Machine Learning Aided Modeling of Granular Materials

A B.Tech project report submitted in partial fulfillment of the requirements for the award of the degree of

Bachelor of Technology

by

Ananya Bansal

(2022CHB1041)

Under the Supervision of

Dr. Saikat Roy

Department of Chemical Engineering

INDIAN INSTITUTE OF TECHNOLOGY, ROPAR

May, 2025

Declaration

I affirm that this submission reflects my original ideas, expressed in my own words. Where I have included the ideas or words of others, I have appropriately cited and referenced the original sources. I also confirm that I have upheld all principles of academic honesty and integrity, and have neither misrepresented nor fabricated or falsified any idea, data, fact, or source in this work. I understand that any breach of these principles may result in disciplinary action by the Institute and could also lead to legal consequences from improperly cited sources.

Ananya Bansal
Entry No: 2022CHB1041
Date: 12/05/2025
Place: IIT Ropar

Acknowledgements

I would like to take this opportunity to express my heartfelt gratitude to everyone who has supported and guided me throughout this project. I am especially thankful to my supervisor, Dr. Saikat Roy, for his invaluable guidance and time and for providing me with the opportunity to work on such a meaningful and insightful project. This project has truly been a challenging yet incredibly rewarding journey, and pushed me to step out of my comfort zone and learn something new, which I'm thankful for. I also acknowledge the efforts of Mr. Divas Singh Dagur, who helped me with the procurement of data required for this project.

Ananya Bansal

Abstract

This project aims to develop a Graph Neural Network (GNN) model to predict the contact forces in granular assemblies under uniaxial compression. Building on the work of Cheng and Wang (2022), we represent the granular assemblies as graphs, where particles are nodes and contacts are edges. The GNN model is designed to learn from this graph structure and predict normalized particle maximum normal contact forces (NPMNCFs).

The methodology involves using Discrete Element Method (DEM) simulations to generate data on granular assemblies, creating graph representations, and developing a GNN model with edge features, node features and target features. While the model development is ongoing, initial implementation has been completed, and challenges in achieving desired accuracy have been identified.

While initial models faced limitations, refinement through data normalization and targeted strain selection led to significant improvements. A final model trained on a single strain level achieved a Pearson correlation coefficient of 0.999 across training, validation, and testing sets, indicating strong predictive performance and the potential for application in broader particulate systems. This report details my methodology, the challenges encountered throughout the semester, the strategies implemented to address them, and both preliminary and final results.

Contents

List of Figures

Abbreviations

GNN	Graph Neural Network
NPMNCF	Normalized Particle Maximum Normal Contact Force
DEM	Discrete Element Method
MLP	Multi-Layer Perceptron

Symbols

G	Graph
V	Set of vertices (nodes)
E	Set of edges
w_v	Node weights
w_e	Edge weights

Chapter 1

Introduction

1.1 Background

Granular materials play a crucial role in various industries and natural processes. Understanding their behavior under different conditions is essential for applications ranging from Chemical Engineering to pharmaceutical manufacturing. Recent advancements in machine learning, particularly Graph Neural Networks (GNNs), have opened new avenues for modeling complex systems like granular materials.

1.2 Problem Statement

The primary challenge addressed in this project is the accurate prediction of interparticle contact forces in granular materials based on particle kinematics and contact data. Traditional methods often rely on computationally intensive simulations or simplified models that may not capture the full complexity of granular systems.

1.3 Project Objectives

The main objectives of this study are:

- Develop a GNN model to predict NPMNCFs in granular materials under uniaxial compression.
- Validate the model using data from DEM simulations.
- Analyze the model's performance and identify areas for improvement.

1.4 Scope of Work

This mid-term report covers:

- Literature review on granular materials and GNNs.
- Methodology for data generation and model development.
- A brief summary of the work done until mid-semester.
- Preliminary results and challenges encountered.
- Steps taken to overcome challenges and final results.

Chapter 2

Literature Review

2.1 Introduction

This chapter presents a comprehensive review of relevant literature, focusing on the current state of knowledge regarding granular materials, force transmission in particulate systems, and the application of machine learning techniques, particularly Graph Neural Networks, in material science.

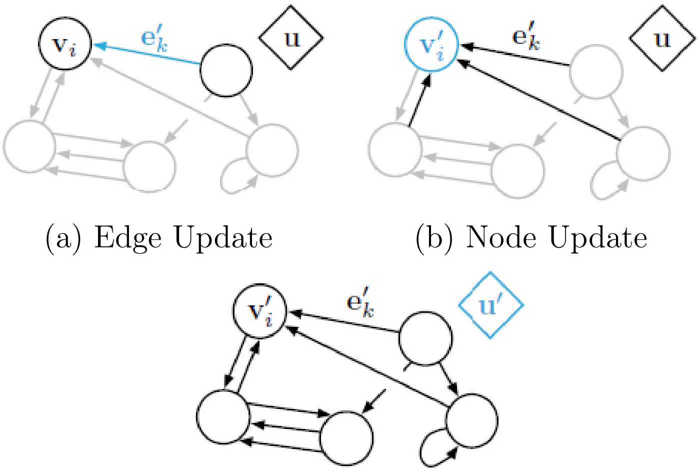
2.2 Literature Review

Extensive research has been conducted on force transmission in granular materials using techniques such as the Discrete Element Method (DEM) and experimental approaches like X-ray tomography. Recent advancements have introduced machine learning-based models, including Support Vector Machines and Neural Networks, to predict inter-particle contact forces. The incorporation of Graph Neural Networks

(GNNs) is a relatively new approach that has shown promising results in capturing complex particle interactions.

Battaglia et. al (2018) introduced a general framework for GNNs, showcasing their ability to capture and learn from intricate relationships within data structures. This approach has been successfully extended to physical systems, demonstrating the versatility of GNNs in simulating complex physics.

Figures 2.1 and 2.2 show the description of the GNN model in Battaglia's paper:



(a) Edge Update

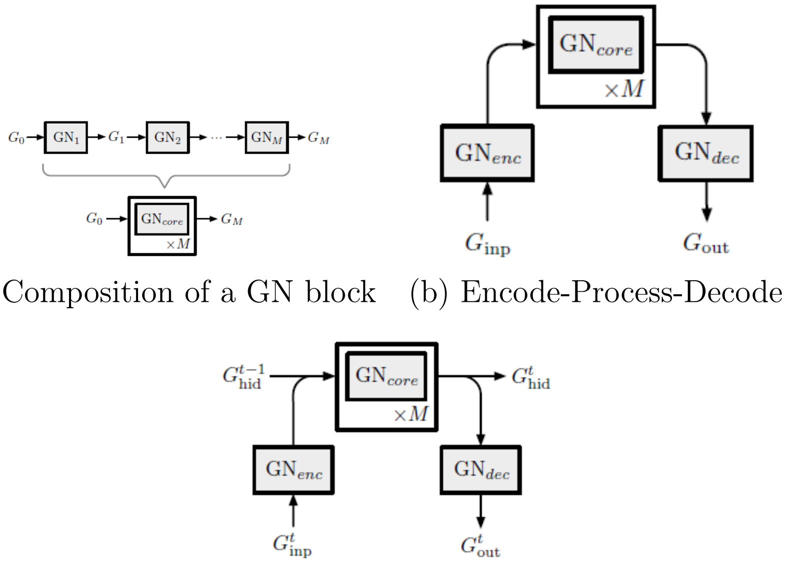
(b) Node Update

(c) Global Function Update

FIGURE 2.1: Updates in a GN block

In the context of granular materials, Zhuang Cheng, Jianfeng Wang (2022) developed a GNN model to predict normalized particle maximum normal contact forces (NPMNCFs) in granular assemblies under uniaxial compression. Their model, which forms the basis of our current project, demonstrated high accuracy in predicting NPMNCFs using particle kinematics and contact data, showing strong agreement with Discrete Element Method (DEM) simulations.

This paper presents a Graph Neural Network (GNN) model to estimate contact forces in granular materials under uniaxial compression. The model predicts normalized particle maximum normal contact forces (NPMNCFs) using particle kinematics and inter-particle contact data during loading increments.



(a) Composition of a GN block (b) Encode-Process-Decode

(c) Recurrent GN architecture

FIGURE 2.2: GNN Model Architecture

The GNN uses an encode-process-decode architecture, with nodes representing particles and edges representing contacts. Node attributes include normalized particle radii, while edge attributes capture contact evolution and relative displacements. As per Zhuang Cheng, Jianfeng Wang (2022), the model is trained on data from 3D discrete element modeling (DEM) simulations of 50 granular assemblies with different initial microstructures. It achieves an average Pearson correlation coefficient of 0.89 between predicted and actual NPMNCFs.

Key findings include:

- Predicted NPMNCFs show a unified probability density function across different compression stresses, consistent with DEM simulations.
- The model accurately captures particle size-dependent characteristics of NPMNCFs.

- Predicted results reflect force transmission patterns (force chains) observed in DEM simulations.

The study demonstrates the GNN model's capability to predict inter-particle contact forces while reflecting statistical and physical laws of granular media. This approach could be applied to experimental data to estimate contact forces in real granular systems or potentially to data generated for colloidal systems.

More recently, Rituparno Mandal, Corneel Casert, Peter Sollich (2022) extended the application of GNNs to predict force chains in jammed solids, further demonstrating the potential of these models in understanding complex force transmission in particulate systems.

2.3 Research Gaps

Despite these advancements, several research gaps remain:

1. **Application to diverse granular systems:** Most studies focus on specific types of granular materials. The generalizability of GNN models to different particle sizes, shapes, and material properties needs further investigation.
2. **Integration with experimental data:** While models have been validated against DEM simulations, their performance with experimental data from real granular systems requires further study.
3. **Application to colloidal systems:** The potential of GNN models in predicting forces in colloidal systems, which exhibit both similarities and differences to granular materials, remains largely unexplored.
4. **Limitations of GNN:** As stated by Battalgia, "GNN cannot be guaranteed to solve some classes of problems, such as discriminating between certain

> non-isomorphic graphs. Certain notions, like recursion, control flow, and conditional iteration are not straightforward to represent with graphs."

2.4 Summary

This literature review highlights the significant progress made in understanding and modeling granular materials, particularly through the application of machine learning techniques. The work of Cheng and Wang provides a solid foundation for our current project, while the identified research gaps inform our objectives and future directions.

Chapter 3

Methodology

3.1 Introduction

This chapter outlines the methodology employed in developing our Graph Neural Network (GNN) model for predicting contact forces in granular materials. We describe the overall approach, data preparation, model architecture, and evaluation metrics.

3.2 Methodology

The methodology follows the key steps as:

1. **Data Generation:** We use DEM simulations to generate data on granular assemblies under uniaxial compression. This includes particle positions, velocities, individual force vectors and contact forces at various stages of compression.

2. **Graph Representation:** We represent each granular assembly as a graph, where particles are nodes and contacts & inter-particle distances define edges. Node attributes include particle properties, while edge attributes capture contact information and relative displacements and velocities.
3. **GNN Model Development:** We implement a GNN model using an encode-process-decode architecture, similar to that described by Zhuang Cheng, Jianfeng Wang (2022).
4. **Model Training and Validation:** The model is trained on a subset of the DEM simulation data and validated on a separate test set.
5. **Performance Evaluation:** We assess the model's performance using metrics such as training & testing loss, mean squared error, R2 score, etc.
6. **Iterative Refinement:** Based on initial results, we refine the model architecture and training process to improve performance.

3.3 Models and Algorithms

3.3.1 Graph Neural Networks (GNNs)

Graph Neural Networks (GNNs) are a class of deep learning models designed to operate on graph-structured data. They are particularly well-suited for analyzing and making predictions about systems that can be represented as graphs, such as granular materials where particles are nodes and contacts between particles are edges. A graph G is denoted by G = (V, E). In the presence of node weights and edge weights, a weighted graph can be defined as G = (V, E, w_v, w_e).

GNNs work by iteratively updating node representations based on information from neighboring nodes. This process, often called message passing, allows the model to capture both local and global structural information of the graph.

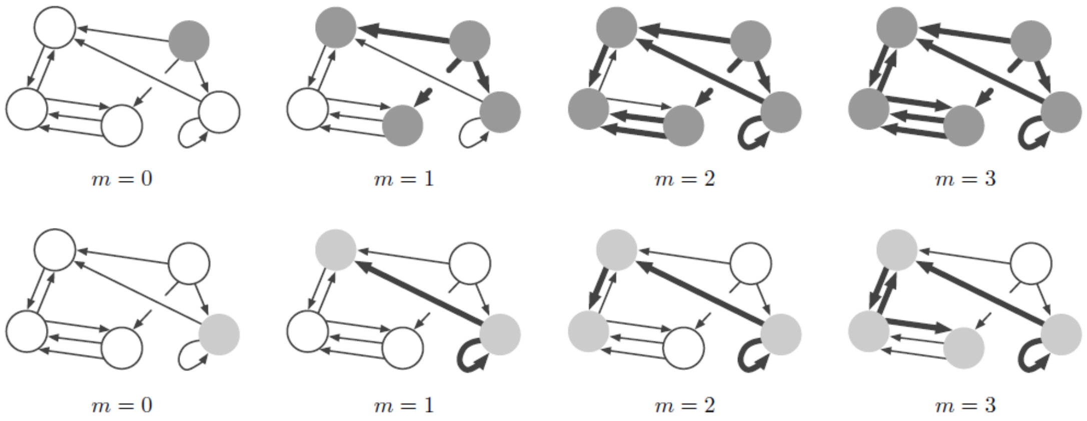


FIGURE 3.1: Example of Message Passing

Key features of GNNs include:

- Ability to handle variable-sized inputs.
- Preservation of graph structure in the learning process.
- Capacity to learn hierarchical representations of graph data.

In this project, we utilize GNNs to predict contact forces in granular materials under compression.

3.3.2 Model Architecture

Our GNN model architecture consists of three main components:

1. **Encoder:** Two-layer Multi-Layer Perceptrons (MLPs) encode node and edge attributes into latent representations.
2. **Processor:** A series of graph network blocks update node and edge representations through message passing.

3. **Decoder:** Another set of MLPs decode the final graph representation to predict NPMNCFs.

The model is implemented using TensorFlow and the Graph Nets library. We use the Adam optimizer for training, with gradient clipping to prevent exploding gradients.

3.3.3 Mathematics behind GNNs

The core operations in a Graph Neural Network (GNN) can be described mathematically as follows:

1. **Message function:**

$$m_{v,w}^{(k)} = M^{(k)} \left(h_v^{(k-1)}, h_w^{(k-1)}, e_{vw} \right)$$

2. **Aggregation function:**

$$a_v^{(k)} = A^{(k)} \left(\{ m_{v,w}^{(k)} : w \in N(v) \} \right)$$

3. **Update function:**

$$h_v^{(k)} = U^{(k)} \left(h_v^{(k-1)}, a_v^{(k)} \right)$$

Where:

- $h_v^{(k)}$ is the feature vector of node v at layer k
- e_{vw} is the feature vector of the edge between nodes v and w
- $N(v)$ is the set of neighbors of node v
- $M^{(k)}, A^{(k)}, U^{(k)}$ are learnable functions at layer k

These operations are performed iteratively for a fixed number of layers or until convergence.

A single GNN layer on a node i can be formulated as:

$$h_i = \sigma(W_1 \cdot h_i + \sum_{j \in N_i} W_2 \cdot h_j)$$

Here, we use the sum aggregation and a simple feed-forward layer as functions F and H. The dimensions of W_1 and W_2 should commute properly with the node embeddings. If $h_i \in \mathbb{R}^d$, then $W_1, W_2 \subseteq \mathbb{R}^{d' \times d}$, where d' is the embedding dimension.

When working with adjacency matrices, the GNN forward pass can be expressed as:

$$H^{l+1} = \sigma(\tilde{A} H^l W^l)$$

Where:

- $\tilde{A} = A + I_N$ is the adjacency matrix with self-loops added,
- H^l is the node feature matrix at layer l,
- W^l is a learnable parameter matrix.

In these equations:

- σ represents an activation function (e.g., ReLU).

- I_N is the identity matrix to account for self-loops.

These mathematical formulations form the foundation of GNNs, enabling them to model relationships in graph-structured data effectively.

In the context of predicting forces in granular materials, this process allows the GNN to learn how forces are transmitted through the material, taking into account both local interactions between neighboring particles and the global structure of the particle network.

3.4 Summary

This methodology combines physics-based simulations with advanced machine learning techniques to create a predictive model for granular materials. The graph-based approach allows us to capture the inherent structure of granular assemblies, while the GNN architecture enables learning complex relationships between particle properties and resulting forces.

Chapter 4

Work Done up to Mid-Semester

4.1 Introduction

This chapter presents a summary of the work done and learnings made up to the mid-term evaluation.

4.2 Initial Approach and Results

My initial approach for developing a GNN model to predict contact forces in granular materials included:

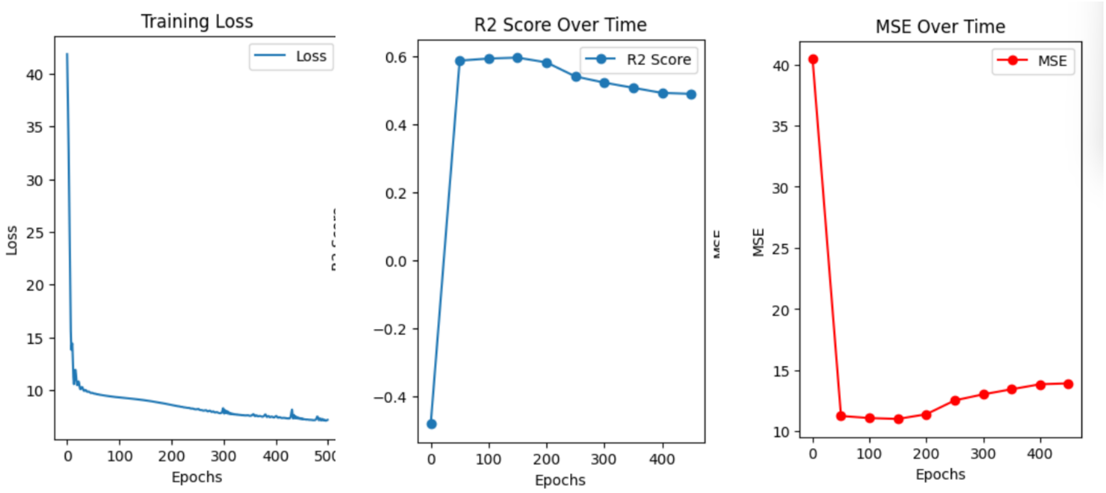
- Using data from a single granular assembly.
- The model was trained on a subset of particles within the assembly and the remaining particles were used for model validation.
- The initial data did not take into account the velocity vectors and assumed a static assembly configuration.

After training the model with this approach, I obtained the following validation metrics:

Best score according to Loss:

- Loss: 7.252278804779053
- R2 Score: 0.489762961864471 44
- MSE: 13.928693771362305

Figure 4.1 shows the plots for Loss, R2 score and MSE for the above model over time/epochs.



(a) Loss over time (b) R2 score over time (c) MSE over time

FIGURE 4.1: Evaluation metrics for preliminary model

These results indicated that while the model showed some predictive capability, its performance fell short of the desired accuracy. Upon analysis, I identified several limitations in my initial approach that likely contributed to the suboptimal performance.

4.3 Next approach that was worked upon

To address the limitations identified in my initial approach and improve the model's performance, I outlined the following steps:

1. Not considering particles as static.
   - Generate data from multiple granular assemblies using Discrete Element Method (DEM) simulations.
   - Incorporate particle dynamics, including velocity vectors, relative distance between the particles, etc.
2. Improved Model Training Strategy:
   - Use data from 5 entire granular assemblies for model training.
   - Use 1 assembly as validation data.
   - Test the model on 4 separate assemblies to ensure generalizability.
3. Performance Evaluation and Iteration:
   - Continuously assess model performance using metrics such as loss, R2 score, and MSE.
   - Iterate on model design and training process based on results.

4.4 Summary

The initial approach to develop a GNN model for predicting contact forces in granular materials used data from a single assembly, trained on a subset of particles, and assumed a static configuration without velocity vectors. This approach yielded suboptimal results. Proposed improvements included using particle dynamics for 10 granular assemblies, refining GNN architecture and optimizing hyperparameters.

Chapter 5

Post-Midsem Progress

5.1 Introduction

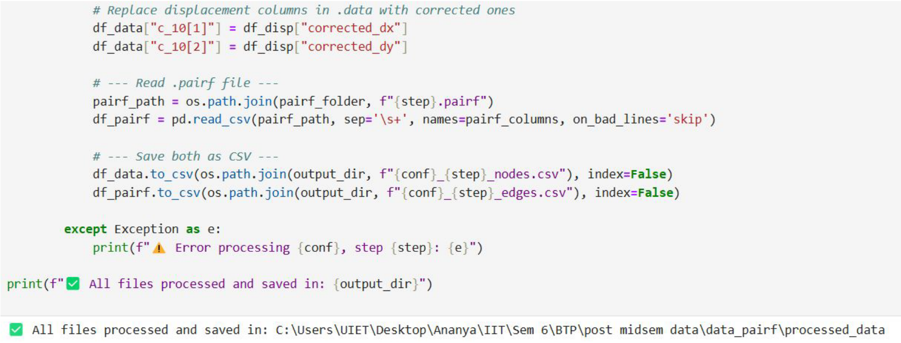
This chapter outlines the work done, approaches used post midsem, challenges faced and steps taken to overcome them, and a comparison between the preliminary and final results.

5.2 Data Procurement & Preprocessing

Data now procured was of 10 different configurations, wherein each nodes file had 4000 particles and each configuration had about 200 nodes files for different values of strain. Accordingly, there were around 200 files each for edge features of every configuration with variable data size. This gave a total of about 4000 files and a huge dataset of size approximately equal to 10^7 data points.

The preprocessing for these files was done as follows:

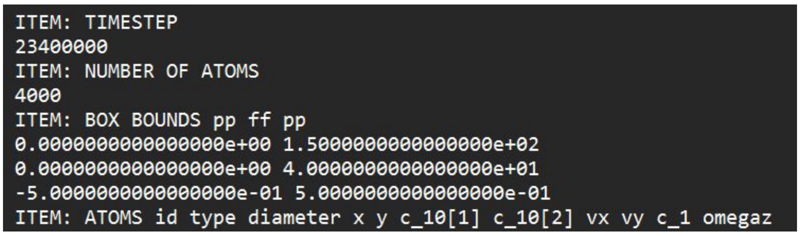
- Cleaned the node and pairf (edge) data and converted all 4000 files to csv format, putting them inside 1 folder for effective training.
- Replaced previous displacements in .data files (node files) with corrected values of displacements in x and y coordinates, which needed to be calculated after each step again.



```
            # Replace displacement columns in .data with corrected ones
            df_data["c_10[1]"] = df_disp["corrected_dx"]
            df_data["c_10[2]"] = df_disp["corrected_dy"]

            # --- Read .pairf file ---
            pairf_path = os.path.join(pairf_folder, f"{step}.pairf")
            df_pairf = pd.read_csv(pairf_path, sep='\s+', names=pairf_columns, on_bad_lines='skip')

            # --- Save both as CSV ---
            df_data.to_csv(os.path.join(output_dir, f"{conf}_{step}_nodes.csv"), index=False)
            df_pairf.to_csv(os.path.join(output_dir, f"{conf}_{step}_edges.csv"), index=False)

        except Exception as e:
            print(f"⚠ Error processing {conf}, step {step}: {e}")

print(f"✅ All files processed and saved in: {output_dir}")
```

```
✅ All files processed and saved in: C:\Users\UIET\Desktop\Ananya\IIT\Sem 6\BTP\post midsem data\data_pairf\processed_data
```

- Removed unnecessary headers and text from all files for model building, and added appropriate headers for all.



```
ITEM: TIMESTEP
23400000
ITEM: NUMBER OF ATOMS
4000
ITEM: BOX BOUNDS pp ff pp
0.0000000000000000e+00 1.5000000000000000e+02
0.0000000000000000e+00 4.0000000000000000e+01
-5.0000000000000000e-01 5.0000000000000000e-01
ITEM: ATOMS id type diameter x y c_10[1] c_10[2] vx vy c_1 omegaz
```

- Added relative distances, velocities etc. columns to .pairf (edges) files.

5.3 Development & Implementation of the GNN Model

The initial GNN model was based on the approach mentioned above, incorporating all of the data procured.

5.3.1 Challenges Faced and Solutions

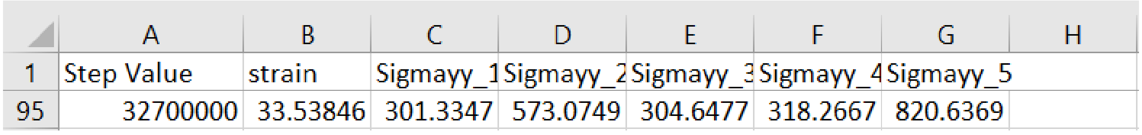
During the implementation and refinement of the GNN model, several challenges were encountered, necessitating adaptive solutions to ensure project success. These challenges and their respective solutions are detailed below:

1. **Problem:** The edges (.pairf) files did not have a fixed number of data points, which made it difficult to read and use all the files.

   Solution: Implemented a code to get the data size for all edges files and ran the model iteratively.

2. **Problem:** Owing to the huge data size (10^7 data points), the model could not effectively train and kept facing crashes, as we had expected in an earlier discussion.

   Next Approach: Working on our discussed next approach, one specific strain & stress step was chosen for all the 10 configurations and the GNN model was then implemented for that particular strain value. This way, the data used now comprised of about 100,000 data points, which would still be a manageable number.



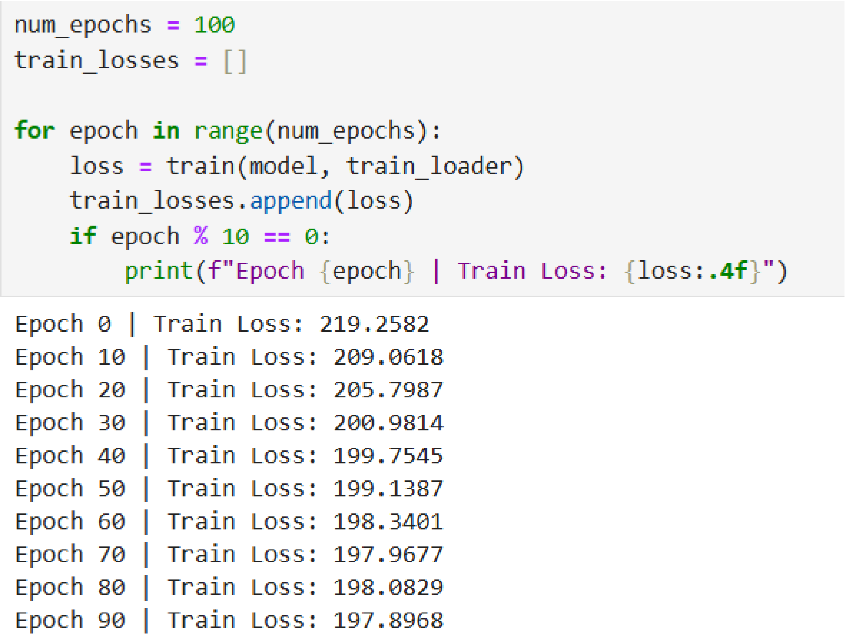
	A	B	C	D	E	F	G	H
1	Step Value	strain	Sigmayy_1	Sigmayy_2	Sigmayy_3	Sigmayy_4	Sigmayy_5	
95	32700000	33.53846	301.3347	573.0749	304.6477	318.2667	820.6369	

FIGURE 5.1: Step value chosen along with the strain & stress values for 5 configurations

5.3.2 Single Step Value Approach

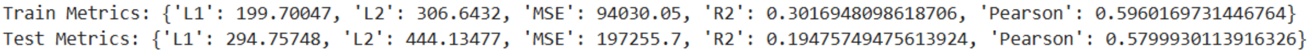
As shown in figure 5.1, the step value of 32700000 was chosen for realistic and useful strain and stress values. The model trained successfully and showed a decreasing loss over epochs, however, it gave sub-par results that needed to be significantly improved.

The results obtained were as shown in 5.3.2:



```
num_epochs = 100
train_losses = []

for epoch in range(num_epochs):
    loss = train(model, train_loader)
    train_losses.append(loss)
    if epoch % 10 == 0:
        print(f"Epoch {epoch} | Train Loss: {loss:.4f}")
```

```
Epoch 0 | Train Loss: 219.2582
Epoch 10 | Train Loss: 209.0618
Epoch 20 | Train Loss: 205.7987
Epoch 30 | Train Loss: 200.9814
Epoch 40 | Train Loss: 199.7545
Epoch 50 | Train Loss: 199.1387
Epoch 60 | Train Loss: 198.3401
Epoch 70 | Train Loss: 197.9677
Epoch 80 | Train Loss: 198.0829
Epoch 90 | Train Loss: 197.8968
```

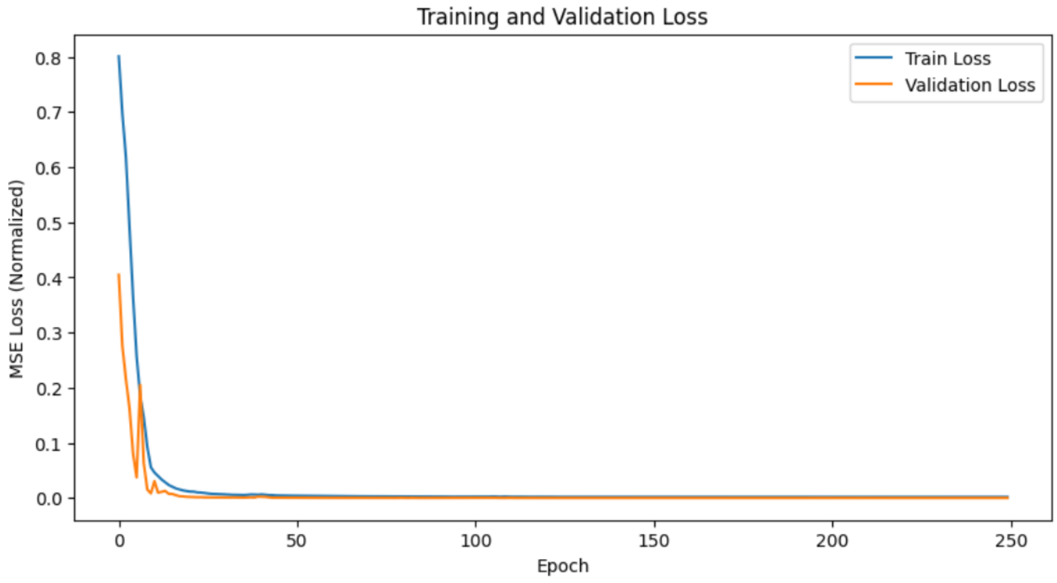


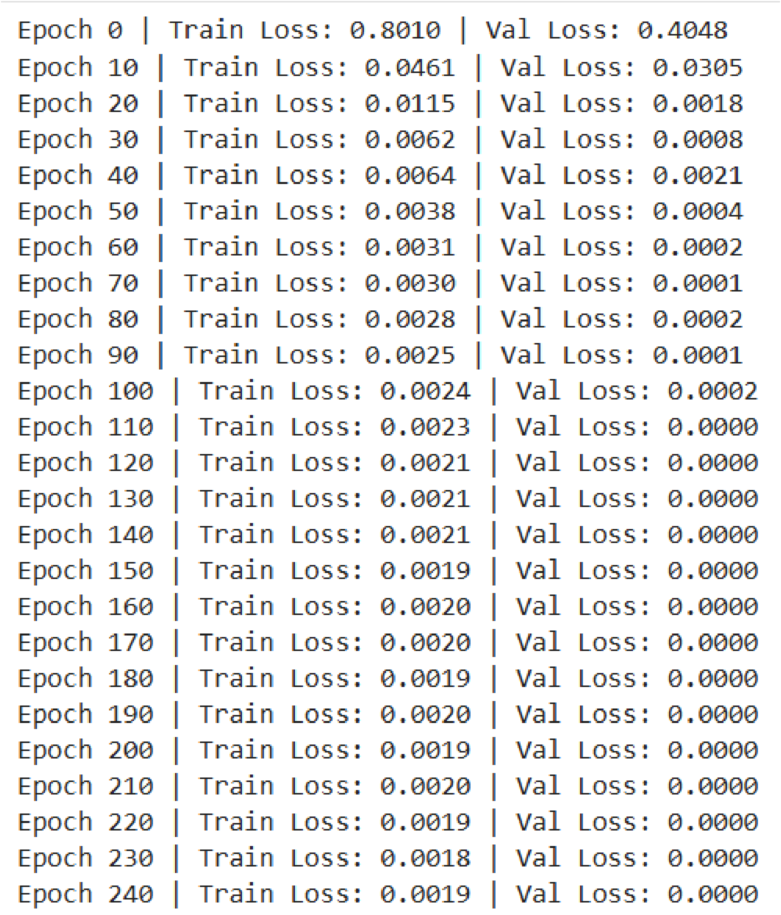
```
Train Metrics: {'L1': 199.70047, 'L2': 306.6432, 'MSE': 94030.05, 'R2': 0.3016948098618706, 'Pearson': 0.5960169731446764}
Test Metrics: {'L1': 294.75748, 'L2': 444.13477, 'MSE': 197255.7, 'R2': 0.19475749475613924, 'Pearson': 0.5799930113916326}
```

As we can see, this showed that despite the model showing a promising start as GNN does require extensive hyperparameter tuning, there was a great need for certain measures to improve the performance. Upon deliberation, the following solutions were worked out:

- Need to increase the number of training epochs so as to minimize loss.
- Since the Normal Contact Force is a huge number and would not match the order of the training features, there is a massive need to normalize the entire data before training (i.e. scale down from -1 to 1 or 0 to 1), to get an accurate measure of the error.
- Hyperparameter tuning for number of hidden layers,etc. required.
- Incorporation of Adam optimizer to get the most optimal results, as used in the Zhuang Cheng, Jianfeng Wang (2022) paper.

Upon implementing all of the above, I obtained a drastic improvement in my results, showing successful implementation of the GNN model, as shown in the figures below :

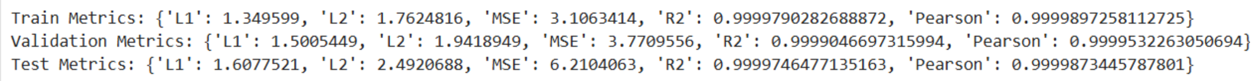




```
Epoch 0 | Train Loss: 0.8010 | Val Loss: 0.4048
Epoch 10 | Train Loss: 0.0461 | Val Loss: 0.0305
Epoch 20 | Train Loss: 0.0115 | Val Loss: 0.0018
Epoch 30 | Train Loss: 0.0062 | Val Loss: 0.0008
Epoch 40 | Train Loss: 0.0064 | Val Loss: 0.0021
Epoch 50 | Train Loss: 0.0038 | Val Loss: 0.0004
Epoch 60 | Train Loss: 0.0031 | Val Loss: 0.0002
Epoch 70 | Train Loss: 0.0030 | Val Loss: 0.0001
Epoch 80 | Train Loss: 0.0028 | Val Loss: 0.0002
Epoch 90 | Train Loss: 0.0025 | Val Loss: 0.0001
Epoch 100 | Train Loss: 0.0024 | Val Loss: 0.0002
Epoch 110 | Train Loss: 0.0023 | Val Loss: 0.0000
Epoch 120 | Train Loss: 0.0021 | Val Loss: 0.0000
Epoch 130 | Train Loss: 0.0021 | Val Loss: 0.0000
Epoch 140 | Train Loss: 0.0021 | Val Loss: 0.0000
Epoch 150 | Train Loss: 0.0019 | Val Loss: 0.0000
Epoch 160 | Train Loss: 0.0020 | Val Loss: 0.0000
Epoch 170 | Train Loss: 0.0020 | Val Loss: 0.0000
Epoch 180 | Train Loss: 0.0019 | Val Loss: 0.0000
Epoch 190 | Train Loss: 0.0020 | Val Loss: 0.0000
Epoch 200 | Train Loss: 0.0019 | Val Loss: 0.0000
Epoch 210 | Train Loss: 0.0020 | Val Loss: 0.0000
Epoch 220 | Train Loss: 0.0019 | Val Loss: 0.0000
Epoch 230 | Train Loss: 0.0018 | Val Loss: 0.0000
Epoch 240 | Train Loss: 0.0019 | Val Loss: 0.0000
```

FIGURE 5.2: Normalized loss over epochs

The performance metrics, for the actual unscaled data were as follows:



```
Train Metrics: {'L1': 1.349599, 'L2': 1.7624816, 'MSE': 3.1063414, 'R2': 0.9999790282688872, 'Pearson': 0.9999897258112725}
Validation Metrics: {'L1': 1.5005449, 'L2': 1.9418949, 'MSE': 3.7709556, 'R2': 0.9999046697315994, 'Pearson': 0.9999532263050694}
Test Metrics: {'L1': 1.6077521, 'L2': 2.4920688, 'MSE': 6.2104063, 'R2': 0.9999746477135163, 'Pearson': 0.9999873445787801}
```

Hence, I was able to now obtain near perfect results for a total dataset of 10 configurations. The accuracy might be very slightly compromised if more of these configurations are taken, but the model would still give quite meaningful results.

The plots for actual vs predicted data for 2 configurations each of testing and training data are shown below:

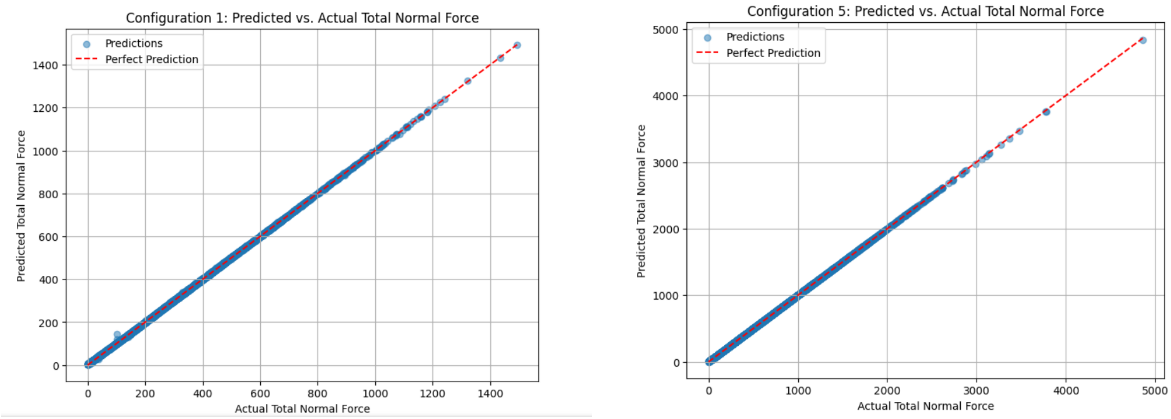


FIGURE 5.3: Training configurations

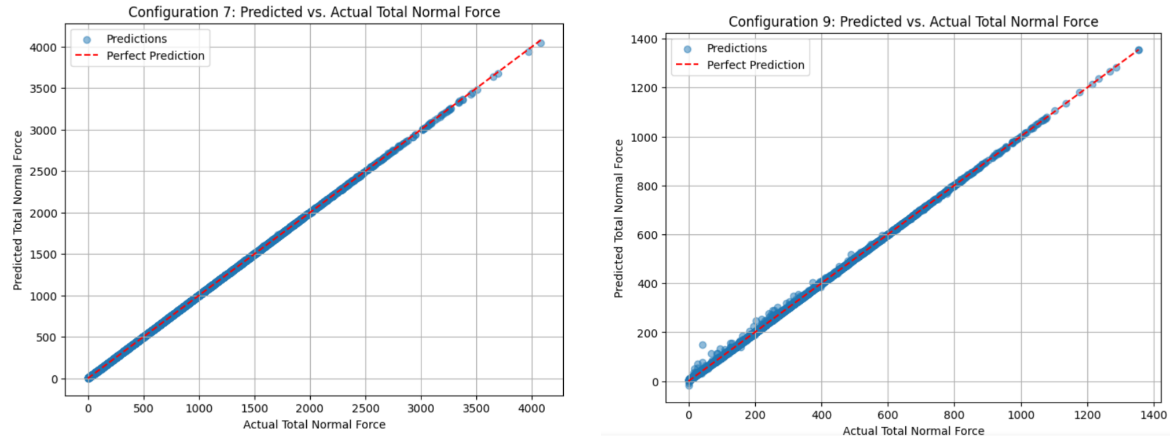


FIGURE 5.4: Testing configurations

5.4 Summary

This chapter detailed the systematic progression from data procurement and preprocessing to the successful development and implementation of a Graph Neural Network (GNN) model for predicting normal contact forces in granular materials. Initially, challenges related to data volume, file inconsistencies, and model instability were identified. These were addressed through a series of targeted interventions, including the selection of a single representative strain step to reduce computational load, comprehensive data cleaning, and incorporation of relative feature enhancements.

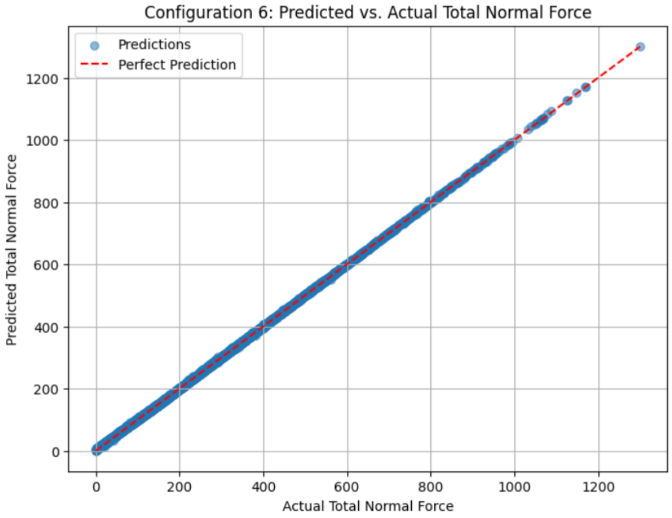


FIGURE 5.5: Validation configuration

Further improvements such as feature normalization, optimizer selection, and hyperparameter tuning led to a significant enhancement in model performance. The final model achieved near-perfect prediction accuracy across training, validation, and testing datasets, with a Pearson correlation coefficient close to 0.999. These results validate the feasibility of using GNNs for contact force estimation and pave the way for extending this approach to more complex systems in Chemical Engineering, such as colloidal suspensions, in future work.

Chapter 6

Conclusions and Discussion

6.1 Introduction

This chapter outlines the conclusions of the study, as well as the practical implications that can be drawn from the results.

6.2 Study Findings

- The final Graph Neural Network (GNN) model achieved near-perfect accuracy in predicting normalized particle maximum normal contact forces (NPM-NCFs).
- Performance metrics across training, validation, and testing datasets indicated high reliability and generalization.
- The use of normalized features and strategic selection of a single strain step significantly improved model accuracy.

- Results showed strong agreement with physical behavior observed in Discrete Element Method (DEM) simulations, which are considered to be 99% accurate.

6.3 Research Contributions

My work contributes to research as:

- It shows successful implementation of a GNN model for force prediction in granular materials using normalized and structured data.
- It demonstrated the feasibility of reducing computational complexity by using a representative strain step.
- It showed the capability of GNNs in capturing inter-particle physics without the need for full DEM simulation.
- Contributed a workflow that could potentially be extended to other particulate systems.

6.4 Practical Implications

- The model can serve as a fast and accurate surrogate for DEM in applications requiring rapid force estimation.
- This can benefit various engineering domains such as chemical, civil, and materials engineering where granular materials are commonly used.
- Offers a new data-driven approach for evaluating stress distributions in granular packings.

6.5 Study Limitations and Scope for Future Work

The limitations faced & future scope include:

- Need for validation with experimental data from real granular systems.
- GNN cannot always solve some classes of problems, like discriminating between certain non-isomorphic graphs. (Battaglia et. al, 2018)
- Potential for extension to dynamic loading conditions and more complex particle shapes, given systems with higher computational power.
- Exploration of the model's applicability to colloidal systems, accounting for additional inter-particle forces.

6.6 Summary

This final summary reinforces that a robust and highly accurate GNN model was developed for force prediction in granular systems. It marks the project as a success in demonstrating the practicality of machine learning in material science and highlights potential for further development in both research and industrial applications.

References

Battaglia et. al, 2018. Relational inductive biases, deep learning, and graph networks. URL: https://arxiv.org/abs/1806.01261.

Rituparno Mandal, Corneel Casert, Peter Sollich, 2022. Robust prediction of force chains in jammed solids using graph neural networks. Nature Communications 13. URL: https://doi.org/10.1038/s41467-022-31732-3.

Zhuang Cheng, Jianfeng Wang, 2022. Estimation of contact forces of granular materials under uniaxial compression based on a machine learning model. URL: https://www.researchgate.net/publication/356441184_Estimation_of_contact_forces_of_granular_materials_under_uniaxial_compression_based_on_a_machine_learning_model. also available at: https://link.springer.com/journal/10035/volumes-and-issues/24-1.