Project Definition¶

Domain Background¶

In the field of instrumental analytical chemistry, there are many techniques for identifying unknown substances. This field is primarily researched for the detection of explosives, drugs and chemical weapons, however, there are many other applications for it. In the healthcare field, researchers are using it to detect cancer and diseases. Chromatography is a method by which a substance is separated into different components and split across time. For example, if you sample a mixture that is made up of multiple substances you could separate those substances and detect them individually with chromatography. One method of chromatography is using what is called a column. This device is what will separate out the components of a mixture, however, it is extremely slow. It takes minutes to work.

Mass spectrometry is the technique of identifying the chemical structure of a substance by separating ions into differing mass and charge. The three essential components of a mass spectrometer are the ion source, the mass analyzer, and the detector. The ionizer ionizes the sampled substance. The analyzer sorts and separates the ions by mass and charge. The detector measures or counts the separated ions. Ions can be displayed as a histogram of mass-to-charge ratio(m/z) or can be displayed across time to see line curves where the peaks would be where the max quantity of ions were detected. There are many different techniques for each component in performing the identification of substances.

The most popular analytical technique today is ion-mobility spectrometry (IMS). This technique separates and identifies ionized molecules based on their mobility within a carrier gas. It is extremely fast and takes milliseconds to achieve a result, but it is less sensitive than other techniques. This technique is very popular because you can make an IMS device for relative low cost compared to other techniques and the device can be small enough to be hand-held.

The final technique that we will discuss is triple quadrupole mass spectrometry (TQMS). This is a tandem mass spectrometry technique, which just means it has two analyzers. The components in order are the ion source, a quadrupole mass analyzer, a quadrupole that acts as a collision cell to fragment the molecules entering it, a second analyzer that analyzes the resulting fragments, and finally the detector. The quadrupole works by creating an electro-magnetic field that separates the ions and makes them follow a

trajectory based on their mass-to-charge ratio (m/z). This technique in theory is the most sensitive and will achieve results in seconds. These devices tend to be very expensive and large. This is the device a team and I are working on.

The above techniques discussed can all be combined to solve problems depending on the application. There are trade-offs that must be made for each technique. Cost and weight are always a major factor. In some cases, the science is not well understood.

Problem Statement¶

My team and I are currently working on a triple quad mass spectrometer that is cheaper and smaller so that we can address new markets and applications that TQMS was unable to address previously. Our current instrument displays mass-to-charge ratios over time. We have in the past used peak thresholds to determine what is a detected substance. This technique only gives us an accuracy of 40% with a very high rate of false positives. We are trying to achieve an accuracy of 90% with no more than a 2% false positive rate. We have talked about adding some filtering techniques, but there will be a tradeoff in time and cost. We need to complete our analysis in under 10 seconds. Ideally, we can solve our problem with purely algorithms. First, I would like to see if we can use our existing approach of classifying compounds based on peak features of relevant mass pairs. We need to first assess what peak features distinguish our detections from noise. If that does not work, then I should be able to use a 1D CNN to learn the mass pair intensity shapes.

Data sets and Input¶

The datasets that will be used in this project were generated from collected samples from our instrument. The instrument was sent out for testing and 12 different substances were tested. The data files we got back are the results from that testing. The datasets are generated from these data files. The datasets have been modified to abstract out any sensitive details such as the substance name and mass pairs. Most importantly the intensities are all generated to mimic the shape of the collected data. The data has also been filtered to remove any malformed data because of a hardware or any other error. There is no proprietary data associated with this project. The model that will be built will need to be re-trained on the actual proprietary dataset to have it work with our instrument. The generated data should be more than adequate in evaluating a model. In most cases, I have between 50 and 80 samples for each substance I am testing for. I realize that this may not be enough, but I am also trying to gauge how many samples will be needed if extending out the substance library. If a compound is performing poorly from lack of samples, I will remove it from the test.

Each data file consists of multiple components. First, there will be a mass pair transition id. A mass pair transition consists of an ion charge (+ or -), parent mass, a daughter mass, and collision energy i.e. +123->456(78). The ion charge is from the ion source, the parent mass is from the first quad, the daughter mass is from the third quad and the collision energy is applied at the second quad. Instead of seeing this transition you will see a number 0 to n-1 where n is the total quantity of specified transitions (i.e. n=51). After the id there will be a sample id associated to the dataset, a comment field which will specify if another substance was combined the tested for substance, and what substrate it was sampled on. Substrate could have the value direct, or a harvest code like Perf5. Direct means the substance was inserted into the instrument through a syringe. Direct should be the most stable result. If the substance was harvested off material Perf5 that means the substance was applied to the material and then swiped off with one of our swabs. Theoretically when a substance is harvested it should measure lower ions than direct because you may not of collected the entire amount of the substance. After the substrate field, there will be detection field and an association field. The detection field will have an array of numbers specifying what compounds are detected within that dataset. The association field will specify what mass pairs are associated to which compounds according to our chemist, for example mass pair 1 is associated to compounds [1,3]. After the associated field there will be a height, width, area, and position of the mass pair peak. These values are acquired by applying a smoothing filter to reduce the noise of the signal. Finally, there will be a time series of intensities over a specified number of time steps or scans (i.e. 23). For now, the scan count is fixed to 23, however, in the future it would be better to have scan count be variable in case we want to stop early. So, for each data file you would have that structure per row times the amount of transitions for example 51x33.

mass_pair_id	sample_id	comment	substrate	detection	association	peak_height
0	21321	Blank	None	None	[1,2]	1000.0

...

peak_width	peak_area	peak_position	timestep1	...	timestep_n
5.2	3000.2	5.1	0.0	...	100.5

Used packages:

- Matplotlib
- Scikit-learn
- Scipy
- Tensorflow
- Tensorflow-gpu
- keras

Number of Samples: 641
Max number of mass pairs: 51

None	11.0
[10]	58.0
[13]	68.0
[14]	36.0
[15]	68.0
[18]	77.0
[19]	4.0
[21, 0, 18, 4]	75.0
[21]	79.0
[22]	79.0
[3]	19.0
[7]	9.0
[8]	58.0

Compound 7 and 19 do not have enough samples. I will perform data analysis anyways. I will consider dropping later.

Metrics¶

Benchmark Model¶

The model can only be benchmarked against the previous solution which yields a total accuracy of 40%. Minimum peak height was the only threshold where it was manually set based on internal lab testing. for example, compound 1 could have a minimum

height threshold of 1600 ion counts. If the peak signal was below this amount it was deemed noise and ignored. If it was above it would be part of some additional logic that required all associated mass pairs to be above their limits to raise a substance detection alert. Below is most of a confusion matrix, but TNR is missing.

Compound ID	TPR	FPR	FNR
Compound 3	5.00%	4.55%	95.00%
Compound 4	30.34%	52.81%	32.58%
Compound 7	0.00%	87.50%	12.50%
Compound 8	13.33%	20.00%	66.67%
Compound 10	12.31%	7.69%	86.15%
Compound 13	0.00%	21.21%	84.85%
Compound 14	5.00%	0.00%	95.00%
Compound 15	20.37%	12.96%	68.52%
Compound 18	59.65%	28.95%	14.91%
Compound 19	20.00%	70.00%	50.00%
Compound 21	72.90%	26.17%	0.93%
Compound 22	70.27%	17.12%	15.32%

Accuracy: ~40%

Evaluation Metrics¶

To evaluate our models, we should be using a weighted accuracy metric such as f-beta score and a confusion matrix to see our false positive rate. According to our requirements we could miss a detection 10% of the time if we have a false positive rate below 2%. Since it is more important to be precise than have high recall, we should have our beta be a value of 0.5.

Analysis¶

Data Exploration¶

I need to build a compound to mass pair lookup table in order to find the mass pairs that are important.
{-1: [0... 50],
0: [22, 23, 25, 26, 33, 34],
3: [20, 47, 49, 50],
4: [2, 11],
7: [16, 18, 27, 30, 46],
8: [16, 18, 39, 46],
10: [0, 3, 4, 19, 20, 22, 23],
13: [41, 42, 46],
14: [16, 18, 46],
15: [36, 39, 46],
18: [19, 20, 21, 40],
19: [7, 8],
21: [22, 23, 25, 26, 33, 34],
22: [35, 37, 38]}

Explore peak properties first. If data ends up being gaussian then we can just use a non-parametric solution. We will need to scale values because they are disproportionate and large across feature columns. I need to see a description of the data by mass pair id to compound so I can determine what features look the most promising.

Abbreviated. All graphs are under external document EDA graphs

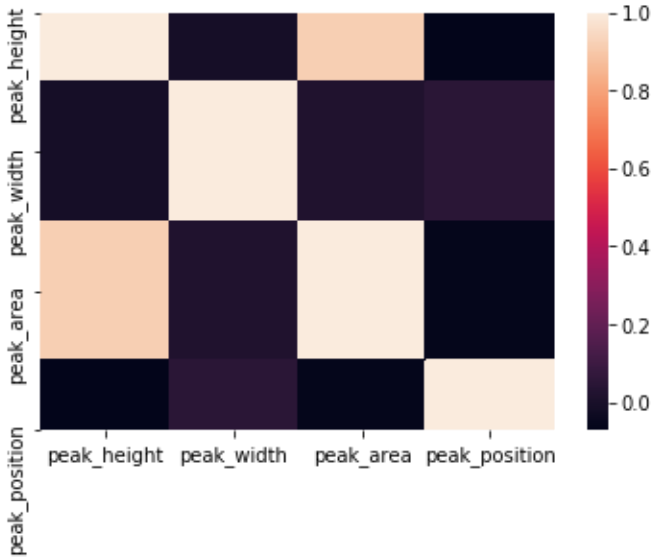
Mass pair ID: 0
Compound id: 10

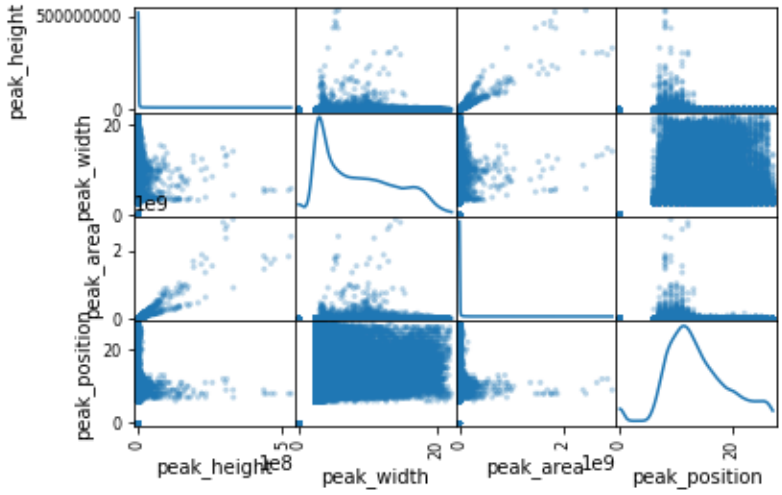
	peak_height	peak_width	peak_area	peak_position
count	58.000000	58.000000	58.000000	58.000000

mean	11550.699613	2.947469	24126.172809	9.603448
std	21829.875428	0.466799	45804.076400	3.631923
min	161.671196	2.455954	504.394940	7.000000
25%	775.652289	2.720582	2211.525918	8.000000
50%	3186.260561	2.839728	7821.474898	9.000000
75%	12483.160226	3.170623	24369.551928	9.000000
max	113501.014255	5.594501	235102.816555	24.000000

After seeing a description of the data, it is obvious that width and position seem to be the most stable features with the least amount of variance. By looking at the data it seems there may be some correlation between height to area and width to position. I need to test to make sure so that I can remove the features that are correlated to improve my future model's accuracy. I'm going to look at all data first, then break it down.

Exploratory Visualization¶

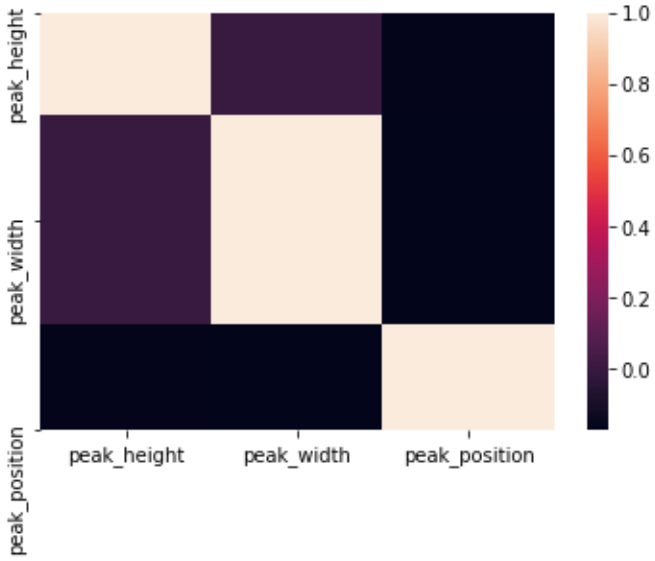


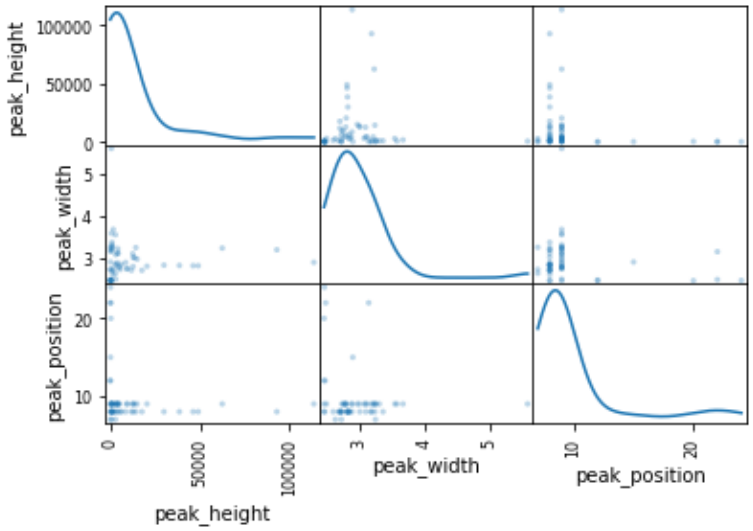


You can tell there is a strong correlation between peak height and area. This makes sense that area under our peak would be calculated based on height. Let's remove area and see if we can see any other correlations once we break down the data to important compound to mass pair associations. I will use a heatmap and scatter matrix to determine correlations and the distribution of data related to each feature.

Abbreviated. All graphs are under external document EDA graphs

Mass pair ID: 0
Compound id: 10

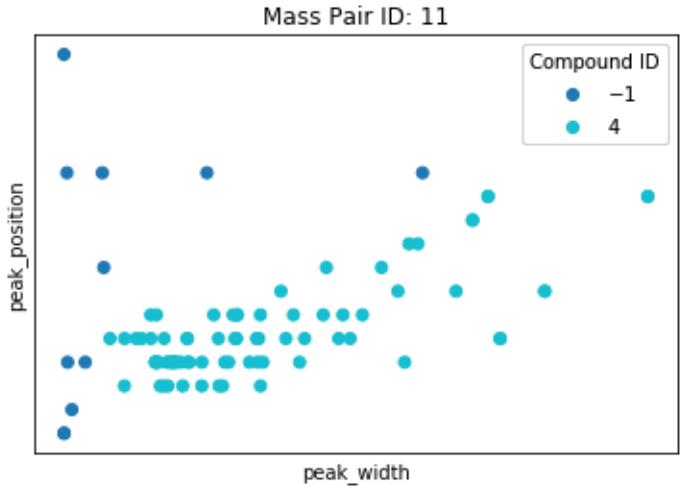


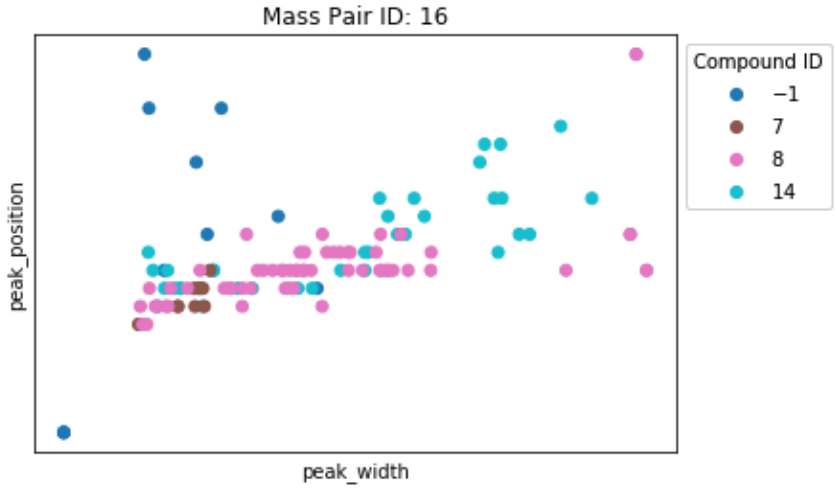


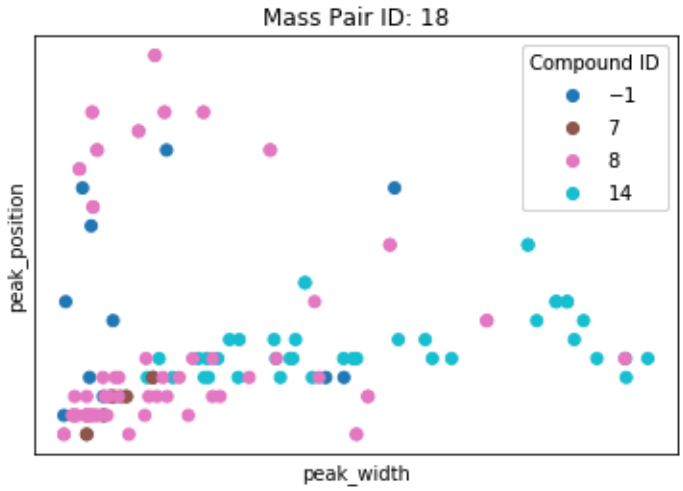
In some cases, the transformation of the data works well. In most cases it does not. It seems that either I do not have enough samples to represent the total population of samples or the current features are not deterministic enough to determine detections.

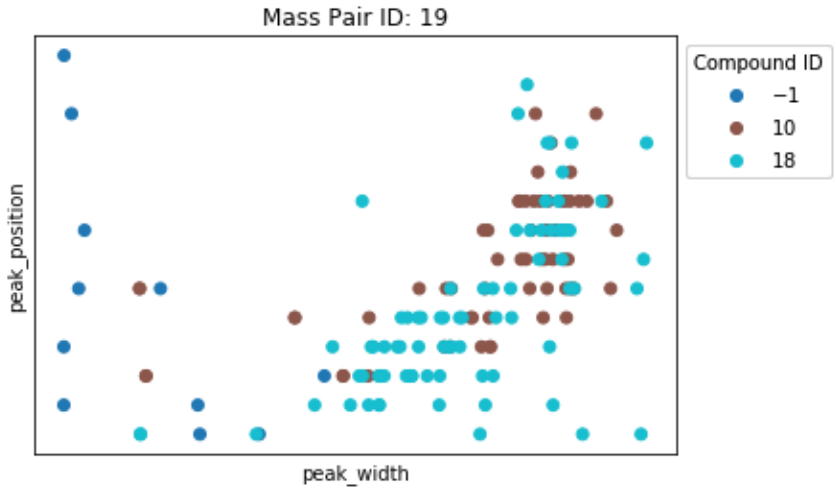
For the most part, width seems to be a very stable and independent feature. Position and height even though not correlated tend to mimic each other so I could probably get away with using either just height or position. I will try position because it previously had the least amount of standard deviation. Some of the mass pair to compounds are much more stable than others. On all cases, there are some outliers. Depending on the algorithm I use, I may need to remove these outliers to not overfit my model. Next, I want to plot and label all my mass pair data to get an idea of how well my algorithm will do on test data.

Abbreviated. All graphs are under external document EDA graphs









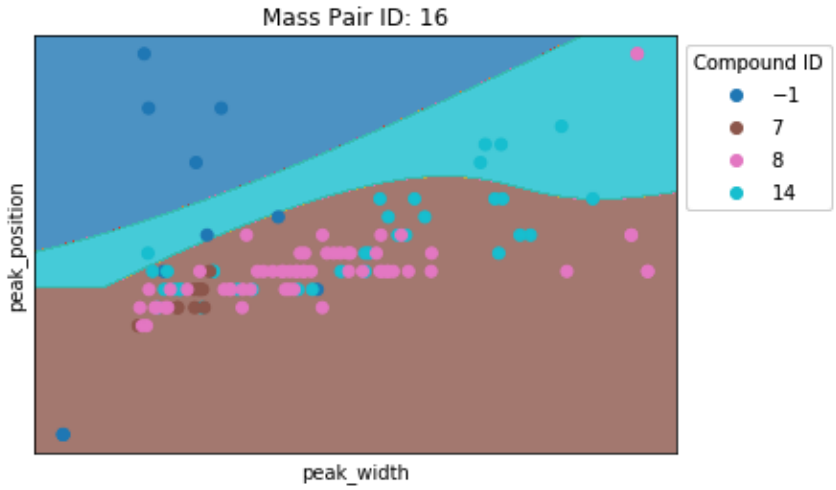
Algorithms and Techniques¶

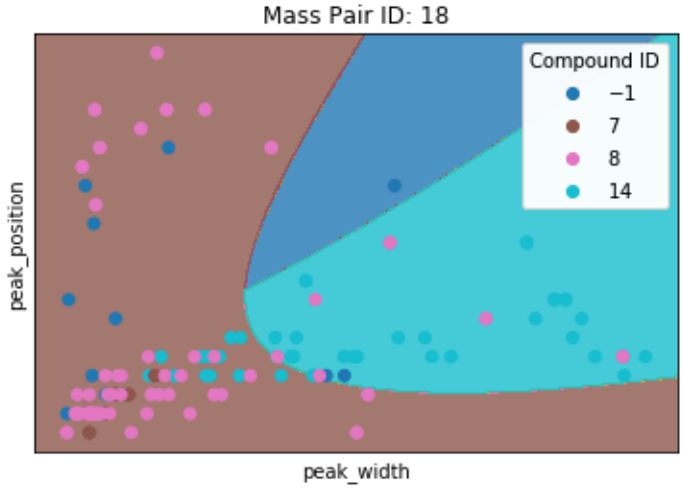
In most cases, graphically height looks a lot worse than position as a feature. In both cases, the features seem less than ideal. Let's apply a polynomial support vector machine to the data. My hope is to create a very general classification that combined with all the associated mass pairs we can gain good results. Based on the data, it appears we could probably draw circles around pieces of the data, therefore, we could use a support vector machine (SVM) of either polynomial or RBF. RBF may overfit so I

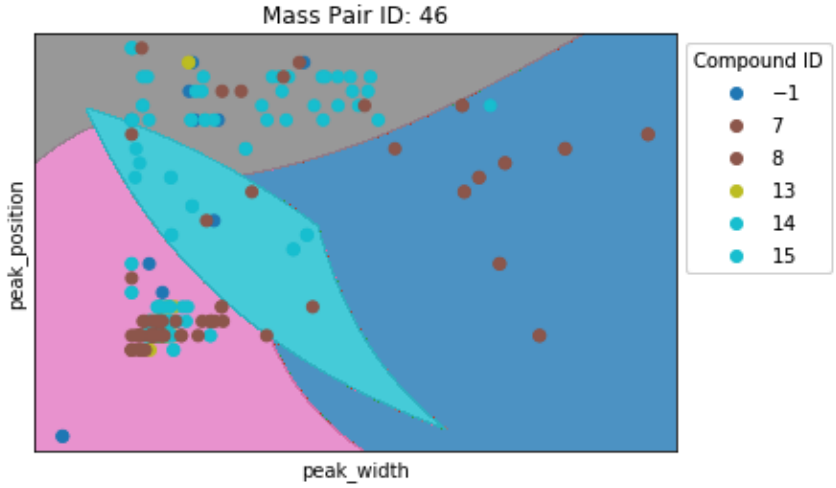
will first try polynomial and see the results. I will graph the classifications so I can see how well the algorithm fits per mass pair.

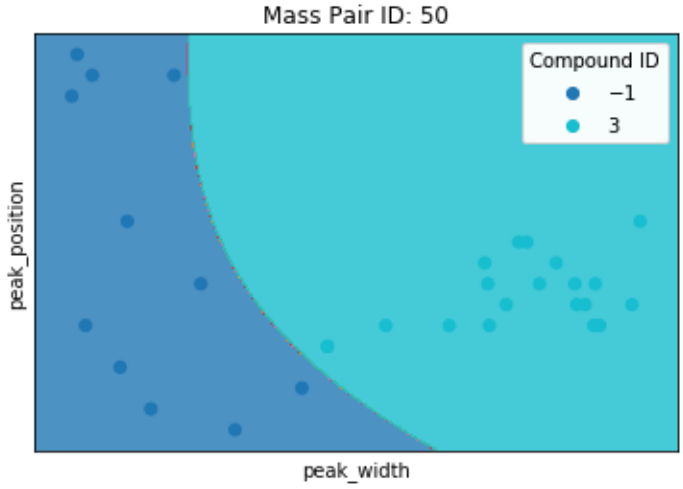
There are multiple reasons I would choose an SVM over other parametric algorithms. In most cases, the data is not linearly separable. This would rule out using a logistic regression algorithm or even a decision tree because a decision tree would probably overfit the data. Now another problem, I have is most of the data is not gaussian. Since it is not gaussian, I must be weary of creating classification lines that are too tight resulting in overfitting of the data. SVMs by design do not overfit the data because they try to maximize the margin between our 2 classes. By using a polynomial SVM I can transform the existing data into a higher dimension resulting in data that is more separable than it was before. This idea is called the kernel trick. In simplified terms, it works by taking the inner product of our data and using those values to increase our dimensionality hopefully creating a distinct decision boundary.

Abbreviated. All graphs are under external document EDA graphs




Mass Pair ID: 18
Compound ID
−1
7
8
14
peak_position
peak_width


Mass Pair ID: 46
Compound ID
−1
7
8
13
14
15
peak_position
peak_width


Mass Pair ID: 50
Compound ID
−1
3
peak_position
peak_width

After reviewing each mass pair graph, it appears that in some cases we can distinguish quite well, while in other cases we cannot. Looking at mass pair id 46 we can see that the regions drawn are quite elaborate. This is very concerning because it could indicate that the current feature set, we are using will not scale well if we were to expand our compound library. Peak shape may be a possible distinguisher. I will have to use a CNN on the intensities to see if they can be classified accordingly.

Benchmark¶

As previously discussed, our benchmark will be a true positive rate of greater or equal to 90% and a false positive rate less than or equal to 2%.

Methodology¶

Data preprocessing¶

I want to build a data structure that would allow me to easily switch between different algorithm prototypes. The data structure will be a dictionary of compound id to associated mass pairs. Each associated mass pair will have a lookup of their timestep intensity values that will be passed into its respective model. I do not need to pad the time steps because they are currently all the same. We will investigate later if smoothing the time step intensities has any impact. We will normalize the timesteps when passing them into the model. The normalization will work by taking the max intensity of all relevant mass pairs to a compound and dividing all intensities by that max value therefore making all intensities values between 1 and 0. This normalization allows us to retain the shape of our intensities while allowing us to scale the values down for faster learning.

Compound 0
Mass pairs [22, 23, 25, 26, 33, 34]
Compound 3
Mass pairs [20, 47, 49, 50]
Compound 4
Mass pairs [2, 11]
Compound 7
Mass pairs [16, 18, 27, 30, 46]
Compound 8
Mass pairs [16, 18, 39, 46]
Compound 10
Mass pairs [0, 3, 4, 19, 20, 22, 23]
Compound 13
Mass pairs [41, 42, 46]
Compound 14
Mass pairs [16, 18, 46]
Compound 15
Mass pairs [36, 39, 46]
Compound 18
Mass pairs [19, 20, 21, 40]
Compound 19
Mass pairs [7, 8]
Compound 21
Mass pairs [22, 23, 25, 26, 33, 34]
Compound 22
Mass pairs [35, 37, 38]

Implementation¶

I will build a temporal CNN using 1D convolution layers. I could build one using 2D layers, however, I think we could simplify the model by using just the intensities without absolute accurate time coordinates.

Model design: Our current data follows the format mass pairs to intensities or timesteps. Our model needs to be timesteps to mass pairs. For ease of use we allow the data to be passed in its original format and we will permute the data to reformat it for our needs. Next, we will follow the pattern of using 2 convolutional layers and then a pooling layer to extract and simplify features and the dataset. By using 2 iterations of this design there should be enough features to determine if there are any consistent patterns that allow for accurate classification. I finished the model with a dropout layer as to not overfit our data.

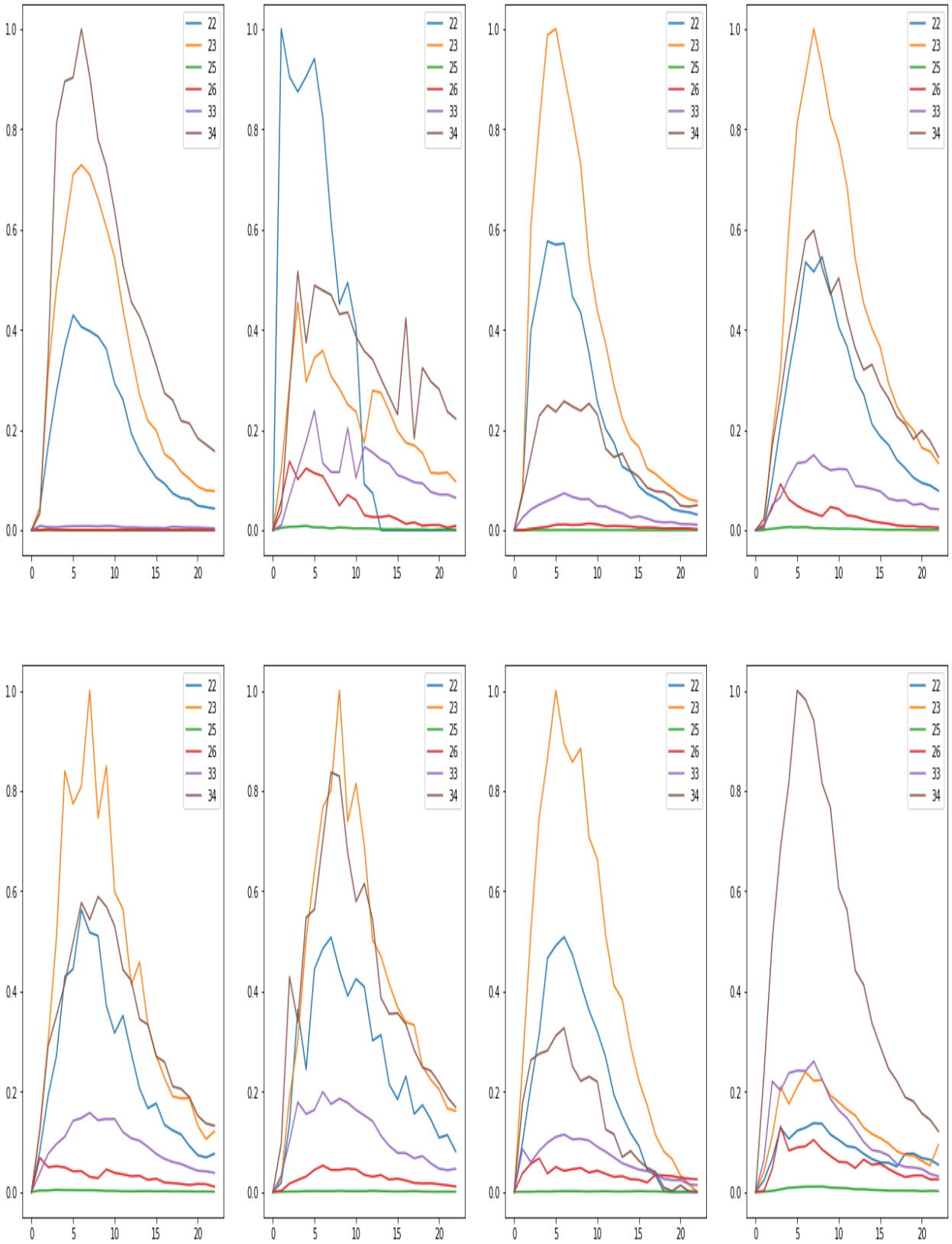
Since I want to have a model made per compound id our last layer will be a dense node of 1 with a sigmoid activation. If we wanted to have a multilabel model we would have the final layer be the number of classes and an activation function of softmax.

Next, create an algorithm that passes all associated mass pair intensities into the created model. We could create a multilabel model, but for experimentation purposes, I would like to create a dictionary of compound to model. I will compile each model using the Adam optimizer for updating the graph weights based on the training data and our loss function will be binary cross entropy because our output is theoretically binary. Our output is a probability that we will need to apply a threshold limit to determine if we should mark it as a true detection or no detection. We will use a ROC curve on the training data to determine the best threshold with the parameters of having less than 2% false positives. I will need to stack the associated mass pairs to intensities in order to pass them into my created model.

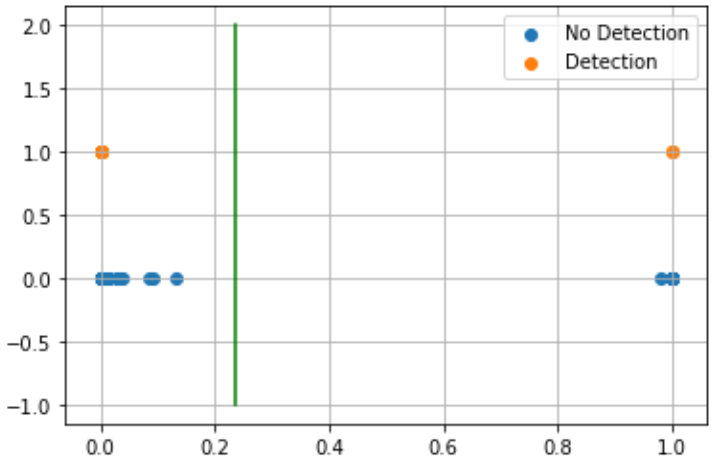
Abbreviated. All graphs are under external document CNN graphs

Compound ID 0

[22, 23, 25, 26, 33, 34]


22
23
25
26
33
34
0.0
0.2
0.4
0.6
0.8
1.0
0
5
10
15
20

Train on detected 61 non-detected 451
Train time in seconds: 5.907068490982056



AUC 0.9743011886154629
Predicted TPR 0.9508196721311475
Predicted FPR 0.0022172949002217295
Threshold 0.23654556274414062
----------End of training----------

Test on detected 14 non-detected 115
F0.5 92.86%

Confusion Matrix:
True Negative 99.13%
False Positive 0.87%
False Negative 7.14%
True Positive 92.86%

Classification Report

	precision	recall	f1-score	support
blank	0.991	0.991	0.991	115
detected	0.929	0.929	0.929	14
accuracy			0.984	129
macro avg	0.960	0.960	0.960	129
weighted avg	0.984	0.984	0.984	129

----------end of testing----------

Compounds	Tr-TPR	Tr-FPR	Threshold	ROC AUC Score	Test-TPR	Test-FPR
0	95.08%	0.22%	23.65%	97.43%	92.86%	0.87%
3	93.75%	0.81%	5.35%	96.47%	100.00%	3.97%
4	100.00%	0.44%	55.87%	99.78%	85.71%	0.00%
7	100.00%	1.19%	8.86%	99.40%	0.00%	1.56%
8	38.30%	1.51%	28.88%	68.40%	27.27%	1.69%
10	72.34%	1.94%	7.07%	85.20%	54.55%	3.39%
13	60.71%	1.54%	62.28%	79.59%	25.00%	1.71%
14	38.71%	1.66%	27.67%	68.52%	40.00%	4.03%
15	98.21%	1.75%	4.16%	98.23%	91.67%	4.27%
18	92.06%	1.78%	63.36%	95.14%	57.14%	6.09%
19	0.00%	0.00%	100.00%	50.00%	0.00%	0.00%
21	96.72%	2.00%	13.49%	97.36%	92.86%	0.87%
22	100.00%	0.67%	0.04%	99.66%	100.00%	0.00%

The above results have come out far better than I expected. The compounds that I am skeptical of are compound 7 and 19. Compound 19 has only trained on 3 true

detections and compound 7 only has 8 true samples. There probably needs to be more sample detections for this result to be accurate. I will ignore this result for now. By plotting one sample we can predict what models will do well. For example, compound ID 8 has 2 sets of intensities that should probably be removed from its association. I'm speaking of the red and orange lines because they are very noisy and do not have one clear peak. Let's review our test data before making any further decisions on how to refine our data and/or algorithm

Some challenges I ran into in implementation were how to normalize my data in order to create trainable and consistent signals. Normalizing and scaling will always be a problem for us especially if we must start quantifying how much of a compound we have detected. Our instruments vary in sensitivity by some percent, so it took some thought in deciding the best way to preprocess our data in a way that would allow our trained model to work across instruments. I ended up choosing to min-max scale our values based on the largest intensity in our window. This would scale all the values between 1 and 0 which would allow us to train a model faster and it would keep the integrity of the mass pair intensities in relation to each other. The only thing I worry about is having volatile intensities that change the look of our intensities. For example, if mass pair 46 has a huge intensity peak when it normally does not it will change the look of our signals for the compounds that use that mass pair. Now the signal that is usually the largest only reaches half the size it normally does because it is now not the highest peak. This constraint makes it very important when we select what mass pairs are important for each compound. It could make the difference between a 40% detection rate to a 90% detection rate.

Refinement¶

Compounds 0, 21, 22 currently meet our goal.

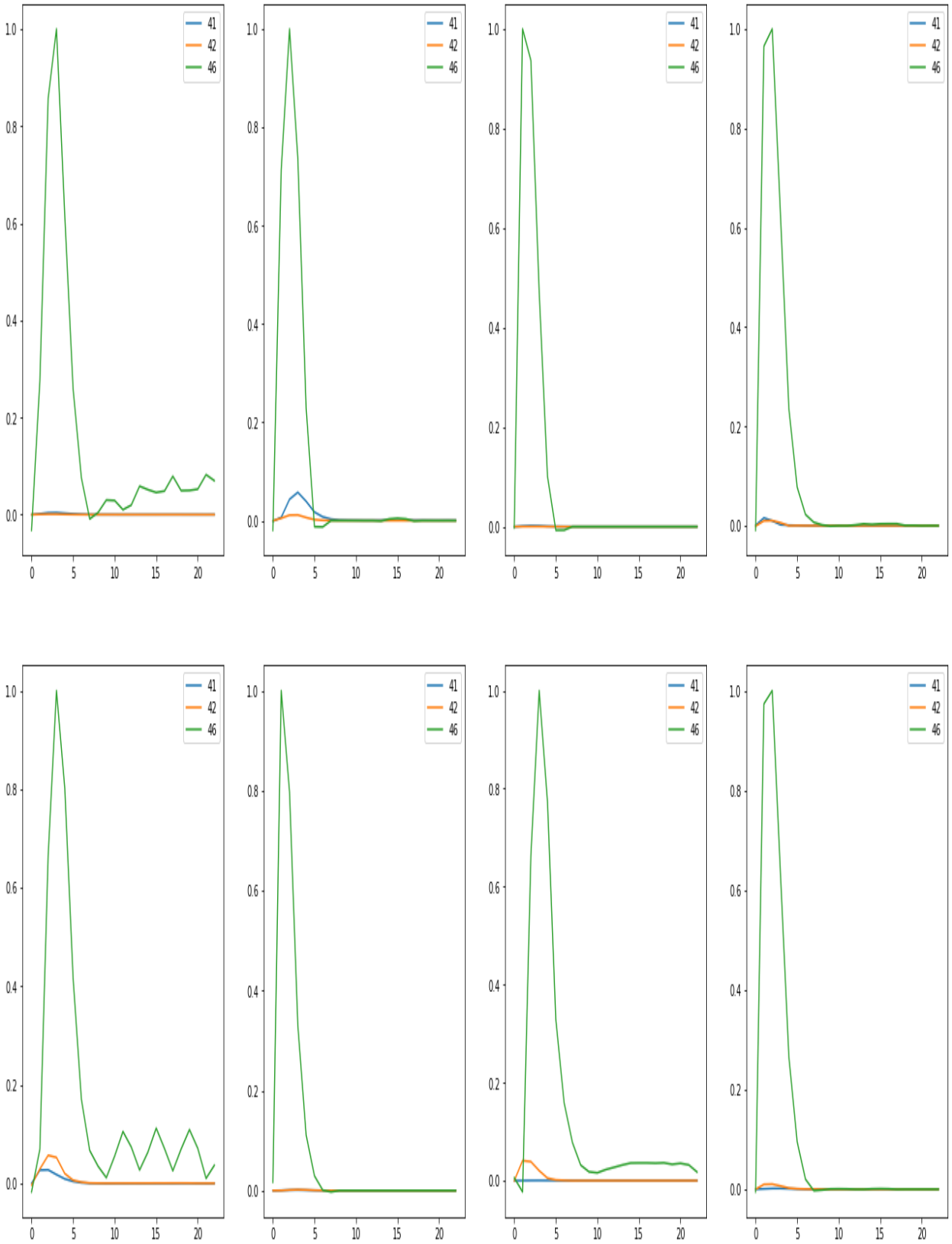
The rest need more data or further refinement to meet our goal.

Compound 7 and 19 do not have enough sample data so we will discard them for now.

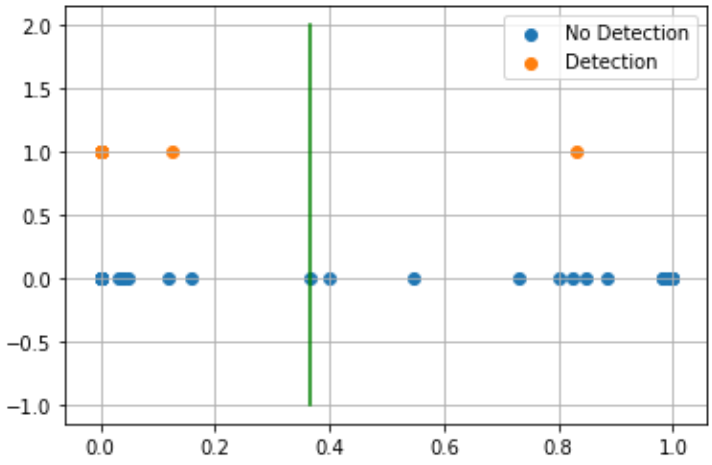
I will go through each one and see if some improvements can be made by removing some associated mass pair data that might be hurting our results. I also need to consider the noise of a signal. If the mass pair intensity is too noisy then I will not be able to learn the shape. I will probably need to apply some sort of noise filtering per mass pair intensity to achieve consistent results. Let's try filtering first.

Abbreviated. All graphs are under external document CNN graphs

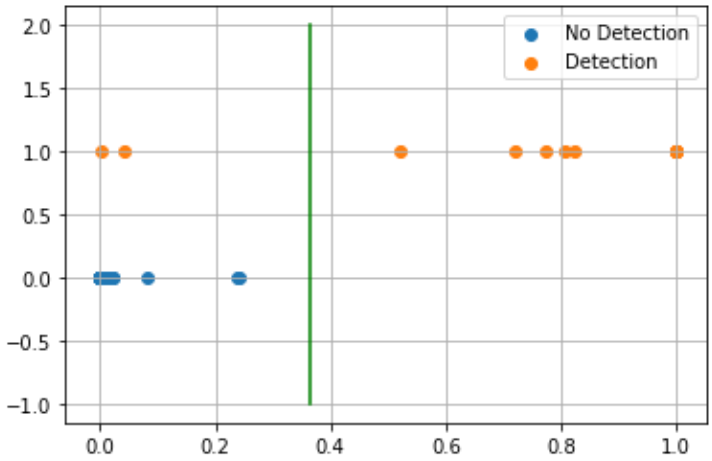
Compound ID 13
[41, 42, 46]


41
42
46
1.0
0.8
0.6
0.4
0.2
0.0
0
5
10
15
20

Train on detected 56 non-detected 456
Train time in seconds: 8.213072299957275



AUC 0.9956140350877193
Predicted TPR 1.0
Predicted FPR 0.008771929824561403
Threshold 0.36573362
----------End of training----------



Test on detected 12 non-detected 117
F0.5 96.15%

Confusion Matrix:
True Negative 100.00%
False Positive 0.00%
False Negative 16.67%
True Positive 83.33%

Classification Report

	precision	recall	f1-score	support
blank	0.983	1.000	0.992	117
detected	1.000	0.833	0.909	12
accuracy			0.984	129
macro avg	0.992	0.917	0.950	129
weighted avg	0.985	0.984	0.984	129

----------end of testing----------

Original

Compounds	Tr-TPR	Tr-FPR	Threshold	ROC AUC Score	Test-TPR	Test-FPR
0	95.08%	0.22%	23.65%	97.43%	92.86%	0.87%
3	93.75%	0.81%	5.35%	96.47%	100.00%	3.97%
4	100.00%	0.44%	55.87%	99.78%	85.71%	0.00%
7	100.00%	1.19%	8.86%	99.40%	0.00%	1.56%
8	38.30%	1.51%	28.88%	68.40%	27.27%	1.69%
10	72.34%	1.94%	7.07%	85.20%	54.55%	3.39%
13	60.71%	1.54%	62.28%	79.59%	25.00%	1.71%
14	38.71%	1.66%	27.67%	68.52%	40.00%	4.03%
15	98.21%	1.75%	4.16%	98.23%	91.67%	4.27%
18	92.06%	1.78%	63.36%	95.14%	57.14%	6.09%
19	0.00%	0.00%	100.00%	50.00%	0.00%	0.00%
21	96.72%	2.00%	13.49%	97.36%	92.86%	0.87%
22	100.00%	0.67%	0.04%	99.66%	100.00%	0.00%

Filtered

Compounds	Tr-TPR	Tr-FPR	Threshold	ROC AUC Score	Test-TPR	Test-FPR
0	95.08%	1.77%	15.14%	96.65%	92.86%	0.87%
3	93.75%	0.40%	4.26%	96.67%	100.00%	0.79%
4	96.72%	1.11%	1.47%	97.81%	78.57%	2.61%
7	100.00%	1.79%	15.97%	99.11%	0.00%	1.56%
8	61.70%	1.51%	37.70%	80.10%	27.27%	3.39%
10	95.74%	0.65%	13.21%	97.55%	72.73%	2.54%
13	100.00%	0.88%	36.57%	99.56%	83.33%	0.00%
14	35.48%	1.46%	16.94%	67.01%	20.00%	5.65%
15	94.64%	0.00%	2.84%	97.32%	91.67%	0.85%
18	76.19%	1.78%	27.57%	87.20%	57.14%	3.48%
19	0.00%	0.00%	100.00%	50.00%	0.00%	0.00%
21	95.08%	0.67%	50.97%	97.21%	92.86%	0.87%
22	100.00%	0.00%	87.79%	100.00%	100.00%	0.00%

Interesting, compounds 4, 10, 13 and 21 do worse and compounds 8, 14, and 15 do better. Mass pairs 16, 18, 46 tend to be very volatile with lots of apparent noise. If I just apply filtering to those compounds, I would be interested to see what happens.

Abbreviated. All graphs are under external document CNN graphs

----------end of testing----------

Compounds	Tr-TPR	Tr-FPR	Threshold	ROC AUC Score	Test-TPR	Test-FPR
8	48.94%	1.94%	50.07%	73.50%	27.27%	2.54%
13	83.93%	1.97%	64.48%	90.98%	83.33%	3.42%
15	98.21%	1.32%	4.59%	98.45%	91.67%	3.42%

Compound 8 was the only one that got better. I wonder if I remove mass pairs that are adding noise to our graphs if that will help stabilize our results. Also, I will try adding mass pairs that are shared across compounds to see if that can help differentiate compounds. For example, compound 8 and 15 share mass pairs. After adjusting mass pairs, it may be worth revisiting mass pair filtering.

Abbreviated. All graphs are under external document CNN graphs

----------end of testing----------

Compounds	Tr-TPR	Tr-FPR	Threshold	ROC AUC Score	Test-TPR	Test-FPR
0	93.44%	1.55%	2.44%	95.95%	78.57%	2.61%
3	93.75%	0.40%	41.39%	96.67%	100.00%	0.00%
4	100.00%	0.44%	38.52%	99.78%	85.71%	0.00%
8	63.83%	1.94%	28.37%	80.95%	54.55%	4.24%
10	100.00%	0.00%	86.57%	100.00%	100.00%	0.00%
13	98.21%	0.88%	58.66%	98.67%	83.33%	3.42%
14	22.58%	1.87%	16.00%	60.35%	40.00%	2.42%
15	100.00%	0.44%	93.78%	99.78%	83.33%	0.00%
18	84.13%	1.78%	41.21%	91.17%	64.29%	4.35%
21	80.33%	2.00%	51.31%	89.17%	85.71%	2.61%
22	100.00%	0.00%	98.79%	100.00%	100.00%	0.00%

Compound 14 seems to be very noisy and needs many more samples in order to try to distinguish when we have a detection or not. I will need to consider removing this compound for now. Some of the compounds do better others do worse. I will try adding filtering to see if we can get even better results.

Abbreviated. All graphs are under external document CNN graphs

Original:

Compounds	Tr-TPR	Tr-FPR	Threshold	ROC AUC Score	Test-TPR	Test-FPR
0	95.08%	0.22%	23.65%	97.43%	92.86%	0.87%
3	93.75%	0.81%	5.35%	96.47%	100.00%	3.97%
4	100.00%	0.44%	55.87%	99.78%	85.71%	0.00%
7	100.00%	1.19%	8.86%	99.40%	0.00%	1.56%
8	38.30%	1.51%	28.88%	68.40%	27.27%	1.69%
10	72.34%	1.94%	7.07%	85.20%	54.55%	3.39%
13	60.71%	1.54%	62.28%	79.59%	25.00%	1.71%
14	38.71%	1.66%	27.67%	68.52%	40.00%	4.03%
15	98.21%	1.75%	4.16%	98.23%	91.67%	4.27%
18	92.06%	1.78%	63.36%	95.14%	57.14%	6.09%
19	0.00%	0.00%	100.00%	50.00%	0.00%	0.00%
21	96.72%	2.00%	13.49%	97.36%	92.86%	0.87%
22	100.00%	0.67%	0.04%	99.66%	100.00%	0.00%

Filtered:

Compounds	Tr-TPR	Tr-FPR	Threshold	ROC AUC Score	Test-TPR	Test-FPR
0	95.08%	1.77%	15.14%	96.65%	92.86%	0.87%
3	93.75%	0.40%	4.26%	96.67%	100.00%	0.79%
4	96.72%	1.11%	1.47%	97.81%	78.57%	2.61%
7	100.00%	1.79%	15.97%	99.11%	0.00%	1.56%
8	61.70%	1.51%	37.70%	80.10%	27.27%	3.39%
10	95.74%	0.65%	13.21%	97.55%	72.73%	2.54%
13	100.00%	0.88%	36.57%	99.56%	83.33%	0.00%
14	35.48%	1.46%	16.94%	67.01%	20.00%	5.65%
15	94.64%	0.00%	2.84%	97.32%	91.67%	0.85%
18	76.19%	1.78%	27.57%	87.20%	57.14%	3.48%
19	0.00%	0.00%	100.00%	50.00%	0.00%	0.00%
21	95.08%	0.67%	50.97%	97.21%	92.86%	0.87%
22	100.00%	0.00%	87.79%	100.00%	100.00%	0.00%

No Filtering Adjusted:

Compounds	Tr-TPR	Tr-FPR	Threshold	ROC AUC Score	Test-TPR	Test-FPR
0	93.44%	1.55%	2.44%	95.95%	78.57%	2.61%
3	93.75%	0.40%	41.39%	96.67%	100.00%	0.00%
4	100.00%	0.44%	38.52%	99.78%	85.71%	0.00%
8	63.83%	1.94%	28.37%	80.95%	54.55%	4.24%
10	100.00%	0.00%	86.57%	100.00%	100.00%	0.00%
13	98.21%	0.88%	58.66%	98.67%	83.33%	3.42%
14	22.58%	1.87%	16.00%	60.35%	40.00%	2.42%
15	100.00%	0.44%	93.78%	99.78%	83.33%	0.00%
18	84.13%	1.78%	41.21%	91.17%	64.29%	4.35%
21	80.33%	2.00%	51.31%	89.17%	85.71%	2.61%
22	100.00%	0.00%	98.79%	100.00%	100.00%	0.00%

Filtering Adjusted

Compounds	Tr-TPR	Tr-FPR	Threshold	ROC AUC Score	Test-TPR	Test-FPR
0	90.16%	1.11%	62.86%	94.53%	85.71%	0.00%
3	93.75%	0.81%	35.64%	96.47%	100.00%	0.00%
4	98.36%	1.33%	19.52%	98.52%	85.71%	1.74%
8	93.62%	0.86%	16.07%	96.38%	81.82%	1.69%
10	100.00%	0.00%	87.70%	100.00%	100.00%	0.00%
13	98.21%	1.10%	54.00%	98.56%	75.00%	5.98%
14	12.90%	1.66%	16.43%	55.62%	40.00%	2.42%
15	100.00%	0.22%	63.91%	99.89%	91.67%	0.00%
18	76.19%	1.78%	31.68%	87.20%	57.14%	4.35%
21	83.61%	1.55%	32.26%	91.03%	78.57%	2.61%
22	100.00%	0.00%	90.65%	100.00%	100.00%	0.00%

The results are mixed. In some cases, the results are better and in others worse. For results that are the same as the original I am inclined to take the adjusted because it creates a simpler model. Most compounds perform better or equal with intensity smoothing/filtering. The adjusted/filtered mass pairs can do much worse meaning the removed or filtered mass pairs have some important features that distinguish them as detections. We will need to adjust our filtering and adjusted mass pairs on a compound basis.

Results¶

Model Evaluation and Validation¶

We have not exactly reached our intended goal, but we have achieved some good results. I would like to see if I could extend my neural network to add additional features that may increase my prediction accuracy. This will allow me to evaluate how good our model is.

Abbreviated. All graphs are under external document CNN graphs

----------end of testing----------

Compounds	Tr-TPR	Tr-FPR	Threshold	ROC AUC Score	Test-TPR	Test-FPR
0	0.00%	0.00%	146.25%	50.00%	0.00%	0.00%
3	0.00%	0.00%	146.50%	50.00%	0.00%	0.00%
4	0.00%	0.00%	130.43%	50.00%	0.00%	0.00%
7	100.00%	0.60%	12.61%	99.70%	0.00%	1.56%
8	0.00%	0.00%	119.21%	50.00%	0.00%	0.00%
10	2.13%	1.51%	17.62%	50.31%	0.00%	0.85%
13	0.00%	0.00%	138.59%	50.00%	0.00%	0.00%
14	0.00%	0.00%	146.50%	50.00%	0.00%	0.00%
15	0.00%	0.00%	133.90%	50.00%	0.00%	0.00%
18	12.70%	1.34%	16.48%	55.68%	0.00%	1.74%
19	100.00%	0.00%	36.72%	100.00%	0.00%	0.00%
21	95.08%	0.89%	51.95%	97.10%	92.86%	0.87%
22	0.00%	0.00%	145.96%	50.00%	0.00%	0.00%

Attempting to incorporate width and position into the network has resulted in much worse results. It seems that the best option is to go ahead with the previously refined model. I do believe that my refined model could do even better if we had a larger sample set.

Compound 0
Substrate BG1 # of samples 0
Substrate BG4 # of samples 0
Substrate CB4 # of samples 0
Substrate Med4 # of samples 0
Substrate None # of samples 75
Substrate Per3 # of samples 0
Substrate Per5 # of samples 0
Substrate Teflon # of samples 0
Compound 3
Substrate BG1 # of samples 0
Substrate BG4 # of samples 0
Substrate CB4 # of samples 0
Substrate Med4 # of samples 0
Substrate None # of samples 3
Substrate Per3 # of samples 0
Substrate Per5 # of samples 11
Substrate Teflon # of samples 5
Compound 4
Substrate BG1 # of samples 0
Substrate BG4 # of samples 0
Substrate CB4 # of samples 0
Substrate Med4 # of samples 0
Substrate None # of samples 75
Substrate Per3 # of samples 0
Substrate Per5 # of samples 0
Substrate Teflon # of samples 0
Compound 7
Substrate BG1 # of samples 0
Substrate BG4 # of samples 0
Substrate CB4 # of samples 0
Substrate Med4 # of samples 0
Substrate None # of samples 9
Substrate Per3 # of samples 0
Substrate Per5 # of samples 0
Substrate Teflon # of samples 0
Compound 8
Substrate BG1 # of samples 9
Substrate BG4 # of samples 8
Substrate CB4 # of samples 8
Substrate Med4 # of samples 9
Substrate None # of samples 10
Substrate Per3 # of samples 7
Substrate Per5 # of samples 7

Substrate Teflon # of samples 0
Compound 10
Substrate BG1 # of samples 8
Substrate BG4 # of samples 9
Substrate CB4 # of samples 5
Substrate Med4 # of samples 10
Substrate None # of samples 9
Substrate Per3 # of samples 8
Substrate Per5 # of samples 9
Substrate Teflon # of samples 0
Compound 13
Substrate BG1 # of samples 10
Substrate BG4 # of samples 10
Substrate CB4 # of samples 10
Substrate Med4 # of samples 10
Substrate None # of samples 8
Substrate Per3 # of samples 10
Substrate Per5 # of samples 10
Substrate Teflon # of samples 0
Compound 14
Substrate BG1 # of samples 0
Substrate BG4 # of samples 0
Substrate CB4 # of samples 0
Substrate Med4 # of samples 0
Substrate None # of samples 18
Substrate Per3 # of samples 0
Substrate Per5 # of samples 14
Substrate Teflon # of samples 4
Compound 15
Substrate BG1 # of samples 10
Substrate BG4 # of samples 10
Substrate CB4 # of samples 10
Substrate Med4 # of samples 10
Substrate None # of samples 10
Substrate Per3 # of samples 9
Substrate Per5 # of samples 9
Substrate Teflon # of samples 0
Compound 18
Substrate BG1 # of samples 10
Substrate BG4 # of samples 10
Substrate CB4 # of samples 9
Substrate Med4 # of samples 10
Substrate None # of samples 19
Substrate Per3 # of samples 10

Substrate Per5 # of samples 9
Substrate Teflon # of samples 0
Compound 19
Substrate BG1 # of samples 0
Substrate BG4 # of samples 0
Substrate CB4 # of samples 0
Substrate Med4 # of samples 0
Substrate None # of samples 4
Substrate Per3 # of samples 0
Substrate Per5 # of samples 0
Substrate Teflon # of samples 0
Compound 21
Substrate BG1 # of samples 0
Substrate BG4 # of samples 0
Substrate CB4 # of samples 0
Substrate Med4 # of samples 0
Substrate None # of samples 75
Substrate Per3 # of samples 0
Substrate Per5 # of samples 0
Substrate Teflon # of samples 0
Compound 22
Substrate BG1 # of samples 10
Substrate BG4 # of samples 10
Substrate CB4 # of samples 10
Substrate Med4 # of samples 9
Substrate None # of samples 20
Substrate Per3 # of samples 10
Substrate Per5 # of samples 10
Substrate Teflon # of samples 0

Above is a list of all the detection samples broken down by substrate. We know that depending on the substrate and the tested compound the shapes of the lines do change. According to this data we should perform best on the compounds that have a high number of samples within each substrate. For example, compounds 0 and 4 should perform well because they have samples only within the No substrate category therefore, they will perform well because there are many good samples to train, validate and test on. Our model struggled on compound 8 giving an average accuracy between ~27-54%. By looking at the distribution of data we can understand why. This might be the kind of compound that needs more than 10 samples per category to train a good model. Depending on how the data is shuffled for train, validate, test our model may not do very well. Same reasoning goes for compounds 13 and 18.

Justification¶

Comp.	TPR	FPR	Thresh.	ROC Score	Test-TPR	Test-FPR	Type
0	96.72%	2.00%	6.85%	97.36%	92.86%	0.87%	Orig.
3	93.75%	1.21%	46.94%	96.27%	100.00%	0.79%	Non-fil. Adj.
4	100.00%	1.11%	7.73%	99.45%	92.86%	0.87%	Non-fil. Adj.
7	100.00%	1.39%	8.00%	99.31%	0.00%	0.78%	Orig.
8	63.83%	1.29%	28.02%	81.27%	54.55%	1.69%	Filt. Adj.
10	100.00%	0.00%	91.87%	100.00%	90.91%	0.00%	Filt. Adj.
13	100.00%	0.88%	46.77%	99.56%	83.33%	0.85%	Fil.
14	41.94%	1.87%	21.62%	70.03%	40.00%	4.03%	Orig.
15	100.00%	0.00%	48.92%	100.00%	91.67%	0.00%	Filt. Adj.
18	84.13%	1.56%	33.73%	91.28%	57.14%	4.35%	Orig.
19	100.00%	0.00%	33.06%	100.00%	0.00%	0.00%	Fil.
21	95.08%	0.22%	28.07%	97.43%	92.86%	0.87%	Orig.
22	100.00%	0.00%	75.91%	100.00%	100.00%	0.00%	Filt. Adj.

If we exclude the results of compounds, 7, 14 and 19, six out of the remaining 10 compounds meet our benchmark of 90% or higher true positive rate and have a false positive rate of less than 2%. Compounds 4 and 13 are close and may just need more sample data to reach our benchmark. I would have thought compound 4 and 18 would have exceeded our benchmark as well, but for some reason they have not. One thing I think I would change is making the samples that have the detection of [21, 0, 18, 4] compounds be a separate model rather than split up into their respective models. This kind of sample is known as a confidence check and is a premade solution that is used to verify the system is calibrated for use. I have noticed that the shapes of our mass pair lines do slightly change when tested within a confidence check and by themselves. This can be observed in compounds 18 and 21. I think perhaps that both these compounds would perform even better without the addition of the confidence check data. By creating a confidence check model there will probably be more features that our CNN can use to better determine a detection. As previously stated, the compounds that performed poorly did not have enough sample data to train on. Looking at how well some compounds did do I think it is a good indicator that the model will work well. With some additional samples and refinement, I could meet my intended benchmark.

I think the models that I have created are trustworthy. I have trained on 60% of my data, validated on 20% of my data and tested on the final 20% of my data. The test results you see are the result of unseen data by my models. There is some robustness to the models as well. If you look at a subset of our sample data by compound, you can see patterns however they are not the same. Filtering helps in some cases where the CNN cannot account for the variability in the signal. The models are quite good at learning different signal patterns by varying substrate granted the right amount of sample data is provided. Models for compounds 10 and 15 validate this statement. Our designed

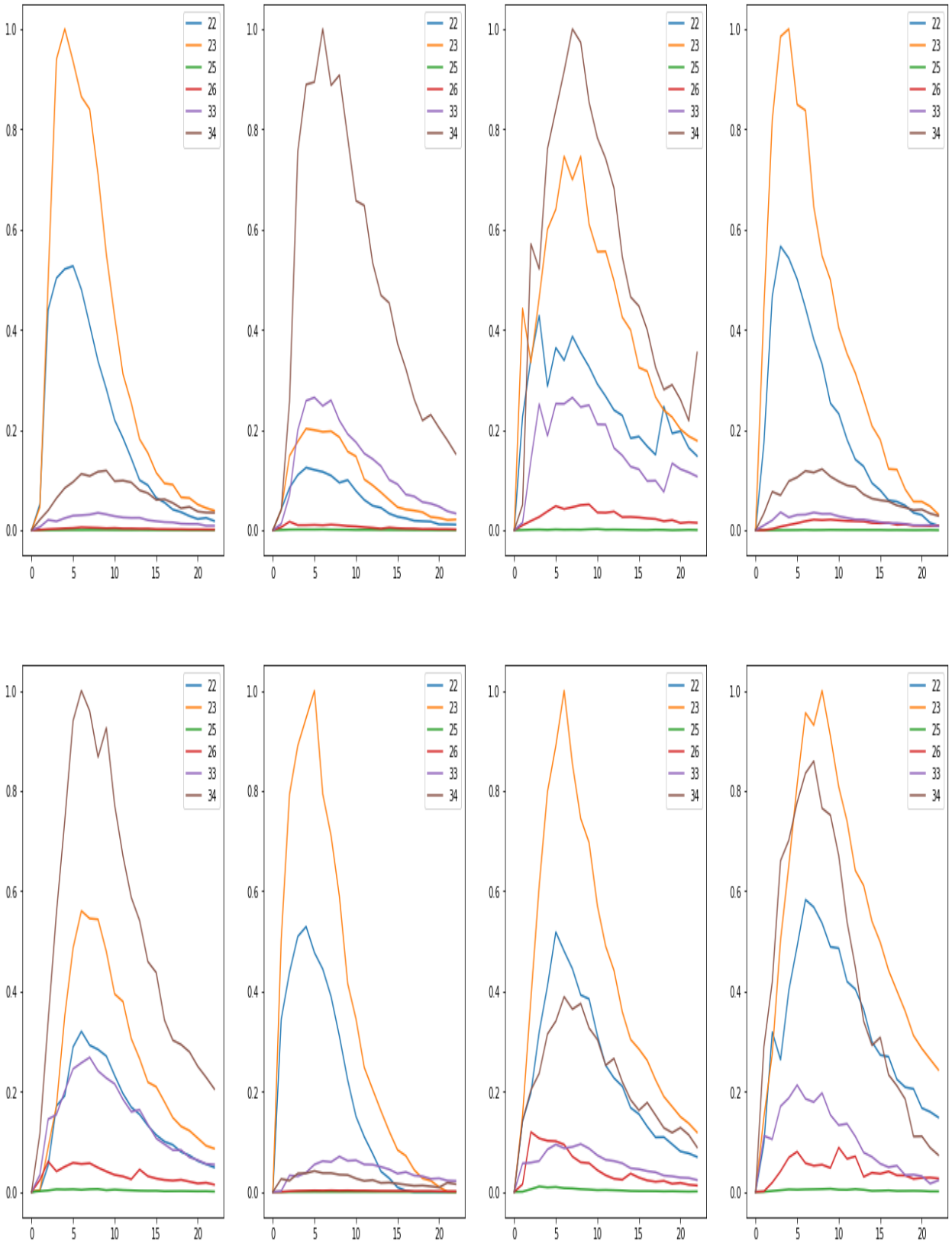
models are quite robust given the correct amount of training data and the right amount of filtering and mass pair selection.

Conclusion¶

Free form Visualization¶

Abbreviated. All graphs are under external document CNN graphs

Compound 0
Substrate None


22
23
25
26
33
34
0.0
0.2
0.4
0.6
0.8
1.0
0
5
10
15
20

Above is a visualization of what the graphs look like per compound per relevant mass pair. As you can see in some cases the graphs look very similar. In other cases, you can see that there might be a mass pair line that is very noisy and would be better off removed. In most cases the mass pair 46 is very noisy (compound 8) and in other cases it has a consistent look (compound 13). It seems we should further explore if filtering mass pair lines by compound would result in even better results. Because of the consistency we see in the data, we can also see why a CNN performs so well on the dataset.

Reflection¶

Before starting this project, our scientists would scratch their heads and say "I don't understand why our algorithms are not detecting the compounds. If I can see it, the algorithms should be able to detect it." We used to use the flawed approach that we assessed in our exploratory data analysis section. This was where we tried to find a peak and characterize it using width, height, position, and area. This was flawed because the approach does not account for noise very well and because of that creates all kinds of problem, such as, inaccurate peak characterization and poor peak selection when there are multiple peaks to choose from.

After exploring the data, I knew I would have to use a CNN to get the best results. The first challenge I ran into was deciding how many hidden nodes should I use for my CNN. There is a tradeoff between training speed and amount learned. Currently there is no rule of thumb or feedback from a trained model on how many nodes you will need or are using other than with experimentation. I have settled on my current model purely by doing a manual parameter grid search and using my results and training time as a metric to find the optimal model. Even if I automated it with a grid search, I still find it troubling that I must basically use a trial and error approach of determining the optimal model.

One area I found interesting was when I was trying to assess how filtering would affect the detection outcome or my CNN. I really wanted to use a grid search starting at 5 and going to 23 in increments of 2 or 4, but after implementing and testing, the grid search took forever, and I was getting mixed results. Using my GPU, it would take an hour and using my CPU it would theoretically take 5 hours. I realize I am complaining about the time when there are models that take a day or more to train, but it made me realize that machine learning is not a fast process. If you worked in a fast pace environment, you could struggle to keep up purely because most of the time you are waiting around for trial and error procedures to finish so that you can assess the results.

By using a CNN, we can learn the characteristics of our mass pair lines to better decide whether we have a detected compound or just chemical noise. Even with a CNN we still must account for noisy lines that can prevent the CNN from working well, but it is much easier to filter or remove a line to see how it impacts the CNNs learning. At least now, we can meet our scientist needs where our detection capabilities can match their detection abilities.

Improvement¶

Some improvements I have alluded to all revolve around parameterized grid searching. I could add grid searching to the parameters in the model to find the optimum model for results and train time. I could add grid searching on filtering to find the best filtering on a compound to mass pair basis. From the results of the filtering and/ or mass pair removal I could try to come up with an algorithm in the future that determines whether to remove, filter, or do nothing to the selected mass pair intensities. This would greatly increase the learning of our models especially when adding new compounds to our detection library.