Parameter combinations used for the experiment

```
default_params

params_one = { "random_state": 33, "activation": "relu", "hidden_layer_sizes": (12, 24, 48, 96), "learning_rate_init": 0.001, "batch_size": 20 }

params_two = { "random_state": 33, "activation": "relu", "hidden_layer_sizes": (12, 24, 48, 96), "learning_rate_init": 0.001, "batch_size": 50 }
```

Other Model 1

This model was intended to test the performance of the model on the given dataset without applying any feature engineering or transformation techniques. This is to understand the impact of some of the hyper params on the basic model.

Features used: All features of the given data set were used. **Data split:** 70% of data is used for training, with 10% of validation and 20% for testing. **Evaluation metrics:** RMSE and MAE scores were used to evaluate the performance of the model. **Observations:**

Model Params	Train (rmse)	Validate (rmse)	Test (rmse)	Evaluate (rmse)
default_params	0.010235990698238237	0.01048744262284605	0.009904215868987822	0.010028905979234757
params_one	0.006932223912952257	0.006729120103675562	0.006602403372255023	0.006681047017901092
params_two	0.007888354736637352	0.007768264985211542	0.007381068902074671	0.007521259696380296

Other Model 2

This model was intended to test the performance of the model, after feature engineering and standardisation.

Features used: The candle positions and robot positions were used to calculate the euclidean distance between each candle and robot, which were used along with collision and orientation of robot.
Data split: 70% of data is used for training, with 10% of validation and 20% for testing. **Evaluation metrics:** RMSE and MAE scores were used to evaluate the performance of the model. **Observations:**

Model Params	Train (rmse)	Validate (rmse)	Test (rmse)	Evaluate (rmse)
default_params	0.005878661190833931	0.005844193387682432	0.006093833720059899	0.0059242788541034415
params_one	0.004978276907360225	0.005092708145074442	0.005229524517010376 5	0.005039975722926263
params_two	0.004997878889105543	0.005126282787313408	0.005290319052123 8	0.005069214445085569

Other Model 3

This model evaluates the model after addition of more features and uses standardisation and transformation techniques.

Features used: The candle positions and robot positions were used to calculate the distance of each robot with the candles along the x and y axis seperately, which were used along with collision and orientation of robot.

Data split: 70% of data is used for training, with 10% of validation and 20% for testing.

Evaluation metrics: RMSE and MAE scores were used to evaluate the performance of the model.

Observations:

Model Params	Train (rmse)	Validate (rmse)	Test (rmse)	Evaluate (rmse)
default_params	0.006857456851737323 5	0.007020733591741029	0.007275347894259288	0.00695737272582952
params_one	0.006120345806405142	0.006308155320899452	0.006538339639785879	0.00622273576352644
params_two	0.006276799679741773	0.006518804555387753	0.006686550020099288	0.00638296085149503 1

Final Model

This model evaluates the model after addition of more features and uses standardisation and transformation techniques.

Features used: Distance between robot and candles, both horizontal and vertical distances as well as Euclidean distances, along with collision and orientation is used.

Data split: 70% of data is used for training, with 10% of validation and 20% for testing.

Evaluation metrics: RMSE and MAE scores were used to evaluate the performance of the model.

Observations:

Model Params	Train (rmse)	Validate (rmse)	Test (rmse)	Evaluate (rmse)
default_params	0.006796070283839948	0.007285995978415394	0.007284185282007731	0.00694270051595419 65
params_one	0.004792246505649399	0.004999201637891569	0.005078654152523908	0.00487023134673264 3
params_two	0.005080281136352966	0.005388566824830438	0.005476499096853347 5	0.00519036430561773 2

Key Findings / Observations

More experiments were performed on the dataset, Only the best params that works for these datasets were only recorded in this report.

- `relu` is the best activation function, which provides the highest performance scores
- the best achitecture in my experiments is `(12, 24, 48, 96)`
- smaller batch sizes performes well
- learning rate of `0.001` gives best results
- feature engineering is key to better results