Deep Learning for Proteomics Data for Feature Selection and Classification

Sahar Iravani, Tim Conrad

Deep Learning for Proteomics Data for Feature Selection and Classification

Sahar Iravani[1][0000−0002−8923−326X] and Tim O.F. Conrad[2,3][0000−0002−5590−5726]

[1] Zuse Institute Berlin, Takustrasse 7, 14195 Berlin, Germany
iravani@zib.de
[2] Zuse Institute Berlin, Takustrasse 7, 14195 Berlin, Germany
[3] Free University of Berlin, Arnimallee 6, 14195 Berlin, Germany
conrad@math.fu-berlin.de

Abstract. Todays high-throughput molecular profiling technologies allow to routinely create large datasets providing detailed information about a given biological sample, e.g. about the concentrations of thousands contained proteins. A standard task in the context of precision medicine is to identify a set of biomarkers (e.g. proteins) from these datasets that can be used for disease diagnosis, prognosis or to monitor treatment response. However, finding good biomarker sets is still a challenging task due to the high dimensionality and complexity of the data and the often quite high noise level.

In this work, we present an approach to this problem based on Deep Neural Networks (DNN) and a transfer learning strategy using simulation data. To allow interpretation of the results, we compare different approaches to analyze the learned DNN. Based on these interpretation approaches, we describe how to extract biomarker sets.

Comparison of our method to a state-of-the-art L1-SVM approach shows that the new approach is able to find better biomarker sets for classification when small sets are desired. Compared to a state-of-the-art ℓ_1-support vector machine (ℓ_1-SVM) approach, our method achieves better results for the classification task when a small number of features are needed.

Keywords: Deep learning · Attribution · LRP · Interpretation · Feature selection · Transfer learning · Mass spectrometry · Proteomics

1 Introduction

High throughput omics methods (such as proteomics) are often used in various settings to gain a better understanding of the molecular background of human diseases. In most cases, these studies are focused on the identification of so-called biomarkers that can be used for diagnosis or prognosis of a disease [26, 1]. Due to the wide range of disease-relevant processes that are influenced by proteins and recent advances in proteomics technologies such as mass-spectrometry (MS), proteomics has fostered a wide availability of this technology. Thus, the need for analyzing MS proteomics dataset has been increasing rapidly.

The overall idea of biomarker detection - also known as feature selection - is to generate proteomics mass spectra from a control group of healthy individuals and from patients carrying a specific disease. In this situation, the usual approach is to find the differences between these two groups, which can then be studied from a bio-medical perspective. The aim is to detect the best but as-small-as-possible set of discriminating features to reduce time-consuming validation studies in the wet-lab needed for each detected difference. However, due to the nature of the high-throughput mass-spectrometry acquisition process, the generated data is very high-dimensional and contains random and systematic noise, which makes analyzing this kind of data a challenging task.

Many approaches based on state-of-the-art methods such as SVM [7], Lasso [11], or ElasticNet [39] have been adapted to classify and select discriminating features from MS data [25]. Other approaches include SPA [5] that addressed classification and feature selection using compressed sensing [8] or rule mining approaches (e.g. [21]) where relevant features are identified by adapting a disjunctive association rule mining algorithm to distinguish emerging patterns from MS data. With the advances of deep learning (DL) techniques, the research to date has tended to integrate the advantage of deep learning scalability to different biomedical areas. So far, however, little attention has been paid to use DL for classification and feature selection for MS proteomics data - mainly due to the lack of enough samples to train a deep network. In this paper, we address this problem of very high dimensional MS proteomics data classification using a DNN in the case where only few training samples are available. Further, we aim to select a proper interpretable DNN approach that can be utilized to identify biomarkers. To set the stage, we will first briefly review the background of DNNs and methods for interpreting their results.

1.1 DNN Classification and Interpretation

There has been great effort on using DNNs since a DNN-based method for the first time significantly outperformed other approaches in the well known ImageNet challenge [24]. Dozents of different network topologies were proposed since then to improve the performance of DNNs for various applications, e.g. by varying layers and filter sizes [37, 32], development of the inception module [36], or adding additional connectivity between layers [15]. Furthermore, effects of different training techniques [16, 19], better activation units [14], different stochastic optimization method [9, 22], faster training method [20], and different connectivity pattern between layers [18] have improved the DNN efficiency. Parallel to advances of training deep networks, there has been quite an improvement on methods for interpreting classification decisions of a trained deep network and even first steps to go beyond this [17]. Available interpretation methods can be divided into three categories: *function*, *signal*, and *attribution* methods.

The *function technique* analyzes the operation that the network function uses to reach the output. For example, in [31] the authors proposed a class saliency map that takes the gradient of a class prediction neuron with respect to the input pixel. This can show how much a small change to each pixel would affect the prediction. However, the sensitivity maps based on the raw gradient are rather noisy. To improce this situation the Smooth grad method [33] enhanced the saliency map by smoothing the gradient using a Gaussian kernel.

The *signal technique* analyzes the components of the input that mainly influence the last layer decision. For example, the Deconvnet [38] approachmaps all the activities of a network back to the input looking for a pattern in input space that causes a specific activation in the feature maps. A given activation is propagated back through un-pooling, rectifying and filtering (transpose of learned features in a forward path) to the input layer. To un-pool for max-pooling, the switches (the position of the maximum within each pooling region) are recorded on the forward pass. Other work, e.g. Guided Backpropagation [34] suggests to ignore the pooling layer and use convolution layers with strides larger than 1. Therefore, it does not need to record the switches in the forward path.

Finally, the *attribution technique* aims for computing the importance of an input feature during the classification decision. An example is the integrated gradient method [35] that computes the partial derivatives of the output with respect to each input feature. However, instead of evaluating the partial derivative just at the given input x as in input×gradient [30], it computes the average of it while the input is changing along a linear path from a baseline x. In [3] this issue was addressed more generally such that it can be applied to a wide range of structures. This methodology called *layer-wise relevance propagation* (LRP) tells how much and to what extent each feature in a particular input contributes to the classification decision. The neuron activation on the decision layer is distributed iteratively to the previous layers until the input layer is reached.

1.2 Contribution

In this work, we present a DL-based method for classifying very high-dimensional proteomics data with the goal of biomarker (feature) identification using and comparing several methods for DNN interpretation.

Unfortunately, almost all available good quality public MS-proteomics datasets contain only up to a hundred samples - which is not enough to train a robust and generalized deep neural network. To deal with this problem, we show how transfer learning using simulated data can improve the situation significantly. Secondly, we adapted the LRP interpretation method to allow identification of the parts of the input that mainly contributes to the classification decision. These identified parts are used for feature selection. We compare the feature selection efficiency of different DNN interpretation methods (attribution, signal, and function) on labeled real datasets. We compare our results to SVM-based method that is a state-of-the-art approach for MALDI-MS feature selection [25].

2 Method

Let x_n, $n = 1,..,N$, $x_n \in \mathbb{R}^D$ and y_n, $n = 1,..,N$, $y_n \in \{0,1\}$ be the classifier input vectors in a very large D-dimensional feature space and the corresponding class labels, respectively. The aim is to find a small (if possible minimal) sized subset of features from the input data $\hat{x} \in \mathbb{R}^d$ $(d << D)$, which can be used to build a classifier f. Ideally, f - which is based only on a subset of all available features - possess the same classification performance as a classifier based on all features. Our approach for feature selection makes use of interpretability analysis for DNNs. The first step is to adapt a DNN to train a generalized model. The last layer of the trained network contains the class probabilities of the given input data. This information is propagated back through the network to the first layer using *layer-wise relevance propagation* (LRP). We use this information to identify the parts of the given input contributing the most on the DNN classification decision. We define the most contributed part of the input over all training data as discriminating features.

2.1 DNN Structure for Proteomics Data Classification

We rather used Multilayer perceptrons than convolutional neural network since our input data is a vector of wide and local dependent features whose shape should be remained the same. Otherwise, the local dependencies might be lost through reshaping the data; hence, the interpretation of the trained classification network will not be meaningful at the points where the features are separated. DNN or multilayer perceptrons are characterized by the depth and the width of the layers. Depth refers to the number of layers and width determines the number of neurons on those layers. Depth and width are selected depending on the complexity of the task while more neurons usually lead the network to learn more complex functions.

Our experiments with DNNs of different depth and width show that even though mass spectrum samples can be classified with only a few DNN layers, using more layers leads to a decreasing generalization error. However, we observe that almost all architecture, ranging from shallow to deep networks, fail to generalize correctly due to the limitation in available labeled spectra in public datasets. To tackle this challenge, we integrate the idea of transfer learning to improve this situation. The idea in transfer learning is to take the representation of a neural network that has learned from one task and transfer that representation to a new task. In this study, we use the Maldiquant library [12] in R to simulate the needed labeled data. A network with multiple fully connected layers, all followed by a rectified linear unit (ReLU) function [27] is designed to classify the simulated data. ReLU adds nonlinearity and consequently more complexity to the network. Besides the proper architecture, training the DNN is demanding to set some hyperparameters that - along with the selected structure - lead to convergence, such as learning rate l_r, optimization method of gradient descent, and proper batch size.

Setting up the proper depth, width, activation function, and hyper-parameters leads to high classification performance on the simulated dataset and consequently the weights that can be used to initialize the training process for the real mass proteomics data. We then retrain the whole network on the mass proteomics data resulting in a robust and generalized network.

2.2 DNN Interpretablity for Feature Selection

In most publicly available MS proteomics datasets, the number of samples is far too small given the number of features ($N << D$) to hope for a generalizable classifier. However, most of the features in different categories do not make a considerable effect on the classification decision. Moreover, because of the noisy nature of MS data, using all available features (dimensions) usually degrades the classification performance. Additionally, considering all data dimensions is computationally expensive. Therefore, we would like to identify the minimal sized set of input features that account for the differences of the classes (e.g. features that are only relevant in the diseased case). Our main idea is to find those features by analyzing the feature relevance during the DNN classification.

Layer Wise Relevance Propagation LRP [3] is a methodology for understanding classification decision made by multi-layer neural network. This method identifies which dimensions of the given input data contributed the most to make the classification decision, given a trained network. The LRP method consists of two main steps: after a neural network is trained, a sample is presented to the network and each neurons' activation is computed. A part of the output corresponding to the desired class is considered as the relevance score of the last layer $R^{(L)}$ that is equal to the real-valued prediction output of the classifier f. This is done using Eq. 1 (known as LRP.ϵ, see [3] for details) where $R^{(L)}$ is distributed onto its input neurons at the previous layer, such that $R_k^{(l+1)} = \sum_{\text{i: i is input for neuron k}} R_{i\leftarrow k}^{(l,l+1)}$ holds.

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

where, $\text{z}_{ij} = x_i w_{ij}$, $\text{z}_j = \sum_i z_{ij} + b_j$ and $x_j = g(z_j)$. g is a non-linear activation function. For each layer R_i is calculated for $i = 1, ..., \text{num\_neurons}$.

Alternatively, the LRP.$\alpha\beta$ rule according to Eq. 2 (see [29] for details) allows to control the importance of positive and negative values that leads to demonstrate contradicting evidence in the input (such that $\alpha - \beta = 1$) . They are typically chosen as $\alpha = 2$ and $\beta = 1$.

$$R_{i\leftarrow j}^{(l,l+1)} = R_j^{(l+1)}.(\alpha.\frac{z_{ij}^+}{z_j^+} + \beta.\frac{z_{ij}^-}{z_j^-}) \tag{2}$$

where "+", "−" denote the positive and negative parts. For $\alpha = 1$, $\beta = 0$ the propagation rule is equivalent to LRP.z^+ rule as in Eq. 3.

$$R_{i \leftarrow j}^{(l,l+1)} = R_j^{(l+1)} \frac{z_{ij}^+}{z_j^+} \tag{3}$$

Iterating every equation down to the first layer yields the relevance scores of all input dimensions, $R_i^{(1)}$.

Feature Selection $R_i^{(1)}$ gives a score for each dimension of the input vector demonstrating their strength in decision making. It means that the values assigned to each dimension indicate the importance of these features on the overall classification decision. Therefore, the high ranked dimensions represent the most discriminating features. Considering offsets, the presence of noise and different peak indices on samples belonging to different categories, we look through the entire sample relevance distributions, $R_{in}^{(1)}$ for $n = 1, ..., N$. The normalized relevance values are added up through the entire dataset. The high weighted dimensions show the strength of each individual feature to differentiate the classes. However, for MS proteomics data, in most cases the identified features are wide and all the indices around are assigned with high values as well (see Fig. 1). To deal with this effect, we establish a post-processing step to locally detect the strongest individual features. The post-processing works as follows: we first select the best feature in the whole spectra, which are determined by weights from the relevance values. Then, the neighbor's features in the determined window are removed. We then select the second best feature and iterate the process until a stopping criterion is met, e.g. when the classification reaches the whole data classification accuracy.

3 Results and Discussion

3.1 DNN Training Setup for Mass Spectra Classification and Feature Selection

Our DNN architecture is characterized by 5 fully connected layers (FCL) of 100 neurons followed by 4 more FCL of 10 neurons and a prediction layer of 2 neurons to classify two classes. All the neurons are activated by a ReLU nonlinear activation function. Neurons at the last layer are fed to the soft-max activation function, which gives the probability of the given input belonging to the healthy and diseased classes. We trained the network with cross-entropy loss function that is minimized using the momentum variant of the stochastic gradient descent optimizer [28]. Learning rate and batch-size are set to $l_r = 0.00001$ and $b = 2$. We train the network on the simulated data for 40 epochs, and then retrain it on real data for 40 epochs followed by another real data set for 10 epochs for fine-tuning. Afterward, the LRP analyzer is applied to each sample that activates the network neurons to get the most relevant parts used by the DNN for the classification decision. Due to the noisy nature of MS data and mass shift of samples the relevance values are calculated for the entire spectrum in the dataset. Finally, the average of normalized relevance values are post-processed with a window-size of 50 on the result.

3.2 Implementation Detail

All the experiments in our proposed method are implemented in python using Keras [4] with Tensorflow backend and innvestigate library [2] on a machine with a 3.50GHz Intel Xeon(R) E5-1650 v3 CPU and a GTX 1080 graphics card with 8 GiB GPU memory.

3.3 Results on Spiked Data

In this section, we compare different methods for DNN interpretation such as Gradient method (grad), variants of LRP (LRP.z, LRP.ϵ, and LRP.$\alpha\beta$ rules), input×gradient, integrated gradient(int_grad), guided back-propagation (guided), deconvenet (dCN), and smooth grad (smgrad) through peak detection (see [2] for more details on the methods). With this comparison, we aim to evaluate the impact of interpretation methods using a public dataset known as *spiked data* [10, 23]. The spiked data-set contains proteomics mass spectra of control and case groups from human blood samples. The case group has been spiked with a protein-mix of different concentration. The amplitudes of 6 spiked peaks differentiate the spectra into case and control and their known m/z (position) values can be used as ground-truth [6]. Thus, the main aim in this part is to investigate how well an algorithm can detect the m/z positions of the known 6 individual spiked peeks among all 42.381 dimensions. The data contains 95 samples of 50 case and 45 control spectra. The experiments are carried out on two concentration levels, 12.21nMol/L and 0.76nMol/L, referred as *spiked160* and *spiked80* in this paper.

The results of our approach, i.e. the selected spiked peaks, are shown in table 1 and 2. The reported peaks are the closest ones to the spiked peaks ground-truth among almost 30 high-ranked features. From these two tables we can clearly see that LRP variants (attribution method), inp×grad, and int_grad are far more capable than signal (grad and smoothgrad) and function (guided and dCN) methods. It can also be seen from the results that, while there is no considerable difference between the variant of LRP in this application, one small peak (m/z 3149) can only be detected using LRP.z. Further studies are needed to investigate the reason for this.

Prior to feature selection using the described DNN classification analyzer, the network should become generalized enough to allow the application of interpretation methods. This is what we addressed with transfer learning for the cases when only a few labeled samples are available to train a DNN. In this situation, a simulated dataset of 5000 samples [12] is fed to the network. The dataset contains two equal-size groups spectra as control and case. Each simulated spectra has more than 40 thousands of mass values as the real data and simulated data spectra have. In addition, each one has 412 peaks in which 24 are discriminating. They are equally spread in two groups and are set in fixed positions trough entire dataset. After training, the network re-trained on a real-world dataset of 81 samples and then fine-tuned on spiked data. Initializing the network weights this way should lead to better results since it is less likely that the optimizer gets trapped in a bad local minimum.

We observe from training the network that, while the objective function can not converge on some subsets of samples, the pre-trained network can avoid that. Pre-trained weights lead to a more robust network that resulted in 97.1% ($CI \pm 2.68$) and 96.5% ($CI \pm 3.6$) generalization accuracies on spiked160 and spiked80, respectively. The seemingly large confidence intervals (CI) results from misclassification of one sample on different subsamples during training. Iterating training (train and validation) on 90% of randomly selected spiked160 (95 samples) and inferring on the rest, each time leads to 100% or 88% testing accuracies. This means when the network perform 88% on testing 1 spectrum out of 9 ones was misclassified.

Table 1. Detected spiked peaks using the 9 DNN interpretation methods on spiked160 as the top 35 high ranked features.

peaks	grad	LRP.z	LRP.$\alpha\beta$	LRP.ϵ	inp×grd	`int_grad`	guided	dCN	smoothgrad
1047.20	-	**1047.91**	1046.76	**1047.91**	**1047.91**	**1047.91**	-	-	-
1297.51	-	1300.67	**1298.23**	1300.67	1300.67	1300.67	-	-	-
1620.88	1623.6	1621.91	**1620.48**	1621.91	1621.91	1621.91	-	-	1623.6
2466.73	-	2467.63	**2466.51**	2467.63	2467.63	2467.63	2463.63	-	-
3149.61	-	-[*]	-	-	-	-	-	-	-
5734.56[**]	-	-	-	-	-	-	-	-	-

[*] Although m/z 3149 is not selected as top high ranked features because of its insignificant peak in comparison to larger peaks in the spectra (as illustrated in Fig. 1), it is selected as the 94th feature with our method using LRP.z. The other LRP rules can also select this peak but later as the less important feature. However, inp×grad and int_grad could not find this small peak. This is the reason why we analyzed the noisy P.CA data and the visualizations by adapting the LPR.z rule.

[**] The mean height of the signal in this peak is less than 40 that is comparable to the level of noise in both spiked160 and spiked80 data-sets [5]. Therefore, this peak cannot be selected as a discriminating feature.

We further explained the results in Fig. 1 by visualizing the output of one of the interpretation methods . The figure shows the mean of the normalized LRP.z values of a spiked160 spectrum overlaid on the distribution of case and control spectra of the dataset around the selected spiked peaks. The visualization around the spiked peaks, as shown in these plots, indicate the wide peak range that causes the deviation on the selected features from the spiked ground truth peaks in tables 1 and 2.

Table 2. Detected spiked peaks using 9 DNN interpretation methods on spiked80 as the top 30 high ranked features.

peaks	grad	LRP.z	LRP.$\alpha\beta$	LRP.ϵ	inp×grd	`int_grad`	guided	deCN	smoothgrad
1047.20	-	1040.61	1041.76	1040.61	1040.61	1040.61	-	-	-
1297.51	-	1298.35	**1298.0**	1298.35	1298.35	1298.35	-	-	-
1620.88	-	**1620.87**	1619.7	**1620.87**	**1620.87**	**1620.87**	-	-	-
2466.73	-	**2467.63**	2468.6	**2467.63**	**2467.63**	**2467.63**	-	-	-
3149.61	**3151.25**	-	-	-	-	-	-	-	**3151.25**
5734.56	-	-	-	-	-	-	-	-	-

The spiked peaks that are amongst the top 30 selected features using our pipeline are supposed to be selected as the most discriminating features. However, in Fig. 2 we illustrate that the selected features that are ranked better than the true spiked peaks are more discriminating. For example, it is apparent from the plot that the difference of intensity values of the case and control samples around feature 1021 is larger than their corresponding difference around feature 1047. Therefore, the DNN tends to rely more on these areas in order to make the classification decision. It can also be learned from this plot that not only the individual features are important for the DNN to make a classification decision, but a Gaussian range around high ranked ones also plays a crucial role. For example, relevance values around the m/z 1021 are considerably higher than the relevance value of individual m/z 1047. Therefore, we can not expect a DNN to classify the two groups based on only individual features.

3.4 Results on Pancreas Cancer Data

The Pancreas Cancer dataset (P. CA) is another pulicly available data-set [10]. It contains 81 spectra having 42391 features collected from pancreatic cancer patients and apparently healthy control patients. As described previously, due to the lack of sufficient training samples on the public dataset for training a deep network we retrain the network on real data from the network trained on simulated data. We achieved 98%-95% training-testing average accuracy while almost all the structures of DNN we tried from shallow to deep and narrow to wide could not become generalized correctly. The classification decision is interpreted using LRP.z rule to extract the important parts. Fig. 3 illustrates the average of normalized LRP.z over the entire dataset around two of the high ranked features. The relevance values are overlaid on top of the mean of the case and control spectra. These two features are illustrated due to the large impact on the classification decision after feature selection (see Fig. 4).

We compare our feature selection method with benchmark methods on the same dataset as follows. A BinDA-algorithm-based method [13] reported 30 peaks m/z 4495, 8868, 8989, 1855, 4468, 8937, 2023, 1866, 5864, 5946, 1780, 2093, **5906**, 5960, 8131, **1207**, 4236, **2953**, 9181, 1021, **1466**, **4092**, 4251, 5005, 8184, 1897, **3264**, 2756, 6051, and 1264, with m/z 8937 as the most discrimi-

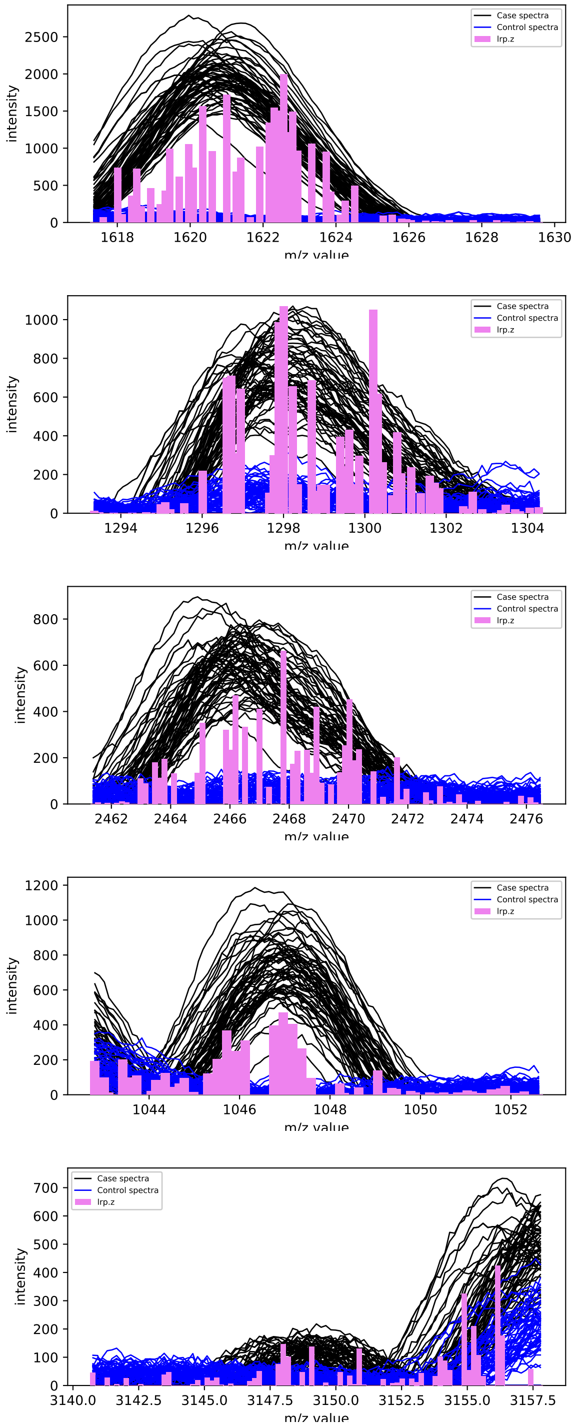

Fig. 1. Visualization of the relevance values around the spiked peaks. Black and blue show the diseased and healthy spectrum of spiked160, and the bars are the average of the normalized LRP.z values over the entire samples. The bars are scaled to the maximum intensity of the spectrum.

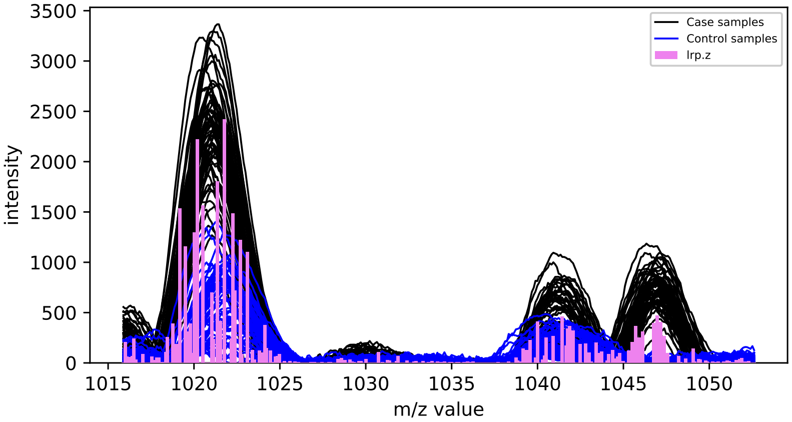

Fig. 2. Comparison visualization of two selected peaks. This plot illustrates the selected spiked 1047 in a wider range to include the selected feature 1021. This illustration shows that m/z 1021 is selected prior to the ground truth m/z 1047 since network sees larger differences between the two classes. Black and blue show the diseased and healthy spectrum of spiked160, and the bars are the averge of the normalizer LRP.z values over the entire samples. The bars are scaled to the maximum intensity of the spectrum

nating features for pancreatic progenitor cell differentiation. Note that, the bold m/z values indicate the features that are also discovered by our method.

In [5] a compressed sensing-based approach was used to identify peaks with m/z **1464**, **1546**, **1944**, **5904**, **1619**, **4209**, and m/z **2662** as the most important features to distinguish the healthy and diseased spectra. In this study, peaks with m/z values **4212.36**, **1465.43**, **3264.36**, **2661.37**, **5909.96**, **4092.18**, **1616.98**, **1545.91**, 4647.56, 6636.87, 3191.41, 2934.34, 5338.51, **2953.42**, 1060.26, and m/z 3242.47 are ranked as the most discriminating features to achieve the state-of-the-art classification accuracy of 95% [5]. The mass shift of 1 to 3 Dalton on the m/z axis among the identified peaks over different study is likely arising from different pre-processing and post-processing procedures.

3.5 Feature Selection Comparison

We compare the discriminating accuracy of our feature selection method with the state-of-the-art ℓ_1-SVM approach for MALDI-MS feature selection [25]. Fig. 4 shows how the classification accuracy is changing for both approaches when the number of features used by the classifier is increased. The experiments are carried out on the two spiked data-sets and the P. CA data-set. As can be seen from the plots, while both methods reach the maximum performance, our method outperforms the ℓ_1-SVM approach when only very few features are used.

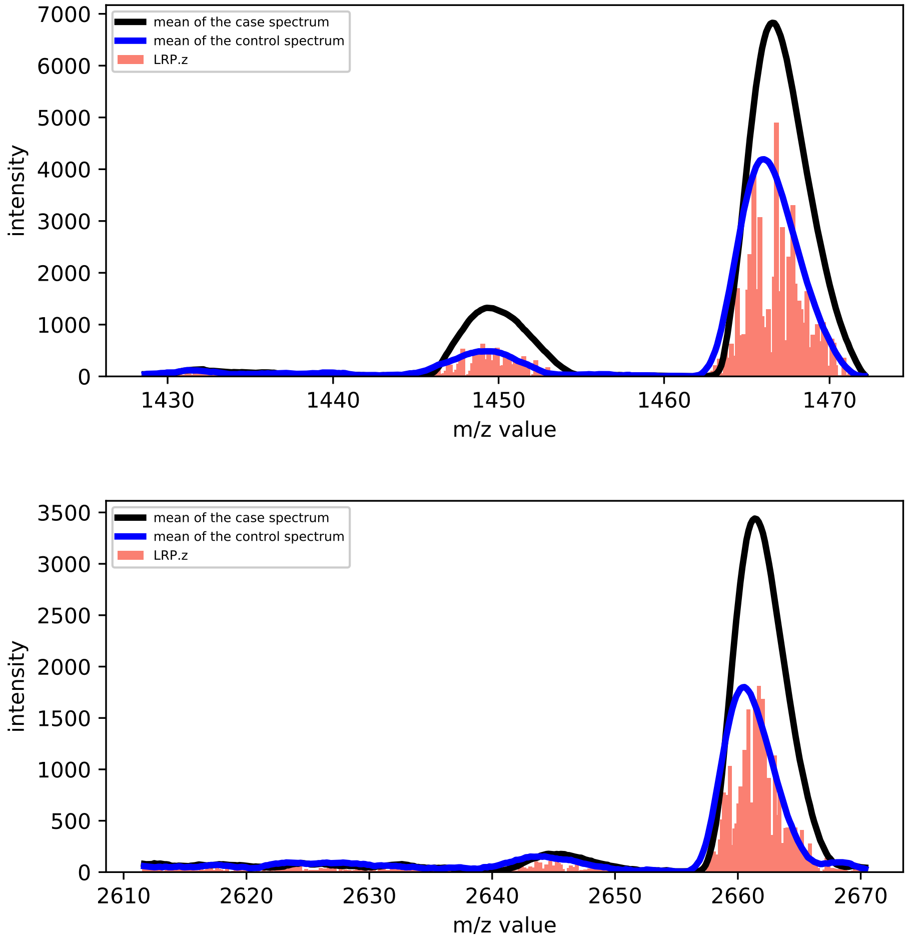

Fig. 3. Illustration of the relevance values around the second (m/z 1465) and forth (m/z 2661) high ranked features of P.CA data. These features are picked for illustration due to their largest impact on the classification accuracy after feature selection, which is apparent from the last row of figure 4. The means of the case and healthy spectrum are shown in black and blue, respectively.

This is an important property in the situation, where more selected features lead to higher costs in the following steps in some bio-medical pipeline, where each selected feature must be validated in expansive wet-lab experiments.

We further investigate the DNN classification performance using the individual features by adding the selected features to the dataset. Despite SVM, it shows significant deduction on the results since as it is shown in the Fig.

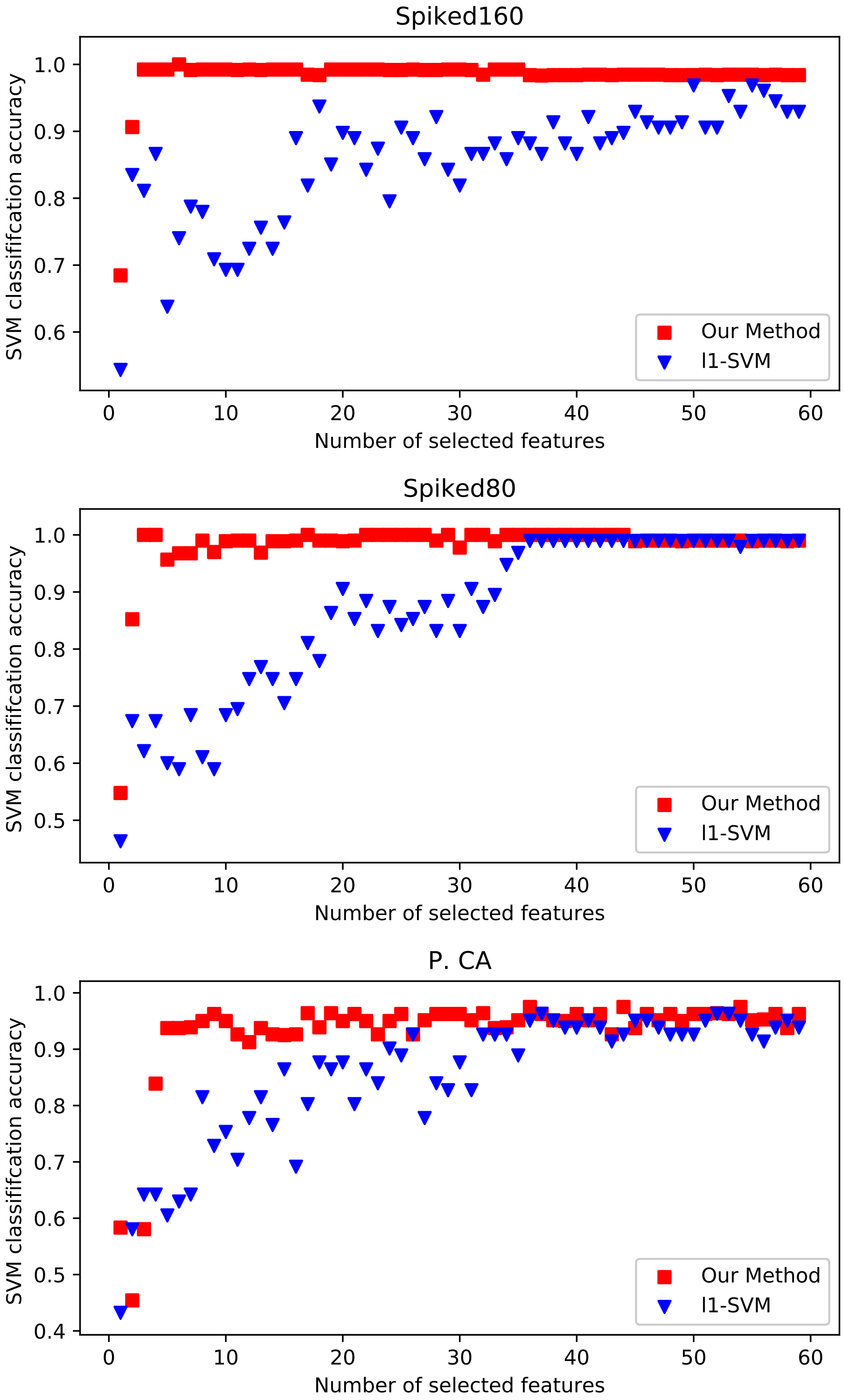

Fig. 4. Generalization accuracies with increasing the number of features to the dataset. Plots show the strength of selected features on spiked160 (first row), spiked80 (second row), and P. CA (third row) using our method in red-square and ℓ_1-SVM in blue-triangle.

2 and explained in previous section DNN sees a wide window around selected individual features for making decision rather than single features.

4 Conclusion

This paper presents a new feature selection method based on deep neural networks (DNN) and a transfer learning strategy using simulated data for very high dimensional MS proteomics data. We compare different DNN interpretation methods and show that the attribution based methods perform best for this application. We also demonstrate that there is no considerable difference between the variant of LRP (ϵ, $\alpha\beta$, and z rules) for identifying the important parts of proteomics data for a classification decision. The results suggest that our approach has a significantly better performance than classical approaches on the classification task, where quite a few numbers of features are favorable.

Acknowledgments

This study was funded by the German Ministry of Research and Education (BMBF) Project Grant 3FO18501 (Forschungscampus MODAL) and Project Grant 01IS18037I (Berlin Center for Machine Learning).

References

1. Aebersold, R., Mann, M.: Mass spectrometry-based proteomics. Nature **422**(6928), 198 (2003)
2. Alber, M., Lapuschkin, S., Seegerer, P., Hägele, M., Schütt, K.T., Montavon, G., Samek, W., Müller, K.R., Dähne, S., Kindermans, P.J.: innvestigate neural networks! Journal of Machine Learning Research **20**(93), 1–8 (2019)
3. Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.R., Samek, W.: On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. PloS one **10**(7), e0130140 (2015)
4. Chollet, F., et al.: Keras. https://keras.io (2015)
5. Conrad, T.O., Genzel, M., Cvetkovic, N., Wulkow, N., Leichtle, A., Vybiral, J., Kutyniok, G., Schütte, C.: Sparse proteomics analysis–a compressed sensing-based approach for feature selection and classification of high-dimensional proteomics mass spectrometry data. BMC bioinformatics **18**(1), 160 (2017)
6. Conrad, T.O., Leichtle, A., Hagehülsmann, A., Diederichs, E., Baumann, S., Thiery, J., Schütte, C.: Beating the noise: new statistical methods for detecting signals in maldi-tof spectra below noise level. In: International Symposium on Computational Life Science. pp. 119–128. Springer (2006)
7. Cortes, C., Vapnik, V.: Support-vector networks. Machine learning **20**(3), 273–297 (1995)
8. Donoho, D.L., et al.: Compressed sensing. IEEE Transactions on information theory **52**(4), 1289–1306 (2006)
9. Duchi, J., Hazan, E., Singer, Y.: Adaptive subgradient methods for online learning and stochastic optimization. Journal of Machine Learning Research **12**(Jul), 2121–2159 (2011)

10. Fiedler, G.M., Leichtle, A.B., Kase, J., Baumann, S., Ceglarek, U., Felix, K., Conrad, T., Witzigmann, H., Weimann, A., Schütte, C., et al.: Serum peptidome profiling revealed platelet factor 4 as a potential discriminating peptide associated with pancreatic cancer. Clinical Cancer Research **15**(11), 3812–3819 (2009)
11. Friedman, J., Hastie, T., Tibshirani, R.: Regularization paths for generalized linear models via coordinate descent. Journal of statistical software **33**(1), 1 (2010)
12. Gibb, S., Strimmer, K.: Maldiquant: a versatile r package for the analysis of mass spectrometry data. Bioinformatics **28**(17), 2270–2271 (2012)
13. Gibb, S., Strimmer, K.: Differential protein expression and peak selection in mass spectrometry data by binary discriminant analysis. Bioinformatics **31**(19), 3156–3162 (2015)
14. Glorot, X., Bordes, A., Bengio, Y.: Deep sparse rectifier neural networks. In: Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics. pp. 315–323 (2011)
15. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
16. Hinton, G.E., Srivastava, N., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.R.: Improving neural networks by preventing co-adaptation of feature detectors. arXiv preprint arXiv:1207.0580 (2012)
17. Holzinger, A., Langs, G., Denk, H., Zatloukal, K., Müller, H.: Causability and explainabilty of artificial intelligence in medicine. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery p. e1312 (2019)
18. Huang, G., Liu, Z., Weinberger, K.Q., van der Maaten, L.: Densely connected convolutional networks. arXiv preprint arXiv:1608.06993 (2016)
19. Huang, G., Sun, Y., Liu, Z., Sedra, D., Weinberger, K.Q.: Deep networks with stochastic depth. In: European Conference on Computer Vision. pp. 646–661. Springer (2016)
20. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: International Conference on Machine Learning. pp. 448–456 (2015)
21. Jayrannejad, F., Conrad, T.O.: Better interpretable models for proteomics data analysis using rule-based mining. In: Towards Integrative Machine Learning and Knowledge Extraction, pp. 67–88. Springer (2017)
22. Kingma, D., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
23. Kratzsch, J., Fiedler, G.M., Leichtle, A., Brügel, M., Buchbinder, S., Otto, L., Sabri, O., Matthes, G., Thiery, J.: New reference intervals for thyrotropin and thyroid hormones based on national academy of clinical biochemistry criteria and regular ultrasonography of the thyroid. Clinical chemistry **51**(8), 1480–1486 (2005)
24. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. pp. 1097–1105 (2012)
25. Liu, Q., Sung, A.H., Qiao, M., Chen, Z., Yang, J.Y., Yang, M.Q., Huang, X., Deng, Y.: Comparison of feature selection and classification for maldi-ms data. BMC genomics **10**(1), S3 (2009)
26. Marrugal, Á., Ojeda, L., Paz-Ares, L., Molina-Pinelo, S., Ferrer, I.: Proteomic-based approaches for the study of cytokines in lung cancer. Disease markers **2016** (2016)

27. Nair, V., Hinton, G.E.: Rectified linear units improve restricted boltzmann machines. In: Proceedings of the 27th international conference on machine learning (ICML-10). pp. 807–814 (2010)
28. Qian, N.: On the momentum term in gradient descent learning algorithms. Neural networks **12**(1), 145–151 (1999)
29. Samek, W., Montavon, G., Binder, A., Lapuschkin, S., Müller, K.R.: Interpreting the predictions of complex ml models by layer-wise relevance propagation. arXiv preprint arXiv:1611.08191 (2016)
30. Shrikumar, A., Greenside, P., Shcherbina, A., Kundaje, A.: Not just a black box: Learning important features through propagating activation differences. arXiv preprint arXiv:1605.01713 (2016)
31. Simonyan, K., Vedaldi, A., Zisserman, A.: Deep inside convolutional networks: Visualising image classification models and saliency maps. arXiv preprint arXiv:1312.6034 (2013)
32. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
33. Smilkov, D., Thorat, N., Kim, B., Viégas, F., Wattenberg, M.: Smoothgrad: removing noise by adding noise. arXiv preprint arXiv:1706.03825 (2017)
34. Springenberg, J.T., Dosovitskiy, A., Brox, T., Riedmiller, M.: Striving for simplicity: The all convolutional net. arXiv preprint arXiv:1412.6806 (2014)
35. Sundararajan, M., Taly, A., Yan, Q.: Axiomatic attribution for deep networks. In: Proceedings of the 34th International Conference on Machine Learning-Volume 70. pp. 3319–3328. JMLR. org (2017)
36. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1–9 (2015)
37. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: European conference on computer vision. pp. 818–833. Springer (2014)
38. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: European conference on computer vision. pp. 818–833. Springer (2014)
39. Zou, H., Hastie, T.: Regularization and variable selection via the elastic net. Journal of the Royal Statistical Society: Series B (Statistical Methodology) **67**(2), 301–320 (2005)