DAE Symposium on Nuclear Physics

Indian Institute of Technology Indore, Madhya Pradesh

December 09-13, 2023

Background:

Muon tracking, long a cornerstone in particle physics, has unveiled an innovative methodology for detecting radioactive materials. Muons, subatomic particles, display unique interactions when they traverse different materials. These interactions are significantly influenced by the atomic structure and density of the encountered substance. By positioning a radioactive material in front of a scintillator, we can meticulously capture and analyze these muon interactions. This metric offers critical insights into the interaction patterns and serves as an accurate means to pinpoint the presence and location of radioactive sources.

Objective:

Our primary objective is to harness machine learning techniques to enhance the accuracy of pinpointing the location of radioactive materials. By analyzing the patterns of muon interactions and their timings with photomultiplier tubes (PMTs), we aim to refine and optimize our models, ensuring a more precise and reliable detection method with far-reaching implications in fields like security, environmental monitoring, and nuclear safety.

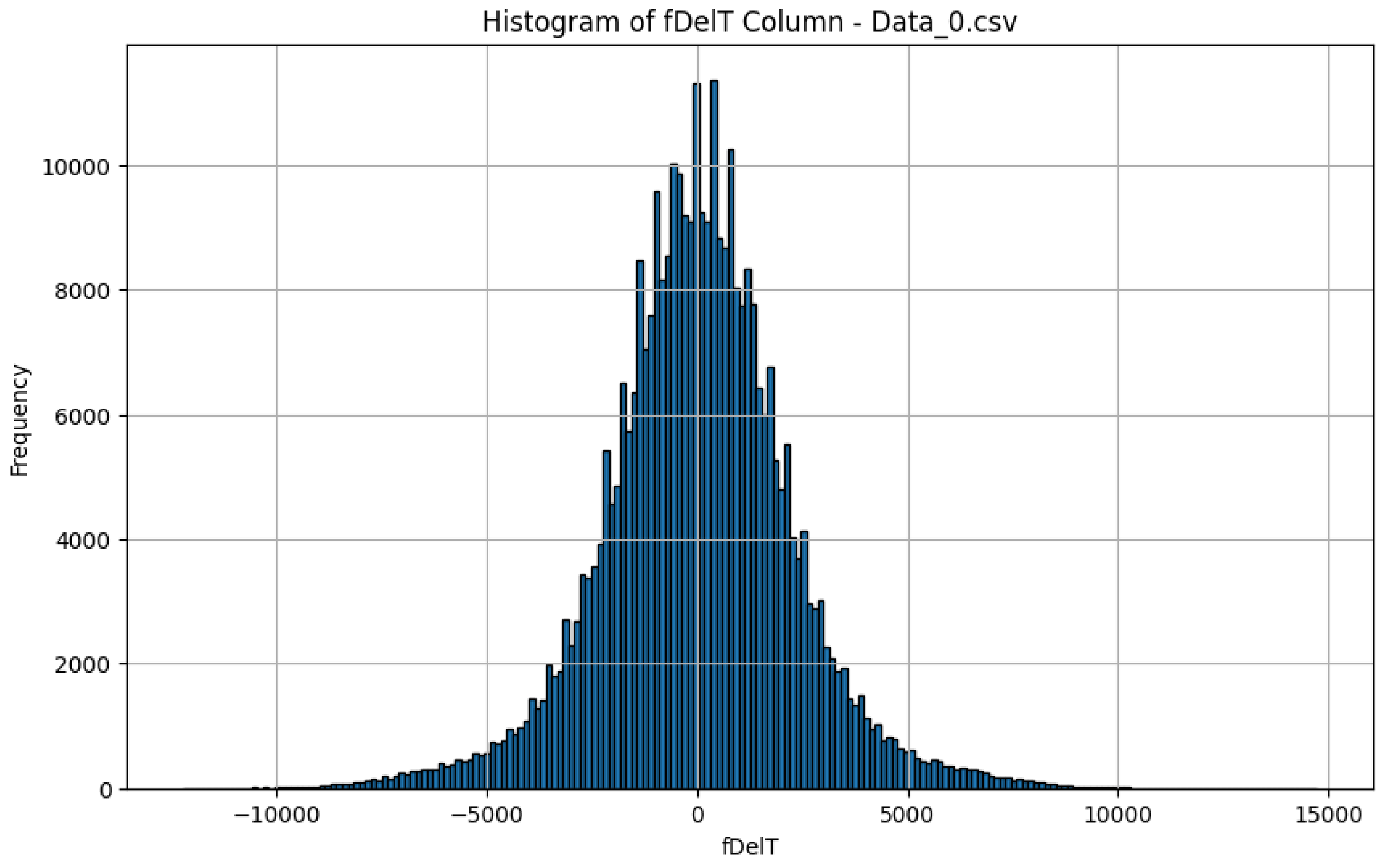
Methodology and Challenges:

The Below image depicts a detector subjected to muon bombardment. As a muon impacts a specific location on the scintillator, it is at a distance x from one PMT and $100-x$ from the other PMT. A potential challenge arises due to potential inaccuracies in measuring x, represented by m. When multiple detectors (y) are aligned sequentially, this inaccuracy is amplified, leading to an accumulated error of $m\times y$.

The attached electronics to the PMT not only record the time of interaction ($t1$,$t2$) but also capture the energy registered during the event ($q1$,$q2$). By delving deep into vast datasets containing $t1$,$t2$,$q1$, and $q2$ values, we aim to hone the accuracy of x calculation.

Data Collection: Muon tracking data was collected using the [specific scintillator type/model] which is 100 cm long. object is placed in front of the detector. When a muon particle hits the detector. the time taken for the secondary particles to reach each of the photomultiplier tubes (PMT) is recorded as $t1$ and $t2$. The differential time, termed *fDelT*, is calculated as $t1-t2$ and stored in binary root files. The central position of the detector is denoted as '0', and it's surrounded by both negative and positive incremental positions, ranging from -50 to +50.

Understanding the histogram Muons bombards the scintillator uniformly. Muons that impact the center will produce an *fDelT* value of 0, since they have equidistant paths to travel to both PMTs. Muons that strike other parts of the scintillator will yield varying *fDelT* values depending on their impact point. As the distance from the center increases, the particle count decreases, culminating in a Gaussian curve for the *fDelT* histogram.



Data Pre-processing: Histograms represent the distribution of *fDelT* values for each position. For consistency in our analysis, the histogram for each position was corrected based on the baseline corrections identified at position 0. Specifically, the average *fDelT* value at position 0 was subtracted from the *fDelT* values of each dataset for each position. This process ensures that the data is centered around a consistent reference point. This alignment notably recentered the peak of the *fDelT* histogram for position '0'. //histogram pic

From the corrected histograms, two metrics were predominantly used: the average and standard deviation of *fDelT*. These metrics are pivotal in our machine-learning model.

Outliers: While the data exhibited outliers, we chose to retain all data points in their original form.

Method:

In our pursuit to understand muon behavior, we investigated various machine learning models. While several models were tested, we'll particularly emphasize two. The second model offers the best results we've observed, while the first highlights challenges inherent to applying machine learning techniques in this domain. Our primary tool for modeling was the Scikit-learn library.

Data Preparation:

The pre-processed data provided two foundational metrics: the average and standard deviation of

fDelT.

Average Time Difference (*avg_fDelT*): The mean of corrected *fDelT* values, representing the central tendency.

Standard Deviation (*std_dev_fDelT*): A measure of dispersion from the *avg_fDelT*.

Our data, captured in CSV files, offers a comprehensive view, with each file detailing muon behavior under specific conditions. Given the dataset's manageable size (under 10 lakh records), we opted for CSV over the more complex root files.

Machine learning models

1st Model: Linear Regression Analysis

While the domain remains relatively unexplored in the context of machine learning, beginning with a foundational approach is essential. To this end, we've employed Linear Regression—our '1st Model'. This serves as both our initial probe into the data and the groundwork from which we evolve and enhance our understanding. Using the insights and limitations of this basic model, we will seamlessly transition into the complexities and enhancements encapsulated in our '2nd Model', showcasing how foundational concepts can be built upon to achieve refined results.

The initial approach for our study was grounded in a basic linear regression model. The model's primary objective was to discern a linear relationship between the average of *fDelT* (denoted as *avg_fDelT*) and the muon's position.

The core idea is to find the best-fitting straight line, known as the regression line, which predicts the output values within a range.

Mathematically, in its simplest form (for one independent variable), the linear relationship can be expressed as:

$y=\beta+mx+\epsilon$

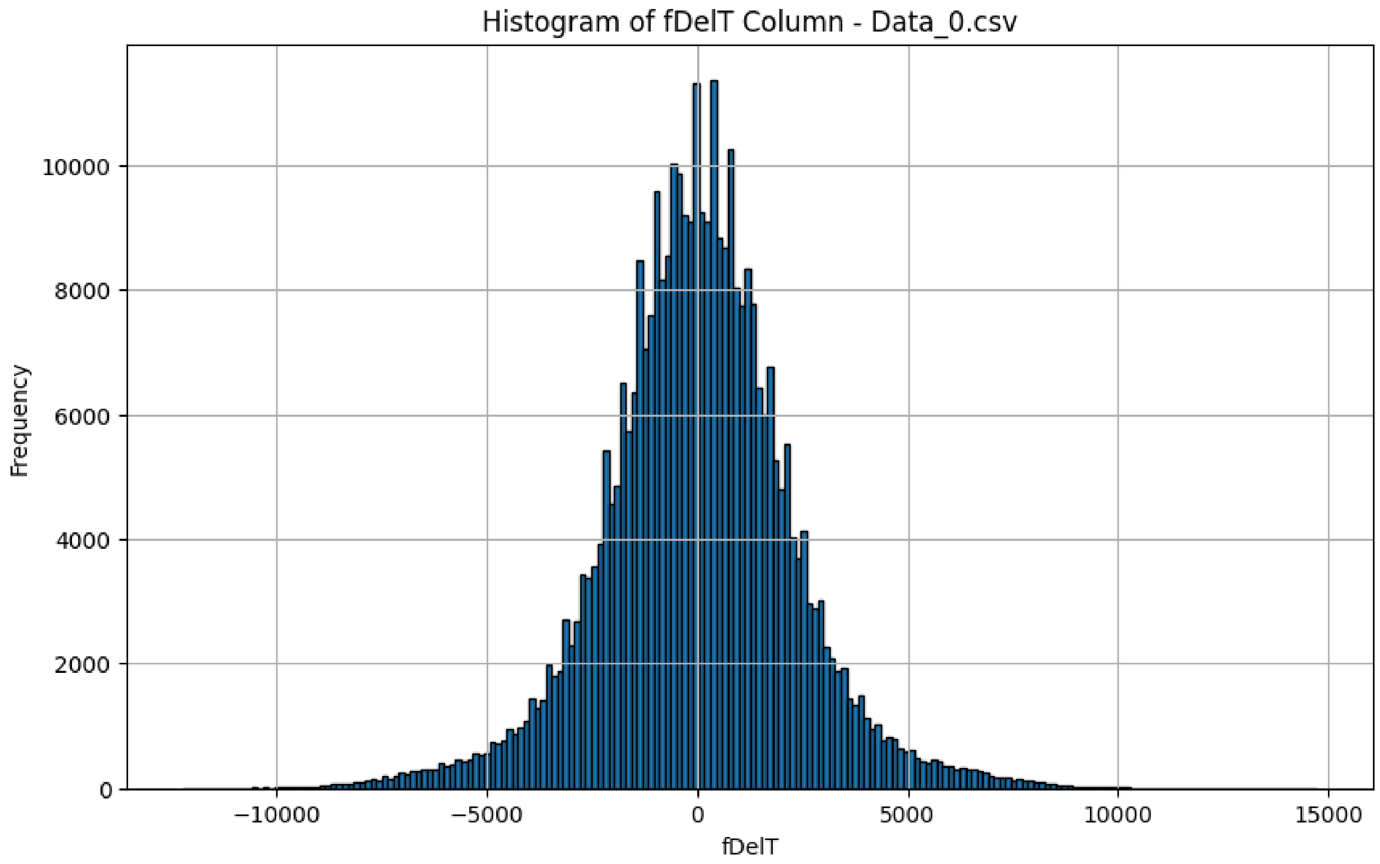
Where:

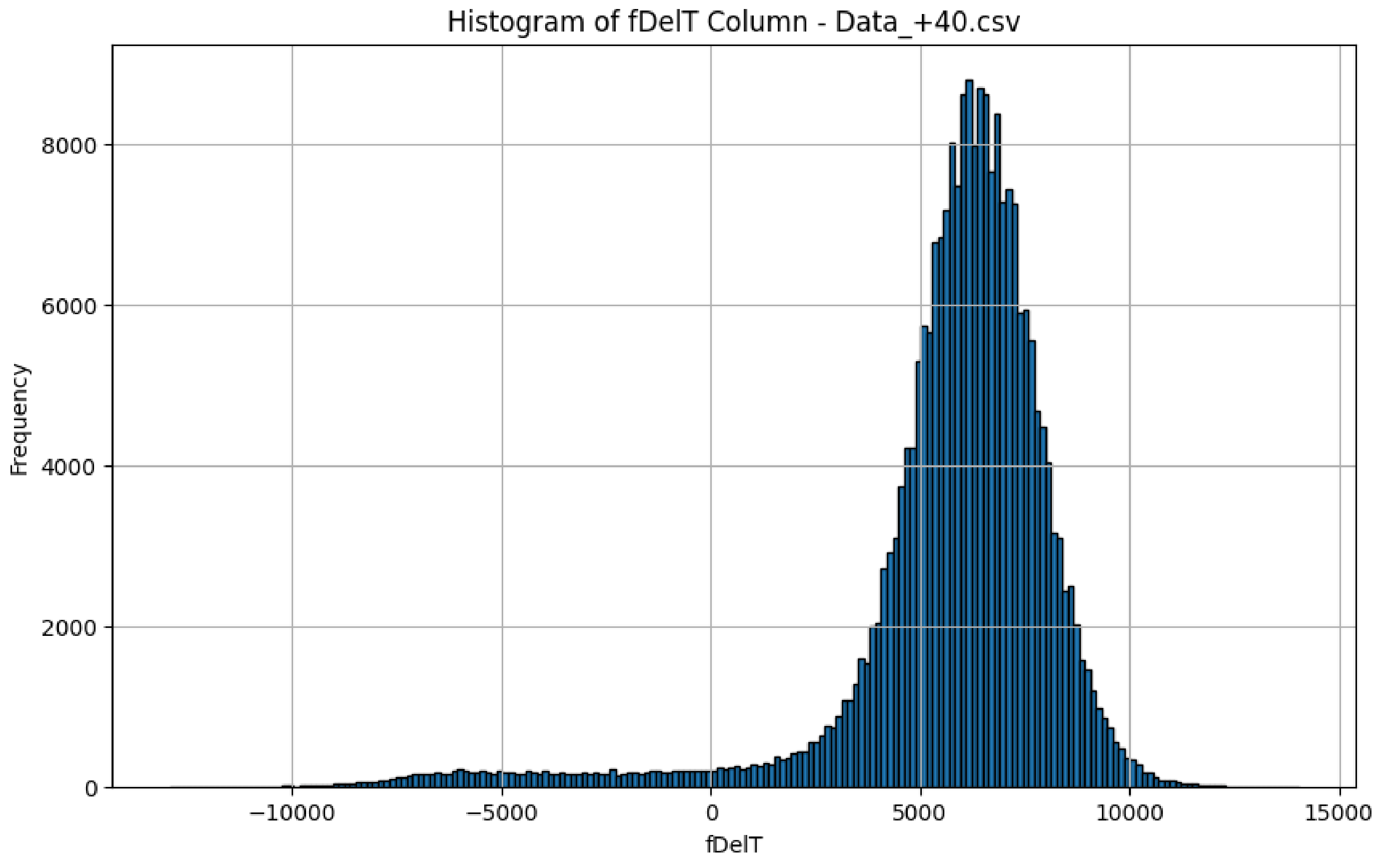
- y is the dependent variable (the variable we're trying to predict: position).
- x is the independent variable (*avg_fDelT*).
- β is the y-intercept of the line.
- m is the slope of the line.
- ϵ represents the error term (the difference between observed and predicted values).

The primary goal of the model is to determine the values of β and $\beta 1$ that minimize the sum of the squared differences (errors) between the predicted and actual observed values in the dataset.

Why It Wasn't Ideal:

Skewed Data Distributions: A pronounced skewness in the histograms became evident as the object's position diverged from the central point. This skewness manifested more intensively with a broader deviation from the center, causing the data distribution to resemble a skewed Gaussian shape. This non-linearity posed challenges to the assumptions of our linear regression model.

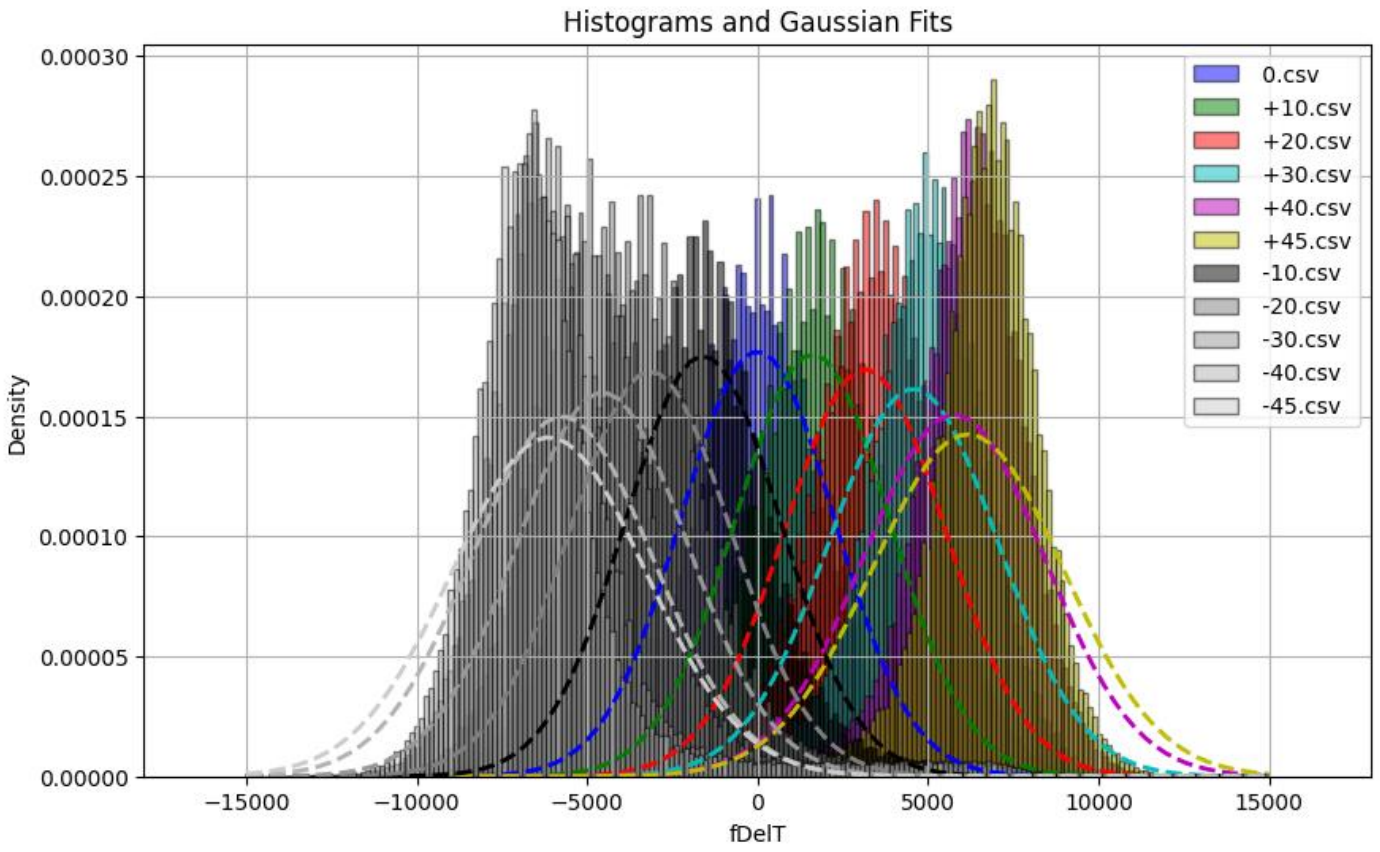




Challenges with Higher-Degree Polynomial Regression: An intuitive approach might be to leverage polynomial regression to accommodate the nonlinear patterns observed. However, models with polynomial terms beyond degree 2 presented a risk of overfitting, capturing not just the underlying trend but also the noise in the data.

To address this, we introduced the standard deviation of *fDelT* (denoted as *std_dev_fDelT*) as an additional feature. This helped the model to be more attuned to the variability in the data across different positions.

Overlap Challenges: Another complexity was the overlapping of Gaussian curves when predicting positions not present in the training set. Specifically, the Gaussian curve of a particular position seemed to interfere with those of its adjacent 10-multiple counterparts. This overlap risked inaccuracies in our predictions.as can be seen in +-40 and +-45 below.



As a safeguard against potential overfitting and to ensure the robustness of our model, we tested it against data points in multiples of 5. These data points were deliberately excluded from the training set, serving as a litmus test for the model's generalization capabilities.

2nd Model:

The 2nd model pivots towards using std_avg_fdelt as a feature too. From each dataset or position, we computed the average time difference, **`avg_fDelT`**, and the standard deviation of the time difference, **`std_dev_fDelT`**.The underlying computations for this model were executed using the Scikit-learn library, a powerful tool for machine learning tasks.

Working:

The cornerstone of polynomial regression is transforming the original features into polynomial features, thereby enabling the model to capture nonlinear patterns. Our model utilized 2-degree polynomial features, essentially aiming to model the relationship within a 3-dimensional space defined by **`avg_fDelT`**, **`std_dev_fDelT`**, and the position.

For polynomial regression of degree *n*, the model takes the form:

y=β0+β1x+β2x2+...+βn**xn

how the higher degrees of *x* are included in the equation, allowing for a nonlinear relationship between *x* and *y*.

- *β*0 represents the weights or importance of the respective terms in the equation. They're determined during the training process to minimize the error between the predicted and observed positions.

How It Works:

During training, the algorithm will:

a. Convert the original features into polynomial features. So, if we have a feature *x*, it will introduce *x*2,*x*3,…,*xn* for an nth-degree polynomial regression.

b. Fit the best polynomial line (curve) that minimizes the residual (difference between the observed and predicted values).

c. Determine the best coefficients (like *β*0,*β*1,…,*βn*) using optimization techniques like gradient descent or normal equations.

The result is a curve that can model nonlinear patterns in the data more effectively than a simple straight line.

Training and Testing:

The analysis and models were constructed based on data extracted from a scintillator of model [name]. Our training dataset incorporated various positions: -40, -30, -20, -10, 0, 10, 20, 30, and 40. Each of these positions held an expansive data size with more than 3 lakh rows.

The methodology of choosing the positions based on the "2n+1" sequence is noteworthy. This sequence ensures that the central position is 0 while having an equal number of positions on either side of this center. The "2n" guarantees that both positive and negative data from positions are incorporated, creating a balanced representation. end data i.e. +-50 should also be used if training in the real environment.

Choice of Test Data:

For the validation or testing phase, positions +45 and -45 were deliberately chosen. These positions deviate from the regular 10-unit increment found in the training dataset. This strategy serves two primary purposes:

1. By selecting positions outside the training range, we ensure that the model is predicting entirely unfamiliar data.
2. Testing at positions that aren’t in the 10-unit interval provides a stringent measure of the model's robustness. Any significant prediction error here would indicate the model's tendency to overfit to the training data's specific positions.

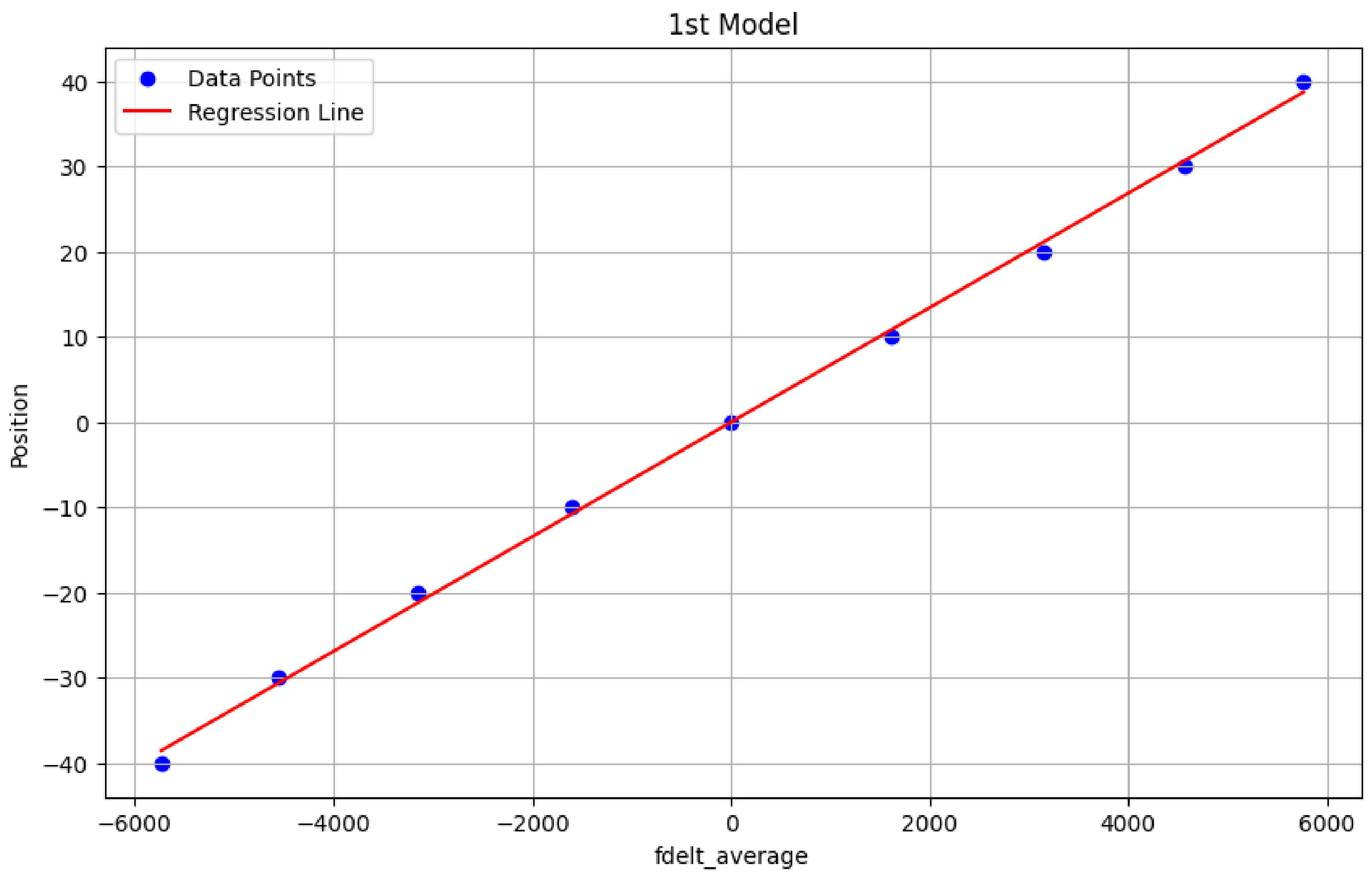
Models Deployed:

1. basic *model:*

This model can be perceived as our foundational approach. Its design was to understand the relationship between individual *fdelt* values for each position and the respective positions themselves.

Results:

- Regression Equation: **`position = 0.00672 * avg_fDelT - 0.05479`**
- Mean Squared Error (MSE) during training: 0.9490



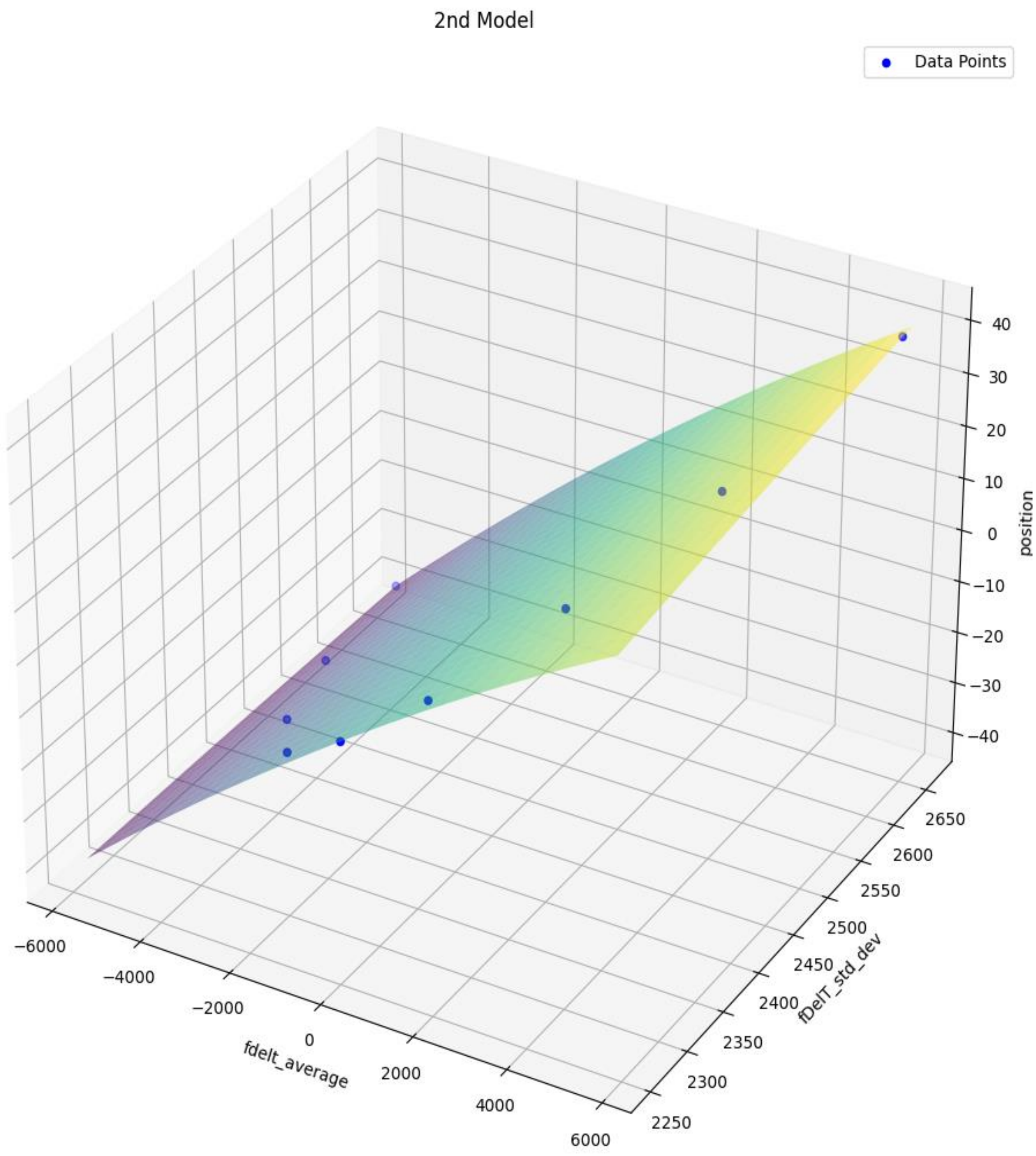
2.

Advanced Model:

Transitioning from the basic model, this version hones in on both the average and the standard deviation of *fdelt*. By combining these two features, the model has a richer set of information to understand the intricacies of muon behavior.

Results:

- Regression Equation: **`position = -55.79589 + 0.00165 * avg_fDelT + 0.03792 * std_dev_fDelT - 0.00001 * (std_dev_fDelT)^2`**
- Mean Squared Error (MSE) during training: `0.003999083152487563`.



Evaluation Metrics:

For our models' assessment, the Mean Squared Error (MSE) was employed during the training phase. MSE's strength lies in its ability to amplify larger errors, making it a stringent metric. In the testing phase, the metric of choice was the mean difference with modulus between the predicted and actual position values. This straightforward metric provides a clear picture of the average deviation in predictions.

Validation/Testing

Traditional Method:

- For `Data_+45.csv`: Predicted distance = 41.2
- For `Data_-45.csv`: Predicted distance = -41.6

- Average mean difference across the test set: 3.6 units

1st Model:

- For `Data_+45.csv`: Predicted distance = 41.38 cm
- For `Data_-45.csv`: Predicted distance = -41.18 cm
- Average mean difference across the test set: 3.72 cm

2nd Model:

- For `Data_+45.csv`: Predicted distance = 45.04 cm
- For `Data_-45.csv`: Predicted distance = -44.21 cm
- Average mean difference across the test set: 0.413 cm
- Mean Absolute Percentage Error (MAPE): 0.9222% Accuracy: 99.0778%

conclusion : The ML model achieved an accuracy of 99.0778%, contrasting with 92% of

conventional method. Its focus on the time feature, unlike conventional methods that also account for charge,

allows for streamlined handling of large datasets and minimal storage requirements.

Discussion and Implications other than accuracy:

Single feature model

An interesting takeaway from our models is their performance is predicated solely on the time feature. This demonstrates the substantial influence of time on muon behaviour in our experiments, as even without leveraging the charge feature, our models achieved noteworthy accuracy.

Data Storage and Computation Efficiency:

A standout implication of our research is the revelation that the entire dataset isn't mandatory for computation. By prioritizing the average and standard deviation for each position iteration, we've fundamentally altered the data storage paradigm. Regardless of whether a dataset for a specific position swells to hundreds of terabytes, our approach ensures that, theoretically, a mere two bytes are ample for storage. This breakthrough not only addresses storage concerns but also promises a substantial decrease in computation time. Importantly, this methodology circumvents the need to engage with every individual data point within a position dataset, further bolstered by our decision to neither remove nor modify outliers.

Groundwork for Future Research:

Towards Scintillator-free Mechanisms:

The horizon looks promising. Our current research trajectory has spurred pivotal debates regarding the plausibility of mapping muon trajectories sans any dependency on scintillators. If realized, this could instigate a seismic shift in the discipline. Furthermore, refining our model to ensure its independence could lead to innovative real-time applications. For instance, by capturing two successive images—one authentic and the other at a position +-(x+5), where 'x' is a tenfold multiple—we might be able to incorporate this data as an additional influential feature in our model.

Charging Ahead with the Charge Feature:

Our current accomplishments lay a robust groundwork for ventures into uncharted territories. An exciting prospect lies in harnessing the 'charge' feature more intensively, potentially paving the way for advanced applications, ranging from intricate imaging processes to the development of unique references or benchmarks.

Glossary for Machine Learning Terms in the Paper

- **Algorithm:** A set of defined steps and computations used to train a model or solve problems with data.
- **Data Pre-processing:** The preliminary stage where raw data is cleaned, transformed, and structured, making it suitable for model training.
- **Dataset:** A structured collection of data points. Each row typically represents a sample, and each column represents a specific feature of the sample.
- **Feature:** A distinct measurable characteristic or attribute of a data point. For this paper, examples include *avg_fDelT* and *std_dev_fDelT*.
- **Histogram:** A graphical representation of data distribution, often using bars of varying heights. In this paper, it's used to visualize the distribution of *fDelT* values across positions.
- **Independent Variable (Predictor Variable):** Variables that are used to predict the outcome or dependent variable's value. In the paper's context, it includes metrics like *avg_fDelT*.
- **Dependent Variable (Response Variable):** The variable that we aim to predict or understand. In this paper, it refers to the muon's position.
- **Linear Regression:** A statistical approach used to model a linear relationship between a dependent variable and one or more independent variables. This is the foundation of the paper's 1st model.
- **Machine Learning:** An analytical methodology allowing computers to refine or develop algorithms by learning from data, hence making informed predictions or decisions.
- **Mean Squared Error (MSE):** A metric that quantifies the average of the squared differences between observed and predicted values, indicating the model's accuracy.
- **Model:** A representation derived from data using a specific machine learning algorithm. It captures patterns and relationships from the data for future predictions.
- **Outliers:** Data points that diverge significantly from the rest of the dataset, often requiring special consideration due to their potential to skew results.
- **Polynomial Regression:** Regression analysis that models the relationship between a dependent variable and one or more independent variables using a polynomial equation. This is integral to the paper's 2nd model.
- **Multiple Regression:** Regression analysis that uses two or more independent variables to predict the outcome of a dependent variable. This is integral to the paper's 2nd model.
- **Regression Equation:** The mathematical representation capturing the relationship between independent and dependent variables.
- **Scikit-learn library:** A widely-used machine learning library in Python, renowned for its diverse tools and functionalities for data analysis and modeling.
- **Training:** The phase where a machine learning model "learns" by processing and analyzing data. It identifies patterns and relationships to make future predictions.
- **Validation/Testing:** The phase where a trained model's accuracy and efficiency are evaluated using a separate dataset, ensuring its predictions are reliable and not just tailored to the training data.