6] R. Madsen, T. Lundstedt, and J. Trygg, "Chemometrics in metabolomics—a review in human disease diagnosis," *Analytica chimica acta*, vol. 659, no. 1-2, pp. 23–33, 2010.

[7] E. C. Y. Chan, P. K. Koh, M. Mal, *et al.*, "Metabolic profiling of human colorectal cancer using high-resolution magic angle spinning nuclear magnetic resonance (hr-mas nmr) spectroscopy and gas chromatography mass spectrometry (gc/ms)," *Journal of proteome research*, vol. 8, no. 1, pp. 352–361, 2009.

[8] J. Lénárt, A. Gere, T. Causon, *et al.*, "Lc–ms based metabolic fingerprinting of apricot pistils after self-compatible and self-incompatible pollinations," *Plant Molecular Biology*, vol. 105, no. 4, pp. 435–447, 2021.

[9] J. Debik, M. Sangermani, F. Wang, T. S. Madssen, and G. F. Giskeødegård, "Multivariate analysis of nmr-based metabolomic data," *NMR in Biomedicine*, vol. 35, no. 2, e4638, 2022.

[10] C. Dass, *Fundamentals of contemporary mass spectrometry*. John Wiley & Sons, 2007, vol. 16.

[11] Y. Djoumbou-Feunang, A. Pon, N. Karu, *et al.*, "Cfm-id 3.0: Significantly improved esi-ms/ms prediction and compound identification," *Metabolites*, vol. 9, no. 4, p. 72, 2019.

[12] L. Specialty Gases & Specialty Equipment, *Mass spectrometry (lc-ms)*. [Online]. Available: `http://hiq.linde-gas.com/en/analytical_methods/liquid_chromatography/mass_spectrometry.html`.

[13] *Human metabolome database*. [Online]. Available: `https://hmdb.ca/`.

[14] *Metlin*. [Online]. Available: `http://metlin.scripps.edu/`.

[15] *Kyoto encyclopedia of genes and genomes*. [Online]. Available: `http://www.genome.jp/kegg`.

[16] J. Rainer, A. Vicini, L. Salzer, *et al.*, "A modular and expandable ecosystem for metabolomics data annotation in r," *Metabolites*, vol. 12, no. 2, p. 173, 2022.

[17] F. Wang, J. Liigand, S. Tian, D. Arndt, R. Greiner, and D. S. Wishart, "Cfm-id 4.0: More accurate esi-ms/ms spectral prediction and compound identification," *Analytical Chemistry*, vol. 93, no. 34, pp. 11 692–11 700, 2021.

[18] S. Tiwary, R. Levy, P. Gutenbrunner, *et al.*, "High-quality ms/ms spectrum prediction for data-dependent and data-independent acquisition data analysis," *Nature methods*, vol. 16, no. 6, pp. 519–525, 2019.

[19] MassBank, *About massbank*. [Online]. Available: `https://massbank.eu/MassBank/About`.

[20] S. User, Apr. 2022. [Online]. Available: `https://www.denbi.de/`.

[21] [Online]. Available: `https://www.ipb-halle.de/en/`.

[22] Apr. 2022. [Online]. Available: `https://www.ufz.de/index.php?en=33573`.

[23] U. d. Luxembourg, *Home*. [Online]. Available: `https://wwwen.uni.lu/`.

[24] D. Romera, E. M. Mateo, R. Mateo-Castro, J. V. Gomez, J. V. Gimeno-Adelantado, and M. Jimenez, "Determination of multiple mycotoxins in feedstuffs by combined use of uplc–ms/ms and uplc–qtof–ms," *Food chemistry*, vol. 267, pp. 140–148, 2018.

[25] M. Olivenboim, E. Cohen, L. Burshtein, *et al.*, "A 90° bend curved light-guide for tof scintillating detectors," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 1018, p. 165 825, 2021.

[26] D. Weininger, "Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules," *Journal of chemical information and computer sciences*, vol. 28, no. 1, pp. 31–36, 1988.

[27] *Rdkit.chem.rdmolfiles module*. [Online]. Available: `https://www.rdkit.org/docs/source/rdkit.Chem.rdmolfiles.html`.

[28] D. Ghosh, S. Chakraborty, H. Kodamana, and S. Chakraborty, "Application of machine learning in understanding plant virus pathogenesis: Trends and perspectives on emergence, diagnosis, host-virus interplay and management," *Virology Journal*, vol. 19, no. 1, pp. 1–11, 2022.

[29] D. Esposito and F. Esposito, *Introducing Machine Learning*. Microsoft Press, 2020.

[30] A. L. Tarca, V. J. Carey, X.-w. Chen, R. Romero, and S. Drăghici, "Machine learning and its applications to biology," *PLoS computational biology*, vol. 3, no. 6, e116, 2007.

[31] X.-X. Zhou, W.-F. Zeng, H. Chi, *et al.*, "Pdeep: Predicting ms/ms spectra of peptides with deep learning," *Analytical chemistry*, vol. 89, no. 23, pp. 12 690–12 697, 2017.

[32] S. Shibayama, G. Marcou, D. Horvath, I. I. Baskin, K. Funatsu, and A. Varnek, "Application of the mol2vec technology to large-size data visualization and analysis," *Molecular Informatics*, vol. 39, no. 6, p. 1 900 170, 2020.

[33] S. Jaeger, S. Fulle, and S. Turk, "Mol2vec: Unsupervised machine learning approach with chemical intuition," *Journal of chemical information and modeling*, vol. 58, no. 1, pp. 27–35, 2018.

[34] D. Rogers and M. Hahn, "Extended-connectivity fingerprints," *Journal of chemical information and modeling*, vol. 50, no. 5, pp. 742–754, 2010.

[35] A. Abraham, "Artificial neural networks," *Handbook of measuring system design*, 2005.

[36] Simplilearn, *What is perceptron: A beginners guide for perceptron [updated]*, Feb. 2022. [Online]. Available: `https://www.simplilearn.com/tutorials/deep-learning-tutorial/perceptron`.

[37] J. Brownlee, *A gentle introduction to the rectified linear unit (relu)*, Aug. 2020. [Online]. Available: https://machinelearningmastery.com/rectified-linear-activation-function-for-deep-learning-neural-networks/#:~:text=The%5C%20rectified%5C%20linear%5C%20activation%5C%20function,otherwise%5C%2C%5C%20it%5C%20will%5C%20output%5C%20zero.

[38] B. Sanchez-Lengeling, E. Reif, A. Pearce, and A. B. Wiltschko, "A gentle introduction to graph neural networks," *Distill*, vol. 6, no. 9, e33, 2021.

[39] A. Daigavane, B. Ravindran, and G. Aggarwal, "Understanding convolutions on graphs," *Distill*, vol. 6, no. 9, e32, 2021.

[40] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," *stat*, vol. 1050, p. 20, 2017.

[41] T. Wood, *Softmax function*, May 2019. [Online]. Available: https://deepai.org/machine-learning-glossary-and-terms/softmax-layer.

[42] M. Labonne, *Graph attention networks: Self-attention explained*, Apr. 2022. [Online]. Available: https://towardsdatascience.com/graph-attention-networks-in-python-975736ac5c0c.

[43] J. M. Heras, Lita, Gabriel, Jose, and Manuel, *Precision, recall, f1, accuracy en clasificación*, Oct. 2020. [Online]. Available: https://www.iartificial.net/precision-recall-f1-accuracy-en-clasificacion/.

[44] J. Han, J. Pei, and M. Kamber, *Data mining: concepts and techniques*. Elsevier, 2011.

[45] *Cfm-id: Spectra prediction*. [Online]. Available: https://cfmid.wishartlab.com/predict.

[46] CSIC, *Espectrometria de masas - museo nacional de ciencias naturales*. [Online]. Available: https://www.mncn.csic.es/docs/repositorio/es_ES/investigacion/cromatografia/espectrometria_de_masas.pdf.

Appendix 1. Programming code

Examples of the codes we have developed for the implementation of the Deep Learning algorithms can be found in the following repository: https://github.com/MaribelPR/TFG_DL_MSMSprediction.git. They are generally private, except if someone requests access to them.

Appendix 2. Additional figures

Dispersion plots of the Precision, Recall, Accuracy and Cosine similarity obtained for the predictions of each molecule of the CFM-ID, compared to those obtained from the prediction of our method using the ANN models with the second threshold type and randomised input data (TH2 Random).

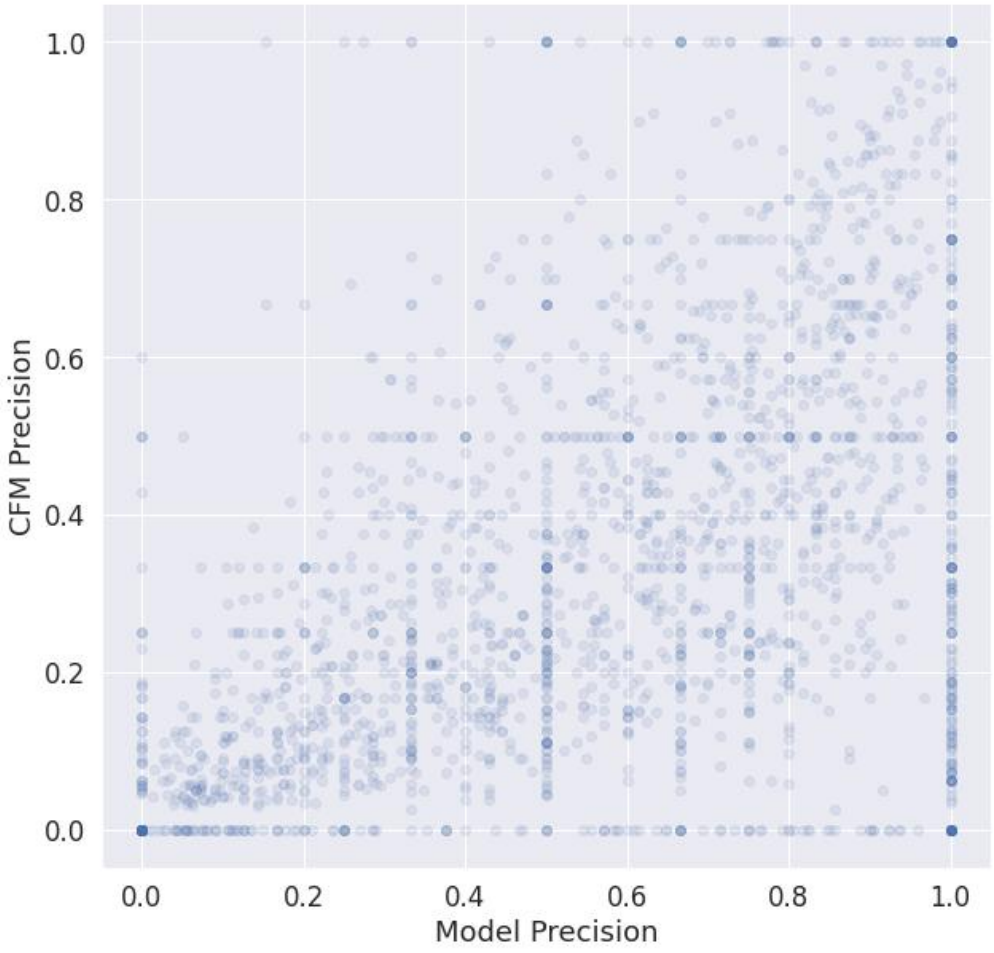

Figure 46. Dispersion of the Precision obtained by the ANN models with the second threshold type and randomised input data (TH2 Random), versus the Precision obtained by the CFM-ID.

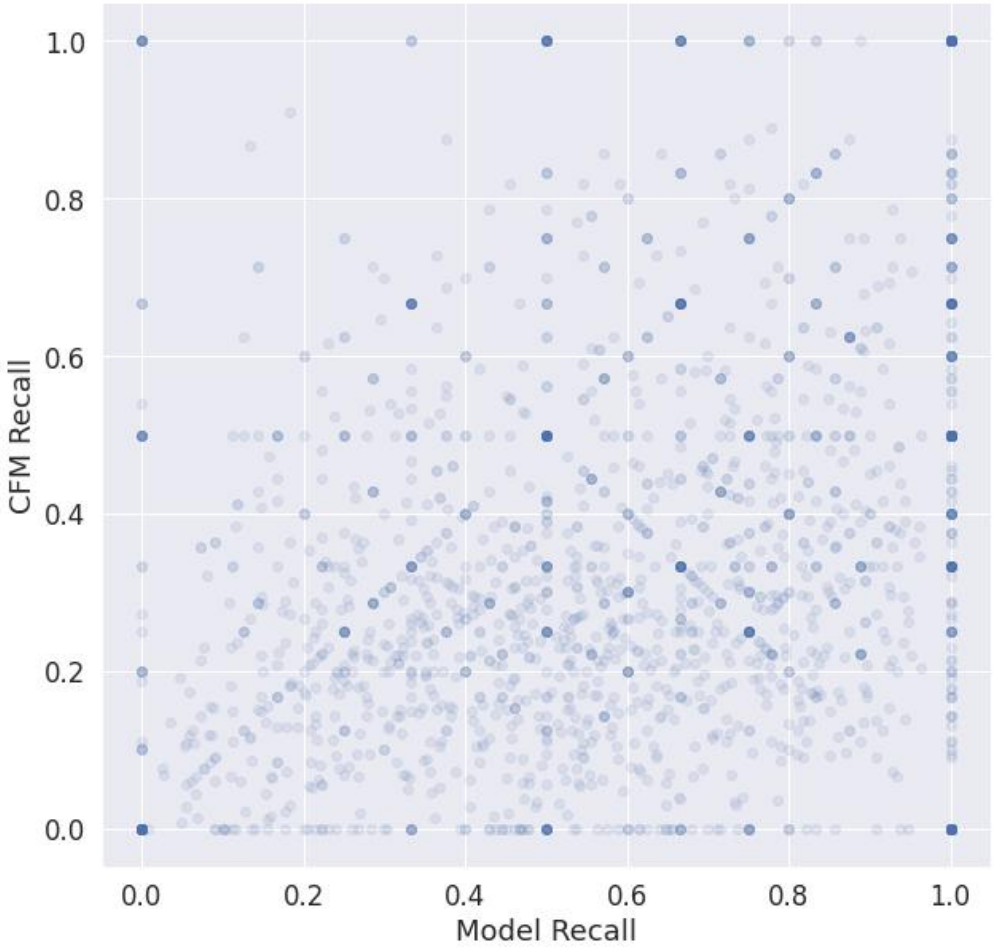

Figure 47. Dispersion of the Recall obtained by the ANN models with the second threshold type and randomised input data (TH2 Random), versus the Recall obtained by the CFM-ID.

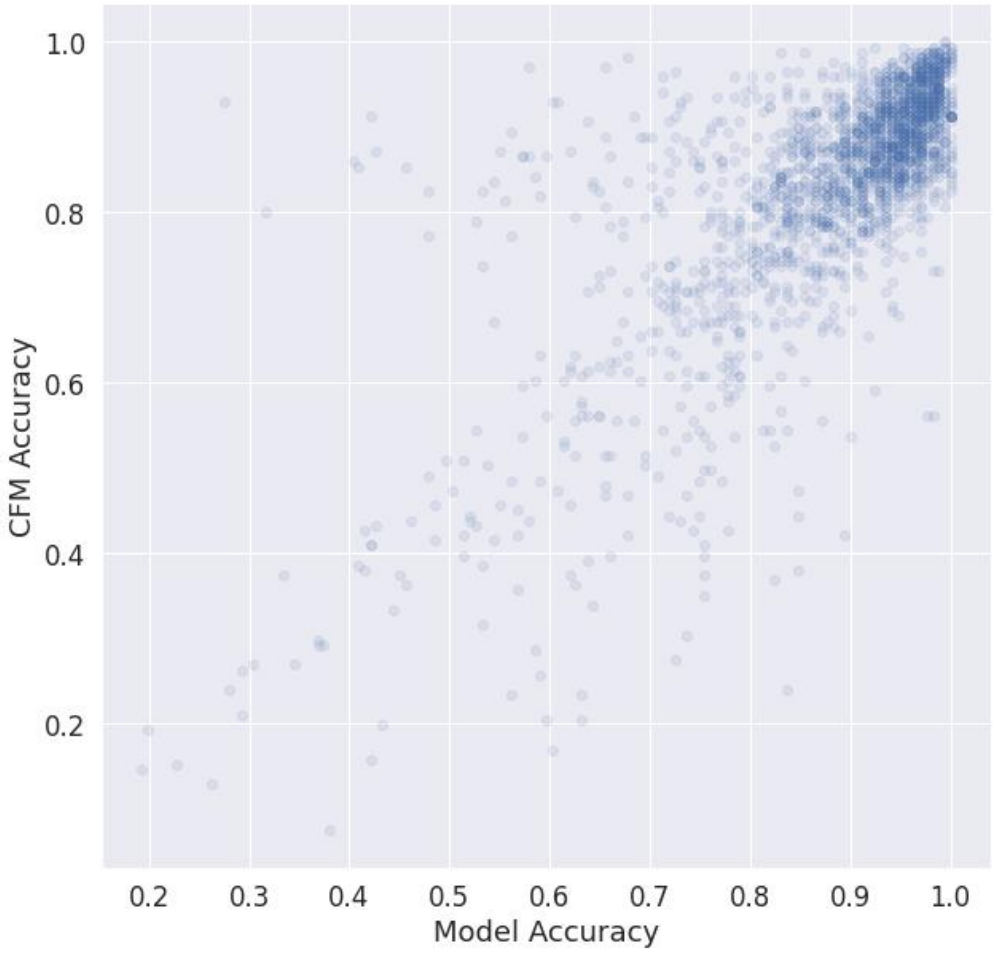

Figure 48. Dispersion of the Accuracy obtained by the ANN models with the second threshold type and randomised input data (TH2 Random), versus the Accuracy obtained by the CFM-ID.

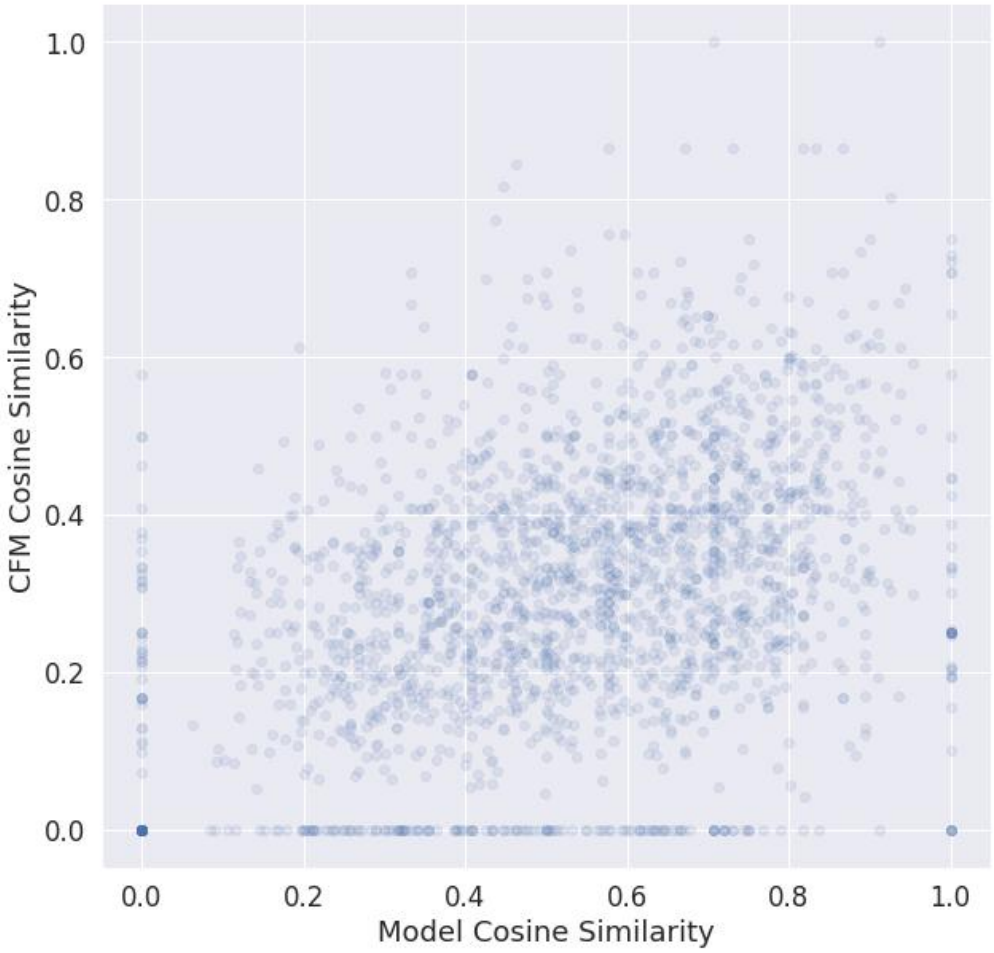

Figure 49. Dispersion of the Cosine similarity obtained by the ANN models with the second threshold type and randomised input data (TH2 Random), versus the Cosine similarity obtained by the CFM-ID.

Dispersion plots of the Precision, Recall, Accuracy and Cosine similarity obtained for the predictions of each molecule of the CFM-ID, compared to those obtained from the prediction of our method using the GNN models with the second threshold type and randomised input data (TH2 Random).

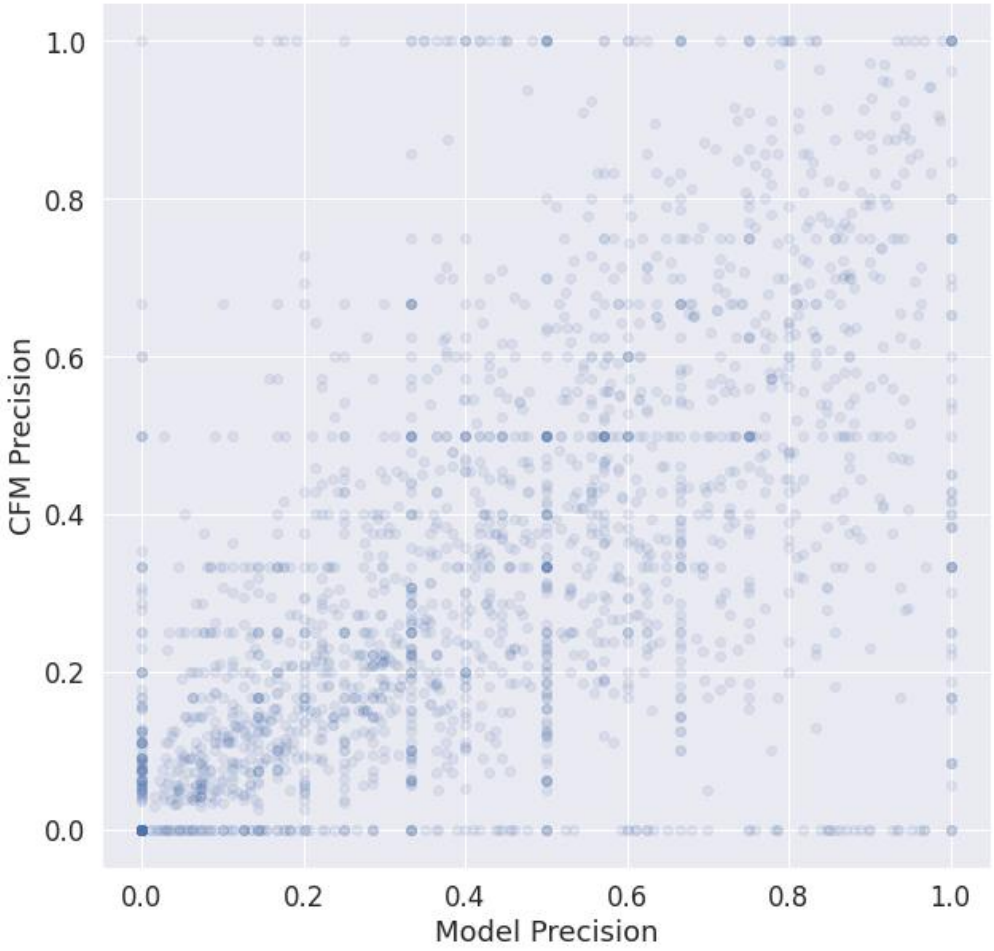

Figure 50. Dispersion of the Precision obtained by the GNN models with the second threshold type and randomised input data (TH2 Random), versus the Precision obtained by the CFM-ID.

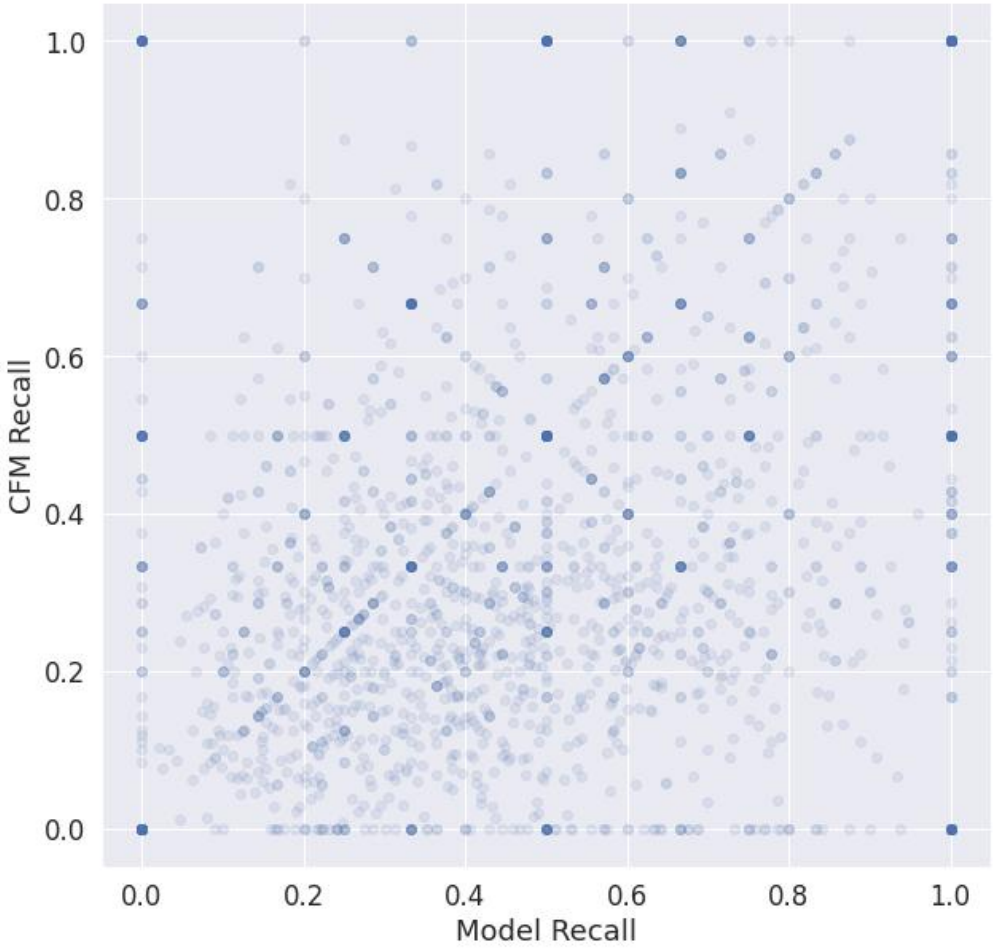

Figure 51. Dispersion of the Recall obtained by the GNN models with the second threshold type and randomised input data (TH2 Random), versus the Recall obtained by the CFM-ID.

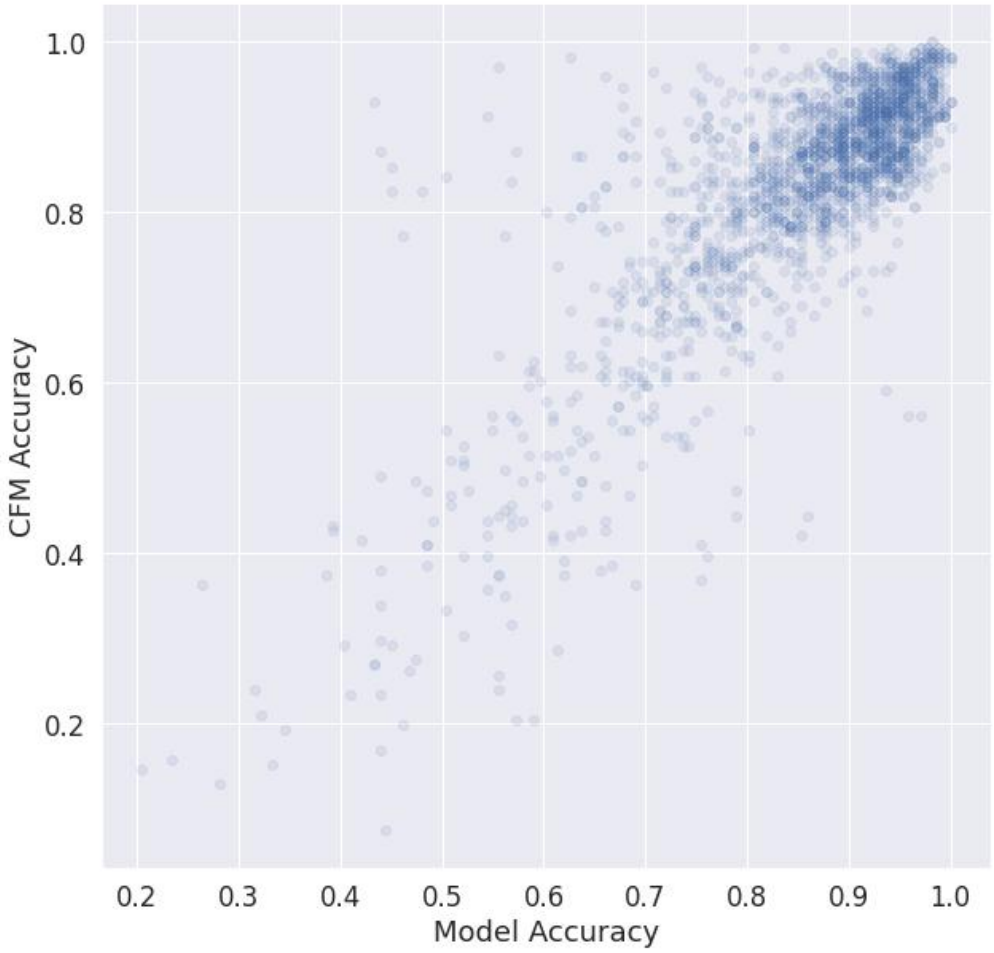

Figure 52. Dispersion of the Accuracy obtained by the GNN models with the second threshold type and randomised input data (TH2 Random), versus the Accuracy obtained by the CFM-ID.

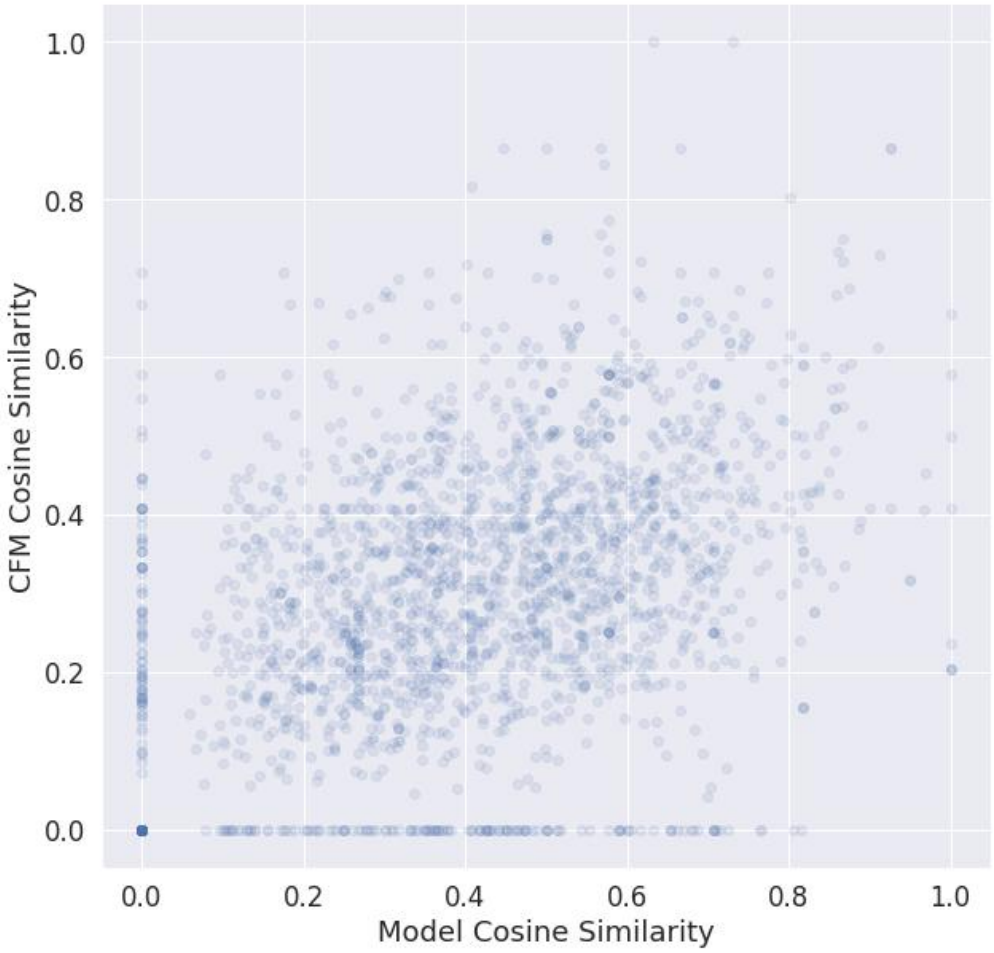

Figure 53. Dispersion of the Cosine similarity obtained by the GNN models with the second threshold type and randomised input data (TH2 Random), versus the Cosine similarity obtained by the CFM-ID.