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.