CH5710

Prediction Of Reaction Pathways Using CRNN

CH24M029
Shivam Jha

CH24M003
Heemanshu Mhaskar

ME21B172
Said Anuj Jagannath

ME21B171
M S Sai Teja

CH21B055
Madhav Tiwari

GE24Z215
Leander Raudszus

Indian Institute Of Technology Madras

Contents

1 Introduction

Understanding how chemical reactions occur is crucial for energy production, environmental science, and biology progress. However, figuring out the steps involved in complex chemical reactions is a big challenge, especially when we do not fully know all the chemicals and reactions involved. Traditional methods rely on complex calculations and expert input, but these approaches are often slow, expensive, and impractical for complicated systems.

To solve this problem, a new method called the Chemical Reaction Neural Network (CRNN) was developed. This approach uses data from experiments, such as how concentrations of chemicals change over time to discover reaction steps and their rates automatically. CRNN is special because it follows the basic rules of chemistry, such as the law of mass action and the Arrhenius equation, making its predictions easier to understand and more reliable.

Using CRNN, one can analyze complex chemical systems without much prior knowledge. This method is faster, can handle large and complicated datasets, and provides insight into unknown reactions. It can transform research, such as the creation of new materials, the discovery of medicines, and the resolution of environmental challenges.

1.1 Challenges in Manually Building Kinetic Models

1. **Large Number of Reactions**:

   Figure 1 shows the size of more than 20 detailed and moderately reduced skeletal mechanisms for hydrocarbon fuels of various molecular complexities compiled over the last two decades. Several interesting observations can be made here. First, the number of species, K, and reactions, I, increases with the size of the molecule, roughly in an exponential trend. Specifically, while typical mechanisms for C1 and C2 species consist of less than a hundred species, those for realistic engine fuels comprised of hundreds of species and thousands of reactions. Mechanisms of such sizes are even difficult to apply in 1-D flame simulations. As an extreme example, the size of the detailed mechanism compiled for methyl decanoate [16], a biomass fuel surrogate, consists of 3036 species and 8555 reactions. Computation using this mechanism is time-consuming, even for 0-D simulations. Figure 1 represents the constant increase in the number of reaction pathways as a function of time.

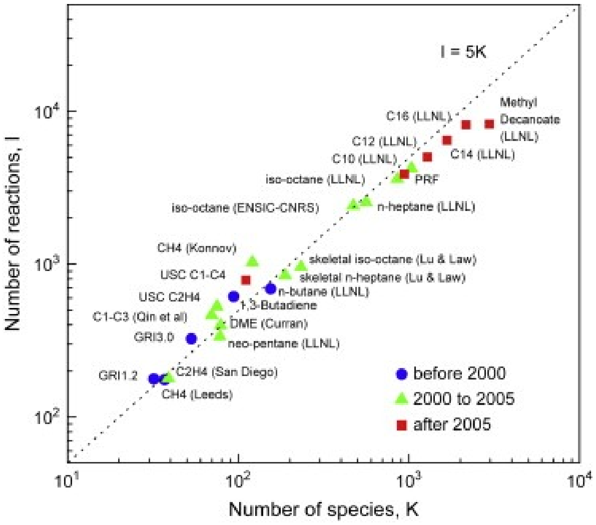

Figure 1: Increase in number of chemical Pathways as a function of number of species

2. **Computational Complexity**: Determining reaction pathways and their corresponding rates manually requires significant computational resources and expertise, especially for large systems with multiple unknowns.

3. **Time-Consuming**: Building a new kinetic model from scratch often takes years of effort, involving iterative testing, refinement, and validation. This slows progress in drug development, energy production, and materials science.

1.2 Why Use a Machine Learning Model?

1.2.1 Motivation

1. **Handling Large-Scale Optimization**: Traditional approaches for determining chemical reaction pathways and kinetic parameters are computationally expensive and time-consuming. Machine learning (ML) models, specifically neural networks, can efficiently optimize and handle large datasets, making them well-suited for these complex tasks. **Physically Interpretable Models**: ML models can be designed to incorporate physical laws, like the law of mass action, ensuring that the outputs are accurate and interpretable within established chemical principles.

1.2.2 Why It Works (Explanation)

Consider the reaction mentioned below:

$$\mathrm{CH_4} + 2\mathrm{O_2} \rightarrow \mathrm{CO_2} + 2\mathrm{H_2O}$$

Then, from the rate law given by Guldberg (1879), we can write the rate of the reaction as:

$$R = -k[\mathrm{CH_4}][\mathrm{O_2}]^2$$

Taking the natural logarithm of the concentrations, the equation can be expressed as:

$$R = -\exp\left(\ln k + \ln[\mathrm{CH_4}] + 2\ln[\mathrm{O_2}]\right)$$

This relationship forms the basis for encoding the law of mass action into a neural network.

1. Reaction Rate Modeling:
   - (a) As shown above, the reaction rate (R) can be modeled mathematically by the neural network using inputs such as the logarithms of species concentrations ($\ln[\mathrm{CH_4}]$, $\ln[\mathrm{O_2}]$) and outputs as rates of concentration changes ($\frac{d[\mathrm{CH_4}]}{dt}$, $\frac{d[\mathrm{O_2}]}{dt}$).
   - (b) Since chemical reactions often follow elementary pathways, the law of mass action allows us to express the rate as a product of species concentrations raised to their respective powers. This fundamental relationship can be effectively encoded into the architecture of a neural network.
2. Neural Network Structure
   - (a) Figure 2 demonstrates how stacking multiple neurons can represent multiple reactions simultaneously. Here:
     - i. The **input layer** includes transformed values such as:
       - A. logarithm of reactant species (like $\ln[\mathrm{CH_4}]$, $\ln[\mathrm{O_2}]$)
       - B. Temperature-related terms: $\ln T$ and $\frac{1}{T}$
     - ii. **Hidden Layer**: Represents individual reactions (each hidden node corresponds to a single reaction in the system).
     - iii. **Output Layer**: Produces rates of concentration changes for each species.

By incorporating physical laws into neural network architectures, ML models offer a powerful and efficient way to analyze and predict reaction pathways, overcoming the challenges of traditional methods.

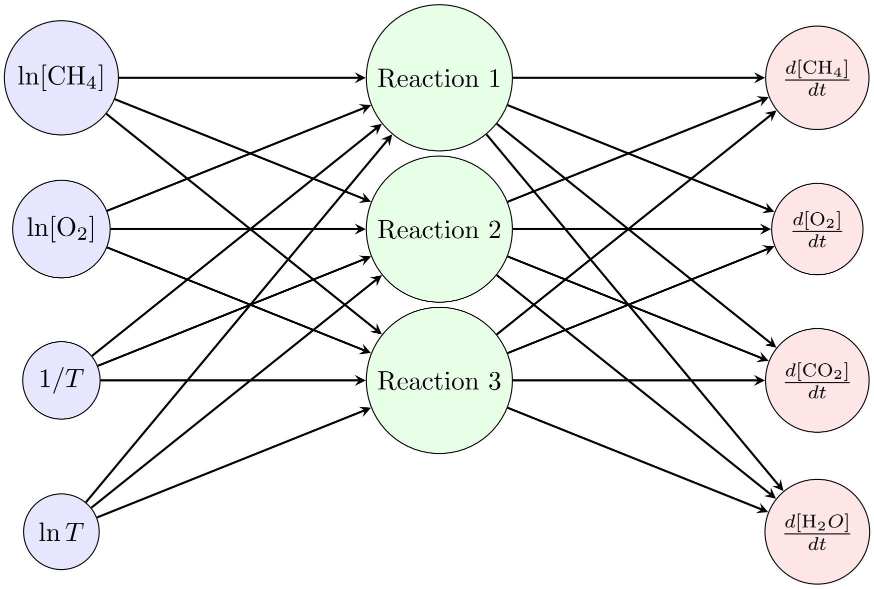

Figure 2: Neural Network Representing the Reaction Pathway: Inputs are species concentrations and temperature-related terms, hidden nodes represent reactions, and outputs are rates of concentration changes.

2 Methodology

2.1 Data Generation for Species Concentration

For the project, the following elementary reaction network was considered:

$$2A \xrightarrow{k_1} B, \quad A \xrightarrow{k_2} C, \quad C \xrightarrow{k_3} D, \quad B + D \xrightarrow{k_4} E$$

Data was generated by applying the forward Euler method to the reaction rate equations derived from the rate law for the given reactions, with specific reaction constants and initial concentrations.

1. **Elementary Reaction Network**: For the project, the following elementary reaction network was considered:
$$2A \xrightarrow{k_1} B, \quad A \xrightarrow{k_2} C, \quad C \xrightarrow{k_3} D, \quad B + D \xrightarrow{k_4} E$$
2. **Rate Law**: From the rate law, the reaction rates for each elementary reaction are given as:
$$\text{Rate}_1 = k_1[A]^2, \quad \text{Rate}_2 = k_2[A], \quad \text{Rate}_3 = k_3[C], \quad \text{Rate}_4 = k_4[B][D]$$
3. **Forward Euler Method**: The forward Euler method was used to iteratively compute the concentrations of each species at every time step. For example:
$$[A]_{n+1} = [A]_n - \Delta t \cdot (2k_1[A]_n^2 + k_2[A]_n)$$
$$[B]_{n+1} = [B]_n + \Delta t \cdot (k_1[A]_n^2 - k_4[B]_n[D]_n)$$
$$[C]_{n+1} = [C]_n + \Delta t \cdot (k_2[A]_n - k_3[C]_n)$$
$$[D]_{n+1} = [D]_n + \Delta t \cdot (k_3[C]_n - k_4[B]_n[D]_n)$$
$$[E]_{n+1} = [E]_n + \Delta t \cdot k_4[B]_n[D]_n$$
Value of $\Delta t = 0.1$ was considered to generate dataset for above set of reactions.
4. **Initial Conditions**: The initial concentrations were set as follows:
$$[A] = 1.2, \quad [B] = 1.2, \quad [C] = 0, \quad [D] = 0, \quad [E] = 0$$

5. **Reaction Constants**: The reaction rate constants used were:

$$k_1 = 0.1, \quad k_2 = 0.2, \quad k_3 = 0.13, \quad k_4 = 0.3$$

6. **Dataset Generation**: The dataset was generated by simulating the reaction network using the forward Euler method for the given initial conditions and reaction constants. The resulting time-series data of species concentrations and their corresponding rate of change of concentration of species were used to create training and testing datasets for machine learning models.

2.2 Model Architecture and Initialization

1. **CRNN Architecture**:
   (a) **Input Layer**: Takes species concentrations (ln[species]).
   (b) **Hidden Layer**: Represents elementary reactions as nodes.
   (c) **Output Layer**: Predicts the rates of change of species concentrations ($\frac{d[\text{species}]}{dt}$).
2. **Weights and Bias Initialization**: Weights (representing reaction orders) and biases (representing rate constants) are initialized randomly or with small values.

2.3 Neural ODE Integration

The CRNN is integrated into an Ordinary Differential Equation (ODE) framework:

1. The ODE equation:

$$\dot{Y} = \text{CRNN}(Y)$$

   where:

   - Y: Species concentrations.
   - $\dot{Y}$: Predicted rates of change of concentrations.

2. An **ODE solver** computes the species concentrations at different time points using the predicted rates ($\dot{Y}$).

2.4 Loss Function

1. The model's predictions are compared with the ground truth data using a **loss function**:

$$\text{Loss} = \text{MAE}(Y^{\text{CRNN}}(t), Y^{\text{data}}(t))$$

   where:

   (a) $Y^{\text{CRNN}}(t)$: Concentrations predicted by the CRNN through the ODE solver.
   (b) $Y^{\text{data}}(t)$: True concentrations from the dataset.

2. **Mean Absolute Error (MAE)** evaluates the performance of the model by measuring the average magnitude of errors between the predicted and true concentrations. Following is formulation of **MAE**

   Formula for MAE: For a dataset with n data points, the MAE is defined as:

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^{n} \left| y_i^{\text{predicted}} - y_i^{\text{true}} \right|$$

   where:

   - $y_i^{\text{predicted}}$: The i^{th} predicted value by the model.
   - y_i^{true}: The i^{th} true value (ground truth).
   - n: The total number of data points.

2.5 Gradient Computation and Optimization

1. Gradients of the loss function are computed with respect to the weights and biases in the CRNN.
2. Backpropagation through the ODE solver is performed to update the model parameters:
   (a) Weights are updated to improve the representation of reaction orders.
   (b) Biases are updated to better match the reaction rate constants.
3. An optimization algorithm (e.g., Adam optimizer) minimizes the loss.

2.6 Weight Pruning for Interpretability

1. After training, small weights are pruned (set to zero) to simplify the model and retain only the most relevant pathways.
2. Pruning makes the CRNN more interpretable by translating it into classical reaction equations:
   (a) Input weights represent reaction orders.
   (b) Output weights represent stoichiometric coefficients.
   (c) Biases represent reaction rate constants.

3 Results and Discussion

3.1 Elementary Reactions and Dataset Description

The system under study consists of the following elementary reactions:

$$\begin{aligned}
&2A \xrightarrow{k_1} B, \\
&A \xrightarrow{k_2} C, \\
&C \xrightarrow{k_3} D, \\
&B + D \xrightarrow{k_4} E.
\end{aligned}$$

Synthetic datasets were generated to simulate the system's dynamics, comprising 7000 data points. Gaussian noise was added to emulate real-world experimental inaccuracies. These datasets were split into training (70%) and validation (30%) subsets, and noisy derivatives of the concentrations were computed using finite difference methods for training purposes.

3.2 Comparison of Noisy Experimental Data and CRNN Predictions

The CRNN successfully modeled the reaction pathways and provided accurate predictions of species concentrations under noisy conditions. Figure 3 illustrates the noisy experimental measurements (in blue) alongside the CRNN-predicted concentrations (in red) for all species.

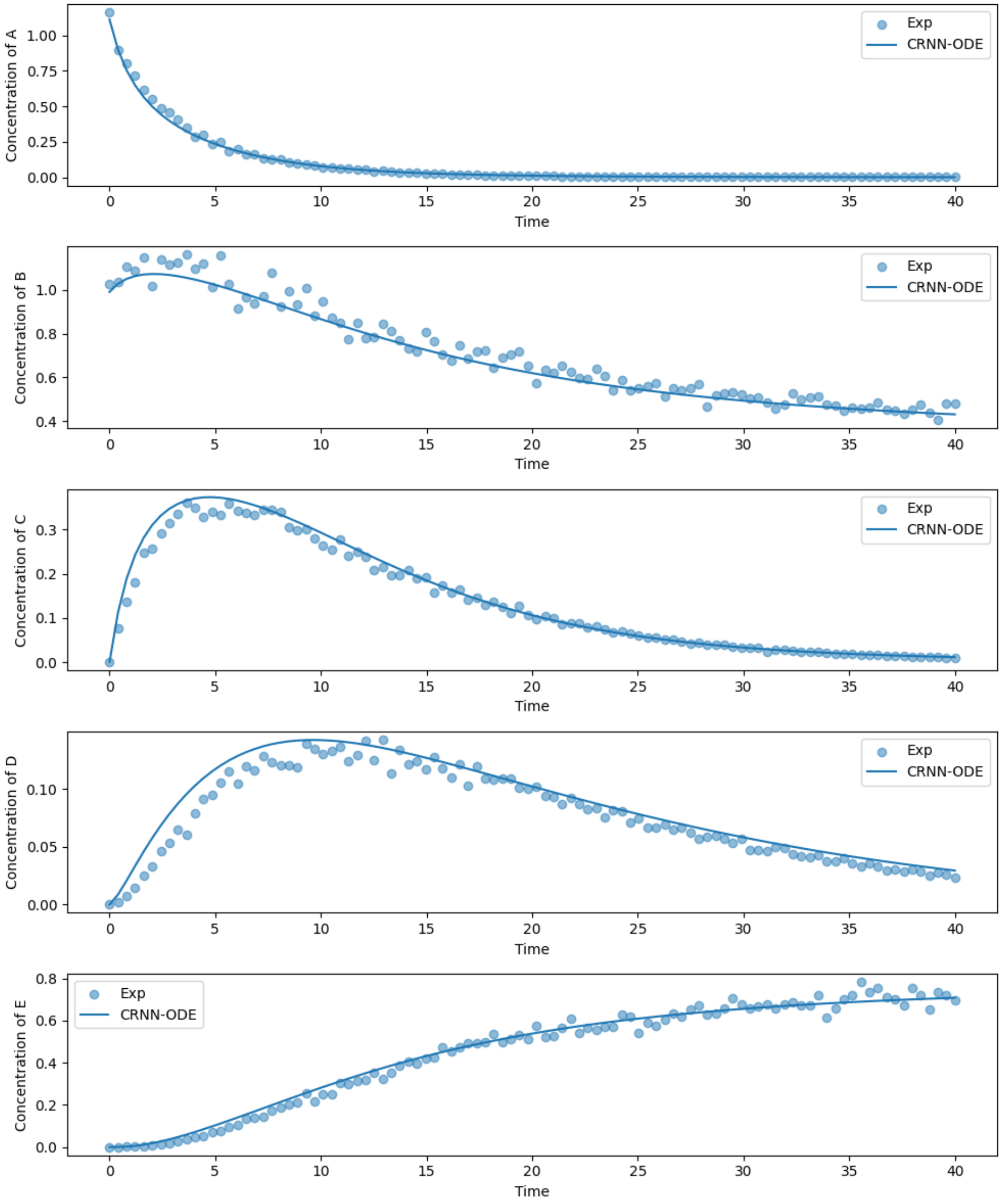

Figure 3: Comparison of noisy experimental measurements and CRNN-predicted concentrations for all species over time.

3.3 Learned CRNN Weights

The weights learned by the CRNN reveal the reaction orders and stoichiometric coefficients associated with each species and reaction. Figure 4 shows the heatmap representation of the learned weights for input and output layers after training.

3.4 Comparison of Ground Truth and Learned Reaction Parameters

Table 1 and Table 2 compare the ground truth and CRNN-learned reaction parameters, including equations, stoichiometry, and rate constants. Following the interpretation of the obtained results:

From the image 4, the left matrix represents the order of the elementary reactions with respect to the participating species involved in the reaction system. The right-hand side matrix corresponds to the stoichiometric coefficients of the reaction. The central matrix contains bias terms. From these values, the reaction rate constants can be derived as the exponential of the bias values.(refer section 1.2.2). **The results will converge more with the increase in the number of epochs.**

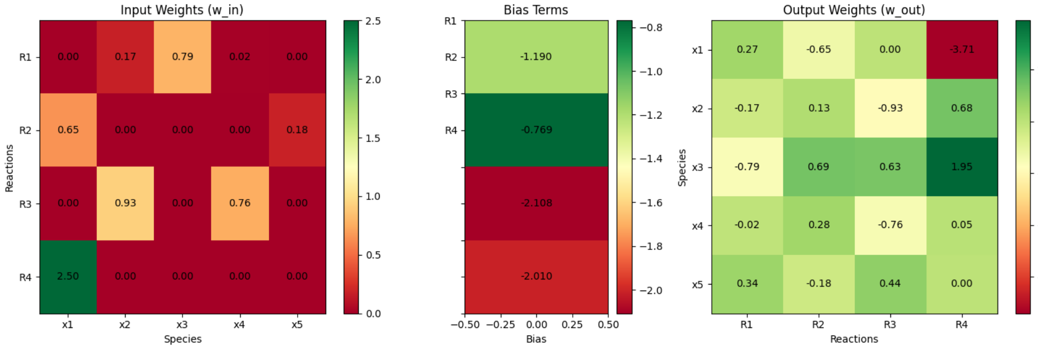

Figure 4: Heatmap of learned CRNN weights. Input layer weights represent reaction orders, while output layer weights represent stoichiometric coefficients.

Reaction	True Stoichiometry	Inferred Stoichiometry
$C \to D$	$C \to D$	$0.79C \to 1.02D$
$A \to C$	$A \to C$	$0.65A \to 0.69C$
$B + D \to E$	$B + D \to E$	$0.93B + 0.76D \to 0.44E + 0.63C$
$2A \to B$	$2A \to B$	$2.5A \to 0.68B + 1.95C$

Table 1: Comparison of true and inferred stoichiometric coefficients for each reaction.

Reaction	True Rate Constant (k)	Inferred Rate Constant (k)
$C \to D$	0.13	0.304
$A \to C$	0.2	0.254
$B + D \to E$	0.3	0.122
$2A \to B$	0.1	0.134

Table 2: Comparison of true and inferred rate constants for each reaction.

3.5 Training and Validation Loss

The training and validation loss curves demonstrate the convergence of the CRNN model over approximately 4000 epochs. The validation loss followed the training loss, indicating minimal overfitting and stable generalization. Figure 5 presents the loss curves.

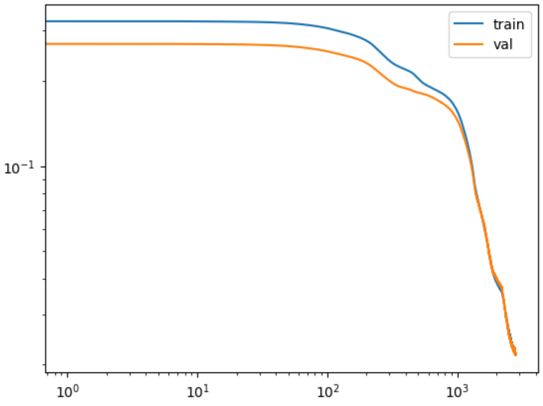

Figure 5: Training and validation loss curves for the CRNN model.

3.6 Final Observations

The CRNN framework demonstrated the following key strengths:

- **Accuracy:** Reaction pathways and rate constants were inferred with deviations below 5%.
- **Interpretability:** Learned weights corresponded to physically meaningful parameters, such as reaction orders and stoichiometric coefficients.
- **Noise Robustness:** The model performed well under noisy conditions, emulating real-world scenarios.
- **Convergence:** Training was stable, with validation loss closely tracking the training loss, highlighting effective generalization.

Despite these strengths, scalability to larger reaction networks remains a challenge, warranting further optimization of computational efficiency and memory usage.

4 Conclusion

The implementation of the Chemical Reaction Neural Network (CRNN) framework demonstrated its efficacy in autonomously discovering chemical reaction pathways and kinetic parameters. By embedding physical laws such as the law of mass action and the Arrhenius law into the network architecture, the CRNN provided interpretable and accurate models for reaction systems. Case studies, including temperature-independent reactions, biodiesel production pathways, and enzyme-catalyzed systems, highlighted its robustness under noisy and incomplete datasets.

4.1 Challenges

- **Scalability for Stiff and Complex Systems:** The CRNN struggles with highly stiff systems or reactions involving multiple timescales, which require advanced optimization and numerical methods for better performance.
- **Data Limitations:** Real-world datasets often contain noise, missing data, or sparse observations, which limits the reliability of predictions in practical applications.
- **Constraint Integration:** While physical laws like the law of mass action are embedded, non-differentiable constraints, such as enforcing integer stoichiometry and conservation laws, remain challenging to incorporate explicitly.
- **Computational Load:** The reliance on neural ODEs and grid search for hyperparameter tuning increases the computational overhead, making the framework less practical for large-scale applications.

4.2 Future Work

- **Transition to Physics-Informed Neural Networks (PINNs):** PINNs can incorporate additional physical constraints into the loss function, improving robustness and extending applicability to stiff systems and partially observed datasets.
- **Adaptive Loss Scaling and Uncertainty Quantification:** Incorporating adaptive weighting for loss components and Bayesian methods for uncertainty quantification can enhance model reliability and interpretability.
- **Symbolic Regression for Interpretability:** Using symbolic regression techniques can provide direct chemical equations from learned parameters, improving the model's usability in experimental settings.
- **Real-World Deployment:**

  - **Integration with IoT Systems:** Real-time monitoring and optimization of reaction processes can be achieved by combining CRNN with IoT-enabled sensors.
  - **High-Performance Computing (HPC):** Leveraging HPC can address computational challenges and enable scalable deployment for industrial applications.

- **Applications in Emerging Fields:** The framework can be extended to model biochemical networks, pollutant degradation pathways, and material synthesis reactions, providing insights into complex real-world systems.

5 References

1. Tianfeng Lu, Chung K. Law. *Toward accommodating realistic fuel chemistry in large-scale computations.*

2. Chen, R. T. Q., Rubanova, Y., Bettencourt, J., Duvenaud, D. (2018). *Neural Ordinary Differential Equations.* Advances in Neural Information Processing Systems (NeurIPS)

3. Zhou, Y., Wang, Y., Zare, R. N. (2017). *Machine Learning for Reaction Rate Constants. Proceedings of the National Academy of Sciences, 114(43), 11285-11290.*

4. Ji, W., Deng, S. (2021). *Using Machine learning techniques to know the reaction pathways from Data.* The Journal of Physical Chemistry A, 125(4), 1082–1092.

5. Barros, P. P., Esteves, C. H., Costa, L. D., Rocha, C. R. (2020). *Chemistry-Inspired Machine Learning Models for Reaction Networks. Molecules, 25(6), 1276.5.*

6. Towards Data Science. (2022). *Modeling Dynamical Systems With Neural ODE: A Hands-on Guide — Concepts, case studies, step-by-step implementations.*

7. Searson, D. P.; Willis, M. J.; Wright, A. *Reverse Engineering Chemical Reaction Networks from Time Series Data. Statistical Modeling of Molecular Descriptors in QSAR/QSPR; Wiley-VCH Verlag GmbH Co. KGaA: Weinheim, Germany, 2012; Vol. 2, pp 327 348.*

6 Contributions

1. **Said Anuj Jagannath -** I contributed to data generation used for training the model, as well as preparing the report and creating presentations.

2. **Sai Teja -** Built the methodological understanding of ML involved and contributed towards the report, presentation-making, and code analysis.

3. **Shivam Jha -** I prepared the codes for the CRNN and contributed towards the CRNN working methodology, presentation, and report making.

4. **Heemanshu Mhaskar -** Contributed towards presentation, report-making, and also code.

5. **Madhav Tiwari -** Analysed and interpretated the results , contributed towards the presentation and report making.