**IMPLEMENTATION**

**MODULES:**

* Data Collection
* Dataset
* Data Preparation
* Feature Extraction
* Splitting the dataset
* Model Selection
* Analyze and Prediction
* Accuracy on test set
* Saving the Trained Model
* Prediction Module
* Model Evaluation Module

**MODULES DESCSRIPTION:**

**Data Collection:**

* In the first module of Employee Attrition prediction using machine learning, we make the data collection process. This is the first real step towards the real development of a machine learning model, collecting data. This is a critical step that will cascade in how good the model will be, the more and better data that we get; the better our model will perform.
* There are several techniques to collect the data, like web scraping, manual interventions. The dataset is located in the model folder. The dataset is referred from the popular dataset repository called kaggle. The following is the link of the dataset
* Kaggle Dataset Link:

https://www.kaggle.com/datasets/jayaprakashpondy/employee-attrition

**Dataset:**

* In this module, we use the dataset which is the primary source of data for the system. This dataset contains 1470 instances and 35 attributes, with a target attribute for classification (Yes vs. No).
* User Input: Data provided by users through the web interface, allowing for real-time malware detection based on user-uploaded files or input data.

**Data Preparation:**

* This module is responsible for preparing the Employee Attrition dataset for analysis. It involves tasks such as data cleaning, normalization, and feature selection. Specifically, 1 relevant attributes are selected from the original 2 attributes to optimize the machine learning models.
* Wrangle data and prepare it for training. Clean that which may require it (remove duplicates, correct errors, deal with missing values, normalization, data type conversions, etc.).
* Randomize data, which erases the effects of the particular order in which we collected and/or otherwise prepared our data.
* Visualize data to help detect relevant relationships between variables or class imbalances (bias alert!), or perform other exploratory analysis.
* NaN values are dropped from the dataset.
* Preprocess your data to handle missing values, outliers, and categorical variables. This step also involves scaling or normalizing numerical features if necessary.

**Feature Extraction:**

* If the dataset contains raw binaries or other non-numeric data, extract features that can be used by the machine learning models. This may involve static analysis (e.g., analyzing the binary's structure) or dynamic analysis (e.g., monitoring the binary's behavior during execution).
* A subset of features (permissions) is selected for model training to reduce dimensionality and focus on relevant attributes.

**Splitting the dataset:**

* Data Splitting and Validation is crucial for training and evaluating the model. This module divides the dataset into training, validation, and testing sets. It ensures that the model's performance is assessed accurately using proper validation techniques like cross-validation. Split the dataset into train and test. 80% train data and 20% test data.

**Model Selection:**

* This module handles the training of the machine learning models using the preprocessed data. It implements the Random Forest classifier and Bagging Classifier

***Random Forest Classifier:***

* Random Forest Classifier is a machine learning algorithm used for Classification tasks.
* Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique
* Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset
* Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.
* The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting

***Bagging Classifier:***

* Bagging Classifier is a machine learning algorithm used for Classification tasks.
* Bagging, also known as bootstrap aggregation, is the ensemble learning method that is commonly used to reduce variance within a noisy data set
* Bagging (or Bootstrap aggregating) is a type of ensemble learning in which multiple base models are trained independently and in parallel on different subsets of the training data
* Each subset is generated using bootstrap sampling, in which data points are picked at random with replacement.
* In the case of the bagging classifier, the final prediction is made by aggregating the predictions of the all-base model using majority voting.

**Analyze and Prediction:**

* This module performs feature selection to identify the most impactful attributes for employee attrition classification. It ensures that the selected features contribute significantly to the model's performance.

**Accuracy on test set:**

* Once the model is trained, it needs to be evaluated for its performance. This module involves splitting the dataset into training and testing subsets and assessing the model's accuracy, precision, recall, and F1-score.
* The Random Forest Classifier achieves a training accuracy of 100% and a testing accuracy of 98%. The Bagging Classifier model attains a training accuracy of 99% and a testing accuracy of 95%.

**Saving the Trained Model:**

* Once you’re confident enough to take your trained and tested model into the production-ready environment, the first step is to save it into an .h5 or .pkl file using a library like pickle.
* Make sure you have pickle installed in your environment.
* Next, let’s import the module and dump the model into .pkl file.

**Prediction Module:**

* This module handles real-time predictions using the trained models. Users can input new data through the frontend, and the module processes this data to classify it as Yes or No.

**Model Evaluation Module**

* This module evaluates the performance of the trained models using the testing dataset. It calculates accuracy metrics and other performance indicators to assess model effectiveness.
* Evaluate model accuracy, precision, recall, and F1-score.
* Generate confusion matrices for both models.
* Compare the performance of the Random Forest classifier and Bagging classifier models.
* Accuracy, precision, recall, and F1-score are used to evaluate model performance.
* Confusion matrix is visualized using seaborn heatmap to understand the classification results.