Functional Requirements

1. Graphical User Interface Module
   1. Description

This module allows the user to provide the details of his/her operating system via a simple interface. The model will work in the backend to analyse the given input and predict if the system will be hit by a malware.

* 1. Input / Output Sequences

The user is allowed to give all the required details of their operating system as input. The input is received via the use of forms in a python GUI framework. The GUI framework also provides an output which indicates the chance of a particular system being hit by malware.

* 1. Functional Requirements

Req 1. Must have a working internet connection that allows him/her to connect to a network which enables connection to the trained model to predict the probability of being hit by malware

Req 2. A system that has the software and hardware capabilities to support the python GUI framework

1. Light GBM Model
   1. Description

Light GBM is a very fitting model to this dataset due to the size of the data and the effectiveness and memory efficiency of this particular method of Gradient boosting. It grows in a tree wise structure. But it differs from other tree-based algorithms due to its horizontal growth structure.

* 1. Input / Output Sequences

The input to the LGBM model is the training data set which is used to calculate and predict the probability of a malware infection. There is also a test dataset of smaller size which is derived from the user.

* 1. Functional Requirements

Req-1 The system must have access to the test dataset to provide accurate output to the user. The system must have full connectivity to the user data to produce an output.

1. XDeepFM Model
   1. Description

Extreme Deep Factorization machine is a relatively new deep learning technique that requires no manual feature engineering. Selection of features is very important and selecting raw features often give suboptimal results. Instead we use a factorization machine before applying a deep neural network for feature selection thus giving highly accurate results.

* 1. Input / Output Sequences

The input to the XDeepFM model is the training data set which is used to calculate and predict the probability of a malware infection. There is also a test dataset of smaller size which is derived from the user.

* 1. Functional Requirements

Req-1 The system must have access to the test dataset to provide accurate output to the user. The system must have full connectivity to the user data to produce an output.

1. Recurrent Neural Network
   1. Description

Recurrent Neural Network is one of the cornerstones of Deep learning. Being one of the oldest yet simplest and reliable model, it works very well for the presented data as computation power is cheaply available. Even if it is not the most efficient method in terms of power consumption, it gives us accurate results with the right hyper parameter tuning. It does have its own issues such as vanishing gradients where the gradient becomes very small. Since the network is feedforward, this reinforces in the future steps making the gradient smaller with each step until it becomes insignificant. This problem is countered using Long Short-Term Memory.

* 1. Input / Output Sequences

The input to the XDeepFM model is the training data set which is used to calculate and predict the probability of a malware infection. There is also a test dataset of smaller size which is derived from the user.

* 1. Functional Requirements

Req-1 The system must have access to the test dataset to provide accurate output to the user. The system must have full connectivity to the user data to produce an output.

1. Data Conversion
   1. Description

The GUI component for python will take data in JSON form which is provided by the testing data. This data however cannot be used in raw form and must be converted to a usable state. Hence this module converts the data from JSON to CSV so that it can be consumed by the GUI component to perform the operations needed on the training data.

* 1. Input / Output Sequences

Input will be the JSON format data acquired from the database which can be extracted as training data. The output is CSV formatted data that can be used by the GUI and supplied to the trained model behind the GUI.

* 1. Functional Requirements

Req-1 The system must have access to the test dataset to provide accurate output to the user. The system must have full connectivity to the user data to produce an output.

Req-2 A conversion framework within python which enables the data conversion is required.