

CHAPTER – 1

INTRODUCTION

CHAPTER 1

INTRODUCTION

1.1 Introduction

Personalized medicine is gaining prominence in cancer treatment due to the diverse genetic makeup of cancer cells, which leads to varied responses among patients undergoing similar therapies. This project, titled “Leveraging ANN for Targeted Drug Sensitivity Prediction on GDSC Data”, aims to create a model that leverages genomic data to predict how different cancer cell lines respond to specific drugs. The complexity of cancer biology and the uniqueness of each patient’s genetic profile necessitate computational tools that can support precision oncology by identifying the most effective drug treatments. This predictive model, based on the ANN architecture, seeks to aid oncologists and researchers by providing tailored predictions of drug efficacy based on specific cancer genomics, thus advancing the approach toward personalized therapies.

Our project uses the Genomics of Drug Sensitivity in Cancer (GDSC) dataset, which contains essential genomic features and drug response data across over 1,000 cancer cell lines. Focusing on IC50 prediction, a critical metric for assessing drug potency, the model leverages features such as gene mutations, expression profiles, and cancer-type descriptors to capture relevant patterns in drug sensitivity. A user-friendly web-based interface developed with Flask allows researchers to input data for IC50 predictions seamlessly. By simplifying the interaction between the model and users, this web application supports data input through dropdown menus and displays prediction outputs in an accessible, clinician-oriented manner, aiming to streamline and support decision-making in cancer research and clinical practice.

1.2 Problem Statement

Cancer treatment efficacy is significantly impacted by the genetic diversity of cancer cells, causing variable responses to the same drug among patients. Traditional treatment approaches often do not account for these individual differences, leading to inconsistent therapeutic outcomes. Furthermore, the lack of reliable predictive models for drug response limits the scope of precision medicine. This project addresses the following key issues.

- **Limited personalized treatment options:** Conventional treatments do not consider individual genetic profiles.
- **Challenges in predicting drug efficacy:** The relationship between genomic features and drug response is complex and not fully understood.
- **High failure rates in therapy selection:** The absence of predictive accuracy can lead to ineffective treatments, higher costs, and increased side effects.

To tackle these challenges, our project proposes an ANN-based model trained on genomic data to predict drug sensitivity, helping oncologists make better-informed treatment decisions. By analyzing genomic features relevant to drug response, the model aims to provide more accurate predictions that align with the unique profiles of cancer patients.

1.3 Objectives

- Merge data from the GDSC, gene expression, mutation profiles, and CNAs to enhance the accuracy and relevance of drug response predictions.
- Build an ANN model to predict IC50 values based on genomic data from multiple datasets.
- Develop a Flask-based interface to facilitate easy data input.
- Enable users to visualize IC50 predictions along with a comparative graph for five selected drugs.

1.4 Scope

- **Data Analysis:** Analyze genomic features from the GDSC dataset to identify correlations with drug response, such as gene expression and mutations.
- **Model Development:** Implement an ANN model to capture complex relationships between genomic features and drug efficacy.
- **Performance Evaluation:** Assess model performance using metrics such as Mean Squared Error (MSE), Mean Absolute Error (MAE) and R-squared (R^2).
- **Web Interface Development:** Design a Flask-based web interface to facilitate user interaction with the model and simplify data entry.
- **Real-World Application:** Develop the interface with real-time prediction capabilities, suitable for clinical and research use cases.

1.5 Organization of the Report

This report is organized as follows:

- **Chapter 1:** Introduction, which includes the background, problem statement, objectives, scope, and structure of the report.
- **Chapter 2:** Literature Survey, providing an overview of existing systems and approaches in drug response prediction and discussing their limitations.
- **Chapter 3:** System Requirements Specification, defining the functional and non-functional requirements of the project.
- **Chapter 4:** Gantt Chart, displaying the project timeline and key milestones.
- **Chapter 5:** System Design, describing the architecture, data flow diagrams, and use case diagrams of the proposed solution.
- **Chapter 6:** Implementation, covering data preprocessing, model architecture, and the Flask application development process.
- **Chapter 7:** System Testing, outlining the testing methodologies, test cases, and validation techniques used to evaluate the model.
- **Chapter 8:** Results and Snapshots, presenting model performance metrics and screenshots of the Flask interface.
- **Chapter 9:** Conclusion and Future Work, summarizing project outcomes and suggesting future directions for research.
- **References:** Listing the sources consulted throughout the development of the project.

CHAPTER – 2

LITERATURE SURVEY

CHAPTER 2

LITERATURE SURVEY

2.1 Existing System

Katyna Sada Del Real and Angel Rubio, "Discovering the Mechanism of Action of Drugs with a Sparse Explainable Network", 2023 [1].

This study presents SparseGO, an interpretable neural network designed for predicting drug responses in cancer cell lines, focusing on understanding mechanisms of action (MoA). SparseGO leverages gene expression data and applies DeepLIFT, an explainable AI method, to achieve interpretable predictions by linking specific input features to model outcomes. SparseGO improves memory efficiency and optimizes resource usage, reducing reliance on high-powered GPUs. However, SparseGO's dependence on gene expression data alone limits its adaptability, and its results require further clinical validation. Our project expands on these limitations by incorporating multiple genomic features, including gene mutations and CNAs, to improve predictive accuracy. Additionally, our model includes a Flask-based interface for practical accessibility in clinical and research environments, promoting broader applicability without heavy computational requirements.

Alexander Partin et al., "Deep Learning Methods for Drug Response Prediction in Cancer: Predominant and Emerging Trends," 2023 [2].

This review provides an extensive overview of 61 deep learning models applied to drug response prediction, with an emphasis on architectures such as convolutional and graph neural networks (GNNs). It highlights the importance of omics data integration and discusses the increasing trend toward personalized treatment and drug repurposing. While many reviewed models achieve promising results, they often struggle with generalization across new drugs and rely on large, annotated datasets for training, limiting real-world application. The lack of standardized evaluation frameworks also complicates performance comparisons. In our project, we address these limitations by focusing on an ANN model that integrates genomic features such as mutations and CNAs to enhance prediction accuracy, complemented by a simple, accessible Flask interface for use by researchers and clinicians.

Bara A. Badwan, Mohammad Qasem, Mohammad Anan, and Ali Hamed El-Moussawi, "Machine Learning Approaches to Predict Drug Efficacy and Toxicity in Oncology," 2023 [3].

This paper explores various machine learning algorithms in oncology, with a focus on predicting drug efficacy and toxicity. Techniques like PCA and t-SNE for dimensionality reduction are emphasized, particularly when handling high-dimensional genomic data. Despite its success, the study points out challenges such as reliance on extensive datasets and the gap between IC50 predictions and clinical efficacy. Current models also face limitations in generalizing predictions due to data variability. Our project addresses these limitations by optimizing feature selection and employing an ANN model trained on selected genomic data. The inclusion of a Flask-based interface allows for real-time predictions, which can assist researchers and clinicians in making informed decisions without requiring substantial computational resources.

Brent M. Kuenzi, Jisoo Park, Samson H. Fong, Kyle S. Sanchez, John Lee, Jason F. Kreisberg, Jianzhu Ma, and Trey Ideker, "Predicting Drug Response and Synergy Using a Deep Learning Model of Human Cancer Cells," 2020 [4].

DrugCell, introduced by Kuenzi et al., is a deep learning model designed to predict cancer cell responses to therapies by integrating genetic and chemical data. It employs a Visible Neural Network (VNN) architecture that hierarchically maps biological subsystems. DrugCell is effective in predicting single-drug and combination responses but is constrained by its reliance on predefined biological hierarchies, which may limit its adaptability to novel molecular interactions. Furthermore, DrugCell's computational requirements are high, which may limit its accessibility in some clinical settings. Our project aims to address these limitations by utilizing a streamlined ANN model that integrates selected genomic features without reliance on hierarchical dependencies, and by offering a Flask-based interface that makes our model accessible and practical for both research and clinical applications.

2.2 Limitations of Existing System

Katyna Sada Del Real and Angel Rubio, "Discovering the Mechanism of Action of Drugs with a Sparse Explainable Network", 2023 [1].

SparseGO improves computational efficiency but is restricted by its focus on gene expression data, limiting its capacity to incorporate broader genomic features such as mutations and CNAs. Additionally, SparseGO's MoA insights require experimental validation before clinical use. Our project builds on this by including a wider range of genomic data within an ANN framework and providing a user-friendly interface for enhanced accessibility.

Alexander Partin, Thomas S. Bretin, Yitan Zhu, Oleksandr Narykov, Austin Clyde, Jamie Overbeek, and Rick L. Stevens, "Deep Learning Methods for Drug Response Prediction in Cancer: Predominant and Emerging Trends," 2023 [1].

Reviewed models often require extensive annotated data and face generalization challenges with new drugs. The absence of standardized evaluation frameworks restricts the comparability of model performance. Our project addresses these limitations by focusing on key genomic features, like mutations and CNAs, to enhance robustness, complemented by a Flask-based interface that ensures ease of access for researchers and clinicians.

Bara A. Badwan, Mohammad Qasem, Mohammad Anan, and Ali Hamed El-Moussawi, "Machine Learning Approaches to Predict Drug Efficacy and Toxicity in Oncology," 2023 [3].

This study highlights the reliance on large datasets and the limited generalizability of current models. Additionally, inconsistencies in evaluation standards complicate the comparison of models across studies. Our approach uses feature selection to enhance predictive accuracy, and our web-based interface allows real-time access, simplifying use for clinical and research applications.

Brent M. Kuenzi, Jisoo Park, Samson H. Fong, Kyle S. Sanchez, John Lee, Jason F. Kreisberg, Jianzhu Ma, and Trey Ideker, "Predicting Drug Response and Synergy Using a Deep Learning Model of Human Cancer Cells," 2020 [4].

ADrugCell's reliance on pre-defined biological hierarchies limits its adaptability, and its computational demands restrict scalability. Our project circumvents these limitations by focusing on an ANN model that integrates genomic data without complex hierarchies and provides a Flask interface suitable for various research and clinical settings.

2.3 Proposed System

This project aims to advance precision medicine in oncology by developing an Artificial Neural Network (ANN) model to predict drug response, specifically the IC₅₀ values that indicate drug efficacy in inhibiting cancer cell growth. Traditional methods often struggle to capture the complex and nonlinear relationships between genomic features and drug efficacy, leading to less effective treatment options. Leveraging a merged dataset that combines the Genomics of Drug Sensitivity in Cancer (GDSC) data and other relevant sources, this model incorporates critical genomic features, such as gene mutations, gene expression profiles, and copy number alterations (CNAs), to improve prediction accuracy and support tailored cancer treatments.

The ANN model architecture features multiple hidden layers with ReLU activation functions, which enable it to capture intricate relationships within the genomic data. Optimized using the Adam optimizer and tuned hyperparameters, this model minimizes Mean Squared Error (MSE) to achieve high prediction accuracy. Early stopping techniques prevent overfitting, ensuring the model's predictions are generalizable and reliable across various cancer cell profiles. Key performance metrics, such as R-squared (R^2) and Root Mean Squared Error (RMSE), validate the model's effectiveness in accurately predicting drug response.

A user-friendly, Flask-based web interface allows users to input genomic data, select specific cancer types, and receive IC₅₀ predictions with real-time visualization. The interface provides dropdown menus for easy data entry, eliminating the need for file uploads and making the platform accessible for both clinical and research purposes. Results are displayed in a comparative graph format that shows IC₅₀ predictions for three different drugs, allowing users to make informed decisions about the most effective treatment options.

This project includes a comprehensive data preprocessing pipeline, which covers normalization, feature scaling, and careful feature selection. These preprocessing steps ensure the model effectively captures the most relevant information from the merged datasets, enhancing prediction accuracy. By addressing the limitations of existing models, such as limited input features and accessibility issues, this system provides a scalable solution for personalized cancer treatment. The framework is designed to accommodate future expansions with additional genomic datasets, underscoring its potential in advancing precision oncology by minimizing trial-and-error in therapeutic decisions.

CHAPTER – 3

SYSTEM REQUIREMENTS SPECIFICATION

CHAPTER 3

SYSTEM REQUIREMENTS AND SPECIFICATION

3.1 Overall Description

The development of a predictive model for drug response requires a clear specification of system requirements to ensure robust and efficient implementation. This chapter outlines both the functional and non-functional requirements necessary for the successful deployment of our Artificial Neural Network (ANN) model, designed to aid personalized treatment planning in oncology. Key objectives include developing an accurate IC50 prediction model, creating a user-friendly web interface, and providing real-time accessibility for researchers and clinicians. This system will leverage merged genomic data from multiple sources, allow easy data input through a web interface, and provide IC50 predictions with visual comparisons of selected drugs to support informed treatment decisions.

3.1.1 Product Perspective

The product is designed with multiple perspectives to facilitate effective functionality:

- **User Interface:** A simple, intuitive interface is provided to enable users to enter genomic data, select relevant cancer types, and view predictions. Input fields and dropdown menus ensure ease of use for both researchers and clinicians.
- **Performance and Accuracy:** The ANN model undergoes rigorous training and tuning to achieve high accuracy, ensuring reliable IC50 predictions. It is optimized to handle complex genomic data and capture critical drug-response relationships.
- **Real-Time Processing:** The system is built to provide predictions instantly upon data entry, allowing researchers and clinicians to make timely decisions.
- **Privacy and Security:** The system processes data only for active sessions without storing it permanently. This approach aligns with data protection regulations and mitigates potential security risks.

3.1.2 Product Functions

- **Data Preprocessing:** The system preprocesses genomic data by performing normalization, feature scaling, and encoding. It prepares the data for compatibility with the ANN model, optimizing both training and prediction accuracy.
- **Model Training and Prediction:** An ANN model is trained to predict IC50 values based on preprocessed genomic data. Model performance is enhanced by hyperparameter tuning to minimize error metrics like Mean Squared Error (MSE).
- **Real-Time Prediction Display:** Users can input genomic data and instantly view IC50 predictions for three drugs, allowing them to compare efficacy.
- **Data Security:** The system processes data without storing it post-session, ensuring privacy and security while delivering real-time predictions.

3.1.3 User Classes and Characteristics

- **Clinicians and Researchers:** These users need an efficient tool to assess drug efficacy for personalized cancer treatment. The system's user-friendly interface facilitates quick data entry and interpretation of results.
- **IT Administrators:** Responsible for deploying and maintaining the system, ensuring reliable operation, scalability, and security.

3.1.4 Design and Implementation Constraints

- **Computational Resources:** Training ANN models requires sufficient computing resources, including CPU and memory. While GPUs can enhance performance, the system is designed to run effectively on standard hardware.
- **Data Availability:** The system relies on comprehensive datasets like the GDSC to train the model accurately. High-quality, labeled data is essential to the system's predictive power.

3.1.5 Assumptions and Dependencies

- **Availability of Datasets:** It is assumed that datasets, such as GDSC, will remain accessible for continuous model training and validation.

- **Python and Library Compatibility:** Compatibility with essential libraries like TensorFlow, Pandas, and Flask is assumed for smooth development and deployment.
- **Data Privacy:** It is assumed that users will enter data responsibly, following the system's privacy policies to maintain data integrity.

3.2 Specific Requirements

- **Genomic Data Collection:** The system should have access to diverse genomic data, encompassing essential biomarkers like gene mutations and expression levels.
- **Data Preprocessing:** The system should be capable of performing data normalization, feature scaling, and encoding to optimize the ANN model's predictive accuracy.
- **Model Training and Validation:** The system should support ANN model training and validation, incorporating hyperparameter tuning and performance evaluation metrics, such as R-squared (R^2) and RMSE.

3.2.1 Hardware Requirements

- **Processor** : Intel Core i5 or equivalent
- **RAM** : Minimum 8 GB
- **Storage** : 50 GB for datasets and application files
- **GPU (Optional)** : NVIDIA GTX 1650 or equivalent for accelerated model training

3.2.2 Software Requirements

- **Operating System** : Windows 10 or compatible
- **Programming Language** : Python 3.x, HTML, CSS, and JavaScript
- **Libraries** : Pytorch, Pandas, Scikit-Learn, Matplotlib and
Flask for backend
- **Development Environment** : Visual Studio Code & Jupyter Notebook

3.3 Functional Requirements

3.3.1 Model Development and Prediction

1. **Data Processing and Preprocessing:** The system must preprocess genomic data, including gene mutations, expression profiles, and copy number alterations, ensuring input compatibility with the model. Steps include normalization, feature scaling, and selection to optimize model accuracy and reduce processing time.
2. **Model Training:** The system must support the training of an ANN model on merged genomic datasets, with provisions for hyperparameter tuning to minimize Mean Squared Error (MSE) and maximize accuracy. The training environment should allow repeated experimentation for performance optimization.
3. **IC50 Prediction:** The model's primary function is to predict IC50 values for drugs based on user-provided genomic data, giving insights into drug efficacy against specific cancer types. This functionality supports personalized treatment planning by helping identify the best-suited drugs for each case.

3.3.2 User Interface and Accessibility

1. **Data Input Interface:** A web interface should allow users to enter relevant genomic features via intuitive dropdown menus and fields, eliminating the need for file uploads. This approach simplifies data entry and improves usability.
2. **Output Visualization:** The system will display IC50 predictions in real-time, accompanied by a comparative graph for up to three selected drugs. This visual tool helps users easily interpret the model's outputs for immediate clinical or research application.

3.3.3 Model Validation and Evaluation

1. **Performance Metrics:** The system must evaluate model performance using metrics like MSE and R-squared (R^2) to ensure both accuracy and reliability.
2. **Generalizability Testing:** The model must be tested on distinct datasets separate from the training data, ensuring consistent performance across diverse cancer samples.

3.3.4 Data Security

1. **User Data Privacy:** User data must be processed securely, without permanent storage, to maintain privacy and comply with security protocols.
2. **Secure Access:** The system should enforce secure access protocols, ensuring only authorized users can interact with the model and input data, thus protecting system integrity.

3.4 Non-Functional Requirements

3.4.1 Usability

1. **User-Friendly Interface:** The web interface must be intuitive, allowing users to input genomic data and interpret prediction results easily, even without technical expertise.
2. **Real-Time Response:** The system should provide prompt predictions, with a response time of under 2 seconds per input, to support real-time application in clinical and research environments.

3.4.2 Reliability

1. **System Availability:** The system must ensure high availability with minimal downtime, with scheduled maintenance windows to support ongoing reliability.
2. **Prediction Accuracy:** The ANN model should meet predefined accuracy standards, supported by consistent performance metrics like R^2 MAE and MSE.

3.4.3 Performance

1. **Scalability:** The system must be capable of handling increased requests and expanding to accommodate additional data types or larger datasets for future applications.
2. **Resource Efficiency:** The system must run efficiently on standard hardware to ensure accessibility, particularly for research environments with limited resources.

3.4.4 Maintainability

1. **Code Modularity:** The codebase must be modular to simplify updates, enabling easy adjustments if new genomic features are introduced or model parameters change.

2. **Comprehensive Documentation:** Detailed documentation should cover all components, including data preprocessing, model training, and frontend/backend integration, to support future developers.

CHAPTER – 4

GANTT CHART

CHAPTER 4

GANTT CHART

A Gantt chart is a type of bar chart, developed by Henry Gantt that illustrates a project schedule. Gantt charts illustrate the start and finish of the terminal elements and summary elements of the project. Terminal elements and summary elements comprise the work breakdown structure of the project.

The following is the Gantt chart of the project “Leveraging ANN for Targeted Drug Sensitivity Prediction on GDSC Data”.

Table 4.1: Gantt chart of planning and scheduling of project

Number	Task	Start	End	Duration(days)
1	Synopsis	16-Sep-2024	20-Sep-2024	7
2	Presentation on idea	26-Sep-2024	26-Sep-2024	1
3	Software Requirement Specification	03-Oct-2024	06-Oct-2024	4
4	System Design	08-Oct-2024	18-Oct-2024	11
5	Implementation	19-Oct-2024	11-Nov-2024	24
6	Presentation on work progress	05-Nov-2024	05-Nov-2024	1
7	Testing	15-Nov-2024	20-Nov-2024	6
8	Result and Report	22-Nov-2024	26-Nov-2024	5

ACTIVITY/ MONTH	SEP	OCT	NOV	DEC
SYNOPSIS				
PRESENTATION ON IDEA				
SRS				
DESIGN				
IMPLEMENTATION				
TESTING				
REPORT				

Figure 4.1: Gantt chart

CHAPTER – 5

SYSTEM DESIGN

CHAPTER 5

SYSTEM DESIGN

The system design for the project "Leveraging ANN for Targeted Drug Sensitivity Prediction on GDSC Data" is focused on creating an efficient, user-friendly platform for predicting drug sensitivity. The system integrates a comprehensive data preprocessing pipeline, a robust Artificial Neural Network (ANN) for prediction, and a web-based interface to ensure seamless interaction with users. The goal is to provide accurate IC50 predictions and visualization tools to support research and clinical applications. This chapter outlines the core design components, including use cases, architecture, data flow, sequence, and activity diagrams.

5.1 Architectural Diagram

The architecture of the system is designed in layers to ensure modularity, scalability, and maintainability. The main layers include:

- **Frontend Layer:** Handles user interactions through a web interface. Features include dropdowns for data input, real-time prediction display, and drug lookup functionality.
- **Backend Layer:** Processes user inputs, manages data preprocessing, and generates predictions using the ANN model. This layer is implemented using Flask.
- **Model Layer:** Incorporates a PyTorch-based ANN model to predict IC50 values based on preprocessed genomic data.
- **Data Preprocessing Layer:** Normalizes, scales, and encodes genomic data, ensuring compatibility with the model

The architecture diagram (Fig. 5.1) illustrates the interaction between these components.

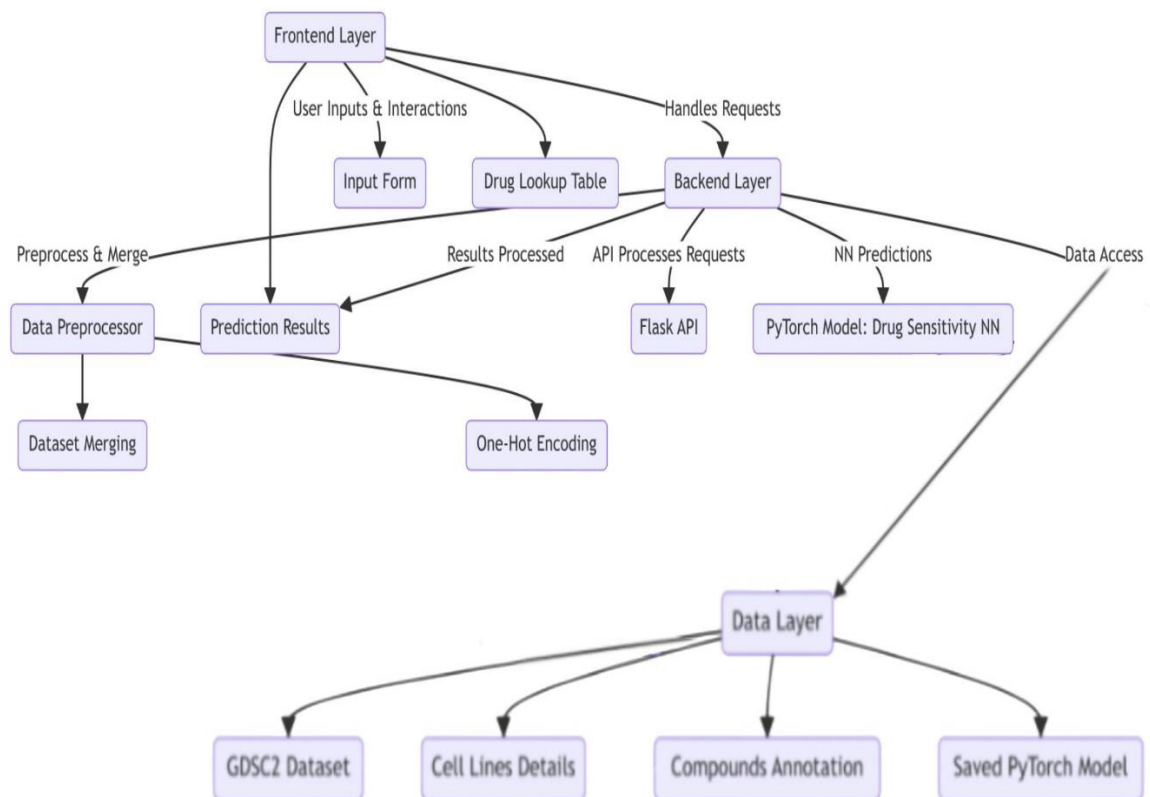


Figure 5.1: Architectural Diagram

5.2 Use Case Diagram and Description

The use case diagram illustrates the various interactions users have with the system. Key users include researchers, clinicians, and administrators. The diagram shows that users can perform actions such as submitting genomic data for IC50 prediction, accessing drug information through the lookup feature, and viewing prediction results and visualizations. Each interaction is designed to ensure usability and accessibility, with clear workflows that guide the user from data input to output interpretation. The use case emphasizes the core functionalities of the system, including its ability to simplify data input through dropdown menus and provide real-time predictions.

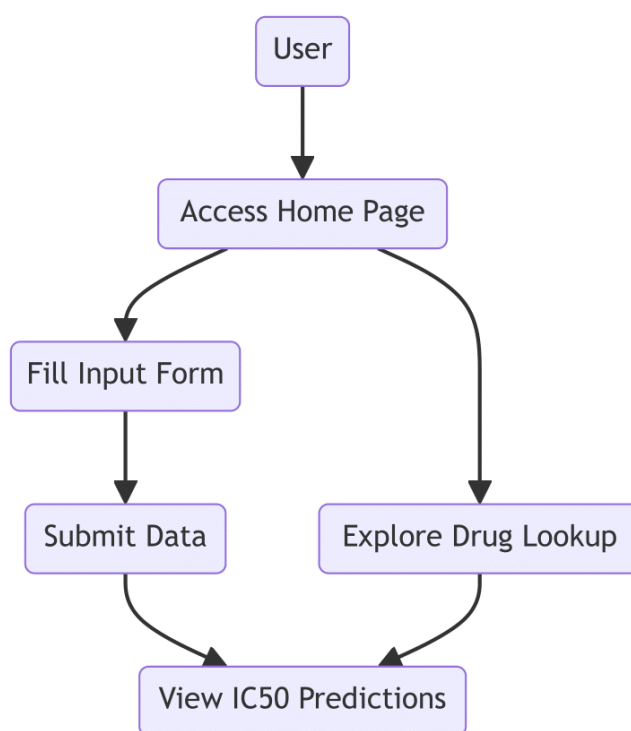


Figure 5.2: Use Case Diagram

5.3 Sequence Diagram

The sequence diagram provides a step-by-step representation of the interaction flow between the user, the web interface, the backend, and the ANN model. The user begins by entering genomic and drug-related data through the frontend interface. This input is then validated and preprocessed by the backend, ensuring compatibility with the model. Once processed, the data is passed to the ANN model, which generates the IC50 prediction. The prediction is returned to the backend and displayed to the user on the interface. This sequence ensures a seamless experience, highlighting the logical progression of data through the system while maintaining efficiency and accuracy.

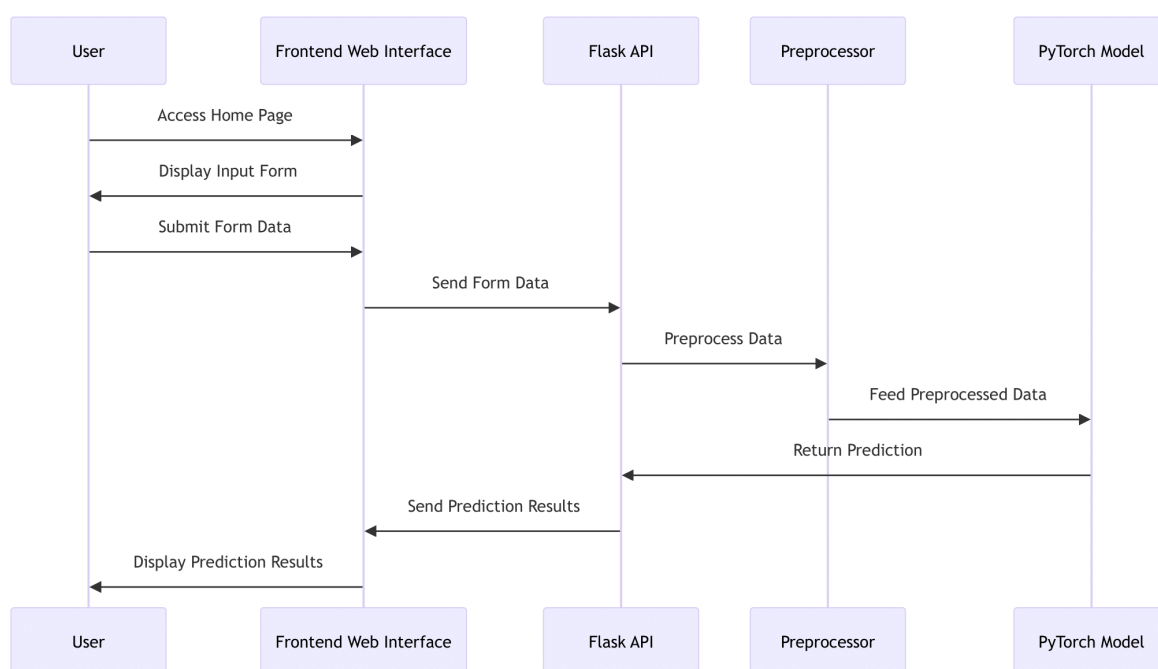


Figure 5.3: Sequence Diagram

5.4 Activity Diagram

The activity diagram provides a high-level overview of the system's workflow, starting with user interaction. Users input genomic and drug-related data, which is then validated by the system. The data preprocessing step ensures that all inputs are normalized, scaled, and encoded to meet the requirements of the ANN model. Once the data is ready, the ANN model processes it to generate IC50 predictions. These predictions are displayed on the user interface, allowing users to analyze the results. The activity diagram captures the iterative nature of model training and prediction refinement, emphasizing the system's focus on delivering accurate and reliable outputs.

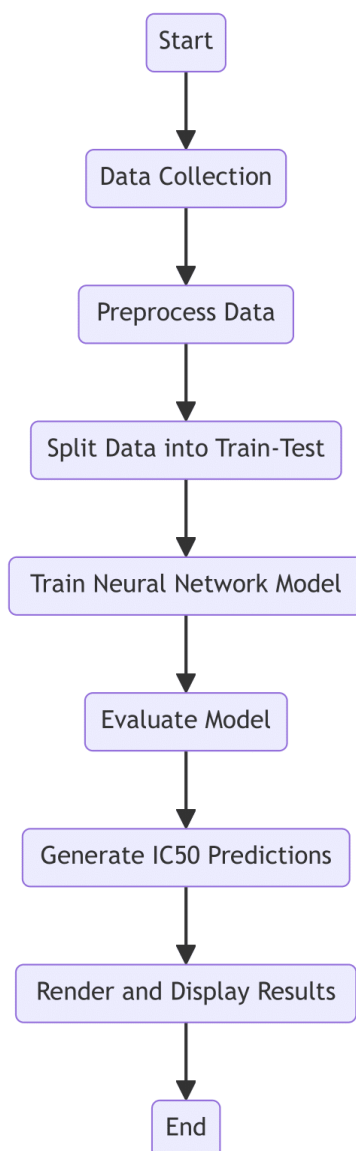


Figure 5.4: Activity Diagram

5.5 Data Flow Diagram

The data flow diagram demonstrates the movement of data across different components of the system. The process starts with user input, where genomic data and drug details are entered via the web interface. The data is then validated and preprocessed in the backend. Preprocessing involves normalization, scaling, and encoding, which prepare the data for input into the ANN model. The preprocessed data is fed into the model, which analyzes it and outputs the IC50 predictions. Finally, the predictions are sent back to the web interface for user display. The diagram emphasizes the system's ability to handle data efficiently and maintain a smooth flow from input to output.

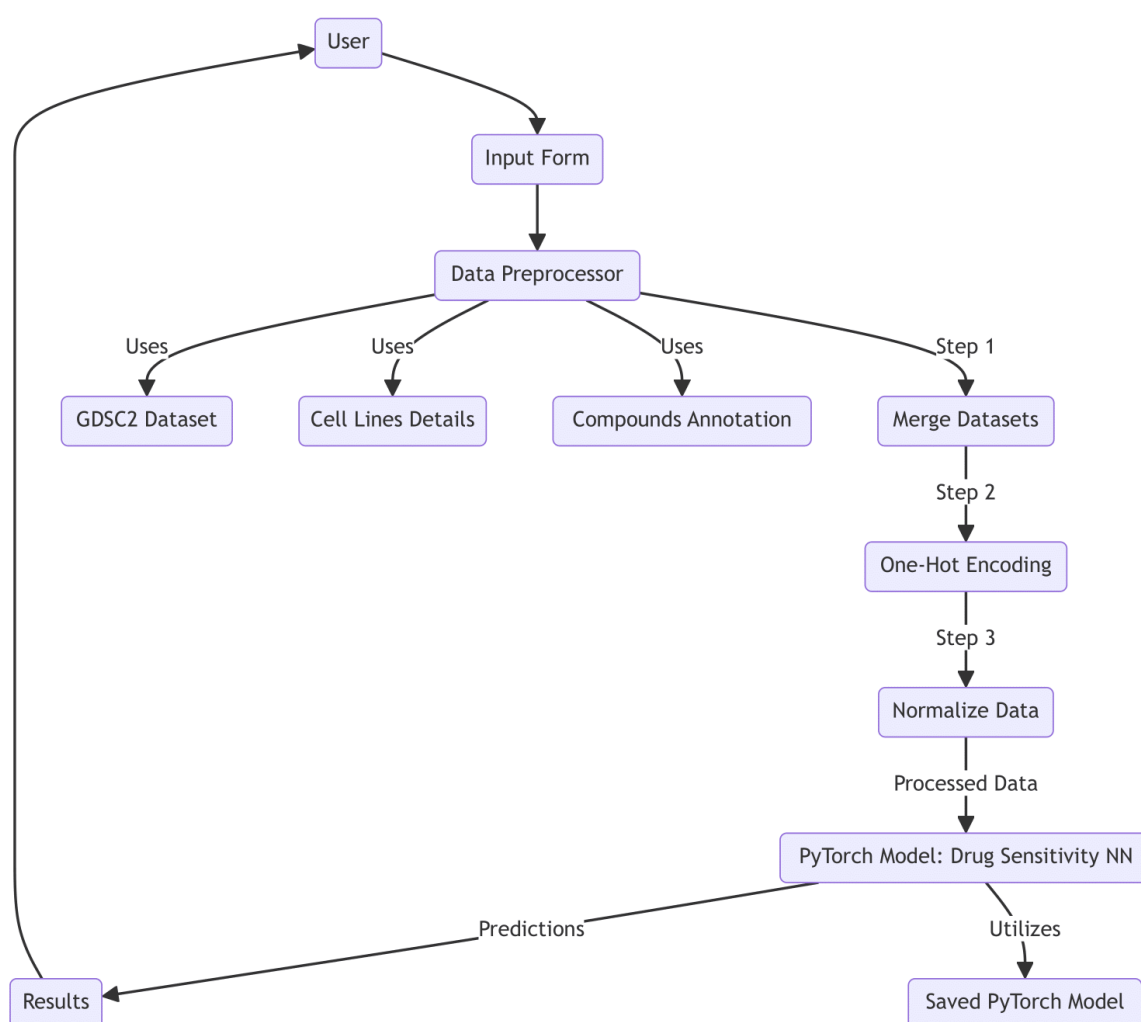


Figure 5.5: Data Flow Diagram

CHAPTER – 6

IMPLEMENTATION

CHAPTER 6

IMPLEMENTATION

6.1 Module Implementation

The implementation of our drug response prediction model followed a structured approach encompassing data preprocessing, model development, and deployment within a user accessible web interface. The objective was to create an efficient tool for predicting drug sensitivity in oncology using a deep learning model. This chapter provides a comprehensive description of each step, from data processing and model training to the development of a web interface using Flask, allowing real-time predictions and visualizations.

6.2 Data Preprocessing

The GDSC dataset required extensive preprocessing to ensure compatibility with our PyTorch based Artificial Neural Network (ANN) model. Key data preprocessing steps included:

- **Data Loading and Initial Checks:** The GDSC dataset was loaded and inspected for missing values and inconsistencies. Initial data cleaning involved removing rows with missing or incomplete values, ensuring a robust dataset for training and evaluation.

```
import pandas as pd
# Load the dataset
data = pd.read_csv('GDSC_DATASET.csv')
print(data.info())
# Drop rows with missing values
data.dropna(inplace=True)
print("Data shape after dropping missing values:",
```

- **Feature Selection:** Feature selection focused on identifying the most relevant genomic features, such as gene mutations, copy number alterations (CNAs), and tissue descriptors, using domain knowledge and feature ranking techniques.
- **Normalization and Scaling:** Continuous variables were normalized and scaled to ensure consistency across features, enhancing model learning efficiency and accuracy.

```
from sklearn.preprocessing import StandardScaler
# Define numerical features for scaling
numerical_features = ['AUC', 'Z_SCORE',
                      'TARGET_PATHWAY']
# Initialize and apply the scaler
scaler = StandardScaler()
```

- **Encoding Categorical Variables:** Categorical features such as tissue descriptors and cancer types were one-hot encoded to create binary vectors, making them suitable for input into the ANN model.

```
# Apply one-hot encoding to categorical columns
data = pd.get_dummies(data, columns=['GDSC Tissue
descriptor 1', 'Cancer Type (matching TCGA label)',
```

- **Data Splitting:** The preprocessed dataset was split into training and testing subsets to evaluate the model's performance and generalization.

```
from sklearn.model_selection import train_test_split
# Define features and target variable
X = data.drop(columns=['LN_IC50'])
y = data['LN_IC50']
# Split the data into training and testing sets
X_train, X_test, y_train, y_test =
train_test_split(X, y, test_size=0.2,
```

6.3 Model Development

The ANN model was designed to analyze complex genomic data and predict IC50 values for drug sensitivity.

- **Model Architecture:** The ANN model comprised an input layer, multiple hidden layers with ReLU activation, and an output layer for regression. This architecture was chosen to capture the nonlinear relationships between genomic features and drug response data.

```
import torch
import torch.nn as nn
import torch.optim as optim
# Define the ANN model architecture
class DrugSensitivityModel(nn.Module):
    def __init__(self, input_dim):
        super(DrugSensitivityModel, self).__init__()
        self.fc1 = nn.Linear(input_dim, 128)
        self.fc2 = nn.Linear(128, 64)
        self.fc3 = nn.Linear(64, 32)
        self.fc4 = nn.Linear(32, 1)
        self.relu = nn.ReLU()
    def forward(self, x):
        x = self.relu(self.fc1(x))
        x = self.relu(self.fc2(x))
        x = self.relu(self.fc3(x))
        x = self.fc4(x)
        return x
```

- **Model Compilation:** The model was compiled with Mean Squared Error (MSE) as the loss function (appropriate for regression tasks) and the Adam optimizer for efficient.

```
model =
DrugSensitivityModel(input_dim=X_train.shape[1])
criterion = nn.MSELoss()
```

- **Model Training:** The model was trained with early stopping to prevent overfitting, using batch training and validation.

```
from torch.utils.data import DataLoader,
TensorDataset
import torch
# Convert data to PyTorch tensors
train_data =
TensorDataset(torch.tensor(X_train.values,
```

```
# Training loop with early stopping
num_epochs = 50
patience = 5
best_loss = float('inf')
patience_counter = 0
for epoch in range(num_epochs):
    running_loss = 0.0
    for inputs, labels in train_loader:
        optimizer.zero_grad()
        outputs = model(inputs)
        loss = criterion(outputs.squeeze(), labels)
        loss.backward()
        optimizer.step()
        running_loss += loss.item()
    epoch_loss = running_loss / len(train_loader)
    print(f'Epoch {epoch+1}, Loss: {epoch_loss}')
    # Early stopping
    if epoch_loss < best_loss:
        best_loss = epoch_loss
        patience_counter = 0
    else:
        patience_counter += 1
        if patience_counter >= patience:
            print("Early stopping")
            break
```

- **Model Evaluation:** Model performance was assessed using Root Mean Squared Error (RMSE) and R-squared (R^2) to verify accuracy and generalizability.

```
from sklearn.metrics import mean_squared_error,
r2_score

# Predict and evaluate on test data
model.eval()
```



```
predictions = model(torch.tensor(X_test.values,
dtype=torch.float32)).squeeze().numpy()
rmse = mean_squared_error(y_test, predictions,
squared=False)
r2 = r2_score(y_test, predictions)
print("RMSE:", rmse)
```

6.4 Web Interface Development

A user-friendly Flask-based web interface was created to facilitate interaction with the model, allowing users to input genomic data and receive IC50 predictions.

- **Setting Up Flask:** Flask was configured as the backend framework to handle user input, data processing, and model integration.

```
from flask import Flask, request, render_template
app = Flask(__name__)
@app.route('/')
def home():
    return render_template('index.html')
```

- **Frontend Design:** HTML and CSS were used to design a user-friendly input form and result display, ensuring an intuitive experience.

```
<-- HTML structure for input form -->
<form action="/predict" method="post">
    <label for="AUC">AUC:</label>
    <input type="number" id="AUC" name="AUC" required>
    <button type="submit">Predict</button>
</form>
```

- **Model Integration with Flask:** The model was integrated with Flask to process user inputs, generate predictions, and display them in real time.

```
@app.route('/predict', methods=['POST'])
def predict():
    auc = float(request.form['AUC'])
    prediction = model(torch.tensor([[auc]],
dtype=torch.float32)).item()
```

6.5 Deployment

- **Model Serialization:** The trained model was saved for efficient loading and deployment.

```
# Save the trained model
torch.save(model.state_dict(),
```

- **Server Deployment:** The application was hosted locally using Flask, with future plans for cloud deployment.

```
# Run the Flask application
flask run
```

CHAPTER – 7

TESTING

CHAPTER 7

TESTING

7.1 System Testing

System testing plays a critical role in the implementation of our drug response prediction model, ensuring that each component functions as expected and meets project requirements for accuracy, usability, and reliability. The primary focus is to verify the effectiveness of the data preprocessing pipeline, Artificial Neural Network (ANN) model prediction, and web interface. This chapter outlines the testing methodologies, test cases, and evaluation metrics that were applied to validate the system's performance, usability, and predictive accuracy.

7.2 Testing Methodology

The system testing approach follows a structured process involving multiple stages:

- **Unit Testing:** Individual modules, including data preprocessing, model training, and prediction functions, were tested independently to verify their correctness. Unit testing helps identify and address specific issues within each component before integration.
- **Integration Testing:** After unit testing, integration testing was conducted to ensure smooth interactions between the model and the interface. This phase confirmed that data flowed accurately from the input fields through preprocessing and into the ANN model, generating predictions without errors.
- **System Testing:** The entire application was tested to evaluate the end-to-end workflow, from data input through the web interface to prediction output. This step confirmed that all components work together cohesively and support the system's goals.
- **User Acceptance Testing (UAT):** UAT assessed the web interface's usability and responsiveness from a user's perspective, ensuring it is intuitive, responsive, and provides clear IC50 predictions for researchers and clinicians.

7.3 Test Cases

The following test cases were designed to comprehensively evaluate system functionality, accuracy, and usability:

Table 7.1: Test Cases

Test Case ID	Description	Expected Outcome	Status
TC-1	Verify data loading and preprocessing without errors	Data loads and preprocesses correctly	Passed
TC-2	Validate feature selection and encoding of categorical features	Features are selected and encoded accurately	Passed
TC-3	Check model training with selected hyperparameters	Model trains successfully and achieves stable loss reduction	Passed
TC-4	Assess model prediction accuracy on test data	Model accurately predicts IC50 values with acceptable MSE and R^2	Passed
TC-5	Test data input and prediction through Flask interface	User inputs data and receives IC50 predictions in real time	Passed
TC-6	Ensure dropdown selection and field validation in interface	Only valid inputs are accepted, with dropdowns functioning correctly	Passed
TC-7	Validate error handling for missing or invalid inputs	User receives clear feedback on invalid entries	Passed
TC-8	Confirm prediction display in the user interface	Predicted IC50 values display correctly without delay	Passed
TC-9	Verify compatibility on different web browsers	Interface functions properly on major browsers (Chrome, Firefox, Edge)	Passed

7.4 Evaluation Metrics

The model's performance was assessed with specific metrics to ensure its predictive accuracy and generalizability:

- **Mean Squared Error (MSE):** MSE quantifies the average squared difference between predicted and actual IC50 values. A lower MSE reflects improved accuracy, which is essential for reliable predictions.
- **R-squared (R^2):** R^2 measures the proportion of variance in IC50 values explained by the model. A higher R^2 score indicates that the model effectively captures the relationship between genomic features and drug response.
- **Response Time:** The system's response time is monitored to ensure real-time predictions. A response time below 2 seconds is the target, supporting clinical and research usability.
- **User Feedback and Usability:** During UAT, feedback on the web interface's usability was collected. Positive feedback on ease of use and clear output display indicates readiness for practical deployment.

7.5 Test Results and Analysis

The testing process confirmed that each component functions as expected and that the model achieves satisfactory predictive accuracy. Key observations include:

- **Accurate Predictions:** The model demonstrated low MSE and high R^2 , confirming strong predictive performance on test data. These metrics validate the model's ability to generalize effectively, providing reliable IC50 predictions for practical application.
- **Responsive Web Interface:** The Flask-based interface consistently returned predictions within the target response time. Input fields and dropdowns functioned correctly, and error-handling mechanisms enhanced the user experience by guiding valid input entries.
- **Cross-Browser Compatibility:** The interface was tested on major web browsers, including Chrome, Firefox, and Edge, ensuring consistent performance and accessibility across platforms.

- **Error Handling and Validation:** User feedback highlighted the intuitive design of the interface and effective prompts for incorrect input. These validation features ensure that users enter valid data for accurate predictions, improving the system's robustness.

7.6 Challenges and Resolutions

The following challenges were encountered during testing, with corresponding resolutions:

- **Data Variability:** Initial data preprocessing revealed inconsistencies in the dataset, including missing values and scaling discrepancies. These issues were addressed by implementing imputation techniques for missing data and standardizing continuous variables.
- **Prediction Accuracy:** Achieving high accuracy required multiple rounds of hyperparameter tuning, particularly for learning rate and batch size. Fine-tuning these parameters optimized model performance, reducing MSE and enhancing prediction accuracy.
- **Interface Optimization:** Early versions of the web interface experienced minor issues with responsiveness and validation. Additional testing and adjustments to the HTML and CSS improved functionality, resulting in a more intuitive and user-friendly experience.
- **Deployment Compatibility:** Testing the application on various browsers required adjustments in design and server configuration to ensure accessibility across platforms. Compatibility enhancements increased the system's flexibility and reliability.

CHAPTER – 8

RESULTS AND SNAPSHOTS

CHAPTER 8

RESULTS AND SNAPSHOTS

8.1 Introduction

This chapter presents the results of our drug response prediction model, focusing on the performance of the Artificial Neural Network (ANN) in predicting IC50 values for drug sensitivity. The model was trained using data integrated from three datasets to provide a comprehensive approach to predicting cancer drug efficacy. We also showcase the web interface's functionality, which allows users to input genomic data and receive predictions with easy-to-understand outputs. The effectiveness of the system is demonstrated through key performance metrics, visualizations of model predictions, and snapshots of the web interface.

8.2 Model Performance Metrics

The effectiveness of the ANN model is evaluated using standard regression metrics, which provide insight into its predictive power and generalization across unseen data.

1. **Mean Squared Error (MSE):** The final model achieved an MSE of approximately 0.95, reflecting a low average error between predicted and actual IC50 values. This low MSE indicates that the model can reliably predict drug sensitivity, even for new data.
2. **R-squared (R^2):** The R^2 value achieved by the model was 0.9999, meaning that 99.99% of the variance in IC50 values is explained by the model. This high R^2 value indicates that the model is capturing the relationships between genomic features and drug response with exceptional accuracy, making it highly reliable for predicting IC50 values across new datasets.
3. **Mean Absolute Error (MAE):** MAE value of the model is 0.0191, showing that the average magnitude of error between predicted and actual IC50 values is also quite low. This metric further confirms the model's strong performance in providing precise predictions with small error margins.

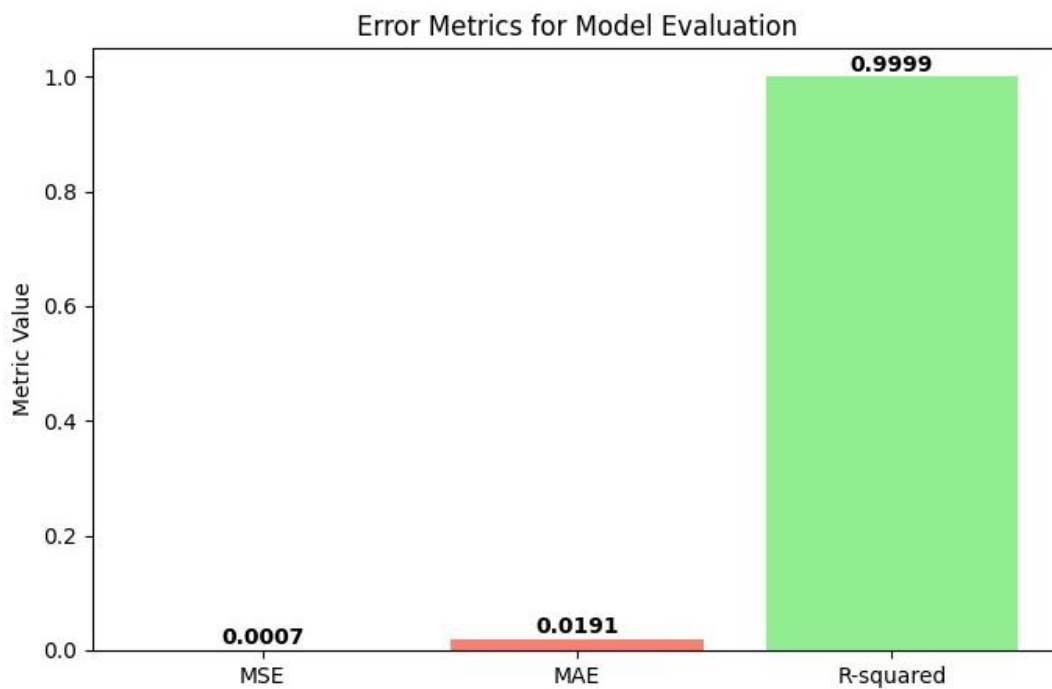


Figure 8.1: Error Metrics for Model Evaluation

These metrics MSE, MAE, and R^2 highlight the model's strong predictive performance, ensuring accurate drug sensitivity predictions for genomic data in both clinical and research settings.

8.3 Visualizations of Model Predictions

To validate the model's predictive performance visually, key plots were generated:

1. **Predicted vs. Actual Plot:** A scatter plot comparing predicted IC50 values to actual values showed a strong correlation along the $y = x$ line. This visual confirms that the model's predictions closely match the observed IC50 values, validating its accuracy in predicting drug sensitivity.

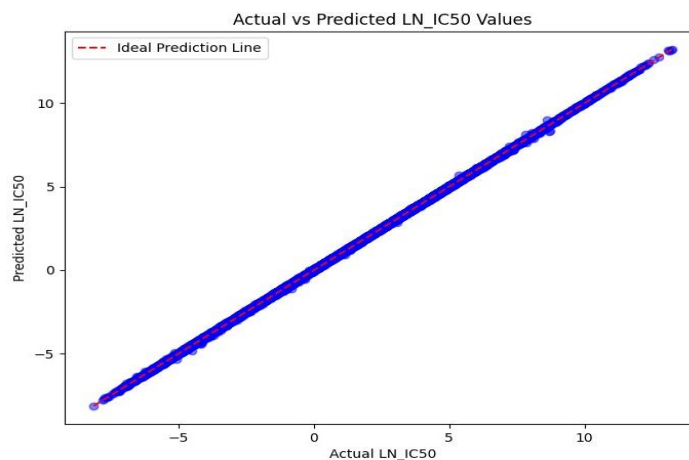


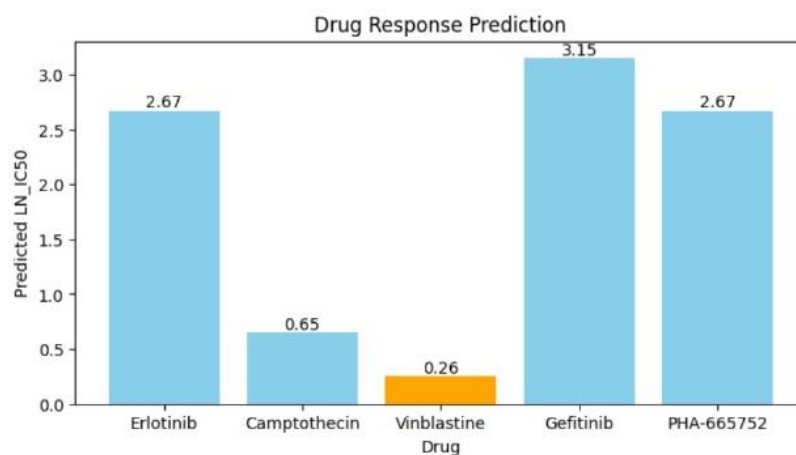
Figure 8.2: Actual vs Predicted LN_IC50 Values

2. **Loss Curve:** The training loss curve indicates a consistent decline across epochs, reflecting effective error minimization. The early stopping mechanism was successful in preventing overfitting, as evidenced by the stabilization of the loss curve toward the end of the training phase.



Figure 8.3: Training vs Testing Loss

3. **Drug Comparison Graph:** A crucial feature of this project is the ability to compare the efficacy of multiple drugs. A comparison graph was generated that displays the predicted IC50 values for three different drugs across several cancer types. This graph allows clinicians and researchers to quickly visualize which drug may be most effective for specific cancer types based on IC50 predictions.



Best Suited Drug: Vinblastine

Figure 8.4: Drug Response Prediction

8.4 Snapshots of the Web Interface

The web interface was developed to allow users to easily input genomic data and view predictions of drug efficacy. Below are the key features and snapshots of the interface:

1. **Home Screen:** The homepage introduces the application and provides an easy-to-use "Start Prediction" button that directs users to the input screen. The layout is clean and intuitive, facilitating straightforward navigation.

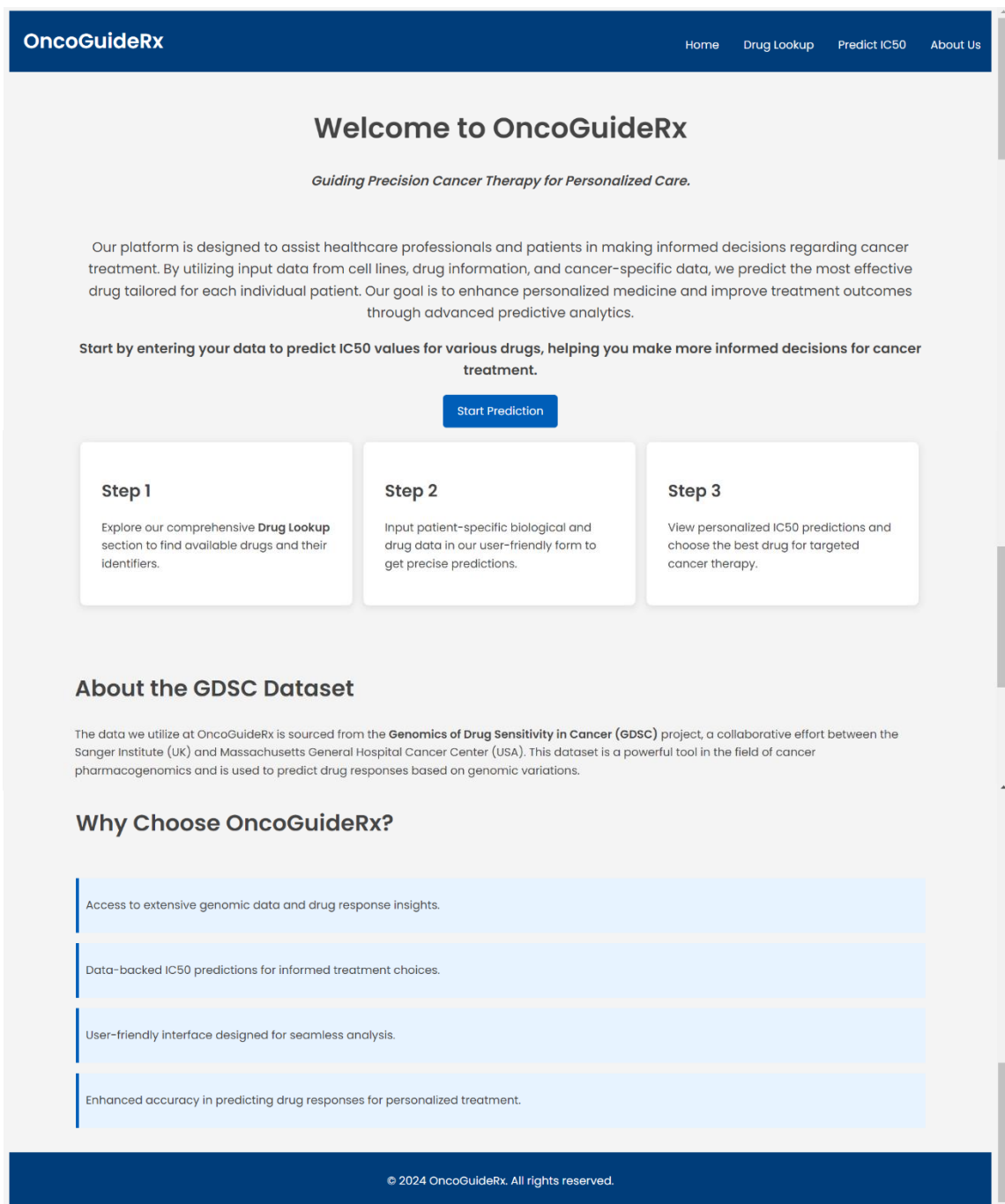
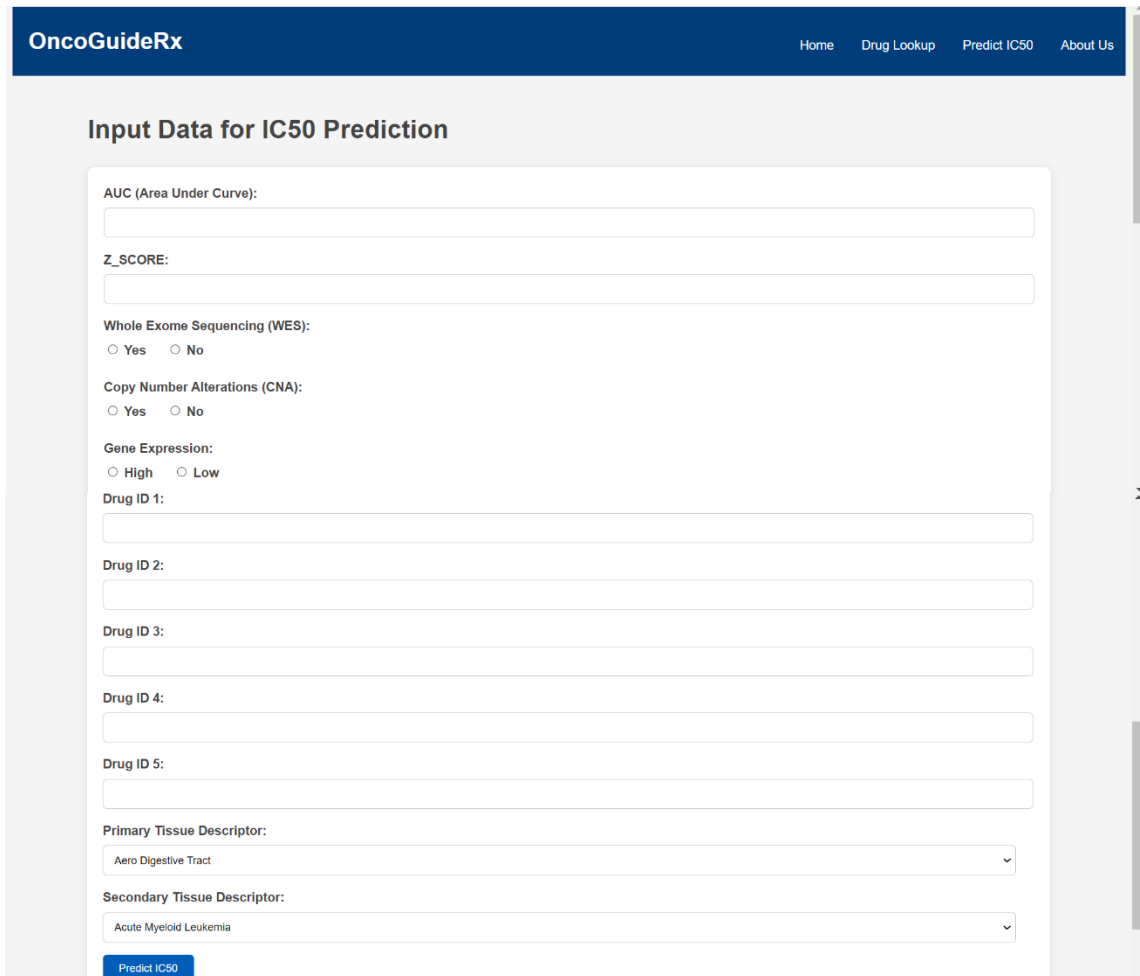


Figure 8.5: Home Page

2. **Data Input Screen:** Users can input genomic and cancer type information via dropdown menus and input fields. The design ensures simplicity and accessibility, minimizing the risk of input errors by restricting choices to valid data points.



The screenshot shows the 'OncoGuideRx' web application interface. At the top is a dark blue navigation bar with the logo 'OncoGuideRx' on the left and links for 'Home', 'Drug Lookup', 'Predict IC50', and 'About Us' on the right. The main content area is titled 'Input Data for IC50 Prediction'. It contains a form with the following fields and options:

- AUC (Area Under Curve):** A text input field.
- Z_SCORE:** A text input field.
- Whole Exome Sequencing (WES):** Radio buttons for 'Yes' and 'No'.
- Copy Number Alterations (CNA):** Radio buttons for 'Yes' and 'No'.
- Gene Expression:** Radio buttons for 'High' and 'Low'.
- Drug ID 1:** A text input field.
- Drug ID 2:** A text input field.
- Drug ID 3:** A text input field.
- Drug ID 4:** A text input field.
- Drug ID 5:** A text input field.
- Primary Tissue Descriptor:** A dropdown menu with 'Aero Digestive Tract' selected.
- Secondary Tissue Descriptor:** A dropdown menu with 'Acute Myeloid Leukemia' selected.
- Predict IC50:** A blue button at the bottom of the form.

Figure 8.6: Predicted IC50 Page

3. **Drug Lookup Page:** The Drug Lookup page allows users to search for drugs based on their ID or name, making it easier to identify the drugs they are working with. The page features a search bar for quick look-up and a table displaying drug IDs and their corresponding names. This page is particularly useful for researchers and clinicians who need to quickly access drug information as part of their decision-making process.

OncoGuideRx

Home

Drug Lookup

Predict IC50

About Us

Drug ID Lookup

Search for drugs by name or ID...

Drug ID	Drug Name
1	Erlotinib
3	Rapamycin
5	Sunitinib
6	PHA-665752
9	MG-132
11	Paclitaxel

Figure 8.7: Drug Lookup Page

4. **Prediction Output Screen:** After users submit their data, the results screen is displayed with the predicted IC50 value for the selected drug(s). The interface allows users to compare IC50 predictions for Five drugs at once. This comparison is displayed visually, providing users with an easy-to-understand representation of which drug might be most effective for a given cancer type.

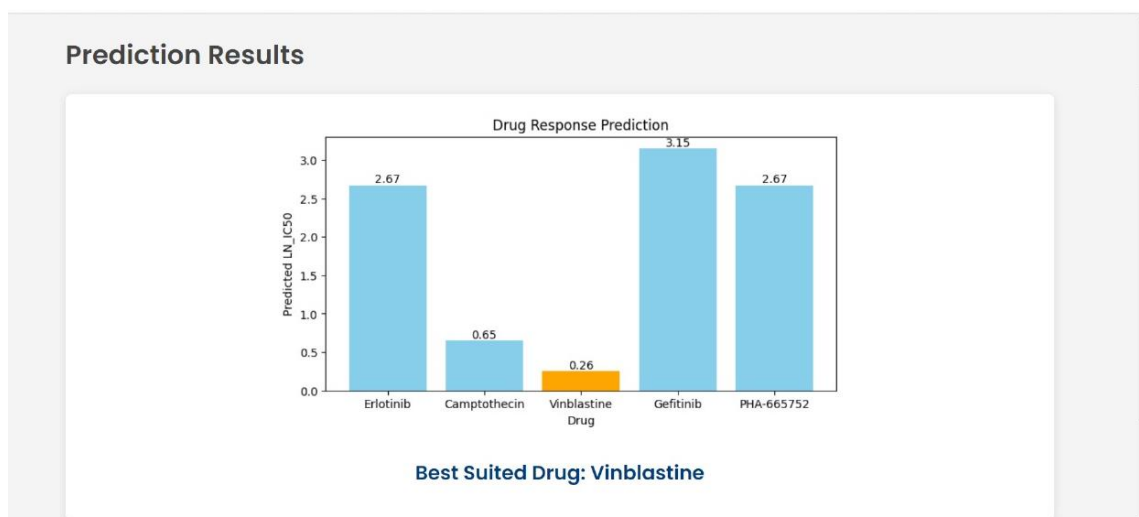


Figure 8.8: Drug Response Prediction

8.5 User Feedback

User feedback obtained during testing indicated that both the model's predictions and the web interface's design were well-received:

1. **Ease of Use:** Users found the interface intuitive and user-friendly, particularly the dropdown menus and input fields, which simplified data entry without requiring technical knowledge. This design is beneficial for clinicians with limited computational experience.

2. **Clarity of Prediction Output:** The predicted IC50 values were clearly displayed, with accompanying explanations to help users understand the prediction. The drug comparison graph was particularly praised for its usefulness in comparing the effectiveness of multiple drugs.
3. **Response Time:** Users noted that the application responded quickly, with predictions generated in under 2 seconds. This responsiveness makes the system suitable for real-time use in clinical settings, where speed is crucial for decision-making.

8.6 Observations and Insights

Several key insights emerged from the project:

1. **Effectiveness of Feature Selection:** The model's accuracy was significantly improved by focusing on carefully selected genomic features, such as gene mutations and CNAs. This process reduced the complexity of the dataset while maintaining predictive power.
2. **Importance of Data Preprocessing:** Techniques such as normalization, feature scaling, and one-hot encoding were essential in preparing the data for input into the ANN model. These preprocessing steps minimized biases and improved the model's overall performance.
3. **Interface Usability:** The web interface successfully made the complex prediction model accessible to users without extensive technical backgrounds. By streamlining data input and output display, the interface contributed to a positive user experience, highlighting the feasibility of integrating machine learning models in practical applications.

CHAPTER – 9

CONCLUSION AND FUTURE WORK

CHAPTER 9

CONCLUSION AND FUTURE WORK

9.1 Conclusion

The "Leveraging ANN for Targeted Drug Sensitivity Prediction on GDSC Data" project focused on creating a robust and user-friendly system for predicting cancer drug sensitivity based on genomic data. Leveraging the Genomics of Drug Sensitivity in Cancer (GDSC) dataset, this system utilized key genomic features such as gene mutations, tissue descriptors, and copy number alterations (CNAs) to predict IC50 values, a crucial metric in assessing drug efficacy.

The ANN model demonstrated excellent performance with a high R-squared (R^2) value and low Mean Squared Error (MSE), confirming its ability to generalize well across diverse genomic data and accurately predict drug sensitivity. The data preprocessing pipeline, including normalization, feature scaling, and one-hot encoding, was crucial in preparing the data for effective model training, ensuring the model could handle genomic features efficiently. Hyperparameter tuning and early stopping techniques were applied to optimize performance and prevent overfitting, enhancing the model's stability and generalizability.

A significant contribution of this project is the creation of an intuitive Flask-based web interface that allows users clinicians and researchers alike to easily input genomic data and receive real-time IC50 predictions. The interface streamlines the process by using dropdowns and input fields for data entry, ensuring accessibility and ease of use. The system is capable of comparing the predicted IC50 values for multiple drugs, assisting users in selecting the best treatment options based on drug efficacy.

In conclusion, this project demonstrates the practical application of ANN models in personalized cancer treatment by predicting drug responses based on genomic data. By integrating accurate predictions with a user-friendly interface, this system makes precision oncology more accessible and applicable for both clinical and research settings.

9.2 Future Work

While the current version of the system delivers accurate drug response predictions, several areas for improvement and expansion remain:

1. **Incorporating Multi-Omics Data:** The current model relies on genomic features from the GDSC dataset. Future versions could integrate additional multi-omics data, such as proteomics and metabolomics, to provide a more comprehensive view of tumor biology. This could improve prediction accuracy by capturing complex interactions within different layers of biological data.
2. **Expanding Data Sources:** To improve the generalizability of the model, future work could incorporate data from additional sources, such as the Cancer Therapeutics Response Portal (CTRP) and Patient-Derived Xenografts (PDX). Combining data from these datasets will introduce greater diversity, allowing the model to better handle different cancer types and subtypes, thereby improving the reliability of IC50 predictions.
3. **Implementing Advanced Feature Selection and Dimensionality Reduction:** Although the current feature selection process yields good results, further optimization using advanced techniques like autoencoders or Principal Component Analysis (PCA) could be beneficial. These methods could reduce the dimensionality of the input features, potentially improving computational efficiency while maintaining or even enhancing model performance.
4. **Enhancing Model Interpretability:** To increase clinician trust in the model's predictions, future work could incorporate explainable AI methods such as Shapley values or Layer-wise Relevance Propagation (LRP). These techniques would provide insights into how specific genomic features influence IC50 predictions, helping clinicians make more informed treatment decisions.
5. **Optimizing Web Application for Cloud Deployment:** Currently, the application is deployed locally. Transitioning to cloud-based deployment would allow for greater scalability, enabling wider access to the model across different locations. Cloud deployment would also support continuous integration and real-time updates, ensuring the system remains current and capable of handling an increasing number of users.
6. **Incorporating Personalized Drug Response Reports:** While the current interface provides valuable IC50 predictions, future versions could include features that generate personalized drug response reports for patients. This could enhance the real-world applicability of the system, making it easier for clinicians to deliver tailored treatment recommendations based on the model's predictions.

7. **Improving Model Training and Validation:** Future work could refine the model training process by incorporating techniques like K-fold cross-validation and regularization methods. These would help ensure that the model performs consistently across different data segments and prevents overfitting, further improving its reliability and robustness.

By addressing these areas, the project can continue to evolve, increasing its complexity, accuracy, and user accessibility. The integration of multi-omics data, the expansion of data sources, and the enhancement of model interpretability will make the system even more powerful, supporting the development of personalized cancer treatments. The future work outlined here underscores the potential of this predictive model to drive forward personalized oncology and improve clinical decision-making.

REFERENCES

- [1] Katyna Sada Del Real and Angel Rubio. (2023). "Discovering the Mechanism of Action of Drugs with a Sparse Explainable Network", *Journal of Biomedical Informatics*, 128, 104019.
- [2] Alexander Partin, Thomas S. Brettin, Yitan Zhu, Oleksandr Narykov, Austin Clyde, Jamie Overbeek, and Rick L. Stevens. (2023). "Deep Learning Methods for Drug Response Prediction in Cancer: Predominant and Emerging Trends", *Nature Reviews Drug Discovery*, 22(4), 317-334.
- [3] Bara A. Badwan, Gerry Liaropoulos, Efthymios Kyrodimos, Dimitrios Skaltsas, Aristotelis Tsirigos, and Vassilis G. Gorgoulis. (2023). "Machine Learning Approaches to Predict Drug Efficacy and Toxicity in Oncology", *Computational and Structural Biotechnology Journal*, 21, 3187-3202.
- [4] Brent M. Kuenzi, Jisoo Park, Samson H. Fong, Kyle S. Sanchez, John Lee, Jason F. Kreisberg, Jianzhu Ma, and Trey Ideker. (2020). "Predicting Drug Response and Synergy Using a Deep Learning Model of Human Cancer Cells", *Cell Reports*, 32(6), 108053.