CS401 Design Report

Equation of State Coefficient Predictor
Course Name: COSC/ECE 401/402
Submitted to: Mentor/Sponsor Names
Date Submitted: November 27, 2024

Executive Summary

This design report outlines the development of a molecular data preprocessing and predictive modeling pipeline, leveraging advanced computational tools and machine learning to analyze molecular structures and properties. The project integrates molecular embeddings derived from SMILES (Simplified Molecular Input Line Entry System) representations with a custom-built deep learning framework to predict molecular properties efficiently and accurately.

The preprocessing pipeline, implemented using Python libraries such as RDKit and PubChemPy, generates molecular descriptors, cleanses the data, and standardizes features to ensure high-quality inputs for the modeling stage. Molecular embeddings are then processed into a regression task to predict target properties, which represent critical metrics for chemical analysis.

The predictive modeling pipeline employs a deep learning approach, designed to process high-dimensional molecular embeddings. A residual neural network architecture, incorporating batch normalization, dropout regularization, and residual connections, was developed to enhance model accuracy and stability. The pipeline uses robust training methodologies, including K-Fold Cross-Validation and early stopping, to minimize overfitting and maximize generalization.

Key features of the pipeline include automated hyperparameter optimization via Optuna, efficient learning rate scheduling, and dynamic validation metrics such as mean squared error (MSE) and R^2 score. Results are visualized through intuitive scatter plots comparing true and predicted values, providing insights into the model's performance.

The pipeline remains exploratory, with ongoing optimization and evaluation aimed at refining its predictive capabilities. This system represents a foundational framework for scalable and efficient molecular property prediction, with applications in cheminformatics, pharmaceutical research, and materials science.

Contents

List of Figures

List of Tables

Chapter 1

Problem Definition & Background

1.1 Problem Statement

Accurate and efficient prediction of molecular properties is critical in applications such as drug discovery, toxicity assessment, and materials engineering. Traditional experimental methods are expensive and time-consuming, limiting their scalability when screening large chemical libraries. Computational approaches offer faster alternatives, but existing methods often struggle with complex molecular structures, lack scalability, or fail to deliver actionable predictions for practical applications.

1.2 Background and Context

Pharmaceutical and materials science industries rely heavily on molecular property predictions to accelerate drug development, optimize material designs, and evaluate compound safety. For example, predicting solubility, toxicity, or chemical stability enables researchers to identify promising candidates before committing to costly laboratory experiments. Current tools often demand extensive preprocessing or lack integration with predictive models capable of handling high-dimensional molecular data, creating bottlenecks in workflows.

By combining cheminformatics and deep learning, this project aims to create an efficient, scalable pipeline for molecular property prediction. The pipeline focuses on generating meaningful molecular embeddings from SMILES strings and using these embeddings to train machine learning models that can predict properties such as molecular weight, logP, or aromaticity, addressing practical needs across various scientific domains.

1.3 Application Areas

- **Drug Discovery:** Rapid screening of chemical libraries to identify drug-like candidates with desired properties such as bioavailability and toxicity profiles.
- **Materials Science:** Predicting properties of new materials, such as thermal stability or conductivity, to optimize performance in applications like electronics or energy storage.

- **Environmental Safety:** Assessing molecular toxicity to minimize environmental impact and comply with regulatory standards.

1.4 Underlying Theory

The project leverages RDKit to compute molecular descriptors and generate SMILES-based embeddings that encode structural and chemical features. These embeddings are processed using a deep learning model with a residual neural network architecture, chosen for its ability to preserve critical information and handle high-dimensional input data effectively. The use of batch normalization, dropout regularization, and residual connections ensures the model remains stable, accurate, and generalizable.

1.5 Prior Work

Machine learning models have shown promise in cheminformatics, particularly for predicting molecular properties. However, most solutions either rely on shallow regression models with limited scalability or complex neural networks that are difficult to optimize. This project integrates preprocessing, embedding generation, and model training into a cohesive, application-focused pipeline designed to address these limitations.

1.6 Who Benefits

- **Pharmaceutical Researchers:** Accelerate drug discovery by identifying compounds with desired pharmacokinetic and toxicity profiles.
- **Materials Engineers:** Optimize material properties for use in cutting-edge technologies like semiconductors, batteries, and nanomaterials.
- **Regulatory Scientists:** Quickly assess chemical safety for environmental and human health compliance.

1.7 Unaddressed Needs

Current computational tools for molecular property prediction are often either too domain-specific or lack the flexibility to adapt to diverse datasets and target properties. This project bridges that gap by offering a scalable, modular pipeline that can be tailored to various application areas, enabling actionable predictions across multiple industries.

Chapter 2

Requirement Specification

2.1 Overview

The primary objective of this project is to develop a scalable and efficient molecular property prediction pipeline integrated with a user-friendly graphical user interface (GUI). The pipeline will preprocess molecular data, generate embeddings, and use a deep learning model to predict target properties. The GUI will enable users to input molecular structures, view predictions, and interpret results interactively.

2.2 Functional Requirements

- **Data Input and Processing:**
  - Users can upload molecular datasets (e.g., CSV files) containing molecule names and/or SMILES strings.
  - Support for manual molecule input via the GUI with real-time validation of SMILES strings.
  - Automatic computation of molecular descriptors and embeddings using RDKit and PubChemPy.
- **Prediction Model:**
  - The system will predict key molecular properties such as molecular weight, logP, and other user-specified metrics.
  - Predictions will be computed using a pre-trained deep learning model (residual neural network).
  - Provide real-time predictions for single-molecule inputs via the GUI and batch predictions for uploaded datasets.
- **Graphical User Interface (GUI):**
  - A user-friendly interface for inputting molecules, visualizing predictions, and exploring molecular properties.

  - Interactive visualization of input molecules and predicted properties (e.g., tables, plots, or 3D molecule views).
  - Intuitive workflows for both novice and expert users, with minimal technical barriers.

- **Visualization and Reporting:**
  - Scatterplots comparing true vs. predicted values for batch datasets.
  - Exportable results in standard formats (e.g., CSV, JSON).
  - Provide confidence intervals or error metrics for each prediction to aid in decision-making.

2.3 Non-Functional Requirements

- **Performance:**
  - Ensure prediction latency of under 2 seconds for single molecule inputs and reasonable processing times for batch predictions.
- **Scalability:**
  - Support datasets with up to 10,000 molecules for batch predictions.
- **Reliability:**
  - Ensure the system gracefully handles invalid inputs and provides meaningful error messages.
- **Usability:**
  - Design the GUI to be platform-independent, accessible via major operating systems (Windows, macOS, Linux).

2.4 Metrics and Targets

- **Prediction Accuracy:** Achieve a mean squared error (MSE) of less than 0.1 and an R^2 score greater than 0.9 on validation datasets.
- **Processing Speed:** Complete predictions for 10,000 molecules in under 10 minutes.
- **GUI Usability:** Receive positive usability feedback (above 90% satisfaction) in user testing with at least 10 participants.

2.5 Standards and Constraints

- Compliance with cheminformatics data standards for SMILES strings and molecular descriptors.
- Use of industry-standard libraries (e.g., RDKit, PyTorch) for implementation.
- Implementation of machine learning models that are compatible with GPU acceleration.

Chapter 3

Technical Approach

3.1 Overview

Our design plan aims to develop an advanced molecular property prediction system integrating multiple machine learning approaches with domain-specific molecular representations and encodings. The pipeline will include molecular data preprocessing, various encoding methods, and multiple predictive models. These models include Convolutional Neural Networks (CNNs) for molecular graph analysis, Physics-Informed Neural Networks (PINNs) incorporating domain-specific equations, and Kernel Attention Networks (KANs) with learnable activation functions to aggregate predictions. The system will also incorporate a graphical user interface (GUI) to facilitate user interaction and accessibility.

3.2 System Architecture

The pipeline consists of the following main components:

- **Data Preprocessing:** Generate molecular encodings and representations such as Coulomb matrices and other molecular embeddings.
- **Predictive Models:**
  - CNN for learning from molecular graph images.
  - PINN utilizing physical equations for properties a and b with respect to P, V, n, R, and T.
  - KAN with learnable activation functions for aggregating predictions.
- **Prediction Aggregation:** Combine predictions from individual models to improve overall accuracy.
- **Graphical User Interface:** Enable molecule input, result visualization, and exploration of predictions.

3.3 Functional Decomposition

- **Data Preprocessing:**
  - Parse input molecules from datasets or user input via the GUI.
  - Generate molecular graph images for CNN input.
  - Compute Coulomb matrix representations and other molecular encodings for model training.
- **Predictive Models:**
  - **CNN:** Analyze molecular graph images to extract spatial features and predict molecular properties.
  - **PINN:** Incorporate the differential equations $a, b = f(P, V, n, R, T)$ to constrain predictions based on physical principles.
  - **KAN:** Use learnable activation functions to adaptively combine predictions from different encodings and models.
- **Prediction Aggregation:**
  - Use KAN to aggregate predictions from CNN, PINN, and other encoding-based models.
  - Provide final property predictions with uncertainty estimates.
- **Graphical User Interface:**
  - Allow users to input molecules as SMILES strings, names, or files.
  - Visualize molecular graphs, Coulomb matrices, and predicted properties.
  - Enable export of predictions in standard formats.

3.4 Major Design Decisions

- **Molecular Representations:**
  - Use molecular graph images for CNNs.
  - Generate Coulomb matrices and other domain-specific encodings for PINNs and KANs.
- **Model Selection:**
  - Employ CNNs for spatial feature extraction from molecular graph images.
  - Incorporate physical constraints via PINNs to improve generalization.
  - Use KANs for adaptive aggregation of predictions from different models and representations.
- **Integration:** Combine all model predictions in a modular framework to ensure extensibility for future improvements or additional encodings.

3.5 Implementation Details

- **Data Preprocessing:**
  - Generate molecular graph images from SMILES strings using RDKit.
  - Compute Coulomb matrices and other relevant molecular encodings.
  - Normalize input data for compatibility with deep learning models.
- **Predictive Models:**
  - **CNN:** Train a convolutional neural network on molecular graph images to predict properties.
  - **PINN:** Train a physics-informed neural network using differential equations for a and b with respect to P, V, n, R, T.
  - **KAN:** Train a kernel attention network to aggregate predictions using learnable activation functions for flexibility.
- **GUI Development:**
  - Build a cross-platform GUI using PyQt or Tkinter.
  - Integrate visualization tools for graph images, Coulomb matrices, and predictions.

3.6 Prediction Aggregation and Evaluation

- Use KAN to aggregate predictions from CNNs, PINNs, and other models.
- Evaluate aggregated predictions using metrics such as mean squared error (MSE) and R^2 score.
- Compare model performance on benchmark datasets and validate with experimental data where possible.

Chapter 4

Design Concepts, Evaluation & Selection

4.1 User Interface Design

The graphical user interface (GUI) is designed to enable users to input molecular structures, initiate predictions, and visualize results interactively. The user flow is as follows:

1. **Input:** Users can draw molecules, upload files, or enter SMILES strings directly.
2. **Prediction:** The system processes molecular encodings and uses machine learning models to predict properties.
3. **Visualization:** Results are displayed as tables, scatterplots, and molecular representations.

To validate the GUI, preliminary usability testing was conducted with researchers and students in cheminformatics. Feedback indicated that the workflow meets the requirements for ease of use and accessibility.

4.2 Model Architecture Decisions

4.2.1 Molecular Representation

Alternative 1: Coulomb Matrix Encoding

- Represents molecules by encoding atomic interactions through electrostatic potential matrices.
- Pros: Captures detailed electrostatic features; suitable for quantum chemistry-based models [1].
- Cons: Computationally expensive for large molecules.

Alternative 2: Molecular Graph Representations

- Uses graph-based encodings where nodes represent atoms and edges represent bonds, enabling convolutional neural networks (CNNs) to extract structural features.
- Pros: Retains spatial and relational information critical for molecular properties [2].
- Cons: Requires additional preprocessing to construct graph representations.

Evaluation: Coulomb matrices are simpler to compute but lack the flexibility of graph-based encodings. Preliminary results suggest graph representations may provide better performance for tasks requiring spatial information.

4.2.2 Prediction Models

Alternative 1: Physics-Informed Neural Network (PINN)

- Incorporates the equation $a, b = f(P, V, n, R, T)$ into the loss function to constrain predictions with physical laws.
- Pros: Embeds domain knowledge, improving generalization for unseen data [3].
- Cons: Requires precise parameter tuning and domain-specific equations.

Alternative 2: Convolutional Neural Networks (CNNs) for Molecular Graphs

- Processes graph-based encodings to predict molecular properties.
- Pros: Captures structural relationships and local interactions efficiently.
- Cons: Computationally demanding for large datasets.

Evaluation: PINNs provide a physics-grounded approach, while CNNs excel in extracting relational features. We plan to use both approaches and evaluate their complementary strengths.

4.3 Prediction Aggregation

Alternative 1: Simple Averaging of Model Outputs

- Combines predictions by averaging outputs from individual models.
- Pros: Simple and computationally efficient.
- Cons: May overlook variations in model performance.

Alternative 2: Kernel Attention Networks (KANs)

- Dynamically weights predictions from each model using attention mechanisms with learnable activation functions.
- Pros: Adaptive to model strengths, improving overall accuracy [4].

- Cons: Computationally complex to implement and optimize.

Evaluation: KANs offer a more robust aggregation strategy by assigning dynamic weights based on model performance. We anticipate that this approach will lead to more accurate and reliable predictions.

4.4 Evaluation Plan

To evaluate the proposed designs, the following plan is established:

- **Test Cases:** Utilize benchmark molecular datasets to test different encodings, models, and aggregation strategies.
- **Metrics:**
  - Prediction accuracy: Mean squared error (MSE) and R^2 score.
  - Usability: User satisfaction surveys on the GUI.
  - Performance: Latency and throughput for large datasets.
- **Comparison:** Evaluate alternative encodings and model architectures for performance under varied datasets and target properties.

4.5 Selected Design Approach

Based on preliminary results and evaluation criteria, the following approach will be implemented:

- Coulomb matrix and graph-based encodings will be used in parallel.
- PINNs and CNNs will serve as core predictive models, complemented by KANs for prediction aggregation.
- The GUI will provide visualization tools for predictions and molecular representations, ensuring usability across diverse user groups.

Chapter 5

Deliverables

5.1 Overview

At the conclusion of this project, we will deliver a comprehensive molecular property prediction system, consisting of a user-friendly graphical interface, robust predictive models, and detailed documentation. This system is designed to enable researchers and industry professionals to efficiently analyze molecular properties and leverage machine learning in their workflows.

5.2 Final Deliverables

The project will produce the following key deliverables:

5.2.1 Software System

- **Molecular Property Prediction Pipeline:**
  - Preprocessing tools for generating molecular encodings (e.g., Coulomb matrices, SMILES-based embeddings).
  - Multiple predictive models, including:
    - Convolutional Neural Networks (CNNs) for molecular graph images.
    - Physics-Informed Neural Networks (PINNs) incorporating domain-specific equations.
    - Kernel Attention Networks (KANs) for aggregating predictions.
  - Integration of prediction models with output aggregation for reliable property estimation.
- **Graphical User Interface (GUI):**
  - Interactive molecule input via drawing, file uploads, or SMILES entry.
  - Visualization tools for molecular graphs, Coulomb matrices, and predicted properties.

- Export functionality for results in CSV or JSON formats.

5.2.2 Documentation

- **User Guide:**
  - Instructions on how to install, configure, and use the system.
  - Examples of workflows for different use cases (e.g., drug discovery, materials optimization).
- **Technical Documentation:**
  - Detailed explanation of the system architecture, model training process, and hyperparameter optimization.
  - Guidelines for extending the system (e.g., adding new encodings or models).

5.2.3 Evaluation Results

- **Performance Metrics:**
  - Validation results for prediction accuracy (R^2, mean squared error) across benchmark datasets.
  - Computational performance benchmarks, including runtime and scalability for large datasets.
- **User Feedback:**
  - Summarized findings from usability testing of the GUI.
  - Recommendations for future iterations based on user feedback.

5.2.4 Source Code and Deployment Tools

- Fully commented source code for the pipeline and GUI.
- Deployment scripts for running the system locally or on a cloud platform.
- Pre-trained models for immediate use without retraining.

5.3 Future Extensions

While not part of the primary deliverables, the system will be designed to facilitate future enhancements, such as:

- Integration with additional molecular encoding methods.
- Support for predicting more advanced molecular properties.
- Scalability to handle larger datasets and distributed computing environments.

Chapter 6

Project Management

6.1 Overview

Effective project management is critical to ensuring the successful and timely delivery of this molecular property prediction system. While the specifics of certain components may evolve, we have developed a preliminary timeline that outlines the major milestones and their corresponding deadlines. This schedule will be refined as the project progresses and new challenges or opportunities arise.

6.2 Project Timeline

The following table summarizes the key milestones and deliverables for the project:

6.3 Team Roles and Responsibilities

To ensure efficient progress, the following team roles have been defined:

- **Project Manager:** Oversees the project schedule, ensures deadlines are met, and coordinates between team members.
- **Data Scientist:** Develops preprocessing tools and ensures the accuracy of molecular encodings.
- **Model Developer:** Focuses on designing, training, and optimizing predictive models (CNNs, PINNs, KANs).
- **UI/UX Designer:** Leads the design and implementation of the graphical user interface.
- **Tester:** Conducts usability and performance testing, ensuring system reliability and user satisfaction.

Milestone	Deadline	Description
Initial Planning and Research	Week 2	Finalize project objectives, gather requirements, and review relevant literature.
Data Preprocessing Pipeline	Week 5	Implement and test tools for generating molecular encodings (Coulomb matrices, SMILES embeddings).
CNN Model Development	Week 9	Train and evaluate the CNN for molecular graph analysis.
PINN Model Development	Week 12	Incorporate physics-informed constraints into a predictive model.
KAN Aggregation Implementation	Week 15	Develop and integrate the kernel attention network for prediction aggregation.
GUI Prototype	Week 18	Build a basic GUI prototype to input molecular data and display results.
System Integration	Week 20	Combine all components (models, preprocessing, GUI) into a cohesive pipeline.
Performance Optimization	Week 22	Optimize models and preprocessing for speed and accuracy.
Usability Testing	Week 24	Conduct testing with end-users to gather feedback and refine the GUI.
Final Deliverables Submission	Week 26	Submit the completed system, documentation, and evaluation results.

Table 6.1: Preliminary Project Timeline

6.4 Contingency Plans

Recognizing potential risks, the following contingency plans are in place:

- **Delays in Data Collection:** Utilize publicly available molecular datasets as a fallback.
- **Model Performance Issues:** Allocate additional time for model optimization and explore alternative architectures if necessary.
- **GUI Development Challenges:** Prioritize core functionality and defer advanced features to future iterations.

6.5 Future Refinements

This timeline and role allocation will be revisited at regular intervals to ensure alignment with project goals and to address unforeseen challenges. Updates will be documented and incorporated into subsequent phases of the project.

Chapter 7

Budget

7.1 Overview

This section outlines the preliminary budget estimates for the molecular property prediction project. The budget considers software, hardware, and personnel costs, with engineering time being the most significant expenditure. The estimates will be refined as the project progresses and specific requirements are finalized.

7.2 Estimated Expenditures

The anticipated costs are categorized as follows:

7.3 Engineering Time Allocation

The project timeline is broken into milestones, with estimated human-hours allocated for each phase:

7.4 Future Adjustments

This budget represents an initial estimate and will be updated as the project evolves. Actual expenditures and time allocations will be recorded and compared to these estimates to ensure accountability and adaptability.

Category	**Estimated Cost (USD)**	**Description**
Software Licenses	500	Includes libraries and tools such as RDKit, PubChemPy (free but maintenance donations), PyTorch, and GUI development frameworks.
Hardware	$2,000$	GPU-enabled machine for training and testing predictive models.
Cloud Services	$1,000$	Computational resources for large-scale training and testing, if required.
Personnel	$15,000$	Estimated engineering time: 500 hours at $30/hour$. Includes data preprocessing, model development, GUI design, and testing.
Miscellaneous	500	Expenses for documentation, presentations, and potential third-party APIs for molecular data.
Total Estimated Cost	$19,000$	

Table 7.1: Preliminary Budget Estimates

Milestone	**Estimated Hours**	**Description**
Initial Planning and Research	50	Literature review, requirements gathering, and project planning.
Data Preprocessing Pipeline	80	Implementing tools for molecular encodings and data validation.
CNN Development	100	Designing, training, and testing convolutional neural networks.
PINN Development	120	Incorporating physics-based equations into a predictive model.
KAN Aggregation Implementation	80	Developing kernel attention networks for model output aggregation.
GUI Development	70	Creating a user-friendly interface for input, prediction, and visualization.
System Integration and Optimization	60	Combining all components and optimizing performance.
Testing and Documentation	40	Usability testing, system evaluation, and preparing project documentation.
Total Estimated Hours	**500**	

Table 7.2: Estimated Human-Hours Allocation

Bibliography

[1] M. Rupp, A. Tkatchenko, K.-R. Müller, and O. A. von Lilienfeld, "Fast and accurate modeling of molecular atomization energies with machine learning," *Physical Review Letters*, vol. 108, no. 5, p. 058301, 2012.

[2] D. Duvenaud, D. Maclaurin, J. Iparraguirre, R. Bombarell, T. Hirzel, A. Aspuru-Guzik, and R. P. Adams, "Convolutional networks on graphs for learning molecular fingerprints," in *Advances in Neural Information Processing Systems*, vol. 28, 2015.

[3] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving partial differential equations," *Journal of Computational Physics*, vol. 378, pp. 686–707, 2019.

[4] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, vol. 30, 2017.

Appendix A

Appendix A

Include supplementary materials such as raw data, detailed calculations, or additional diagrams.