Understanding NuMI Beam Position with a Machine Learning Technique

Carrie R. Cox

Undergraduate SIST Intern, Seattle Pacific University

Author Note

This project was made possible by the SIST (Summer Internships in Science and Technology) program at Fermilab.

Abstract

The NuMI beamline is currently the world's most powerful neutrino beam and supplies neutrinos to the NOvA detector located in Minnesota. There are many detectors and devices in place that monitor the beam's performance and quality by measuring several parameters, including the beam's intensity, profile or overall shape, and the position of its centroid. This project applied a supervised machine learning technique to the data collected from the beam position monitors (BPMs), and the muon monitors located along the NuMI beamline to develop two algorithms that each predict the location of the beam centroid, one using the BPM data and the other using the muon monitor data. The first algorithm developed was a multiple linear regression model, and then a second algorithm was developed using a neural network technique. The predictions made by the models were compared to the centroid location measurements taken by the multi-wire detector to assess their performance.

Keywords: NuMI beamline, machine learning, neural network

Understanding NuMI Beam Position with a Machine Learning Technique

Machine learning (ML) is a branch of artificial intelligence (AI) studies that focuses on using sample data and algorithms to imitate the way a human might learn, gradually improving the accuracy of the model as more sample data is provided and more training is performed. It is a powerful tool in analyzing data that helps users make key insights and discoveries about a set of data and to monitor and improve the performance of a process or system. This project utilizes the data received from some of the detectors used in the NuMI beamline system and a supervised learning ML technique to develop two different models that each predict the location of the beam's center-of-mass, or centroid. One model was developed from the beam position monitor (BPM) data while the other model used the muon monitor data to generate its prediction. The position of the beam's centroid is measured by the Multi-wire sensor, which is positioned just before the target. The BPM system is located before the beam collides with the target, and so is considered the upstream data, while the muon monitors are located after the target system and so is the downstream data. The two predictions made, from the upstream and downstream data, are compared with the measurement made by the multi-wire sensor and to each other. By comparing the two predictions and the measurement, scientists at Fermilab will be able to see which devices are most reliable for recording the beam centroid's position and will be able to find detector anomalies, which will help ensure the quality of the NuMI beamline as it powers cutting-edge neutrino experiments at Fermilab.

Background

The NuMI Beamline

The NuMI (Neutrinos from Main Injector) beamline accepts a beam of protons from the Main Injector. The proton beam then collides with the target, which is approximately 1.2 meters long and made of graphite divided into 47 thin fins. The interactions resulting from the collision of the proton beam with the target produces secondary particles: pions and kaons. These secondary particles are focused using two magnetic focusing horns, and then enter the evacuated pipe. In this region, the particles decay into muons and muon neutrinos. The muons and muon neutrinos continue through the beam absorber where they meet three muon monitors, each positioned between layers of solid rock. The muons are stopped and measured by the muon monitors and layers of earth, while the neutrinos continue to the NOvA detector in Minnesota.

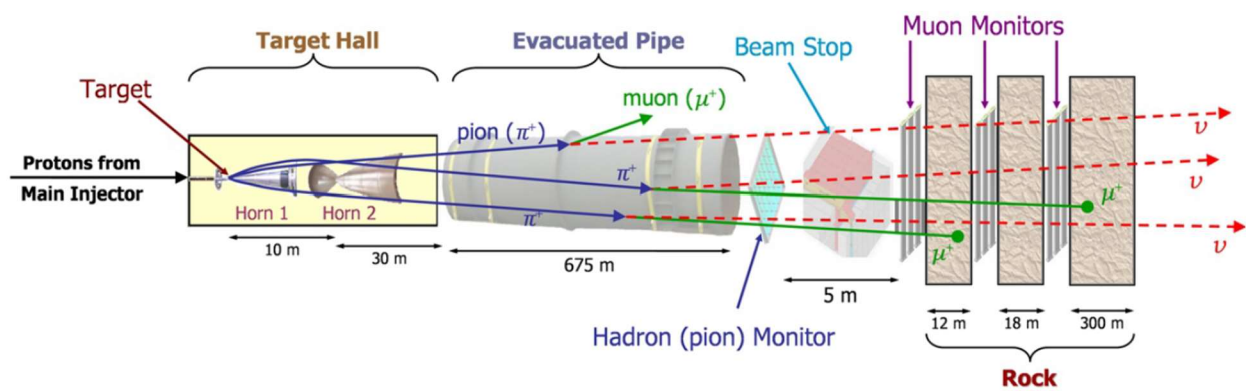

Figure 1: The NuMI Beamline, with components listed above depicted

The Detectors

Several devices and detectors are used to monitor the NuMI beamline's performance. Data collected from just a few of these detectors was used in this project, including from the BPMs, the multi-wire sensor, and the muon monitors. A brief explanation of the method and function of these devices is provided below.

Beam Position Monitors (BPMs).

A BPM is a non-destructive (does not affect the beam's performance) electromagnetic diagnostic device that measures the horizontal and vertical position of a proton beam as the beam passes through the sensor. This is done by measuring the charges induced by the electric field of the beam particles on an insulated metal plate. As the positively charged beam passes through the BPM, the distribution of the electrons on the plate changes, inducing a current of the same magnitude but opposite charge (see Fig. 2). To determine the position of the beam's center-of-mass, four pick-up plates are placed above, below, and on either side of the beam so that the horizontal and vertical position of the beam can be measured (Forck, Kowina, & Liakin, 2007).

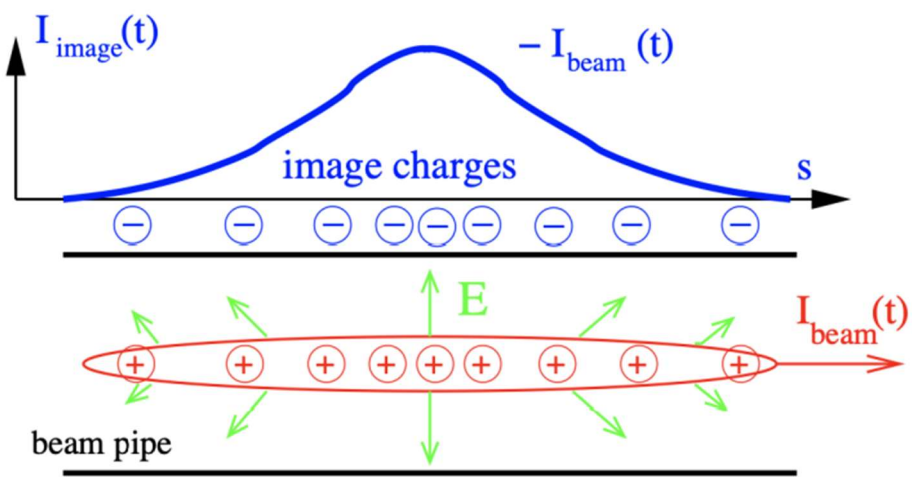

Figure 2: Graph of charge distribution generated by proton beam traveling through BPM; resulting current is measured by BPM

Data taken from two different kinds of BPMs was used in this study, from strip-line BPMs and button BPMs. Strip-line BPMs are hollow and cylindrical, made of stainless steel, and installed using ceramic supports. The conducting plates are installed inside the BPM so that as the beam travels through the tube, the signal is picked up by the plates (Fig. 3). Button BPMs work on a similar principle as other BPMs, but the conductive region on them is curvilinear rather than planar or flat as in the strip-line BPM. Additionally, in order to measure both the vertical and horizontal position of the beam's centroid, four button BPMs must be installed with their curved plates facing inward toward the beam, so that the four button BPMs together form a cylindrical region for the beam to pass through (see Fig. 4)

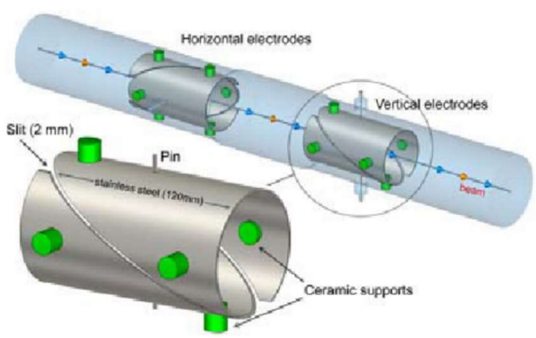

Figure 3: Image of Stripline BPM, showing electrode placement, supports, and path of beam through BPM

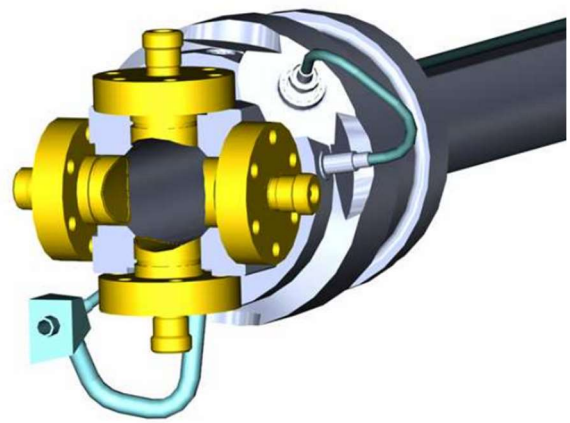

Figure 4: Image of installed button BPM system in beam pipe

Multi-wire Detector.

The multi-wire detector is used to not only measure the horizontal and vertical position of the beam's centroid, but also its profile or shape, its intensity, and its spot size or cross-sectional area. To make these measurements, the proton beam travels through a fine wire mesh, causing ionization to occur, generating a signal. This detector is placed just before the proton beam collides with the target. Though the Multi-wire sensor is capable of measuring more parameters

than the BPM system, unlike the BPM device it does lower the beam's intensity, and so less Multi-wire detectors are used than BPMs.

Muon Monitors.

The final type of detector that data was taken from in this project is the muon monitor. Muon monitors are charged capacitor plates that are filled with hydrogen gas between the plates. As the negatively charged muons pass through the muon monitor, they ionize the hydrogen gas, kicking the electrons out of the hydrogen atoms. The free electrons then ionize other hydrogen atoms, causing an amplifying cascade. The free electrons are attracted to the positively charged capacitor plate, where the signal is received. Muon monitors are composed of nine vertical strips of these capacitor plates, which are also divided lengthwise by nine, totaling at eighty-one evenly sized pixels per muon monitor. Three muon monitors are used in the NuMI beamline system, spaced between layers of rock, so this project utilizes data from a total of 243 muon monitor pixels.

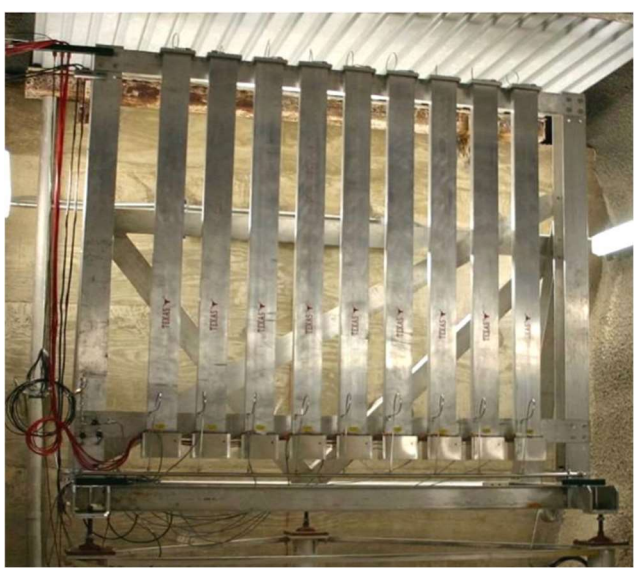

Figure 5: Photograph of a muon monitor

Introduction to Machine Learning

Machine learning is a subfield of artificial intelligence studies that gives computers the ability to learn without being explicitly programmed. There is a wide variety of situations and problems in which machine learning can be applied, and many different algorithms designed to build models that answer these problems. The ultimate purpose of a machine learning model can be descriptive, in which the system uses the data to tell what happened, predictive, in which the model uses the data to predict what will happen or the value of another variable, or prescriptive, meaning the system will make suggestions as to what action to take. Two common types of machine learning problems are classification, in which the model assesses a data sample and outputs either “true” or “false”, and predictive, in which the model receives one or more data points and then predicts the value of another, separate, continuous variable. Additionally, there are supervised and unsupervised machine learning techniques. In supervised learning, the model development algorithm is given a correct answer to compare the output of its model to, while in unsupervised learning no comparison result is given. This project utilizes the data received from the BPM system and the muon monitors and machine learning algorithms to generate predictive models that predict the horizontal and vertical position of the proton beam’s centroid, which are compared to the measurement taken by the multi-wire sensor.

Every machine learning project begins with data and having larger batches of training data will generally lead to better models. The data that is provided to the machine learning algorithm is split into two groups, the training set and the validation, or test set. The training set is used to train the model, or what the model uses to learn, while the validation set is used to

assess how the model performs when it is shown new data. Below is the BPM and multi-wire signal data used in this project.

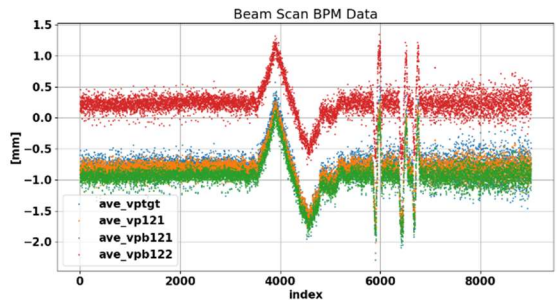

Figure 6: Sample of BPM signal data recording the horizontal beam position

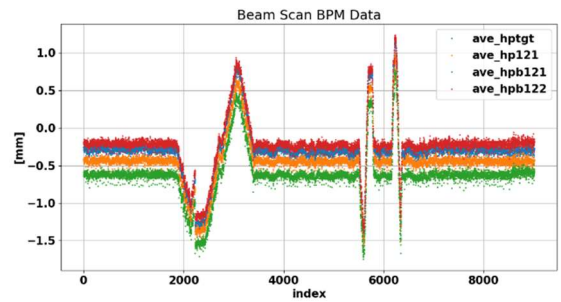

Figure 7: Sample of BPM signal data recording the vertical beam position

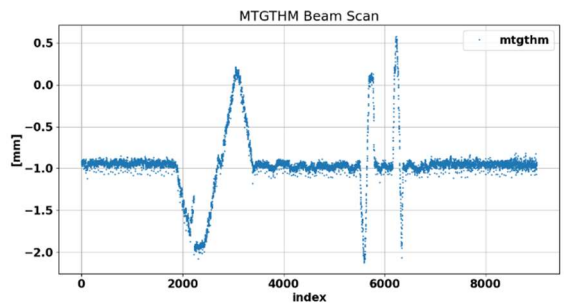

Figure 8: Sample of Multi-wire measurement data of horizontal beam centroid position

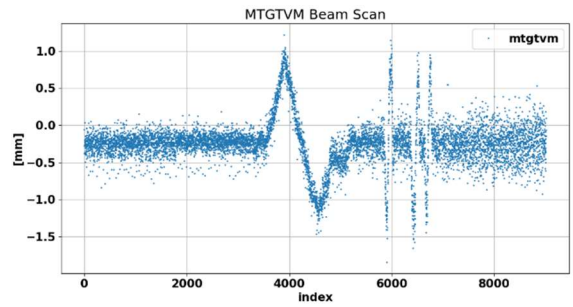

Figure 9: Sample of Multi-wire measurement data of vertical beam centroid position

The correlations between different beam parameters, including the BPM signals, the multi-wire centroid measurements, the muon monitor data, the beam intensity, and the beam spot size, were analyzed using Python to generate comparative plots. Then, different machine learning techniques were implemented, first a linear regression model and then a neural network model, which will be explained in further detail below.

Linear Regression (Ordinary Least Squares)

Linear regression is one of the simpler and more popular machine learning algorithms, and is used to make predictions for continuous variables, in this case the horizontal and vertical position of the beam's centroid. The linear regression algorithm shows a relationship between a dependent, or target, variable and one or more independent, or predictor, variables. In this

application, the multi-wire data is considered to be the dependent variable, while the BPM signals are the independent variables. Linear regression can be represented mathematically as

$$y = a_0 + a_1 x$$

where y is the target variable, x is the predictor variable, a_0 is the intercept of the line, a_1 is the scale factor to each input value. Above is the equation for simple linear regression, however in this situation there is more than one predictor variable, and so a multiple linear regression model is used, pictured below.

$$y = a_0 + a_1 x_1 + a_2 x_2 + \cdots + a_n x_n$$

The result is that a multiple linear regression model does not simply generate a line of best fit, but an equation of a hyperplane of best fit, which can be conceptualized as a line of best fit for data in three or more dimensions.

The linear regression algorithm determines how well the hyperplane fits the data according to the cost function, which is the degree of error between the model's prediction and the actual value of the training data set. There are many different formulas that can be used to calculate cost, but linear regression uses what is known as Mean Squared Error (MSE).

$$J = \frac{1}{n}\sum_{i=1}^{n}(pred_i - y_i)^2$$

J is the cost function, n is the total number of samples in the training data set, $pred_i$ is the prediction the model generated for a certain data point, and y_i is the actual value for that same data point (Pai, Multiple Linear Regression, 2021). In order to minimize the cost function, the model implements a method called gradient descent, in which the partial derivative of the cost function is taken to find the relationship between a single parameter and the cost function, and used to determine how the constant and scaling coefficients must then be updated (Pai, Gradient Descent, 2021). Additionally, the learning rate is a constant that determines how much these

parameters are altered per training iteration. An advantage of having a larger learning rate is that the model will quickly update the constants used, however the level of precision that the model can achieve will be more limited. A smaller learning rate will cause the algorithm to take more time to build an effective model, but may reach a higher degree of accuracy. After taking the partial derivatives of the cost function with respect to a_0 and a_1, the resulting update functions are pictured below.

$$a_0 = a_0 - \alpha * \frac{2}{n}\sum_{i=1}^{n}(pred_i - y_i)$$

$$a_1 = a_1 - \alpha * \frac{2}{n}\sum_{i=1}^{n}(pred_i - y_i) * x_i$$

In both equations, α is the learning rate, n is the total number of data samples, $pred_i$ is the prediction generated by model while y_i is the true value from the training sample, and in the second equation x_i is the value of the independent variable for that data point.

Multiple linear regression has many advantages, such as its overall simplicity, its consistency, and the high levels of accuracy it can reach in certain situations. However, the only parameter that can be altered by the user is the learning rate, so while it is consistent, its limited flexibility means that linear regression can only reach a certain level of accuracy. In contrast, there are many adjustable parameters in a neural network algorithm, which will be described below.

Neural Network

A neural network takes its name from and is inspired by the structure of the human brain, mimicking the process by which biological neurons signal and communicate with each other. Neural networks are comprised of layers of nodes, beginning with the input layer, which is the data itself, the output layer, which is the prediction generated by the model, and layers of nodes in between the input and the output, referred to as hidden layers. Each node is connected to every node of the following layer, and has an associated weight and threshold. The weights determine the importance of a given variable in determining the output. Once all of the inputs, multiplied by their weights are summed, they are passed through an activation function which determines the output of the node, and if the output exceeds a given threshold the node is activated, causing the output of that node to become part of the input of the next layers of nodes (IBM Cloud Education, 2020). An advantage of a neural network model is that there are many tunable hyperparameters, including the number of hidden layers, the number of nodes between layers, the activation functions used, the learning rate, and the number of epochs, or training iterations the algorithm runs through to develop its final model.

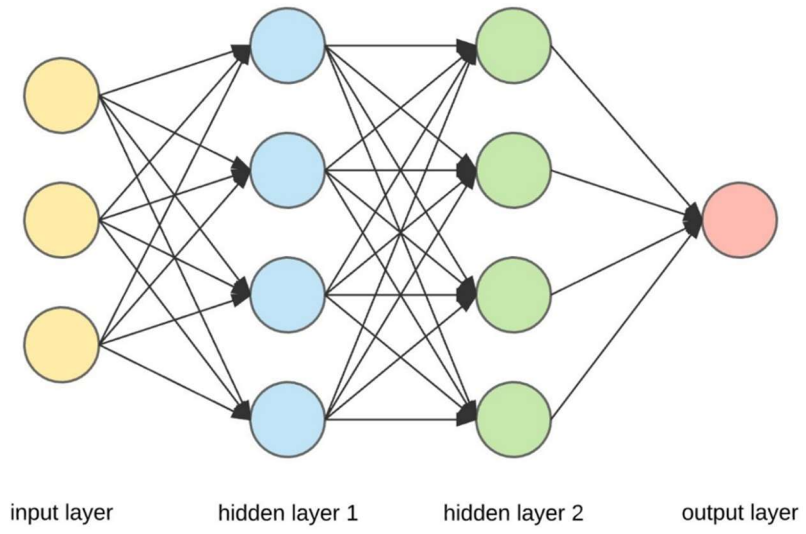

Figure 10: Visualization of a neural network

Method

The initial dataset used for model development was the beam scan data, which shows more variation in the position of the beam centroid than normal operation datasets and so was used as the dataset for training the model. The data was divided into a 70 percent training, 30 percent validation split. This project began with implementing a multiple linear regression algorithm to develop two models that both accepted the BPM data. One model predicted the horizontal location while the other predicted the vertical. The Python sklearn library was used to develop the model (Pedregosa, et al., 2011), and the learning rate applied was 0.001. Then, a neural network algorithm was applied to develop two models that each predicted the horizontal and vertical location of the beam centroid, one from the BPM signals and the other from the muon monitor data. The neural network algorithms implemented the Python tensorflow, keras libraries, calling the sequential model function to build a deep learning model by adding layers to the model, and the Adam optimizer. The hyperparameters used by the neural network models, depicted on the following page, were taken from an optimization algorithm applied to the same data for a separate project. A learning rate of 0.00001 was applied to both algorithms, and they each ran through 3,000 epochs.

BPM Neural Network Hyperparameters

Layer	Nodes	Activation Function
1 (Input)	8	Tanh
2	480	Tanh
3	130	Sigmoid
4	135	Sigmoid
5	11	Sigmoid
6	4	Sigmoid
7 (Output)	2	Linear

Muon Monitor Neural Network Hyperparameters

Layer	Nodes	Activation Function
1 (Input)	243	Tanh
2	480	Tanh
3	130	Sigmoid
4	135	Sigmoid
5	11	Sigmoid
6	4	Sigmoid
7 (Output)	2	Linear

Results

The first results shown are those from using the multiple linear regression algorithm on the BPM data. Two models were developed, one model that predicts the horizontal position and one that predicts the vertical position. The scipy model.score result, which is the coefficient of determination R^2 for the horizontal predictions training and validation sample was 0.999, and also 0.999 for the vertical prediction training and validation data samples[1]. See Appendix A for more detailed plots of the multiple linear regression algorithm result.

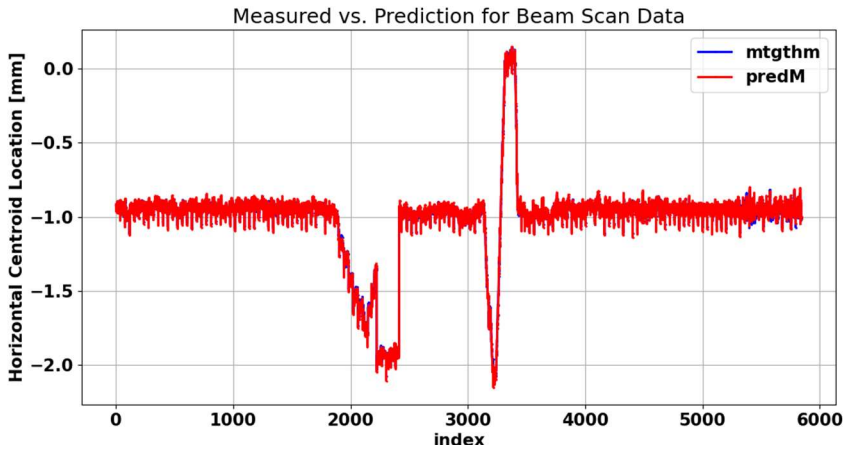

Figure 11: Plot of measurement and prediction of horizontal centroid position from multiple linear regression model

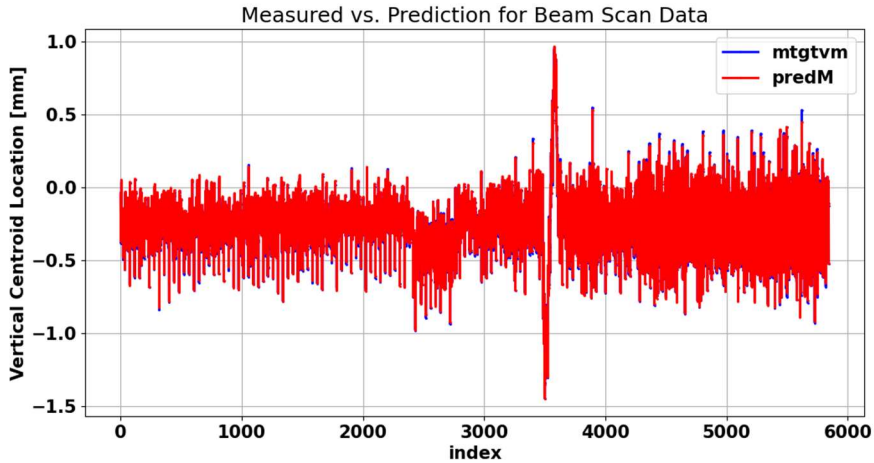

Figure 12: Plot of measurement and prediction of vertical centroid position from multiple linear regression model

---

[1] For more information on the coefficient of determination R^2, see the scikit learn Linear Regression page under sklearn.linear_model.LinearRegression

Second shown are the results of the neural network algorithm. Two models were developed using neural network algorithms that each predicted both the horizontal and vertical location of the beam centroid, one developed from the BPM data and the other from the muon monitor data. The results for both parameters from both models are included, and further depictions of these results can be seen in Appendix B. Important to note is that the predictions for the vertical position of the beam centroid developed from the muon monitor data were significantly less reliable than the same model's predictions for the horizontal position, and the model using the BPM data to predict both parameters, as shown by the offset from the mean and significantly greater standard deviation in Figure 30, Appendix B.

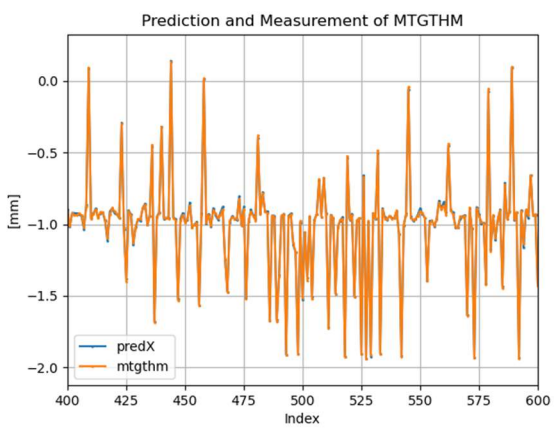

Figure 13: Partial plot of BPM model prediction and multi-wire measurement of the horizontal position

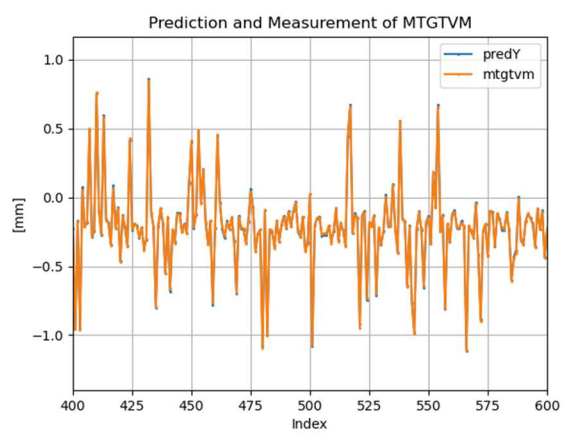

Figure 14: Partial plot of BPM model prediction and multi-wire measurement of the vertical position

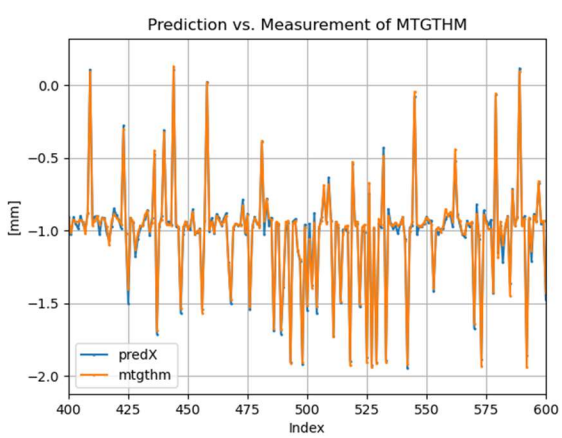

Figure 13: Partial plot of Muon Monitor model prediction and multi-wire measurement of the horizontal position

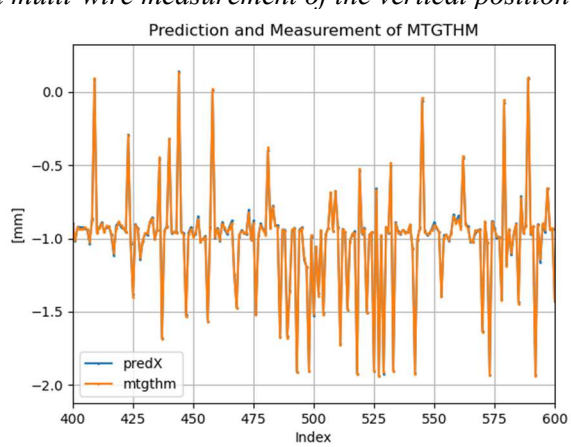

Figure 16: Partial plot of Muon Monitor model prediction and multi-wire measurement of the vertical position

Conclusion and Further Work

This project determined that it is very possible to use the BPM and muon monitor data to predict the location of the beam centroid. Models such as the ones developed in this project and implementing neural network techniques can be applied to the NuMI beamline system's normal operation to assist Fermilab scientists with finding detector anomalies and with determining which measurements taken by the devices are the most accurate by comparing the three independent values of the horizontal and vertical position of the beam centroid, the BPM prediction, the muon monitor prediction, and the multi-wire measurement. If one of these values begins to deviate by a significant degree from the other two, that could indicate there is an anomaly occurring within that measurement system. If one of the three values is consistently differing from the other two, that may mean that the device(s) associated with that value are less reliable. Knowing how stable the NuMI beamline is and which diagnostic devices are most effective will help ensure the overall quality of the NuMI beamline as it continues to power important and exciting experiments.

Further work that could be done on this project includes implementing an optimization algorithm, such as a Bayesian optimization algorithm, that will optimize the hyperparameters used in the neural network algorithms. Another possible next step in this project would be further analysis into possible physical reasons why the vertical position prediction made by the muon monitor data is significantly less reliable than the other predictions generated. Finally, to determine how effective the models are at detecting anomalies and monitoring the beamline, the models should be tested against normal operation datasets before they are implemented into the NuMI system.

Acknowledgements

First, I would like to thank my supervisor, Athula Wickremasinghe, for his continuous support during this project and for helpful suggestions for how to improve. I would also like to thank Katsuya Yonehara for providing valuable background information on the workings of the NuMI beamline. Finally, I want to thank my mentors, Carrie McGivern and Donovan Tooke, for their feedback during this process and the other interns in my group for asking insightful questions and offering good suggestions, and the SIST committee for giving me the opportunity to spend my summer working at Fermilab.

References

Forck, P., Kowina, P., & Liakin, D. (2007). Beam Position Monitors. *Gesellschaft fur Schwerionenforschung GSI*, 1.

IBM Cloud Education. (2020, August 17). *Neural Networks*. Retrieved from IBM Cloud Learn Hub: https://www.ibm.com/cloud/learn/neural-networks

Pai, A. (2021). *Gradient Descent*. Retrieved from Machine Learning Works: https://www.machinelearningworks.com/tutorials/gradient-descent

Pai, A. (2021). *Multiple Linear Regression*. Retrieved from Machine Learning Works: https://www.machinelearningworks.com/tutorials/multiple-linear-regression

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., . . . Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research 12*, 2825-2830.

Appendix A

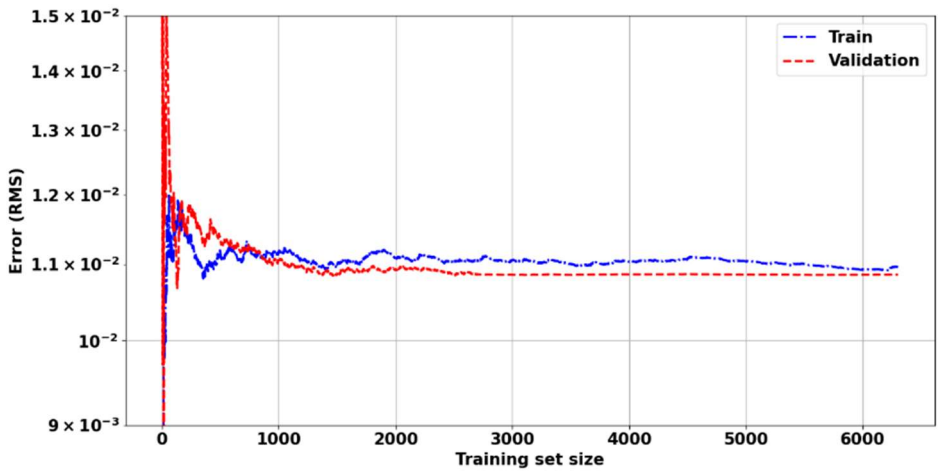

Figure 14: Process plot of error for training, validation data of multiple linear regression for horizontal position prediction model

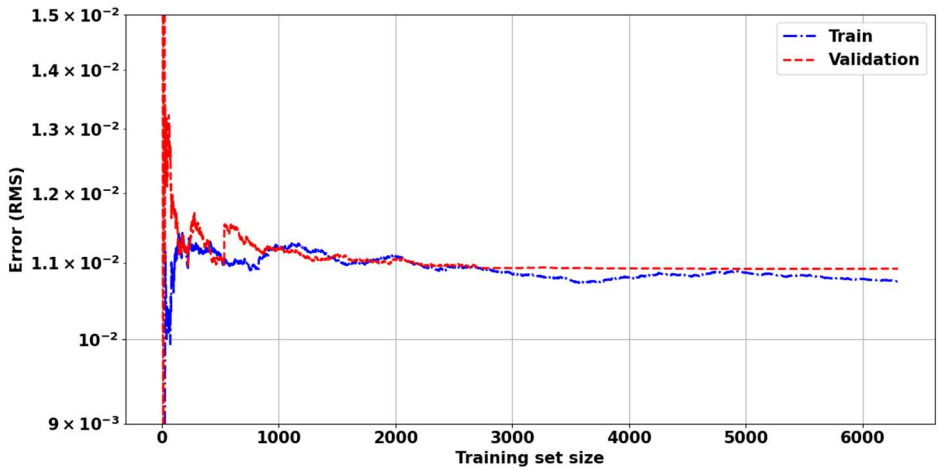

Figure 18: Process plot of error for training, validation data of multiple linear regression for vertical position prediction model

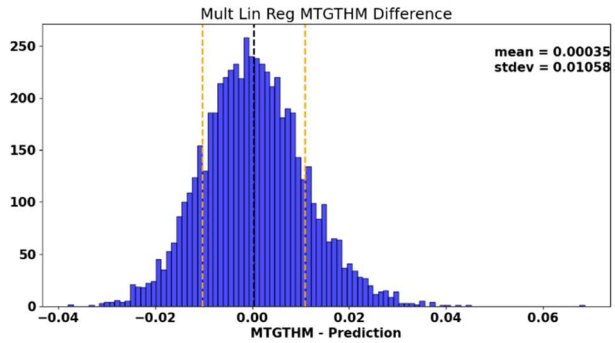

Figure 19: Difference plot, mean, and standard deviation of prediction and measurement of horizontal position

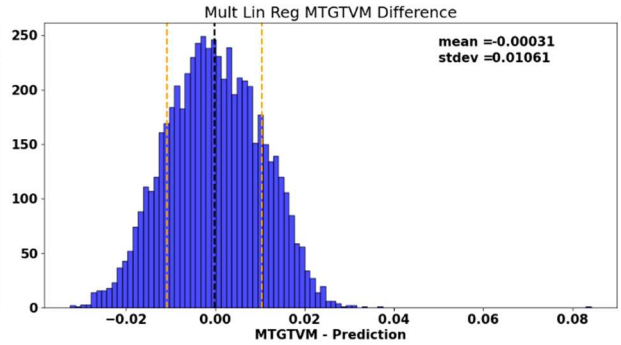

Figure 20: Difference plot, mean, and standard deviation of prediction and measurement of vertical position

Appendix B

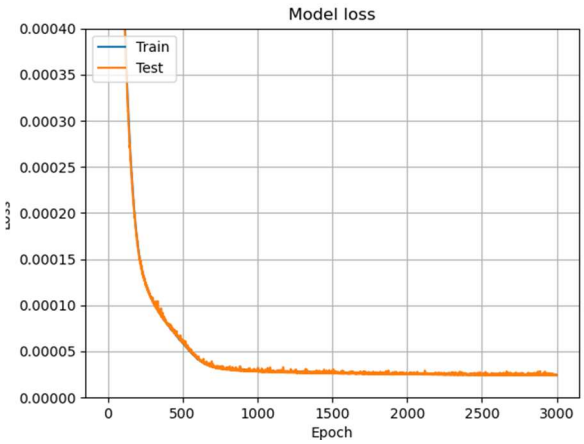

Figure 21: Process plot of neural network algorithm using BPM data

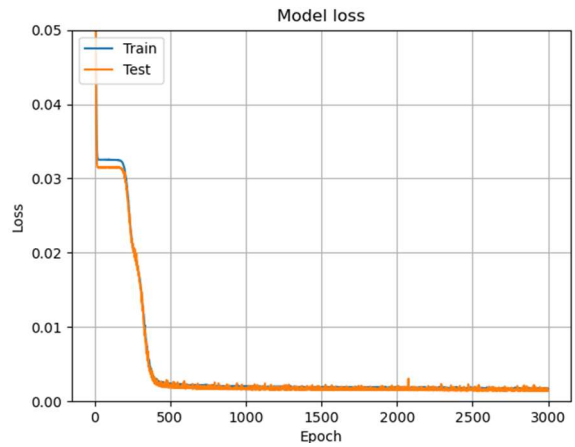

Figure 5: Process plot of neural network algorithm using muon monitor data

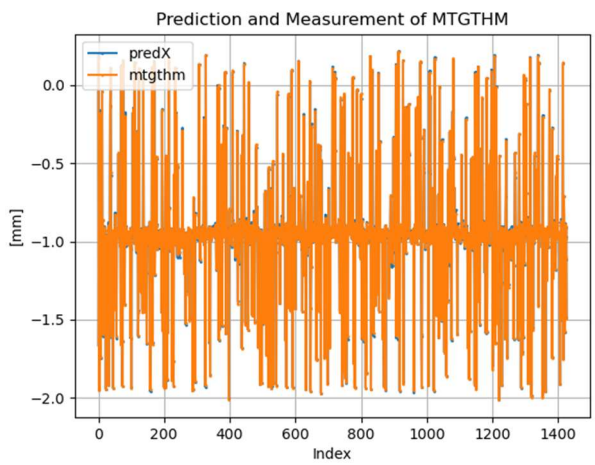

Figure 23: Plot of BPM prediction and multi-wire measurement of horizontal centroid position

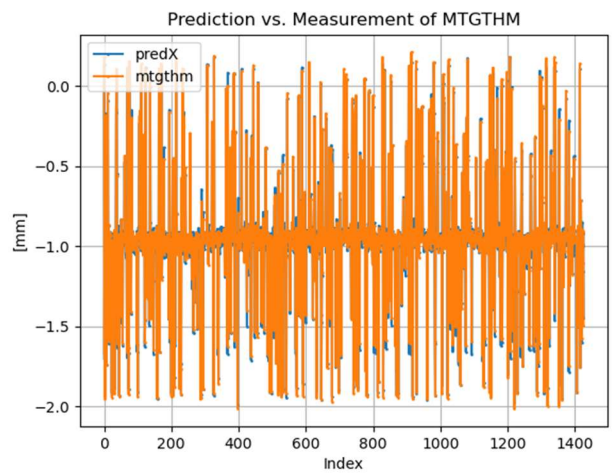

Figure 24: Plot of muon monitor prediction and multi-wire measurement of horizontal centroid position

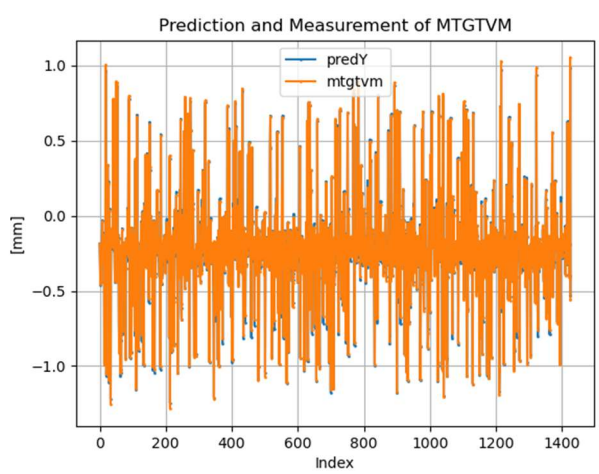

Figure 25: Plot of BPM prediction and multi-wire measurement vertical centroid position

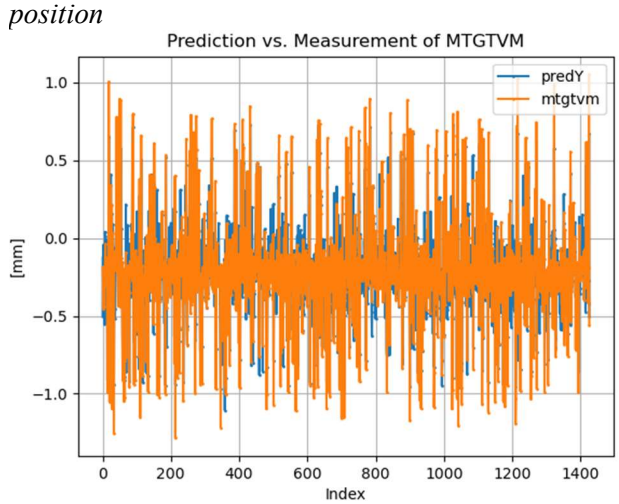

Figure 26: Plot of muon monitor prediction and multi-wire measurement vertical centroid position

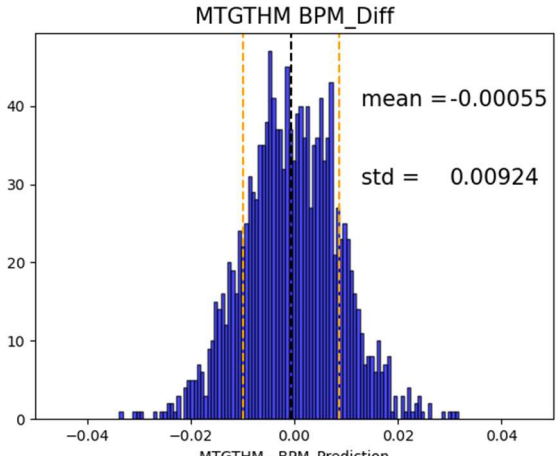

Figure 27: Histogram of the measurement minus the BPM prediction for the horizontal centroid position

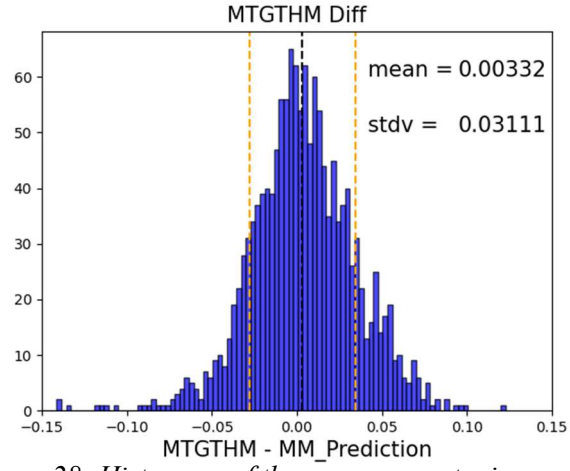

Figure 28: Histogram of the measurement minus the muon monitor prediction for the horizontal centroid position

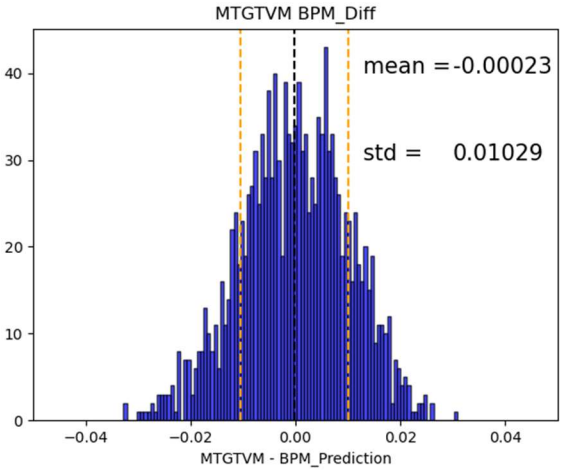

Figure 6: Histogram of the measurement minus the BPM prediction for the vertical centroid position

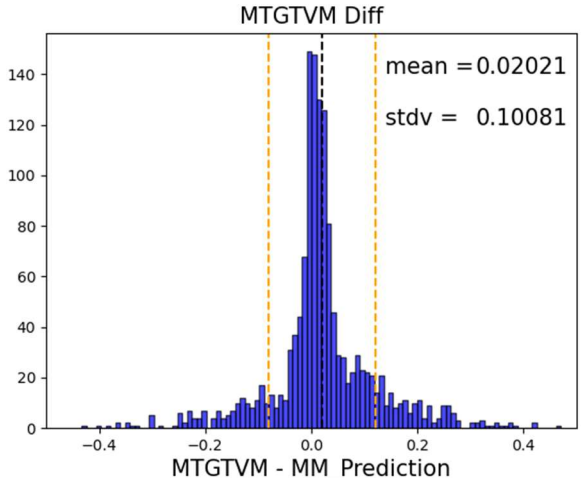

Figure 30: Histogram of the measurement minus the muon monitor prediction for the vertical centroid position

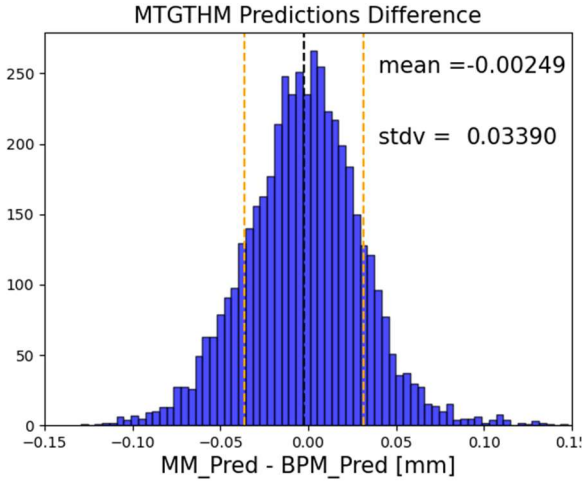

Figure 31: Histogram of the muon monitor prediction minus the BPM prediction for the horizontal centroid position

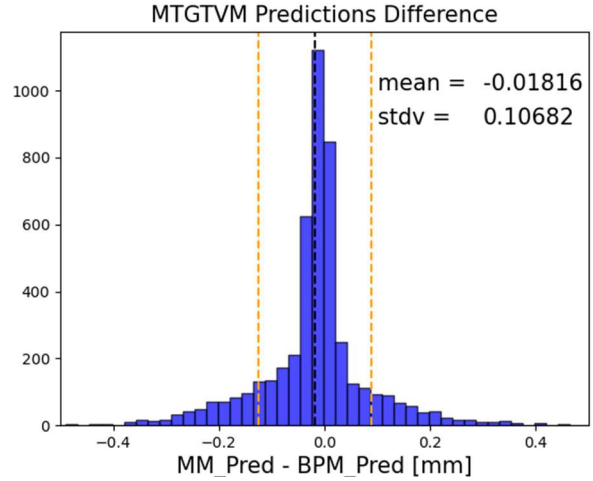

Figure 32: Histogram of the muon monitor prediction minus the BPM prediction for the vertical centroid position