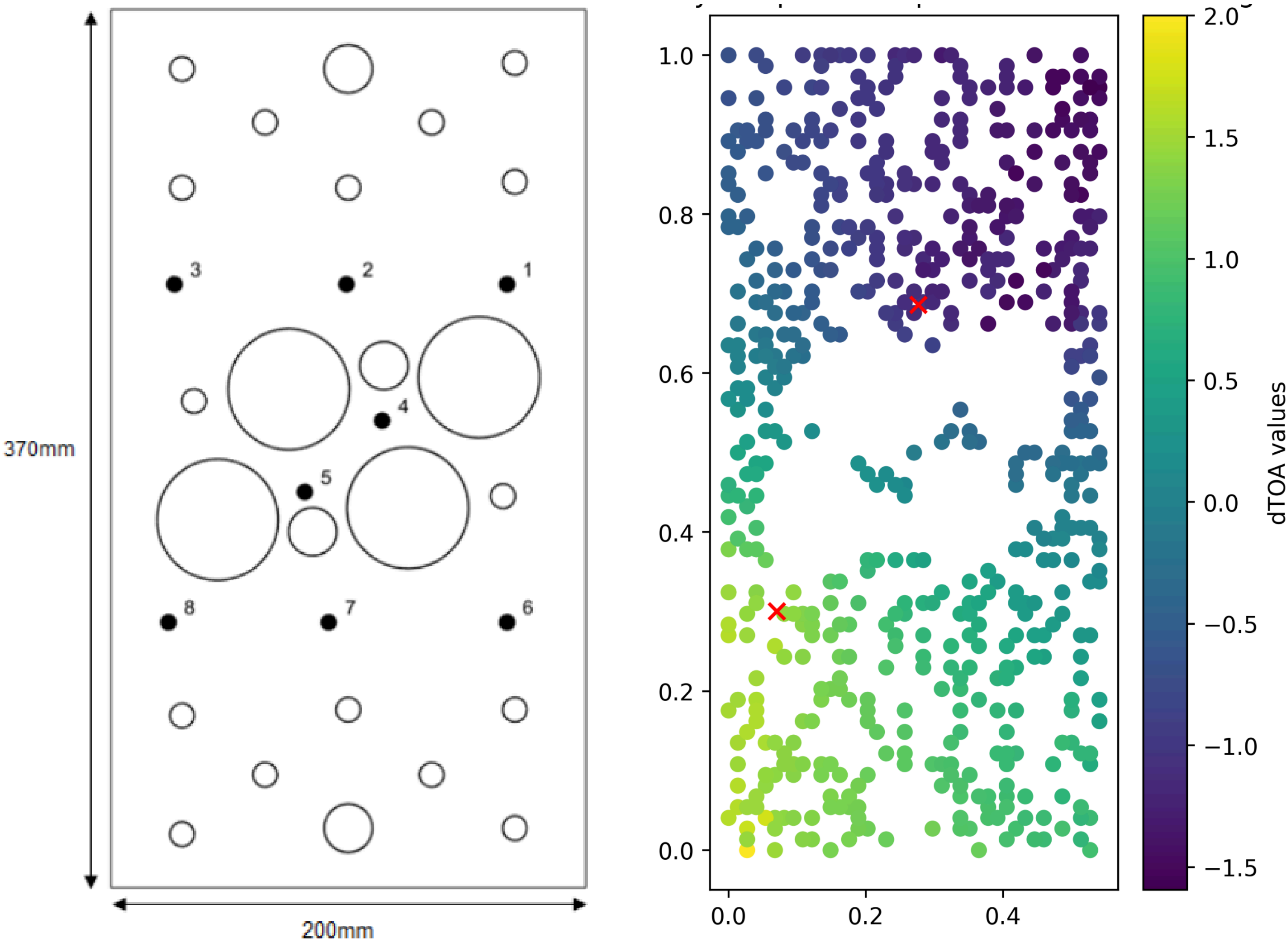
MODEL CARD

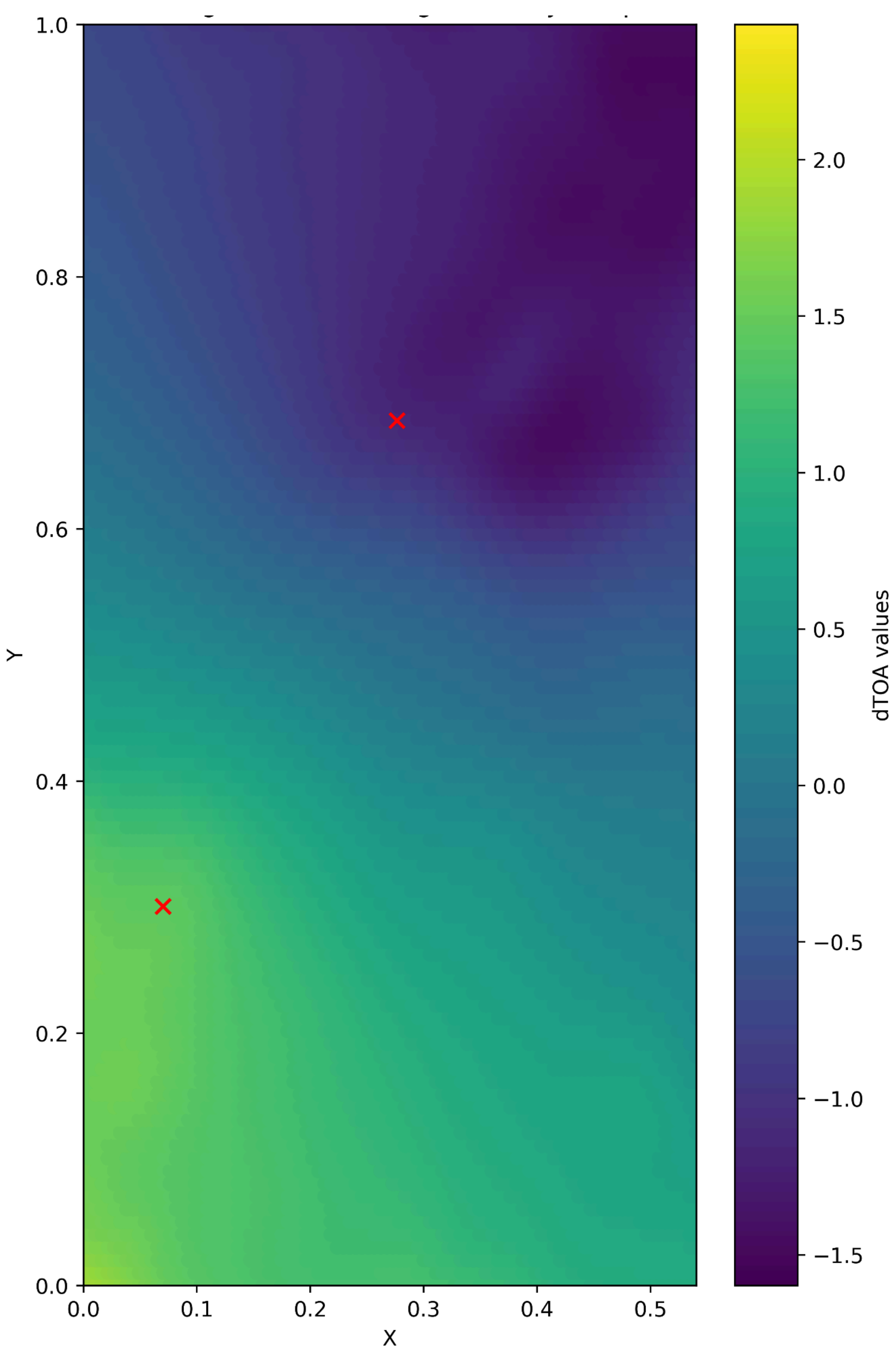
<u>A Gaussian approach to Damage Localisation</u>

Model description

This is a forward model that has been designed to localise damage within a complex structure by using a learnt relationship between acoustic emission source location coordinates (model inputs) and difference-in-time-of-arrival (dTOA) values for a given sensor pair, across the test-piece.



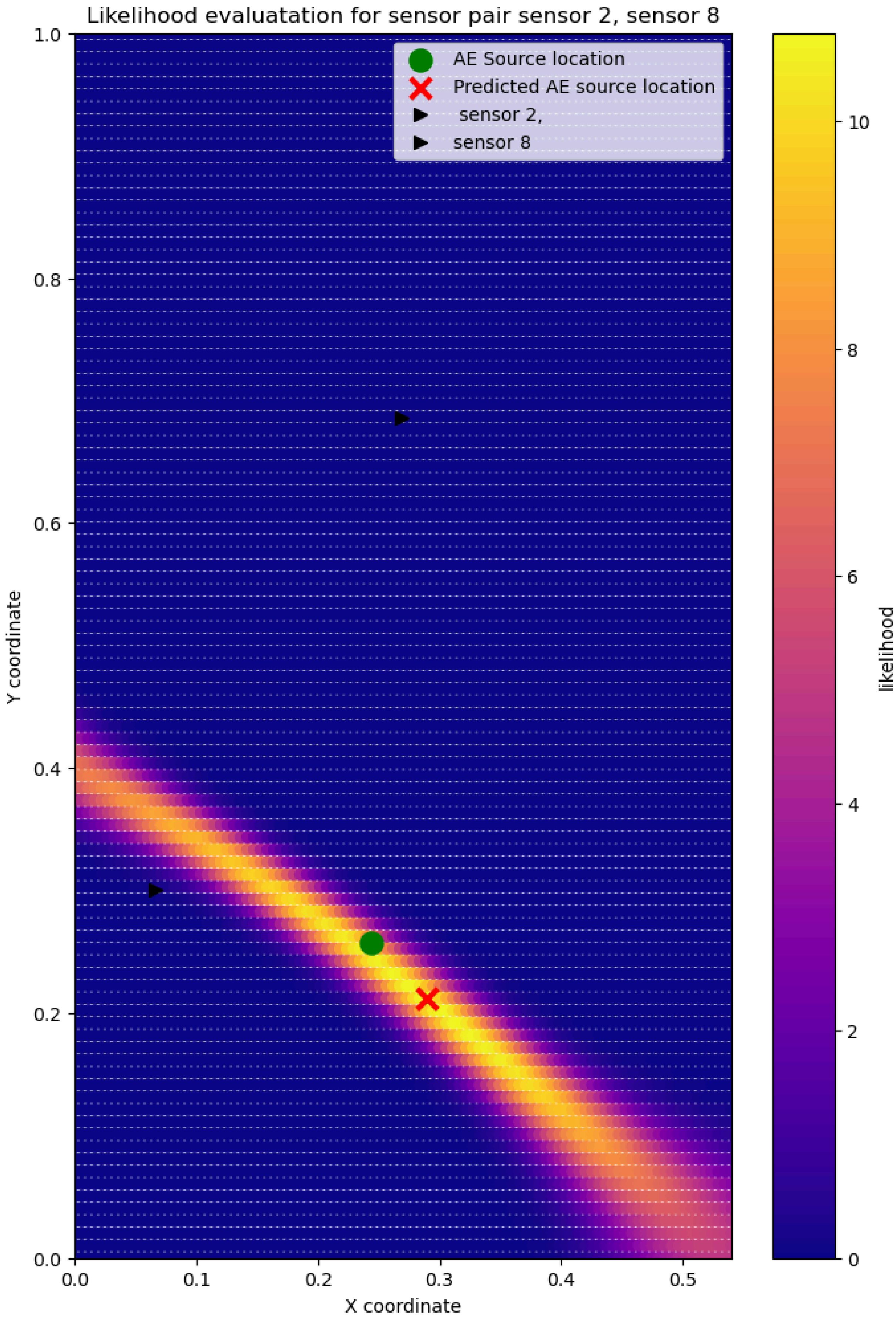
The image on the left represents the original test-piece the dataset was obtained from, and the image on the right represents the model training data used. Examining the image on the right, each dot represents an acoustic emission event. The (X,Y) coordinates of these dots served as the **<u>model input</u>**, whilst the colour, the dTOA value, was the **<u>model output</u>**. The two red crosses represent the exact pair of sensors this data pertains to (sensor pair 2-8).



After using Gaussian Process regression to learn the relationship between the coordinate inputs and the dTOA outputs, the model is then able to make a much more dense prediction of dTOA values across the test-piece.

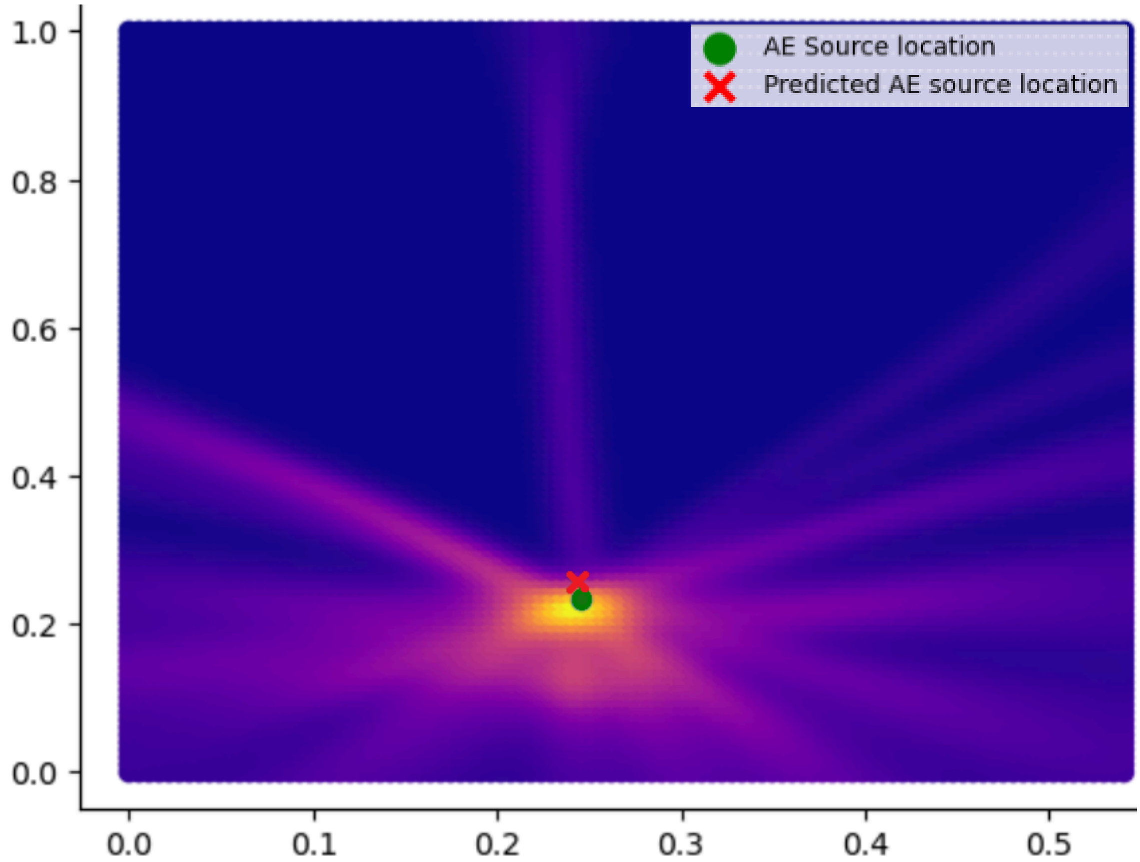
This increases the accuracy of the likelihood assessment without needing to spend more time experimentally obtaining a larger dataset.

After this grid is generated, a prediction of the source likelihood location from an unseen dTOA value can be made, by comparing the unseen value dTOA value with the values across the coordinates in this dense grid.



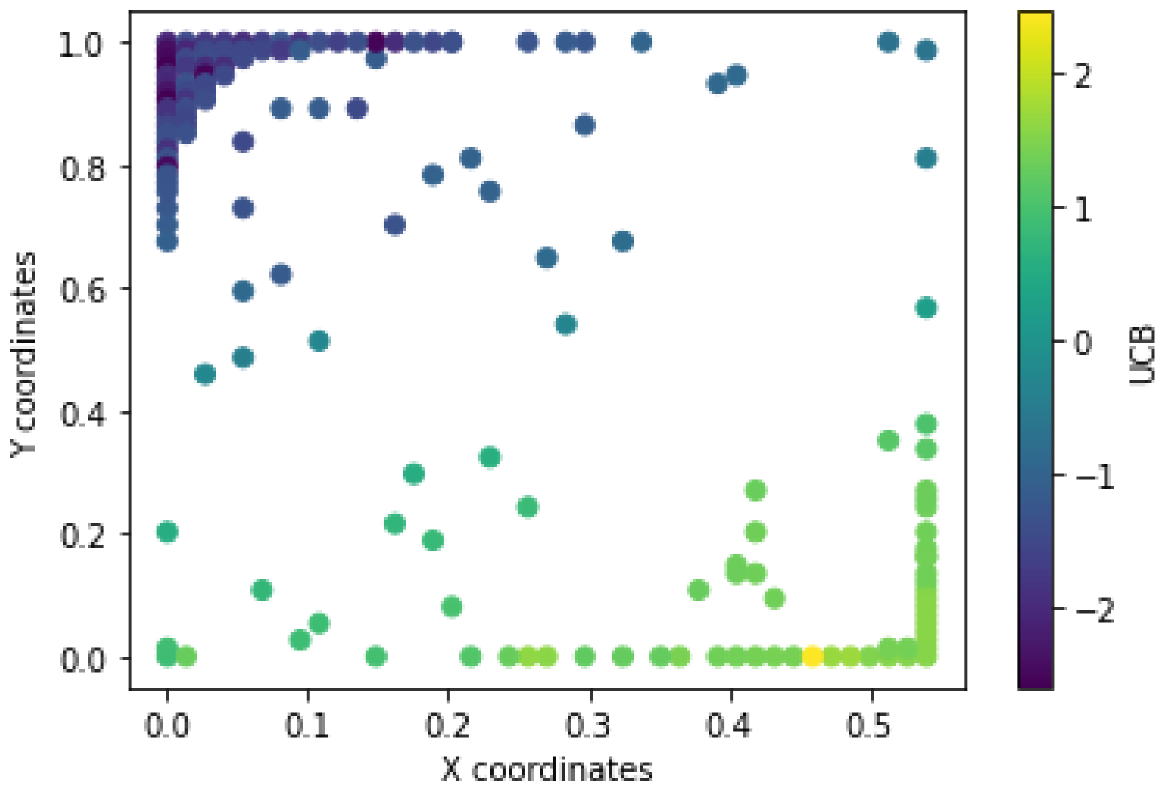
Likelihood evaluatation for sensor pair sensor 2, sensor 8
AE Source location
Predicted AE source location
sensor 2,
sensor 8
Y coordinate
X coordinate
likelihood
0.0
0.2
0.4
0.6
0.8
1.0
0.1
0.3
0.5
0
2
4
6
8
10

Doing so returns a graph like this one. Note that the source location for a singular sensor pair is ambiguous along a line, where the dTOA values between the two sensors will be identical.



By training models that use the data from each sensor pair, an exact source location can be triangulated by marginalising and summing the likelihoods for each location across the grid. That returns the above graph.

Lastly, to increase the model's utility, I implemented a Bayesian Optimisation sampling algorithm, to generate graphs to identify 'rich' training regions. This would inform engineers for future machine learning project where in the structure data collection would be most beneficial, based on regions of high uncertainty and regions which offer substantial model improvement. This was done using the Upper Confidence Bound acquisition function, and iterating the exploration parameter to obtain the minimum negative log marginal likelihood.



Graph identifying training rich regions, generated using bayesian optimisation

Performance

To determine this model's performance, I used the root mean square error, examining the difference between the predicted coordinates and the true coordinates. The coordinates assessed were randomly drawn from the test set, the subset of data that had not been used to train the model.

True AE source coordinates		Predicted AE source coordinates	
X	Y	X	Y
0.189189	0.824324	0.19656	0.79798
0.364865	0.297297	0.37674	0.282828
0.364865	0.283784	0.37128	0.282828
0.067568	0.486486	0.08736	0.474747
0.175676	0.067568	0.17472	0.060606
0.5	0.662162	0.46965	0.636364
0.378378	0.243243	0.3822	0.242424

Examining this dataset of results returns a mean squared error of 3.35%, which indicates an excellent model fit, however, it is not without limitations.

Limitations
This model can be very computationally expensive. This made it difficult to perform a large number of tests, which can result in a skewed performance metric. Secondly, due to the computational costs associated with generating a dense grid of points, only a finite number of coordinates can be assessed. Depending on the true value of the coordinate, it may be hard to predict it accurately if the grid is not dense enough, however, it is hard to justify the computational cost of dTOA value predictions across over 10000 points for a single model.

Trade-offs
Some sensor pair models perform much better than other models. I believe this may be related to where the sensors are located, as some are surrounded by more geometrical complexities (sensors 4 and 5) than others. This may complicate the learning process.

Secondly, the process is computationally intensive. The image of the summed likelihood above incorporates predictions from over 20 models. Whilst it shows excellent accuracy, it is unlikely that this is feasible to apply in real-time. The demonstrative program only uses the models from 4 sensor pairs, for reduced computational costs. This does, however, cause the prediction accuracy to suffer.