Graph Neural Network and Fully Connected Feed Forward Neural Network based Two Phase Composition Prediction

Huseyn Yusifov

A project presented for the Thermodynamics of Subsurface Reservoirs Course
Final Project

ErSE-309
King Abdullah University of Science and Technology
Country
07.05.2024

Abstract

This study delves into the realm of predictive modeling for composition prediction, specifically comparing the effectiveness of Graph Neural Networks (GNNs) and Fully Connected Feed Forward Neural Networks (FCNNs). The research focuses on their application in a two-phase system to accurately predict composition. Leveraging the unique ability of GNNs to process graph-structured data and the robustness of FCNNs in capturing complex relationships, this study aims to uncover insights into their respective strengths and weaknesses.

In the initial investigation, it was found that both GNNs and FCNNs achieved similar accuracy in composition prediction. However, the number of training epochs required differed significantly, with GNNs converging in approximately 2000 epochs compared to 15000 epochs for FCNNs. This suggests that GNNs may be better suited for capturing the complexity of the problem.

Further exploration involved comparing two different graph structures for GNNs: one utilizing Binary Interaction Coefficients and the other without. Results indicated that the graph structure incorporating Binary Interaction Coefficients yielded higher accuracy and faster convergence during training.

Overall, this study provides valuable insights into the comparative performance of GNNs and FCNNs in composition prediction tasks within a two-phase system, highlighting the potential advantages of GNNs in capturing intricate relationships within the data.

1 Introduction

In the realm of predictive modeling, the accurate estimation of composition within complex systems is paramount for numerous applications across various domains, ranging from materials science to environmental engineering. As the demand for precise predictive tools continues to grow, researchers explore advanced techniques to enhance the accuracy and efficiency of composition prediction models.

This study embarks on a comparative investigation into two prominent neural network architectures: Graph Neural Networks (GNNs) and Fully Connected Feed Forward Neural Networks (FCNNs). Focusing on their efficacy in predicting composition within a two-phase system, this research endeavors to shed light on their respective strengths and weaknesses.

To commence the project, data generation was initiated using a gamma distribution for composition (z), with intervals utilized for generating data pertaining to pressure (P) and temperature (T). Subsequently, to establish ground truth data, stability analysis was conducted to isolate two-phase compositions. Successive Submission Method (SSM) was then employed to conduct phase splitting, refining the dataset further. This meticulously curated dataset was then utilized to train neural network models.

GNNs offer a unique capability to process graph-structured data, making them particularly well-suited for tasks involving interconnected components, such as molecular structures or social networks. Conversely, FCNNs excel in capturing complex relationships within data, albeit in a more traditional feed-forward manner.

By dissecting the performance of these architectures, this study aims to provide valuable insights into their applicability in composition prediction tasks. Furthermore, the investigation seeks to unravel the underlying factors influencing their performance, including the nature of the data and the structural characteristics of the predictive models

2 Graph Neural Networks GNNs

In this project, Graph Neural Networks (GNNs) were employed as a key tool for predictive modeling in the realm of composition prediction within a two-phase system. Specifically, a particular subclass of GNNs known as Graph Convolutional Networks (GCNs) was utilized due to its aptitude for processing graph-structured data.

2.1 Graph Convolutional Networks (GCNs)

GCNs are a specialized form of GNNs designed to operate on graph-structured data. Unlike traditional neural networks that operate on grid-like structures such as images or sequences, GCNs are adept at capturing relational information inherent in graph data. The architecture of GCNs enables them to leverage the connectivity patterns between nodes in a graph to perform effective feature extraction and prediction tasks. In Eq. 1, formula of a convolutional layer is shown.

$$\vec{\psi}_i^{n+1} = \sigma(W^n \sum_{j \in N(i)} \frac{\vec{\psi}_j^n}{N(i)}) \tag{1}$$

2.2 Main Components of Graph Data Structures

Graph data structures consist of nodes, which represent entities of interest, and edges, which represent relationships between entities. In the context of composition prediction, nodes may represent individual components or elements, while edges denote interactions or connections between these components. Understanding the underlying graph structure is crucial for effectively leveraging GNNs in predictive modeling tasks.

2.3 Message Passing and Aggregation in GCNs

The fundamental operation in GCNs revolves around the concept of message passing and aggregation (shown in equation 2 and 3 respectively). At each layer of the GCN, information is exchanged between neighboring nodes in the graph through a process known as message passing. This allows nodes to communicate with their immediate neighbors and aggregate information from their local neighborhood. Subsequently, the aggregated information is combined to update each node's feature representation through an aggregation step. This iterative process of message passing and aggregation enables GCNs to capture the relational dependencies and structural characteristics of graph-structured data, thereby facilitating accurate prediction tasks.

$$m_j^{n+1} = W^n \frac{\vec{\psi}_j^n}{N(i)} \tag{2}$$

$$AGG_i^n = \sigma(\sum_{j \in N(i)} m_j^{n+1}) \tag{3}$$

Through the utilization of GCNs, this project aims to leverage the inherent structure of graph data to enhance the accuracy and efficiency of composition prediction within a two-phase system. By elucidating the main components and mechanisms of GCNs, this section provides insights into the underlying principles driving the performance of GNNs in predictive modeling tasks.

3 Successive Submission Method (SSM)

In this section algorithms of Successive Submission Method (SSM) is explained.

1. To calculate vapor phase mole fraction θ_V, the Equilibrium Coefficient K is initialized by using Wilson's Correlation.

$$K_i^{(0)} = \frac{P_{c,i}}{P} e^{5.373(1+\omega_i)(1-\frac{T_{c,i}}{T})} \tag{4}$$

2. Calculate θ_V by using Rachford-Rice Equation by employing Newton-Raphson Method.

$$f_{RR}(\theta_V, \mathbf{K}) = \sum_{i=1}^{N_c} \frac{(K_i^{(k)} - 1)z_i}{1 + (K_i^{(k)} - 1)\theta_V} = 0 \tag{5}$$

3. By using the mass balance equations calculate liquid and vapor phase composition $x_i^{(k)}$ and $y_i^{(k)}$ respectively.

$$x_i^{(k)} = \frac{z_i}{1 + (K_i^{(k)} - 1)\theta_V} \tag{6}$$

$$y_i^{(k)} = \frac{K_i^{(k)} z_i}{1 + (K_i^{(k)} - 1)\theta_V} \tag{7}$$

4. liquid and vapor phase compressibility factors $Z_i^{L,(k)}$ and $Z_i^{V,(k)}$ respectively.

$${Z_i^{\alpha,(k)}}^3 - (1-B){Z_i^{\alpha,(k)}}^2 - (A - 2B - 3B^2)Z_i^{\alpha,(k)} - (AB - B^2 - B^3) = 0 \tag{8}$$

$$A = \frac{a\alpha_m P}{RT^2} \tag{9}$$

$$B = \frac{b_m P}{RT} \tag{10}$$

5. Calculate fugacity for liquid and vapor phases $f_i^{L,(k)}$ and $f_i^{V,(k)}$ respectively.

$$\begin{aligned} ln(\frac{f_i^{\alpha,(k)}}{x_i^{(k)}P}) = \frac{b_i}{b_m}(Z_i^{\alpha,(k)} - 1) - ln(Z_i^{\alpha,(k)} - B) - \\ (\frac{A}{2\sqrt{2}B})(\frac{2\Psi_i^{\alpha,(k)}}{a\alpha_m} - \frac{b_i}{b_m})ln(\frac{Z_i^{\alpha,(k)} + (1+\sqrt{2}B)}{Z_i^{\alpha,(k)} + (1-\sqrt{2}B)}) \end{aligned} \tag{11}$$

6. Check convergency.

$$\sum_{i=1}^{N_c}(\frac{f_i^{L,(k)}}{f_i^{V,(k)}}-1)^2 < 10^{-12} \tag{12}$$

7. If True print out results, If False update K and continue to iteration.

$$K_i^{(k+1)} = K_i^{(k)}(\frac{f_i^{L,(k)}}{f_i^{V,(k)}})^k \tag{13}$$

Figure 1 shows complete Successive Submission Method Algorithm.

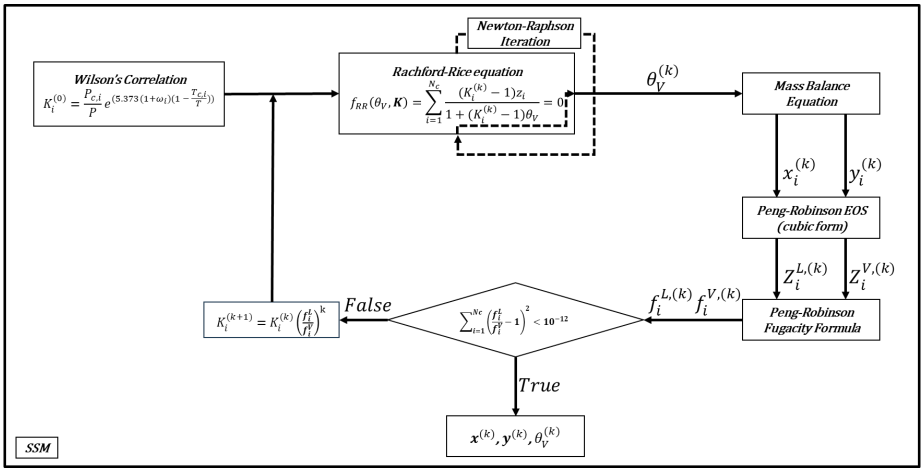
Newton-Raphson Iteration
Wilson's Correlation
$K_i^{(0)} = \frac{P_{c,i}}{P} e^{(5.373(1+\omega_i)(1-\frac{T_{c,i}}{T}))}$
Rachford-Rice equation
$f_{RR}(\theta_V, \boldsymbol{K}) = \sum_{i=1}^{N_c} \frac{(K_i^{(k)}-1)z_i}{1+(K_i^{(k)}-1)\theta_V} = 0$
$\theta_V^{(k)}$
Mass Balance Equation
$x_i^{(k)}$
$y_i^{(k)}$
Peng-Robinson EOS (cubic form)
$Z_i^{L,(k)}$
$Z_i^{V,(k)}$
$f_i^{L,(k)}$ $f_i^{V,(k)}$
Peng-Robinson Fugacity Formula
$\sum_{i=1}^{Nc}\left(\frac{f_i^L}{f_i^V}-1\right)^2 < 10^{-12}$
False
$K_i^{(k+1)} = K_i^{(k)}\left(\frac{f_i^L}{f_i^V}\right)^k$
True
$\boldsymbol{x}^{(k)}, \boldsymbol{y}^{(k)}, \theta_V^{(k)}$
SSM

Figure 1: Successive Submission Method (SSM

4 Single Phase Stability Analysis

Figure 2 shows complete algorithm for Single Phase Stability Analysis

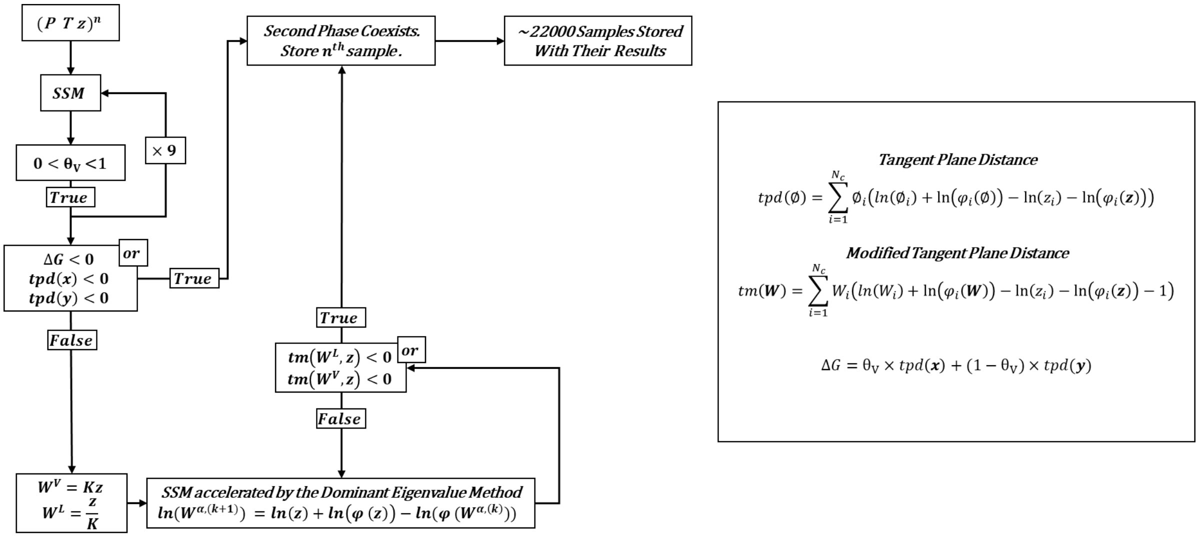
$(P\ T\ z)^n$
SSM
$0 < \theta_V < 1$
$\times 9$
True
$\Delta G < 0$
$tpd(x) < 0$
$tpd(y) < 0$
or
True
False
Second Phase Coexists. Store n^{th} sample.
~22000 Samples Stored With Their Results
True
$tm(W^L, z) < 0$
$tm(W^V, z) < 0$
or
False
$W^V = Kz$
$W^L = \frac{z}{K}$
SSM accelerated by the Dominant Eigenvalue Method
$ln(W^{\alpha,(k+1)}) = ln(z) + ln(\varphi(z)) - ln(\varphi(W^{\alpha,(k)}))$
Tangent Plane Distance
$tpd(\emptyset) = \sum_{i=1}^{N_c} \emptyset_i \left(ln(\emptyset_i) + \ln(\varphi_i(\emptyset)) - \ln(z_i) - \ln(\varphi_i(\mathbf{z})) \right)$
Modified Tangent Plane Distance
$tm(\mathbf{W}) = \sum_{i=1}^{N_c} W_i \left(ln(W_i) + \ln(\varphi_i(\mathbf{W})) - \ln(z_i) - \ln(\varphi_i(\mathbf{z})) - 1 \right)$
$\Delta G = \theta_V \times tpd(\mathbf{x}) + (1 - \theta_V) \times tpd(\mathbf{y})$

Figure 2: Single Phase Stability Analysis

5 Methodology

5.1 GCN Model Architecture

The Graph Convolutional Network (GCN) model employed in this project comprises three layers, each designed to capture and process the relational information embedded within the graph-structured data.

In the first layer of the GCN model, there are three GCN layers. Each GCN layer has an input channel of 6 and an output channel of 24. This configuration allows for the extraction of higher-level features from the input data, effectively transforming the initial representation into a more expressive feature space. Within each GCN layer, the hyperbolic tangent function serves as the activation function, facilitating non-linear transformations of the input data.

Following the first layer, the output of the three GCN layers is multiplied together and passed through the second layer of the model. The second layer consists of two GCN layers, with each GCN layer having an input channel of 24 and an output channel of 12. Similar to the first layer, the activation function employed within each GCN layer is the hyperbolic tangent function. This layer further refines the feature representation obtained from the first layer, extracting increasingly abstract and discriminative features from the input data.

Finally, the output of the second layer is multiplied together and passed through the last layer of the GCN model. The last layer comprises a single GCN layer with an input channel of 12 and an output channel of 2. At this stage, the model synthesizes the extracted features to produce the final prediction output. Once again, the hyperbolic tangent function is utilized as the activation function within the GCN layer, facilitating the transformation of the feature representation into the desired output format.

By leveraging this multi-layered architecture, the GCN model effectively captures the complex relational dependencies present within the graph-structured data, enabling accurate prediction of composition within the two-phase system under investigation.

Figure 3 shows GCN Model Architecture

5.2 FCNN Model Architecture

The Fully Connected Feed Forward Neural Network (FCNN) model employed in this project consists of a total of 7 layers, including an input layer, five hidden layers, and an output layer. This architecture is designed to process the input data and extract relevant features to predict the composition within the two-phase system accurately.

The first layer of the FCNN model accepts an input size of 36, corresponding to the features or attributes of the input data. These features are then propagated through the subsequent hidden layers to extract relevant patterns and relationships. The output layer consists of 18 neurons, representing the predicted composition output of the model.

The FCNN model incorporates five hidden layers, each comprising a different number of neurons. These hidden layers play a crucial role in extracting meaningful representations from the input data, gradually transforming the initial input into a format conducive to accurate prediction. The number of neurons in the hidden layers are as follows: 25, 20, 15, 20, and 25, respectively.

Throughout the FCNN model, the hyperbolic tangent function is utilized as the activation function. This activation function introduces non-linearity into the model, enabling it to learn complex patterns and relationships present within the data. By applying the hyperbolic tangent function at each layer, the FCNN model can effectively capture and represent the non-linear mappings between the input and output variables.

By leveraging this multi-layered architecture and the hyperbolic tangent activation function, the FCNN model endeavors to extract informative features from the input data and generate accurate predictions of composition within the two-phase system under investigation.

Figure 4 shows FCNN Model Architecture

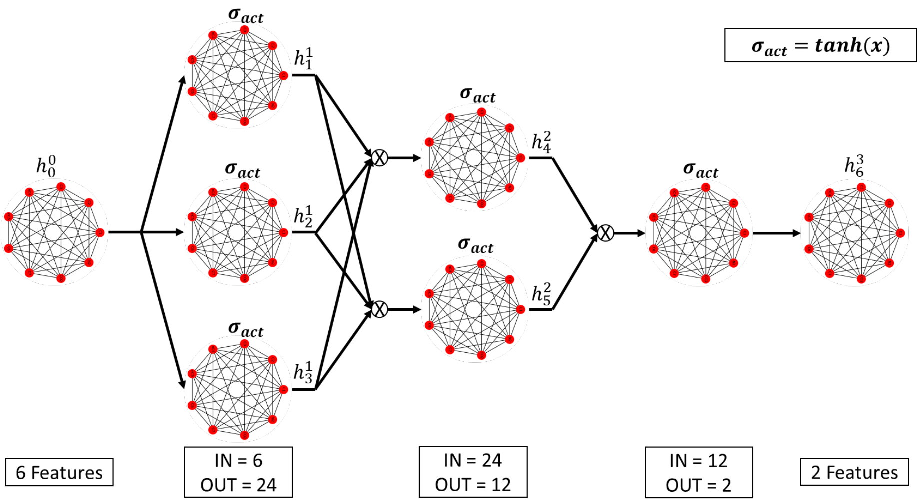

Figure 3: GCN Model Architecture

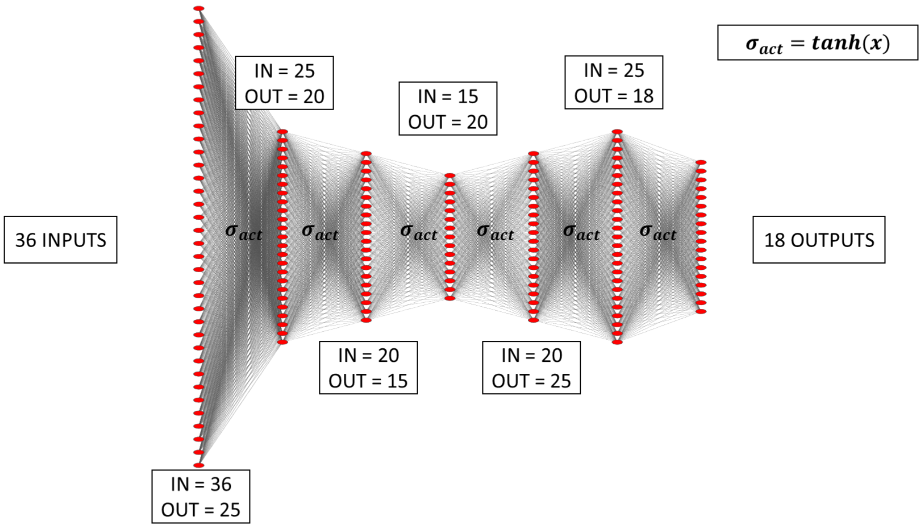

Figure 4: FCNN Model Architecture

5.3 Complete Workflow

1. **Data Generation:** Initially, a dataset comprising 125,000 samples with varying combinations of pressure (P), temperature (T), and composition (z) is generated. This dataset represents a diverse range of conditions within the two-phase system under investigation.

2. **Stability Analysis:** The generated dataset undergoes stability analysis to identify samples where the second phase coexists. This analysis ensures that only relevant samples, indicative of the desired two-phase system, are retained for further processing. As a result, a refined dataset consisting of 23,000 samples is obtained.

3. **Device Configuration and Phase Splitting:** The refined dataset is then transferred to a CUDA-enabled device for accelerated processing. Subsequently, the Successive Submission Method (SSM) is employed to conduct phase splitting, refining the dataset further and generating ground truth data for model training.

4. **Model Training:** The collected data, augmented with ground truth labels obtained from the SSM, is utilized to train deep learning models, including Graph Neural Networks (GNNs) and Fully Connected Feed Forward Neural Networks (FCNNs). These models are trained using the refined dataset to learn the underlying patterns and relationships between the input features and the target composition predictions.

5. **Prediction and Comparison:** Once the models are trained, predictions are made for unseen data samples using each model. The predicted compositions are then compared against the ground truth labels to evaluate the performance of the models. This comparison provides valuable insights into the efficacy of the different modeling approaches and their suitability for accurately predicting composition within the two-phase system.

By following this comprehensive workflow, this project aims to leverage advanced deep learning techniques to accurately predict composition within a two-phase system, ultimately contributing to the advancement of predictive modeling in relevant domains. **Figure 5 shows Complete Workflow Scheme**

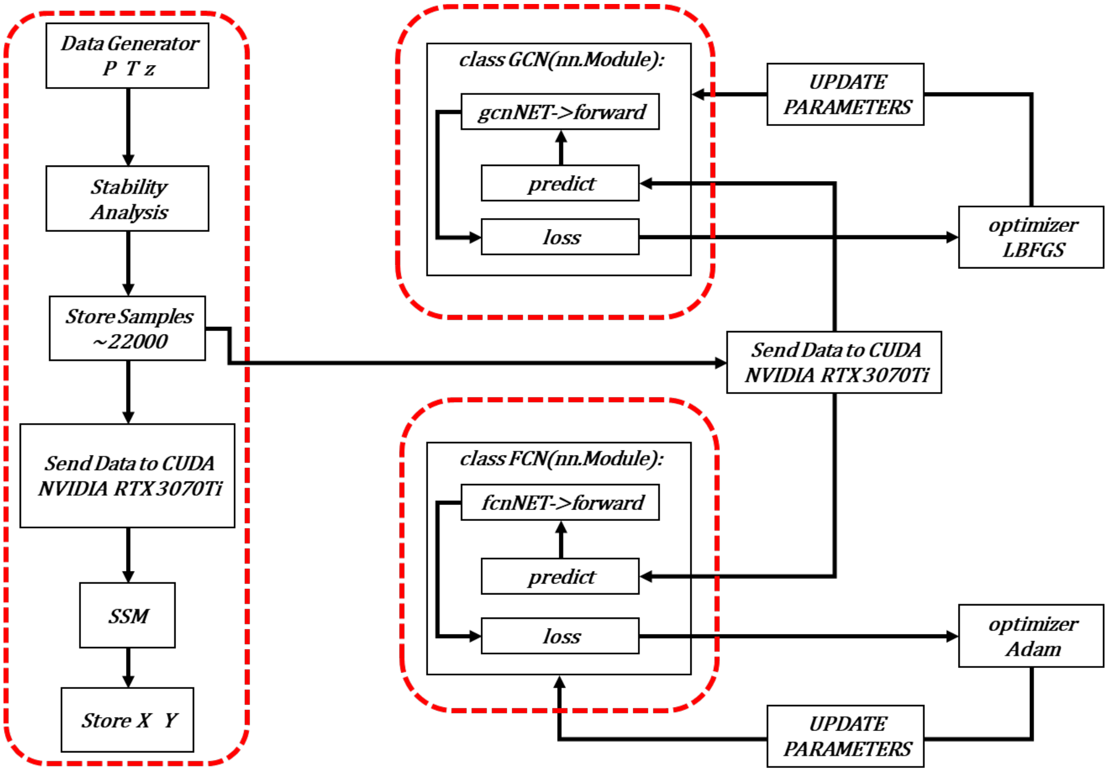

Figure 5: Complete Workflow Scheme

6 Results

Total parity plots for both the Graph Convolutional Network (GCN) and the Fully Connected Feed Forward Neural Network (FCNN) models have been illustrated in Figure 6 and Figure 7, respectively.

GCN Model Performance: Liquid Phase: The GCN model exhibits predictions with an R2 score of 0.983 for the liquid phase. Vapor Phase: For the vapor phase, the GCN model demonstrates predictions with an R2 score of 0.999.

FCNN Model Performance: Liquid Phase: The FCNN model showcases predictions with an R2 score of 0.993 for the liquid phase. Vapor Phase: Similarly, the FCNN model demonstrates predictions with an R2 score of 0.998 for the vapor phase.

These R2 scores serve as indicators of the models' performance in accurately predicting composition within both the liquid and vapor phases of the two-phase system. The high R2 scores obtained for both models underscore their effectiveness in capturing the underlying patterns and relationships within the data, further validating their utility in predictive modeling tasks.

In Figure 8 parity plots of different compositions z in different pressure P and Temperature T is illustrated

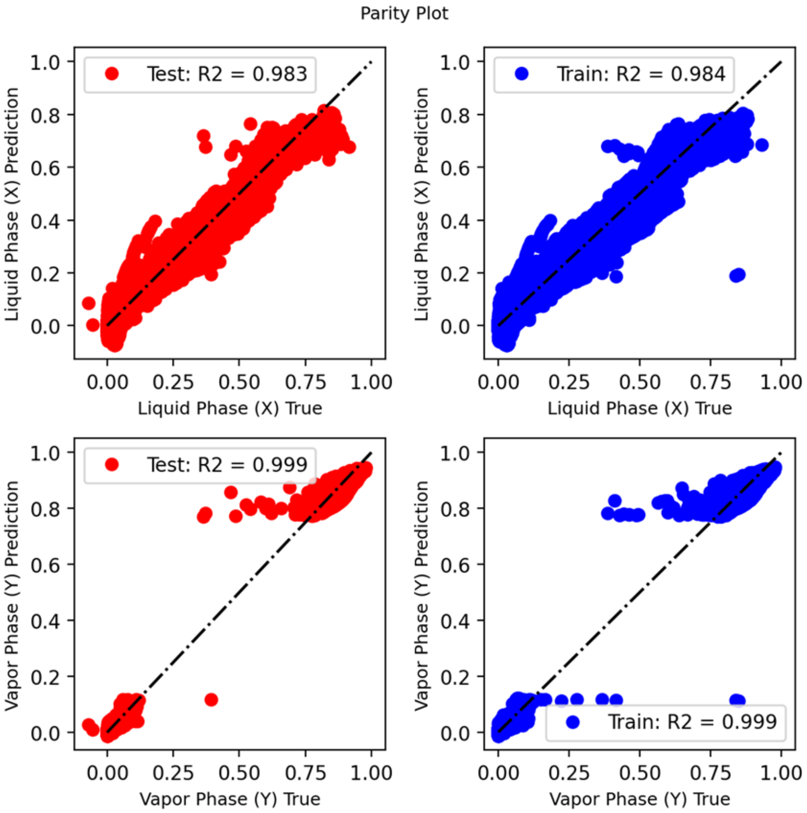

Figure 6: GCN Parity Plot

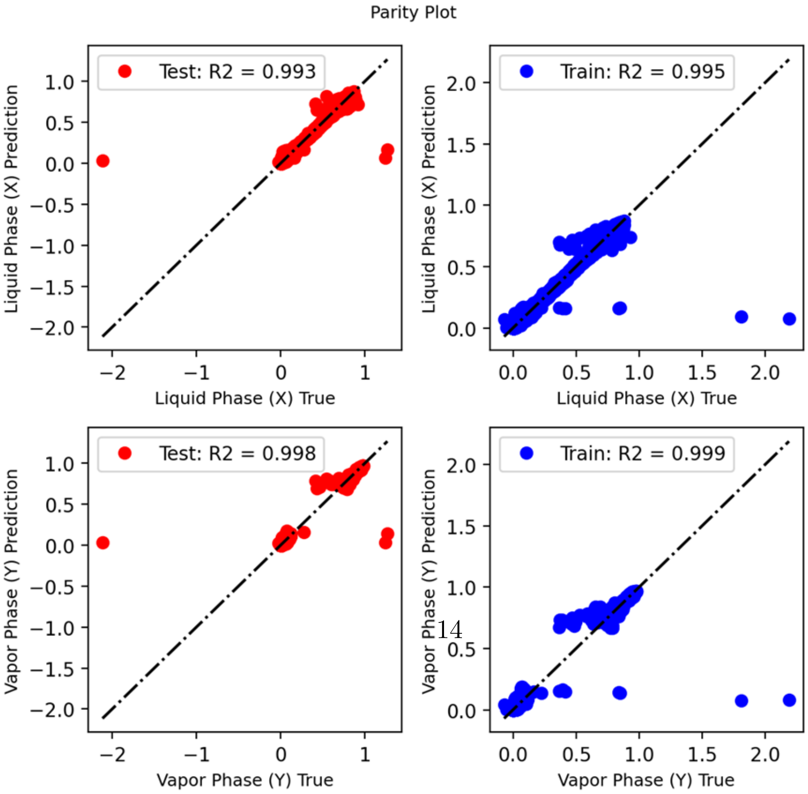

Figure 7: FCNN Parity Plot

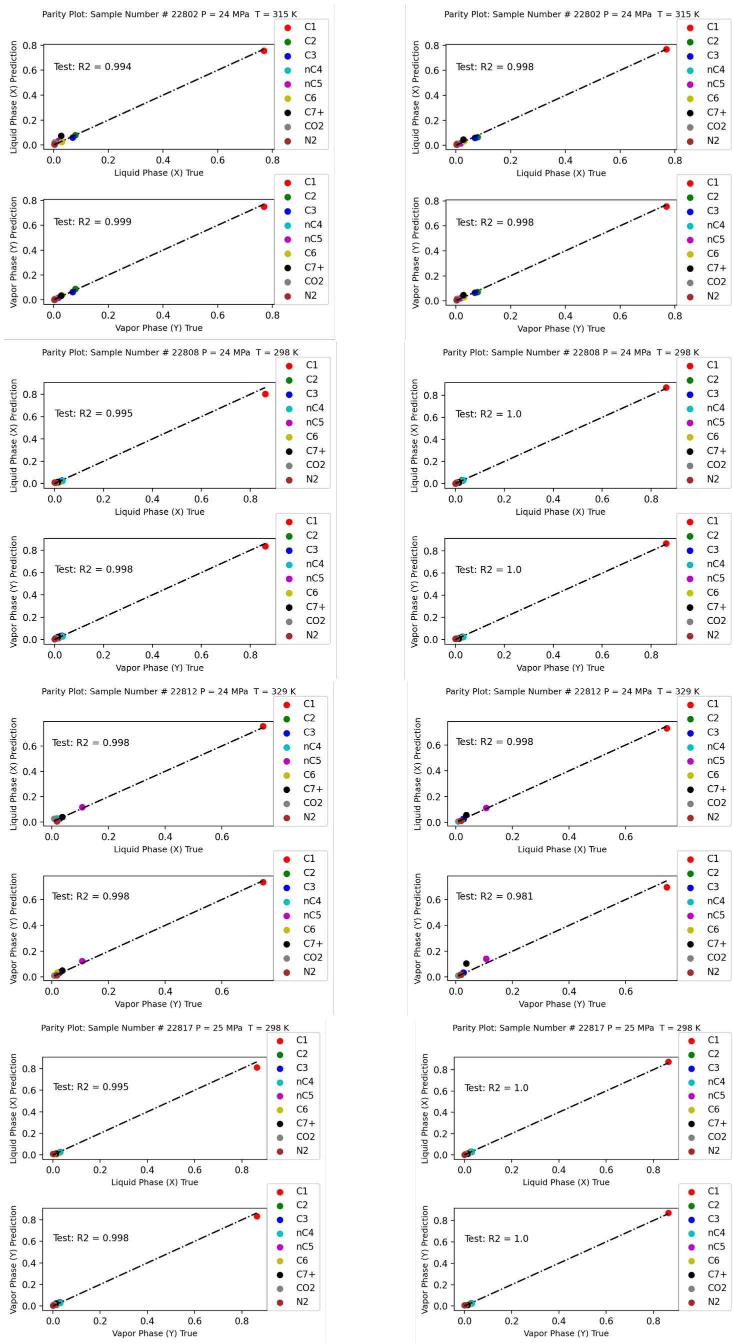

Figure 8: Parity Plots in Different Conditions (Left : GCN, Right: FCNN)

7 Discussion

7.1 Part I

Upon evaluating the efficiency and performance of both the Graph Convolutional Network (GCN) and the Fully Connected Feed Forward Neural Network (FCNN) models, it is evident that both models have achieved satisfactory results in predicting composition within the two-phase system. However, a deeper analysis of their efficiency and ability to capture complexity reveals interesting insights.

Loss Function Sensitivity Analysis: Figure 9 illustrates the sensitivity of the loss function with respect to the number of training epochs for both the GCN and FCNN models. This analysis provides valuable insights into the models' convergence behavior and their respective abilities to capture the complexity of the underlying problem.

GCN Model: The loss function sensitivity graph for the GCN model indicates that the model exhibits sensitivity to changes in the loss function and converges in approximately 2000 epochs. This rapid convergence suggests that the GCN model possesses a high efficiency and a remarkable ability to capture the complexity of the problem within a relatively short training period.

FCNN Model: In contrast, the loss function sensitivity analysis for the FCNN model reveals a different convergence pattern. The FCNN model exhibits sensitivity to changes in the loss function and requires approximately 15000 epochs to converge. This prolonged training period implies that the FCNN model may require more iterations to capture the intricacies and nuances of the underlying data, indicating potentially lower efficiency compared to the GCN model.

7.2 Part II

The utilization of Binary Interaction Coefficients (BICs) plays a crucial role in enhancing the efficacy of graph-based models, particularly in capturing interactions between nodes representing components of the composition within the two-phase system. By evaluating the impact of incorporating BICs into the model architecture, valuable insights can be gleaned regarding the significance of leveraging edge weights to facilitate meaningful connections between components.

Graph-Based Model Capabilities

Graph-based models inherently possess the capability to capture intricate relationships and interactions between nodes within a graph structure. In the context of composition prediction within a two-phase system, each node represents a component, and the edges between nodes signify the interactions or dependencies between these components. By leveraging this inherent structure, graph-based models can effectively capture the complex interplay between different components, thereby facilitating accurate prediction of composition.

Evaluation Using Loss Function Sensitivity

Figure 10 depicts the loss function sensitivity and its convergence for the graph-based model incorporating Binary Interaction Coefficients. The analysis reveals the model's sensitivity to changes in the loss function and its convergence behavior over the training epochs. By comparing the convergence patterns with and without the utilization of BICs, insights can be gained into the importance of incorporating edge weights to capture interactions between components.

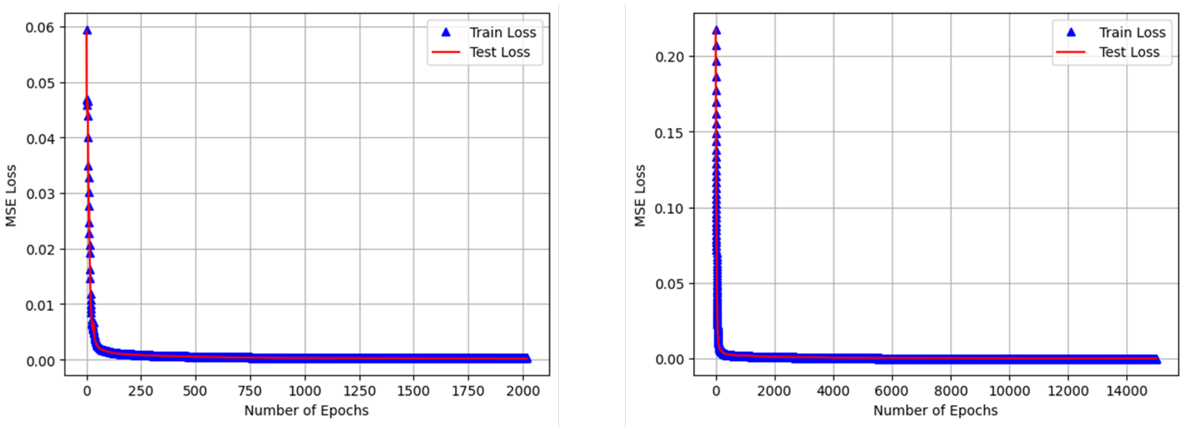

Figure 9: Loss Function Sensitivity Plot (Left: GCN, Right: FCNN)

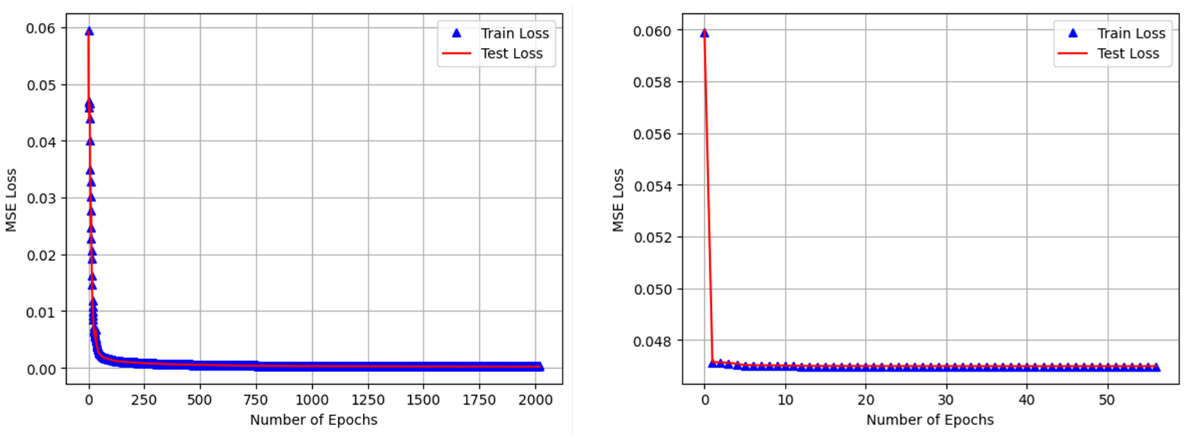

Figure 10: Loss Function Sensitivity. (Left: GCN with BICs, Right: GCN without BICs)

8 Conclusion

While both the GCN and FCNN models demonstrate satisfactory performance in predicting composition within the two-phase system, the efficiency and ability to capture complexity vary between the two models. The GCN model exhibits rapid convergence and efficient capture of complex relationships within the data, as evidenced by its shorter training period compared to the FCNN model. This suggests that the GCN model may offer advantages in terms of efficacy in handling complex predictive modeling tasks.