An Interpretable Deep Learning Approach for Biomarker Detection in LC-MS Proteomics Data

Sahar Iravani and Tim O.F. Conrad

Abstract—Analyzing mass spectrometry-based proteomics data with deep learning (DL) approaches poses several challenges due to the high dimensionality, low sample size, and high level of noise. Besides, DL-based workflows are often hindered to be integrated into medical settings due to the lack of interpretable explanation. We present DLearnMS, a DL biomarker detection framework, to address these challenges on proteomics instances of liquid chromatography-mass spectrometry (LC-MS) - a well-established tool for quantifying complex protein mixtures. Our DLearnMS framework learns the clinical state of LC-MS data instances using convolutional neural networks. Next, based on the trained neural networks, biomarkers can be identified using the layer-wise relevance propagation technique. This enables detecting discriminating regions of the data and the design of more robust networks. We show that DLearnMS outperforms conventional LC-MS biomarker detection approaches in detecting fewer false positive peaks while maintaining a comparable amount of true positives peaks. Unlike other methods, no explicit preprocessing step is needed in DLearnMS.

Index Terms—Biomarker Detection, Mass Spectrometry, LC-MS Proteomics, Deep Learning Interpretation, Layer-Wise Relevance Propagation

1 Introduction

LIQUID chromatography-mass spectrometry (LC-MS) based proteomics allows the analysis of complex biological mixtures, such as body fluids (e.g., blood or urine). Due to the precise and fast quantification process, it is widely used in high-throughput proteomics applications [1]–[3], such as disease diagnosis (or prognosis), biomarker detection, or drug target identification. LC-MS first differentiates the protein components based on their physio-chemical properties, and then separates the ionized components based on their molecular mass and charge, the mass-to-charge ratio (m/z). This process results in a so-called LC-MS map that has two orthogonal dimensions of separation – chromatographic retention time (RT) and m/z.

Expressed proteins on the LC-MS map are large in abundance range, typically highly complex, and contain a high level of noise. These factors make biomarker detection from raw LC-MS data challenging [1], [4]. The idea of biomarker detection - also known as feature selection - is to discover the identification of proteins by which a specific medical condition can be determined. Biomarkers in this study are differentially abundant single peaks specified by m/z and RT on raw LC-MS map. As an advantage of biomarker detection a medical condition can be determined particularly by just focusing on the biomarker related areas, which leads to reduce computational cost and conserve time.

Conventional LC-MS biomarker discovery tools [5]–[8] often start with a peak detection step to extract interesting and informative areas due to the difficulties of processing noisy, sparse, and high-throughput raw LC-MS samples. Some well-known examples for peak detection are the MsInspect Software [5] which identifies peaks using a wavelet additive decomposition, the MZmine 2 software [6] which applies a deconvolution algorithm on each chromatogram to detect peaks, and the Progenesis LC-MS software [8] that uses a wavelet-based approach in such a way that all relevant quantitation and positional information are retained. Other frameworks include XCMS [7] in which the peak detection step is addressed by developing a pattern matching approach on overlaid extracted ion chromatograms with Gaussian kernels; AB3D [9] which iteratively takes the highest intensity peak candidates and heuristically keeps or removes neighboring peaks to form peptide features; MSight [10] which adapts an image-based peak detection on the generated images from LC-MS maps; and MaxQuant [11] in which a correlation analysis involving a fit to a Gaussian peak shape is applied.

Subsequently, the detected peaks are used for biomarker detection through a combination of several steps, including noise reduction, RT alignment [12]–[14], data normalization [15], data filtering [16], baseline correction, and peak grouping. It is likely, however, to miss low-intensity peaks through different levels of processing. Moreover, the tuned parameters may need to be adjusted again for any data from new sources. In this muniscript, we present a biomarker detection approach which reaches overall better performance than mentioned conventional biomarker approaches independent to the aforementioned preprocessing steps.

The success of deep learning (DL)-based methods, often replacing state-of-the-art classical model-based methods, in many fields such as medical imaging [17], biomedicine [18], and healthcare [19], has also encouraged the use of DL models for LC-MS proteomics analysis. To name a few, DeepIso [20] that combines a convolutional neural network (CNN) with a recurrent neural network (RNN) to detect peptide features; DeepNovo [21] and DeepNovo-DIA [22], which use DL-based approach (CNN coupled with RNN) for peptide sequencing on data-dependent acquisition (tandem

- *S. Iravani is with the department of Visual and Data-centric Computing, Zuse Institute of Berlin, Germany.*
  E-mail: iravani@zib.de
- *T. Conrad is with the department of Visual and Data-centric Computing, Zuse Institute of Berlin, Germany and department of Mathematics and Computer Science in Freie Universität Berlin, Germany.*
  E-mail: conrad@zib.de

mass spectra) and data-independent acquisition MS data, respectively; pDeep [23] that adapt the bidirectional long short term memory for the spectrum prediction of peptides; and DeepRT [24] that employs a capsule network to predict RT by learning features of embedded amino acids in peptides. Despite the current successful DL approaches on analyzing LC-MS proteomics, most of the studies are empirically driven, and having a justifiable interpretation foundation is largely missing [25]. Moreover, as machine learning (ML) and DL have been rapidly growing in real-world applications, a concern has emerged that the high precision accuracy may not be enough in practice [26], and interpreting the decisions is important for robustness, reliability, and enhancement of a system. To address this challenge our first contribution consist of leveraging DL interpretability to analyze LC-MS proteomics data.

DNN explanation provides information about what makes a network arrive at a certain decision. Practical post-hoc explanation methods can be divided into four categories: (1) the *function* analysis explains DL model itself through gradient and can show how much changes in input pixel affect the output [27], [28], (2) the *attribution* method interprets the output of the model and explain which features and to what extent contribute to the model's output [29], [29]–[31], (3) the *signal* method tries to find patterns in inputs on which the decision is based [32]–[34], and (4) the *perturbation* analysis that can also be employed for interpreting ML methods calculate the importance of features through measuring the effect of perturbing the elements of inputs on the output [35]–[37]. The perturbation can be a simple occlusion [35], an inpainting occluded pattern using generative models [36], or a meaningful perturbation that is synthesized [37]. An application of DNN explanation employing perturbation analysis has previously studied in metabolomics [38]. However, permutation analysis is not computationally feasible for high-throughput LC-MS analysis. Instead, we employ layer-wise relevance propagation (LRP) [31] in the *attribution* analysis category. The outperformance of LRP against other categories has been demonstrated in our previous work [25] on MALDI-TOF MS data. LRP method propagates back the value of the decision neuron to the input layer, and weights each element based on its contrition. This approach benefits from calculating the weights at once, which makes interpretation fast in appose to perturbation analysis.

We propose DLeanMS, a biomarker detection approach that adapts LRP interpretation strategy to analyze and understand LC-MS data. Given two groups of healthy and diseased LC-MS samples, a CNN is trained , and the decisions are interpreted through LRP. The interpretation highlights the parts of the input on which the network relies to differentiate the two groups. We employ this information to verify the robustness of the network and detect the deferentially abundant peaks. Due to the lack of sufficient labeled datasets, we tune the architecture of the network along with training hyper-parameters on a synthetically generated data through performing systematic series of experiments and quantitatively measuring the interpretations. We evaluate the proposed model on a previously published benchmark dataset. We demonstrate the outperformance of DLearnMS against conventional biomarker detection frameworks without depending on the otherwise necessary preprocessing

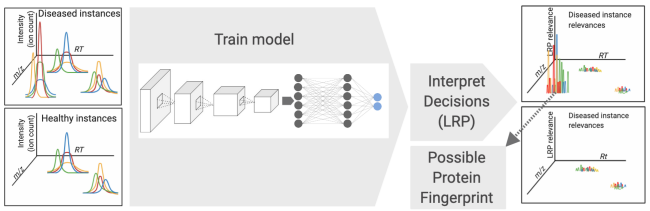

Fig. 1. The schematic of DLearnMS model for discovery of disease related biomarkers.

steps. Nevertheless, LC-MS preprocessing approaches e.g., [39], [40], could be potentially added to our DLearnMS framework that could even further improve the performance results. The stability of the biomarker detection is justified through cross-validation on synthetic data that could hinder disentangling the model errors from interpretation errors. We also discuss the shortcomings of conventional ML models for analysing raw LC-MS data classification and feature selection.

Summarizing, our contribution in this paper lies in the combination of the following triad:

(A) Develop an interpretable DL model for raw LC-MS data feature selection to cover the lack of an explainable DL approach in this field: A local post-hoc explanation method, LRP, is adapted for interpreting a CNN with high inference accuracy for LC-MS classification.

(B) Refine CNN structure to design a robust classification network according to the interpretation outcome: In addition to check the inference accuracy, the robustness is verified if the interpretations are aligned with the ground truth discriminating features.

(c) Detect biomarkers of real LC-MS proteomics data with small sample size: We tackle the insufficient labeled dataset at the class level and biomarker level by adjusting the parameters of the whole proposed pipeline on a synthetically generated data.

2 Designing the Model

Let $I_n \in \mathbb{R}^2$ for $n = 1, ..., N$ be a series of LC-MS maps, which take $O_n \in \{0, 1\}$ as the medical condition labels. Each (x, y) pair on I where $x = m/z$ and $y = RT$, contains ion-count demonstrating features on LC-MS map. The aim of biomarker detection is to find the smallest subset of $(\hat{x}, \hat{y})$ pairs whose ion-counts are differentially abundant between conditions 0 and 1. Our strategy is to design a CNN architecture, modeled as function f, to classify LC-MS samples into two classes, and learn from the prediction behavior to detect $(\hat{x}, \hat{y})$ pairs. Mathematically speaking, a CNN with L layers can be abstracted as $f(I) = f_L \circ ... \circ f_1(I)$ where each layer is a linear function followed by an element-wise non-linear activation, such as the rectified linear unit function (Relu [41]). The power of CNN prediction comes from combining many layers, which at the same time makes it complex and consequently difficult to interpret. Layer-wise relevance propagation technology [31] suggests the use of the layered structure of the neural network to interpret the predictions. The network is assumed to be fully trained and

the predictions are redistributed backward layer by layer to give a score to all the input features. A feature $(\hat{x}, \hat{y})$ will be attributed strong relevance if the function f is sensitive to the presence of that feature. The relevance value of all (x, y) pairs forms the matrix of relevances, R_i^1, known as a heatmap. The goal is to adapt this information for verifying the predicted medical condition and finding most relevant attributions associated with this condition.

2.1 Classification Model and Interpretation

The first step is to design a robust classification CNN on the LC-MS samples of two classes that we are interested in the differences. CNN is characterized by the depth and width of the layers. Depth refers to the number of layers, and width determines the number of filters. We train networks with different width and depth from standard structures like variants of ResNet [42] to customized structures. We observe that training very deep networks like ResNet32 on the LC-MS data (both synthetic and real data) leads to overfitting, while a network with few number of layers fits with high accuracy. The outperformance of the customize network over very deep networks can intuitively be explained by the local dependent characterization of the peaks on the LC-MS map. Very deep networks capture both the local -gained by reach feature representation- and global dependencies -gained by large receptive fields. Therefore, very deep networks may learn some global patterns irrelevant to the data information but relevant to the noise, such as quantification calibration error in data acquisition.

Besides, we observe that by changing a few layers on the architecture of the customized network the training and testing accuracy and loss remain steady. One may select a network with fewer learnable parameters to decrease the computational cost. Whereas, one my select a network with more learnable parameter to increase the capacity in order to achieve a better generalization accuracy. Our strategy to select a proper network is however to leverage CNN interpretation. We quantitatively compare the interpretations of the trained network with different architecture, and select the one whose predictions are the most sensitive to the actual differences between the two groups. DLearnMS employ LRP method using Eq. (1) as the interpretation analysis. Applying LRP on the network's prediction of given input I_n highlights the important parts of I_n through redistributing the neuron score backwards and assigns a relevance to each element of the input. Eq. (1) shows a rule for redistributing the relevances known as LRP.ϵ.

$$R_{i\leftarrow j}^{(l,l+1)} = \begin{cases} \frac{z_{ij}}{z_j+\epsilon}.R_j^{(l+1)}, & \text{if } z_j \geq 0 \\ \frac{z_{ij}}{z_j-\epsilon}.R_j^{(l+1)}, & \text{otherwise} \end{cases} \quad (1)$$

where $z_{ij} = O_i w_{ij}$, $z_j = \sum_i z_{ij} + b_j$, and $O_j = g(z_j)$. g is a non-linear activation function, and w_{ij} defines the weight that connect the neuron j in layer l to the neuron i in layer $l+1$. Other redistribution rules to control the flow of positive and negative relevances include LRP.$\alpha\beta$ and LRP.z [31]. All rules at each step must hold such that $\sum R_{i\leftarrow j}^{(l,l+1)} = R_j^{(l+1)}$, which means all relevance values that flow into a neuron at layer $l+1$ flow out towards the neurons of the layer l. All Relevances, R_i^l, are calculated for $l = 1, ..., \text{num\_layers}$ progressively from last layer, layer after layer, until the input layer is reached and yield R_i^1. Please see [31] for more details. R_i^1 for $i = (x, y)$ demonstrates how much pixel (x, y) - representing m/z and RT - contributes to the decision making. We choose a network whose R_i^1 highlight the differences between the classes the most. The detail explanation on detecting the network architecture tuning is delayed to Section 3.The verified network is then used for feature selection.

2.2 Feature Selection

DLearnMS select the location of important features based on their relevance intensities, R_i^1. Considering offsets, the presence of noise, and different peak indices on the samples, we are interested in interpreting the decision on statistics of the whole training-set. We take the mean of LC-MS samples belonged to the diseased class, D, and healthy class, H, separately. Each mean is given to the trained network, f, and the predictions are interpreted by LRP function. This results in two matrices of diseased relevance values, R_d^1, and healthy group's relevance values, R_h^1.

$$R_d^1 = \text{LRP}(f(\frac{1}{N_d}\sum_{n\in D} I_n)),\ R_h^1 = \text{LRP}(f(\frac{1}{N_h}\sum_{n\in H} I_n))$$

where N_d and N_h are the number of samples in diseased and healthy classes, respectively. The spatial location of peaks on LC-MS map are widely distributed, and the exact location of peaks can be estimated by finding the index with maximum intensity in a predefined window. To this end, DLearnMS first select the peak with strongest relevances on R_d^1. Then, the neighbor's relevances in the window are set to zero. We iterate this process until all the high-intensity relevances are covered. The selected peaks are distinguished as biomarkers if corresponding indices on R_h^1 are attributed non-negative relevances. We will discuss the effect of incorporating R_h^1 along with R_d^1 in Section 3.3.

To extract the biomarker from an unknown sample, the sample is fed to the network to be classified. The peaks are selected locally from LRP interpretation similar to selecting the peaks from training samples. These peaks are distinguished as biomarkers if corresponding indices on R_h^1 are existed and attributed non-negative on R_h^1.

3 Optimizing the Model Parameters on Synthetic Data

In this section, we elucidate how we verify the classification network robustness according to the reliance of the prediction on true discriminating region of the data. We study the influence of varying one or two fully connected layers (FCL), convolutional layers (CL), and max-pooling layers (MPL). Although variation of these settings results in slight differences on the classification performances, this study highlights major improvement in their interpretations. Due to the shortage of annotated real datasets at the biomarker level, the proposed model is developed and tuned on a synthetically generated dataset.

3.1 LC-MS Data Simulation

LC-MS consists of two levels of separations. First, a protein solute (mobile phase) passes through a chromatography column (stationary phase), which effectively separates the components based on the chemical affinity and weight. RT measures the time taken from the injection of the solvent to the detection of the components. Second, each component is ionized and scanned through a mass spectrometer that generates a mass spectrum (MS). Each MS scan measures m/z values of charged particles and peak intensities. Stacking all MS scans on top of each other forms a three-dimensional data whose x, y, and z axes are m/z values, RT, and ion-count intensities, respectively.

To generate the synthetic LC-MS dataset, two groups of samples representing healthy and diseased classes are simulated using UniPort human proteome dataset [43]. The healthy class contains 20 peptides. Two peptides that are independent from the peptides in the healthy samples are added to the peptides in healthy group to form the diseased group. As a results, there are 20 and 22 peptides in healthy and diseased group. The two extra peptides in diseased group define the biomarkers (discriminating features) that we intend to detect on LC-MS map. Investigating such differences is the basis of diagnosis of different biological conditions and disease treatment, e.g., measuring the concentration level of cardiac troponin that enters in the blood soon after a heart attack, or measuring thyroglobulin, a protein made by cells in the thyroid, which is used as a tumor marker test to help guide thyroid cancer treatment.

We use OpenMs [44] and TOPPAS [45] to generate LC-MS samples and convert them into images. The width, height, and pixel intensities of images present m/z, RT, and ion-count intensity, respectively. It should be noted that the images still represent the raw data. The only difference between the matrix of raw data and the converted images is that the ion-count intensity range in raw data is scaled to [0,255]. The dataset contains 4000 samples of each group. 10% of each group is left out for testing, and the rest is used for training and validation.

3.2 Feature Selection Metrics

Here, we introduce selected metrics to evaluate the capability of interpretation heatmap, R_i^1, on reflecting the biomarkers. The metrics should be representative of the percentage of true-positive (TP) and false-positive (FP) peaks. Therefore, we consider intersection over union (IOU), precision, and recall metrics defined as follows:

$$\begin{aligned} \text{IOU} &= \frac{\text{relevant peaks} \cap \text{selected peaks}}{\text{relevant peaks} \cup \text{selected peaks}} \\ \text{Precision} &= \frac{\text{relevant peaks} \cap \text{selected peaks}}{\text{selected peaks}} \\ \text{Recall} &= \frac{\text{relevant peaks} \cap \text{selected peaks}}{\text{relevant peaks}} \end{aligned} \tag{2}$$

where the relevant peaks and selected peaks are ground-truth and predicted peptides peaks. To extract the ground truth on synthetic data, the mean of the images in the diseased group is subtracted from the mean of images in the healthy group and the absolute value of the resulting is taken. The result contains all biomarker peaks and is referred to as ground-truth image (GTI). This is identical to alternatively simulate several replicants of the extra peptide of diseased samples (using OpenMs, and TOPPAS) and take the mean of the replications. We apply a threshold, γ_{gt}, on the GTI to ignore small perturbation generated by LC-MS quantification error. As previously described in Section 2.2 since the spatial location of peaks is distributed widely, we restrict our attention to the peaks with the highest intensities and set to zero a box window with a size of $[w, h]$. To this end, first the index of the highest intensity value on GTI is selected. Second, the surrounding peaks in the window of w and h are set to zero. Next, we iterate this process until all the high-intensity regions are covered. We refer to the resulting as ground truth peak map (GTPM). The selected peaks in Eq. (2) are extracted similar to GTPM from the LRP relevances and form prediction peak map (PPM). The metrics of Eq. (2) can be rewritten as follows:

$$\begin{aligned} \text{IOU} &= 2\left(\textstyle\sum_{(x,y)\in\mathcal{I}} \text{GTPM}(x,y).\text{PPM}(x,y)\right) / \textstyle\sum_{(x,y)\in\mathcal{I}} (\text{GTPM}(x,y)+\text{PPM}(x,y)) \\ \text{Precision} &= \textstyle\sum_{(x,y)\in\mathcal{I}} \text{GTPM}(x,y).\text{PPM}(x,y) / \textstyle\sum_{(x,y)\in\mathcal{I}} \text{PPM}(x,y) \\ \text{Recall} &= \textstyle\sum_{(x,y)\in\mathcal{I}} \text{GTPM}(x,y).\text{PPM}(x,y) / \textstyle\sum_{(x,y)\in\mathcal{I}} \text{GTPM}(x,y) \ , \end{aligned}$$

where $\mathcal{I}$ covers the entire range of (m/z,RT) values.

TABLE 1
Tune the number of fully connected layer (FCL), convolutional layers (CL), max-pooling layers (MPL), and the effect of adding healthy class interpretation information to the biomarker detection analysis. The parameters are tuned according to the intersection over union (IOU), precision, and recall. The effect of incorporating the interpretation of diseased samples' mean (R_d) and the interpretation of healthy samples' mean (R_h) on peak detection is also measured.

# CL	# MPL	#FCL	Samples	IOU	Precision	Recall
6	4	2	R_d	0.3975	0.3814	0.4149
6	4	1	R_d	0.5006	0.4513	0.5621
6	4	1	R_d, R_h	0.6177	0.6188	0.616
4	3	1	R_d	0.6599	0.5985	0.7353
4	3	1	R_d, R_h	0.7008	0.6756	0.7281
4	1	1	R_d	0.7165	0.6171	0.8441
4	1	1	R_d, R_h	**0.8501**	**0.8554**	**0.8448**

3.3 Parameter Tuning

Up to this point in this section, we have introduced the synthetic dataset, and the metrics for network verification and feature selection on this dataset. We explained optimizing the network's architecture (e.g., depth, width, and kernel size) and hyperparameters (e.g., type of optimizer, learning rate, and batch size) with respect to classification loss and accuracy in Section 2.1. We will now discuss how we tune parameters of the network including number of FCL, CL, and MPL. These parameters have not changed the classification accuracy in the variations presented in Table 1; however, significantly had an impact on the focus of the network on the discriminating features for making decisions. This effect is measured through feature selection metrics. The networks that are build by varying aforementioned parameters are trained, interpreted and compared to select a set of parameters that leads to the best IOU, Pr, and Re as feature selection metrics.

We report the effect of the number of FCL, CL and MPL in Table 1. According to recent research in DL field exploiting deeper networks (more CL) are recommended as

they offer richer representation. But it is also important to gain understanding of model's behaviour to check whether the network make decisions according to the regions that we have expected the network to learn. In this study, this regions are the differentially abundant peaks. We measure feature selection metrics by varying the number of FCL, CL and MPL. As a result, among the networks with the same accuracy performance, the one with four CL and one MPL reach the best feature selection performance. This indicate the reliance of the network reflects on diferentially abundant peaks to make classification decisions.

After tuning and designing the structure, we now explain the effect of incorporating the interpretation of healthy predictions along with the interpretation of diseased predictions on feature selection performance. In Section 3.2, we have described in detail how prediction peak map (PPM) is calculated through LRP relevance values. As a recap, To estimate relevance values on the training set, we calculate the mean of the diseased samples, run the trained network on the mean, and calculate the relevances. By convention, positive relevance values are the evidence of existing relevant peaks belong to the respected class. Therefore, in our study, positive relevance values on the interpretation of diseased class are associated with the biomarkers. The feature selection results of utilizing diseased class relevances are presented in Table 1, in which the interpretation column is assigned with R_d. We observe that FP peaks can be reduced by incorporating the interpretation of healthy samples along with the diseased samples. The positive relevances of the interpretation of the healthy group can be explained as the absence of diseased relevant peak, or presence of healthy relevant peaks. Because all the peaks in healthy samples are presented in diseased samples, the positive relevances of this group is just explained as the absence of diseased relevant peak. Accordingly, the indices of high-ranked relevances of the diseased group are selected as biomarkers if the corresponding indices of the interpretation of the healthy group attribute non-negative relevances. The feature selection results are shown in Table 1, in which the interpretation column is assigned with R_d, R_h. As it is apparent, IOU and Pr that are both directly affected by FP in the denominator, considerably improved.

As a result of parameter tuning, the feature selection performance has been improved from 40% to 85% shown in Table 1. Hence, our verified DL network architecture has four CL, one MPL after the second CL, and one FCL on top of the network as the prediction layer. We use the interpretation of this network for biomarker detection as it has been described in Section 2.2.

3.4 Measure Feature Selection Stability using Cross-Validation

Here in this section, we aim at measuring the stability of the detected features which is equivalent to measure the sensitivity of LRP interpretation of similar instances. To this end, the overlaps of selected features using the model trained using cross-validation are measured. The stability of explanation is measured on the synthetic data so that we can avoid problems of disentangling errors made by the model from errors made by the explanation. We allocate 10% of samples for the test set and use five-fold cross-validation on the rest of the samples. The network is trained on training samples and evaluated on the validation set as well as the test set. Hence, the feature selection is evaluated five times on the test set, and once on every validation fold. The intersection of selected features from the test set is shown 99% overlapped features on average. On the validation set, the overlap of selected features between every two folds is calculated and averaged, which result in 98% overlapped features. These results not only justify the stability of the interpretations and the feature selection approach but also imply the robustness of the classification network whose interpretation leads to the features.

4 Results on Real Dataset

In this section, the performance of the proposed method is assessed on a published benchmark LC-MS dataset [4] which we refer to as real dataset. Many other Mass spectrometry datasets are available at repositories such as PRIDE or CompMS. However, the focus of this paper is to assess the feature selection on a raw LC-MS map of samples from two conditions (healthy and control) with known biomarkers presented by their m/z and RT, which is perfectly met in the selected dataset. All the parameters and hyperparameters of the model including the classification, interpretation, and feature selection parts are maintained as they were tuned on the synthetic dataset.

4.1 Real-Data Description

The real LC-MS dataset, consists of two groups. The first group was derived from five serum samples of healthy individuals that have been spiked with a known concentration of spike-in peptides. The second group was obtained from the serum samples only. We refer to the first and second groups as diseased and healthy, respectively. The added peptides to the diseased group are the selection of nine peptides with different concentrations to be representative of real datasets. They have predictable retention behavior and elution order that let the ground truth available in m/z and RT [4]. LC-MS acquisition yields 13 peaks from nine peptides due to the different charges. The specifications of these peaks are presented in Table 2. Please see [4] for more description and visualization of the peaks. The proposed method is intended to detect differentially abundant spike-peaks as biomarkers and to keep detected FP peaks low. The evaluation will be reported as the exact number of TP and FP peaks. We quantize the raw data and form chromatograms matrices. This outcome is then converted into images whose width and height are m/z and RT, respectively. Each RT bin on the y-axis presents seven seconds of the MS level-1 scan, and x-axis covers ions of m/z 350 to m/z 2000. Pixel intensities are demonstrating the ion-counts. LC was run for 240 minutes, however, similar to the benchmark methods, we filter the samples to retain features within 150 minutes because there is no significant peak out of this range. We remove the features with the ion-count intensities less than two as the only noise reduction on the samples.

4.2 Results

Table 3 compares the feature selection of our proposed method on the described real dataset with the benchmark

TABLE 2
Specification of the real data spike-in peptides. Base peak chromatograms of the group with spike-in peptides are presented based on their mass-to-charge ration (m/z), retention time (RT), and ion charge.

Features No.	1	2	3a	3b	4	5a	5b	6	7a	7b	8	9a	9b
m/z	501.25	450.23	530.78	354.19	523.77	648.84	432.89	586.98	624.99	630.35	943.43	712.43	570.15
Charge	2	2	2	3	2	2	3	3	3	3	3	4	5
RT(min) start-end	4-8	45-49	53-56	53-56	59-62	63-67	63-67	73-77	77-81	82-86	79-83	103- 107	103-107

TABLE 3
Feature selection comparison of the proposed method with MZmine 2 [6], Progenesis LC-MS [8], and XCMS [7], which all presented in [4]. The total number of selected features is represented for all methods in the first row. Only features presented in at least two replicates in each group were used for statistical analysis for the baseline methods. The third and forth rows are demonstrating the number of features satisfying two representative criteria including t-test with multiple hypothesis testing (q-value< 0.05), and fold change (FC > 10). The plus sign denotes the combination of different criteria. The numbers written in parentheses indicate the selected biomarker peaks. The effect of incorporating the interpretation of diseased samples (R_d^1) and the interpretation of healthy samples (R_h^1) on peak detection are shown in the two last columns.

	msInspect	MZmine 2	Progenesis	XCMS	DLearnMS: R_d^1	DLearnMS: R_d^1,R_h^1
# All selected features	31168 (12)	12271 (12)	9267 (9)	21486 (13)	8044 (12)	6992(11)
# Features for statistical analysis	6525 (9)	12092 (9)	8415 (9)	8703 (10)	8044 (12)	6992(11)
t-test ($q < 0.05$)	4824 (9)	3505 (7)	4465 (9)	1896 (7)	3985 (11)	3499(11)
t-test ($q < 0.05$) + FC (> 10)	2099 (9)	539 (7)	467 (8)	66 (7)	222 (9)	195(9)

TABLE 4
Real data biomarker detection comparison according to the statistical analysis. Detected differential abundant spike-in peaks are shown by check marks. Note that, our method detects all the features that are commonly selected by all other methods.

Features No.	1	2	3a	3b	4	5a	5b	6	7a	7b	8	9a	9b
msInspect	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	-	-	-
MZmine 2	✓	✓	✓	✓	✓	✓	✓	-	-	-	-	-	-
Progenesis	✓	✓	✓	✓	✓	-	✓	-	-	✓	-	-	✓
XCMS.	✓	✓	✓	✓	✓	-	✓	-	-	✓	-	-	
DLearnMS	✓	✓	✓	✓	✓	-	✓	-	-	✓	-	✓	✓

methods including: msInspect, MZmine 2, Progenesis, and XCMS [4]. The first row in Table 3 demonstrates that our method outperform the other methods in terms of detecting fewer FP peaks. Our analysis does not require the preprocessing steps used in other workflows. We follow the same statistical analysis on the selected peaks, similar to [4]. The t-test for $p < 0.05$ is calculated on each selected feature, and multiple testing correction is applied. The features that satisfy $q < 0.05$ are selected as the discriminating features presented on the third row of Table 3. The fourth row shows the number of selected features satisfied $q < 0.05$ and fold change (FC) > 10. We detect nine biomarker peaks similar to msInspect, while we achieve almost 10 times fewer FP peaks, 195 in comparison with 2099 FP peaks in msInspect. We also outperform MZmine 2 and Prognesis with respect to both evaluation metrics, namely the number of biomarker peaks (seven in MZmine 2 and eight in Prognesis) and FP peaks (539 in MZmine 2 and 467 in Prognesis). Although XCMS achieves the best results with respect to the number of FP peaks, 66, which is the smallest number of FP peaks, its performance concerning the number of detected biomarker peaks, however, has dramatically dropped to seven. The last two columns of the Table 3 demonstrate incorporating healthy samples interpretation, R_h^1, along with the diseased interpretation , R_d^1. The performances show that the number of FP peaks is degraded, although it is not as pronounced as the performance on the synthetic data.

The biomarker peaks that are selected according to the statistical analysis are presented in Table 4. Six peaks that are commonly selected by all four other methods as differentially abundant [4] peaks have also been detected by our method.

5 Conventional Machine Learning Models for LC-MS Proteomics Analysis

In this section, we discuss the challenges that hinder classical ML methods for LC-MS data analysis. Table 5 shows the classification comparison of ML methods including, support vector machine (SVM) with linear kernel, decision tree (DT), and Adaboost with our CNN model. The parameters of the selected methods are tuned using grid search in scikit-learn on synthetic data. We use five-fold and leave-on-out cross-validation for training on the synthetic and real datasets, respectively. As it is apparent from Table 5 there is a huge gap in the classification performance of ML methods between the synthetic data and the real data. One way to investigate the reason is to interpret the results.

There are model agnostic methods that enable estimating the importance of features for decision making by any trained model regardless of the model's complexity, e.g., permutation feature importance, measured by randomly shuffling the feature and tracking the drop in the model's score. LIME [46] is another model agnostic interpretation, which locally interprets any model around a single prediction. Given a trained model, LIME perturb each instance locally, calculates the distance of the perturbed instance from the original sample according to the trained model, and generate a new dataset. A linear model is then fit on the new dataset. The linear model coefficients determine which features are more dominant. These methods, however, are computationally infeasible for analyzing high-dimensional LC-MS data. On

TABLE 5
Classification comparison of the convolutional neural network (CNN) with conventional machine learning methods including, decision tree (DT), support vector machine (SVM), and adaboost. CNN shows significantly better classification performance on the real datasets. The interpretation is not available for weak classifiers. On the synthetic dataset ML methods are as accurate as CNN. However, SVM interpretation demonstrates the overfitting effect. Interpretation on the synthetic data is reported by intersection over union (IOU) between the selected and true peaks. Interpretation on the real data is reported by the amount of true positive peaks from 13 spike-in peaks. '-' shows no interpretation is available for the models.

Synthetic dataset	Accuracy	Sensitivity	Specificity	Interpretation (IOU)
SVM	0.98	0.99	0.98	feature importance(< 0.1)
DT	1.0	1.0	1.0	-
Adaboost	0.99	1.0	0.99	-
CNN	1.0	1.0	1.0	LRP (0.85)
Real dataset	Accuracy	Sensitivity	Specificity	Interpretation (TP/13)
SVM	<0.5	<0.5	<0.5	-
DT	<0.5	<0.5	<0.5	-
Adaboost	<0.5	<0.5	<0.5	-
CNN	0.8	0.8	0.8	12/13

the other hand, inherently interpretable models are not capable of correctly classifying complex LC-MS data. For example, linear models in which the weights of the variables serve as the explanation or shallow decision trees in which the normalized total reduction of the Gini index by every feature yields the explanation.

In Table 5, despite Adaboost that is not inherently interpretable and Decision tree (DT) that is not shallow enough to be interpreted, linear SVM still can be explained by the weights assigned to the features. According to this table, SVM reaches comparable classification performance as the CNN. However, the explanation results in a very poor IOU - less than 10% - between the important features selected by coefficient of SVM model and actual differences. This effect - the high accuracy and weak explanation- resulted by SVM can be explained by low fidelity of the model's interpretation or overfitting of the model caused by some biases or pattern (comes with the simulation), unrelated to actual differences. But, the overfitting effect is more likely since SVM with the same parameter setting trained on the synthetic data results in a very poor classification on the real data. The overfitting effect can also be explained by the Adaboost and DT classification gap between the real and synthetic data as well.

6 Implementation Setup

The experiments in this study are implemented in Python for data analysis, Scikit-learn library [47] for ML analyzes, Keras [48] with Tensorflow backend [49] for DL analysis, and "iNNvestigate" library [50] for DL interpretation analysis on a machine with a 3.50 GHz Intel Xeon(R) E5-1650 v3 CPU and a GTX 1080 graphics card with 8 GiB GPU memory. The classification network is trained for 20 epochs and batch size of two using Adam optimizer [51] with the learning rate of 0.00001, and momentum of 0.9. We use binary cross-entropy as the loss function. The kernel size in all layers is set to 3×3 with the dropout rate of 0.3. The convolution layers in the network are two dimension and contain the following number of kernels: 32 in the first and second layers, 64 in the third layer, and two in the fourth layer. The fully connected layer as the last layer has two neurons for binary classification[1].

1. The datasets and implementation are available upon request from the first author.

7 Discussion

Identifying a set of biomarkers (proteins in this study) from LC-MS data is a standard task in the context of precision medicine. Performing this task on raw data is challenging due to the high dimensionality, complexity, and high noise level. Despite available tools, current workflows require several preprocessing steps to address LC-MS biomarker detection. Besides, the application of DL interpretation is neglected in this area despite the importance of interpretable explanation in biomedical settings. In this study, we introduce a DL method backed by LRP interpretation to address these issues. We design and train a CNN network on the LC-MS map of the healthy and diseased samples in a way that enables extracting biomarker peaks from the interpretation of network decision.

The first challenge with any supervised DL method is that it requires a large labeled dataset for parameter tuning; otherwise, it overfits quickly, particularly on the high dimensional and sparse LC-MS dataset. Due to the insufficient real labeled LC-MS dataset for training, our model was tuned and optimized on a large synthetically generated dataset. Besides, we verified the model robustness by measuring the dependency of the network's decision on true features. The second challenge is that the interpretation of a DL model is not always informative when it comes to very small discriminating peaks in the sparse LC-MS dataset. Therefore, we run systematic experiments using feature selection metrics to quantitatively measure the network's interpretation.

According to the results in Section 2.2, the interpretations of different networks that share similar classification performance - with almost 99% training and testing accuracy - considerably differ. These differences consequently affect biomarker detection. We selected an architecture that fits the best according to the feature selection metrics. Then, we built the biomarker detection on the interpretation of the selected network. Building on this observation, we suggest considering this property of the DL approach not only to detect biomarkers but also to design robust DL networks where measuring the interpretation is attainable. This concept is especially important in the medical application where human health is involved.

We assessed the biomarker detection of the proposed tuned model on a real dataset with predictable spike-in

peptides. We showed our model achieved overall better performance results in comparison with the conventional methods [4], [6]–[8] in terms of detecting fewer FP peaks. Besides, our approach is end-to-end and does not require otherwise necessary preprocessing steps.

Training the DL model on small datasets is not often recommended due to underfitting, overfitting effects, and lack of sufficient evidence (labeled data) to show the model robustness. We showed that a properly designed network can still be reliable through its validation using a proper DL interpretation.

On the synthetic data, we showed that exploiting the interpretation of *both classes* can considerably improve the FP in comparison with the setting when only the diseased class were considered. This observation stressed the importance of understanding the implications that are provided by interpretation analyzes. Leveraging this valuable information can foster more plausible network architectures resulting in a more meaningful conclusion. Recent advances in the image processing field confirm this important fact [26], [31], [52].

The improvement in the FP rate on the real dataset was not as pronounced as the synthetic dataset. This behavior can be statistically explained by the number of samples in the synthetic dataset ($\sim$ 8000) that outnumber the real dataset ($\sim$ 10). We calculated the interpretation analysis on the mean of the samples' intensities. Therefore, the mean intensities on the large set of data is a better representative of whole data distribution than a small set. Consequently, the importance of features belonging to the larger dataset, which are assigned by the network's decision, would be more precise.

According to Section 5, conventional ML models are failed to correctly fit on LC-MS real dataset. Despite high accuracy on the synthetic data, the poor interpretation of linear SVM on synthetic data and the huge gap between classification performance of real and synthetic data demonstrate the overfitting effect.

This study was assessed on the dataset whose biomarkers have been spiked before LC-MS acquisition. To further our research, we plan to apply our proposed method to real diseased cases. This study can be extended to the multi-subject localization of biomarkers. In this case, the interpretation of a robust multi-class classification network on the LC-MS map of samples would highlight the dominant differences of each class from the others. These differences are the potential position of biomarkers. We also consider adapting different LRP rules to different layers of the network due to their confirmed success in machine vision applications [26].

8 Conclusion

We present DLearnMS an interpretable deep learning approach for LC-MS biomarker detection. DLearnMS is built on a generalized convolutional neural network backed by LRP interpretation method. We demonstrate the leverage of the quantification of deep learning interpretation for designing a robust classification network. Towards this end, the lack of labeled LC-MS data is addressed by utilizing synthetically generated data for optimizing and tuning the model. Next, we detect biomarkers on the real data through adapting LRP. DLearnMS surpasses convectional method including msInspect, MZmine 2, Progenesis, and XCMS in terms of detecting fewer false positive peaks while cutting the additional computation load by excluding commonly used preprocessing steps.

Appendix A
Peptides in Synthetic dataset

TABLE 6
The accession number associated with diseased and healthy group in the synthetic dataset.

Classes	Peptide sequences
Healthy	Q9NYW0, Q9NYV9, P59538, P59539, Q96CE8, Q96A56, O75478, Q86TJ2, Q15543, Q15573, Q9H5J8, O00268, Q9UI15, Q9H2K8, Q17R31, P10636, P68366, A6NHL2, Q13509, Q9NVG8
Diseased	Q9NYW0, Q9NYV9, P59538, P59539, Q96CE8, Q96A56, O75478, Q86TJ2, Q15543, Q15573, Q9H5J8, O00268, Q9UI15, Q9H2K8, Q17R31, P10636, P68366, A6NHL2, Q13509, Q9NVG8, Q9HA65, Q9ULP9

TABLE 7
Real data spike-in peptide sequences

Peptide No.	Peptide sequences	charge
1	RGDSPASSKP	2
2	DRVYIHP	2
3a	RPPGFSPFR	2
3b	RPPGFSPFR	3
4	DRVYIHPF	2
5a	DRVYIHPFHL	2
5b	DRVYIHPFHL	3
6	DRVYIHPFHLLVYS	3
7a	WLTGPQLADLYHSLMK	2
7b	WLTGPQLADLYHSLMK	3
8	YPIVSIEDPFAEDDWEAWSHFFK	3
9a	GIGAVLKVLTTGLPALISWIKRKRQQ	4
9b	GIGAVLKVLTTGLPALISWIKRKRQQ	5

Appendix B
Visualize Convergence Distribution

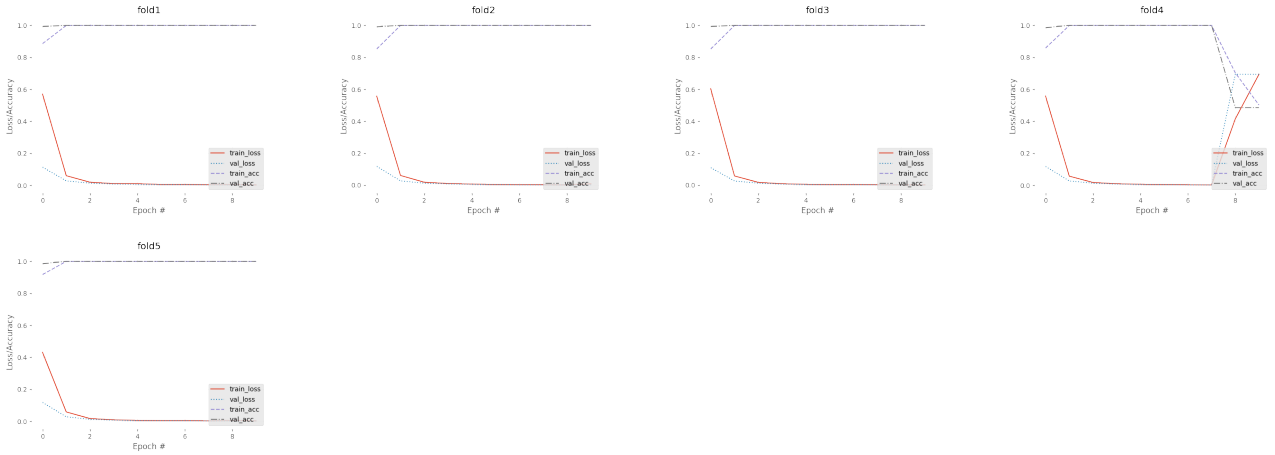

Fig. 2. Training on the simulated data. Training and validation classification accuracy are shown in purple dash line and back dash-doted line, respectively. Training and validation losses are also shown along with the accuracies in red line and blue dotted line. In this plot, we demonstrate the five fold cross-validation training curves for 10 epochs. However, for the classification comparison and continue with interpretation and feature selection the early stopping has been considered. Therefore, training is stopped after five epochs which avoided the divergence on the forth fold.

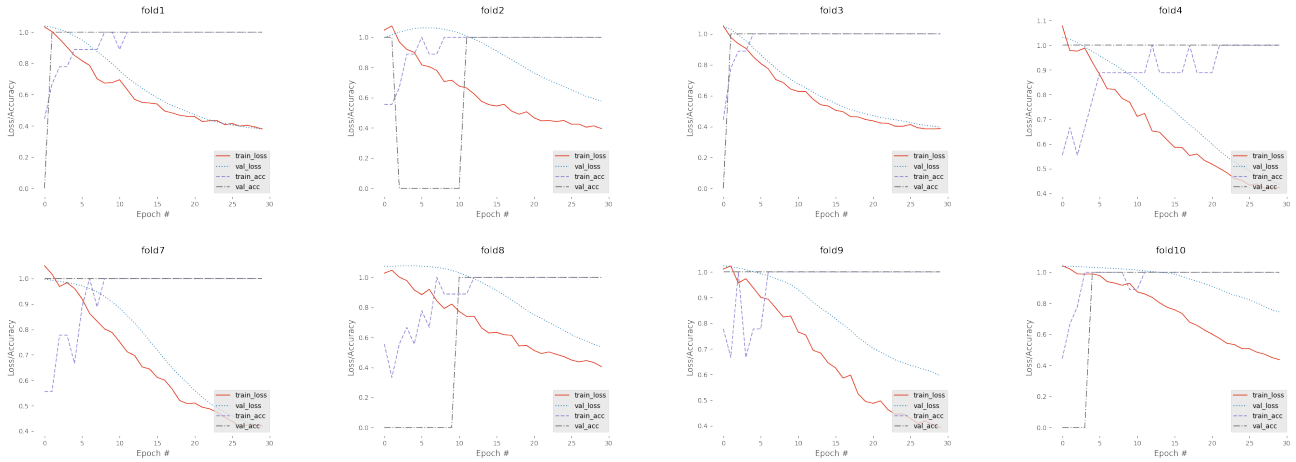

Fig. 3. Training on the real data. Training and validation classification accuracies are shown in dash line and back dash-doted line, respectively. Training and validation losses are also shown along with the accuracies in red line and blue dotted line. The trends are less smooth than simulated data, because of smaller amount of data points in the real dataset than simulated dataset.

Acknowledgment

This work was supported by the German Ministry for Education and Research (BMBF) as Berlin Big Data Center (01IS14013A) and the Berlin Center for Machine Learning (01IS18037I) and within the Forschungscampus MODAL (project grant 3FO18501).

References

[1] H. Wang, T. Shi, W.-J. Qian, T. Liu, J. Kagan, S. Srivastava, R. D. Smith, K. D. Rodland, and D. G. Camp, "The clinical impact of recent advances in LC-MS for cancer biomarker discovery and verification," *Expert review of proteomics*, vol. 13, no. 1, pp. 99–114, 2016.

[2] F. Hoffmann, C. Umbreit, T. Krüger, D. Pelzel, G. Ernst, O. Kniemeyer, O. Guntinas-Lichius, A. Berndt, and F. von Eggeling, "Identification of proteomic markers in head and neck cancer using maldi–ms imaging, lc–ms/ms, and immunohistochemistry," *PROTEOMICS–Clinical Applications*, vol. 13, no. 1, p. 1700173, 2019.

[3] G. H. M. F. Souza, P. C. Guest, and D. Martins-de Souza, "LC-MS e, multiplex ms/ms, ion mobility, and label-free quantitation in clinical proteomics," in *Multiplex Biomarker Techniques*. Springer, 2017, pp. 57–73.

[4] L. Tuli, T.-H. Tsai, R. S. Varghese, J. F. Xiao, A. Cheema, and H. W. Ressom, "Using a spike-in experiment to evaluate analysis of LC-MS data," *Proteome science*, vol. 10, no. 1, p. 13, 2012.

[5] M. Bellew, M. Coram, M. Fitzgibbon, M. Igra, T. Randolph, P. Wang, D. May, J. Eng, R. Fang, C. Lin *et al.*, "A suite of algorithms for the comprehensive analysis of complex protein mixtures using high-resolution LC-MS," *Bioinformatics*, vol. 22, no. 15, pp. 1902–1909, 2006.

[6] T. Pluskal, S. Castillo, A. Villar-Briones, and M. Orešič, "MZmine 2: modular framework for processing, visualizing, and analyzing mass spectrometry-based molecular profile data," *BMC bioinformatics*, vol. 11, no. 1, p. 395, 2010.

[7] C. A. Smith, E. J. Want, G. O'Maille, R. Abagyan, and G. Siuzdak, "Xcms: processing mass spectrometry data for metabolite profiling using nonlinear peak alignment, matching, and identification," *Analytical chemistry*, vol. 78, no. 3, pp. 779–787, 2006.

[8] D. Qi, P. Brownridge, D. Xia, K. Mackay, F. F. Gonzalez-Galarza, J. Kenyani, V. Harman, R. J. Beynon, and A. R. Jones, "A software toolkit and interface for performing stable isotope labeling and top3 quantification using progenesis LC-MS," *Omics: a journal of integrative biology*, vol. 16, no. 9, pp. 489–495, 2012.

[9] K. Aoshima, K. Takahashi, M. Ikawa, T. Kimura, M. Fukuda, S. Tanaka, H. E. Parry, Y. Fujita, A. C. Yoshizawa, S. Utsunomiya *et al.*, "A simple peak detection and label-free quantitation algorithm for chromatography-mass spectrometry," *BMC bioinformatics*,

vol. 15, no. 1, p. 376, 2014.

[10] P. M. Palagi, D. Walther, M. Quadroni, S. Catherinet, J. Burgess, C. G. Zimmermann-Ivol, J.-C. Sanchez, P.-A. Binz, D. F. Hochstrasser, and R. D. Appel, "Msight: An image analysis software for liquid chromatography-mass spectrometry," *Proteomics*, vol. 5, no. 9, pp. 2381–2384, 2005.

[11] J. Cox and M. Mann, "Maxquant enables high peptide identification rates, individualized ppb-range mass accuracies and proteome-wide protein quantification," *Nature biotechnology*, vol. 26, no. 12, pp. 1367–1372, 2008.

[12] J. Listgarten, R. M. Neal, S. T. Roweis, P. Wong, and A. Emili, "Difference detection in LC-MS data for protein biomarker discovery," *Bioinformatics*, vol. 23, no. 2, pp. e198–e204, 2007.

[13] K. Podwojski, A. Fritsch, D. C. Chamrad, W. Paul, B. Sitek, K. Stühler, P. Mutzel, C. Stephan, H. E. Meyer, W. Urfer *et al.*, "Retention time alignment algorithms for LC/MS data must consider non-linear shifts," *Bioinformatics*, vol. 25, no. 6, pp. 758–764, 2009.

[14] S. Gupta, S. Ahadi, W. Zhou, and H. Röst, "Dialignr provides precise retention time alignment across distant runs in dia and targeted proteomics," *Molecular and Cellular Proteomics*, vol. 18, no. 4, pp. 806–817, 2019.

[15] T. Välikangas, T. Suomi, and L. L. Elo, "A systematic evaluation of normalization methods in quantitative label-free proteomics," *Briefings in bioinformatics*, vol. 19, no. 1, pp. 1–11, 2018.

[16] C. Schiffman, L. Petrick, K. Perttula, Y. Yano, H. Carlsson, T. Whitehead, C. Metayer, J. Hayes, S. Rappaport, and S. Dudoit, "Filtering procedures for untargeted LC-MS metabolomics data," *BMC bioinformatics*, vol. 20, no. 1, p. 334, 2019.

[17] A. S. Lundervold and A. Lundervold, "An overview of deep learning in medical imaging focusing on mri," *Zeitschrift für Medizinische Physik*, vol. 29, no. 2, pp. 102–127, 2019.

[18] P. Mamoshina, A. Vieira, E. Putin, and A. Zhavoronkov, "Applications of deep learning in biomedicine," *Molecular pharmaceutics*, vol. 13, no. 5, pp. 1445–1454, 2016.

[19] R. Miotto, F. Wang, S. Wang, X. Jiang, and J. T. Dudley, "Deep learning for healthcare: review, opportunities and challenges," *Briefings in bioinformatics*, vol. 19, no. 6, pp. 1236–1246, 2018.

[20] F. T. Zohora, M. Z. Rahman, N. H. Tran, L. Xin, B. Shan, and M. Li, "DeepIso: A deep learning model for peptide feature detection from LC-MS map," *Scientific reports*, vol. 9, no. 1, pp. 1–13, 2019.

[21] N. H. Tran, X. Zhang, L. Xin, B. Shan, and M. Li, "De novo peptide sequencing by deep learning," *Proceedings of the National Academy of Sciences*, vol. 114, no. 31, pp. 8247–8252, 2017.

[22] N. H. Tran, R. Qiao, L. Xin, X. Chen, C. Liu, X. Zhang, B. Shan, A. Ghodsi, and M. Li, "Deep learning enables de novo peptide sequencing from data-independent-acquisition mass spectrometry," *Nature methods*, vol. 16, no. 1, pp. 63–66, 2019.

[23] X.-X. Zhou, W.-F. Zeng, H. Chi, C. Luo, C. Liu, J. Zhan, S.-M. He, and Z. Zhang, "pdeep: predicting ms/ms spectra of peptides with deep learning," *Analytical chemistry*, vol. 89, no. 23, pp. 12 690–12 697, 2017.

[24] C. Ma, Y. Ren, J. Yang, Z. Ren, H. Yang, and S. Liu, "Improved peptide retention time prediction in liquid chromatography through deep learning," *Analytical chemistry*, vol. 90, no. 18, pp. 10 881–10 888, 2018.

[25] S. Iravani and T. O. Conrad, "Deep learning for proteomics data for feature selection and classification," in *International Cross-Domain Conference for Machine Learning and Knowledge Extraction*. Springer, 2019, pp. 301–316.

[26] W. Samek, G. Montavon, S. Lapuschkin, C. J. Anders, and K.-R. Müller, "Toward interpretable machine learning: Transparent deep neural networks and beyond," *arXiv preprint arXiv:2003.07631*, 2020.

[27] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," *arXiv preprint arXiv:1312.6034*, 2013.

[28] D. Smilkov, N. Thorat, B. Kim, F. Viégas, and M. Wattenberg, "Smoothgrad: removing noise by adding noise," *arXiv preprint arXiv:1706.03825*, 2017.

[29] M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic attribution for deep networks," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017, pp. 3319–3328.

[30] A. Shrikumar, P. Greenside, A. Shcherbina, and A. Kundaje, "Not just a black box: Learning important features through propagating activation differences," *arXiv preprint arXiv:1605.01713*, 2016.

[31] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek, "On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation," *PloS one*, vol. 10, no. 7, 2015.

[32] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *European conference on computer vision*. Springer, 2014, pp. 818–833.

[33] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, "Striving for simplicity: The all convolutional net," *arXiv preprint arXiv:1412.6806*, 2014.

[34] P.-J. Kindermans, K. T. Schütt, M. Alber, K.-R. Müller, D. Erhan, B. Kim, and S. Dähne, "Learning how to explain neural networks: Patternnet and patternattribution," *arXiv preprint arXiv:1705.05598*, 2017.

[35] L. M. Zintgraf, T. S. Cohen, T. Adel, and M. Welling, "Visualizing deep neural network decisions: Prediction difference analysis," *arXiv preprint arXiv:1702.04595*, 2017.

[36] C. Agarwal, D. Schonfeld, and A. Nguyen, "Removing input features via a generative model to explain their attributions to classifier's decisions," *arXiv preprint arXiv:1910.04256*, 2019.

[37] R. C. Fong and A. Vedaldi, "Interpretable explanations of black boxes by meaningful perturbation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 3429–3437.

[38] Y. Date and J. Kikuchi, "Application of a deep neural network to metabolomics studies and its performance in determining important variables," *Analytical chemistry*, vol. 90, no. 3, pp. 1805–1810, 2018.

[39] Q. Liu, D. Walker, K. Uppal, Z. Liu, C. Ma, V. Tran, S. Li, D. P. Jones, and T. Yu, "Addressing the batch effect issue for lc/ms metabolomics data in data preprocessing," *Scientific reports*, vol. 10, no. 1, pp. 1–13, 2020.

[40] E. D. Kantz, S. Tiwari, J. D. Watrous, S. Cheng, and M. Jain, "Deep neural networks for classification of lc-ms spectral peaks," *Analytical chemistry*, vol. 91, no. 19, pp. 12 407–12 413, 2019.

[41] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *ICML*, 2010.

[42] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[43] U. Consortium, "Uniprot: a worldwide hub of protein knowledge," *Nucleic acids research*, vol. 47, no. D1, pp. D506–D515, 2019.

[44] H. L. Röst, T. Sachsenberg, S. Aiche, C. Bielow, H. Weisser, F. Aicheler, S. Andreotti, H.-C. Ehrlich, P. Gutenbrunner, E. Kenar *et al.*, "OpenMS: a flexible open-source software platform for mass spectrometry data analysis," *Nature methods*, vol. 13, no. 9, pp. 741–748, 2016.

[45] O. Kohlbacher, K. Reinert, C. Gröpl, E. Lange, N. Pfeifer, O. Schulz-Trieglaff, and M. Sturm, "TOPP—the OpenMS proteomics pipeline," *Bioinformatics*, vol. 23, no. 2, pp. e191–e197, 2007.

[46] M. T. Ribeiro, S. Singh, and C. Guestrin, ""why should i trust you?" explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 2016, pp. 1135–1144.

[47] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[48] F. Chollet *et al.*, "Keras," 2015.

[49] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, "Tensorflow: A system for large-scale machine learning," in *12th Symposium on Operating Systems Design and Implementation*, 2016, pp. 265–283.

[50] M. Alber, S. Lapuschkin, P. Seegerer, M. Hägele, K. T. Schütt, G. Montavon, W. Samek, K.-R. Müller, S. Dähne, and P.-J. Kindermans, "innvestigate neural networks," *Journal of Machine Learning Research*, vol. 20, no. 93, pp. 1–8, 2019.

[51] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[52] M. Kohlbrenner, A. Bauer, S. Nakajima, A. Binder, W. Samek, and S. Lapuschkin, "Towards best practice in explaining neural network decisions with lrp," in *2020 International Joint Conference on Neural Networks (IJCNN)*, 2020, pp. 1–7.