➢ PROBLEM STATEMENT :

A Machine Learning based Malware Detection System to detect and determine the **presence of malware on a system** and whether it is malicious or not.

➢ MOTIVATION :

As the world is getting more and more digital, the risk of our systems getting attacked by malwares or loss of sensitive information is also increasing.

So, detecting the malwares is very crucial. This motivated me to make a project on the Malware Detection System which could detect whether or not a dataset or set of files is malicious or safe.

Malware detection is crucial with malware's prevalence on the Internet because it **helps for early warning system or detection for the computer secure regarding malwares and cyber threats.** It prevents the information from getting compromised.

➢ INTRODUCTION :

✓ Intrusion :

An intrusion attack can be defined as **any malicious unauthorized activity on a digital network** that intends to gain **unauthorised access to files, privileges, data or money**, and cause **damage** to the computer system.

✓ Malware :

A malware is a malicious software which is intended to **damage and destroy a computer system** by several means, without the consent of the owner.

Malwares can be classified into different categories such as –

- **Viruses**
- **Worms**
- **Trojan Horses**
- **Spyware**
- **Ransomware**
- **Adware**

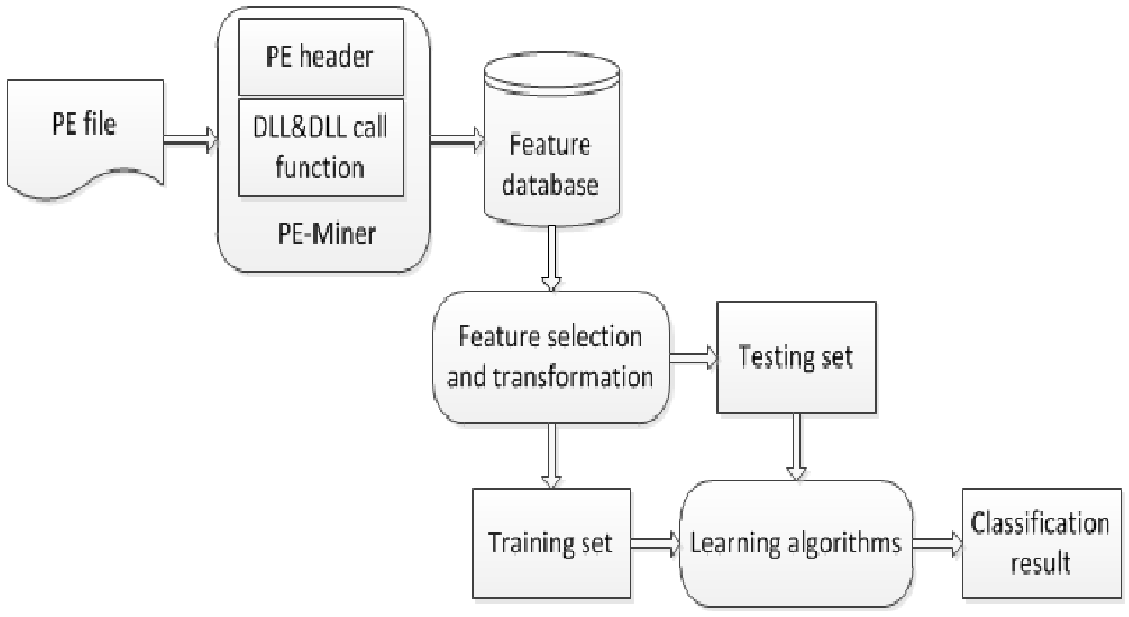
What can a Malware do?

i. Malware can severely **disrupt the operations of a device** and cause **data loss.**
ii. Gain **unauthorized access to sensitive data** and modify, block, or delete the data.
iii. Access **sensitive information** and can grant remote access to predators.
iv. **Steal financial or personal information** from the system.
v. **Slow down** your system noticeably.

As a result, it is important to deploy technologies that continually monitor and detect malware that has entered your system to protect from losses or damage to the system.

✓ Malware(Intrusion) Detection System :

Malware Detection System is a subpart of Intrusion Detection System which is used to **detect and determine the presence of malware on a system** or to distinguish whether a specific program(or dataset or set of files) is malicious or safe. It follows the process of **scanning the computer and files to detect malware.**



➢ TOOLS USED :

- **Google Colaboratory -**

It is a framework and development tool which combines code, output and descriptive text into one document. It does not require a setup and also supports many popular machine learning libraries which can be easily loaded in our notebook.

- **Python libraries –**

1. **Pandas**
   It stands for Panel Data and is the core **Python library for data analysis** and data manipulation. It consists of single and multidimentional data structures for data manipulation.

2. **PeFile module**
   It is a popular Python module to **parse/read and work with Portable Executable Files(PE).**This module provides easy access to the structure of a portable executable.

3. **Scikit-learn (Sklearn)**
   Python module used for **statistical modeling including classification, regression, clustering** and also provides many unsupervised and supervised learning algorithms.
   - ✓ **sklearn.ensemble module**- which includes two averaging algorithms based on randomized decision trees: the **RandomForest algorithm and the Extra-Trees method.**
   - ✓ The **sklearn.feature_selection** module- which implements feature selection algorithms. **SelectFromModel method**- for selecting features based on importance weights.
   - ✓ **Sklearn.model_selection** module – for selecting the appropriate model for our dataset. **train_test_split** module to split arrays or matrices into random train and test subsets.**cross_Validate** module to evaluate metrics by cross validation.
   - ✓ The **sklearn.metrics** module- to implement **several loss, score, and utility functions** to measure classification performance such as accuracy or efficiency.

4. **NumPy**
   It **stands for Numerical Python**, and is a library consisting of **multidimensional array objects** , used for numerical and scientific computing. Using NumPy, **mathematical and logical operations** on arrays can be performed.

➢ METHODOLOGY FOLLOWED :

1.Importing Libraries and pefile

Since we are going to work on pefile headers, we will be installing pefile.

Next , we will be importing Pandas library to be used later on.

Then, we are also going to import the drive module to access the drive contents to access the dataset from there.

2. Load the dataset

Now, we are going to load the dataset and read it using the read_csv function.

We are using a dataset which contains- 41323 legit samples and 96724 malware samples.

3.Preprocessing the data

We will remove or drop out the last column of our dataset (using the drop() function) which is “legitimate”, since we need the model to predict whether a file is legitimate or not.

To be sure, we can display and check whether the “legitimate” column is removed from the dataset.

We are using ensemble learning which can be defined as a method used to enhance the performance of Machine Learning model by combining several rules together.

Now, we will be importing sklearn library as well as some of its modules which we need. These are-

- ✓ **sklearn**.**ensemble** module which includes two averaging algorithms based on randomized decision trees: the **RandomForest algorithm and the Extra-Trees method,** both of which we will be needing to train our dataset.

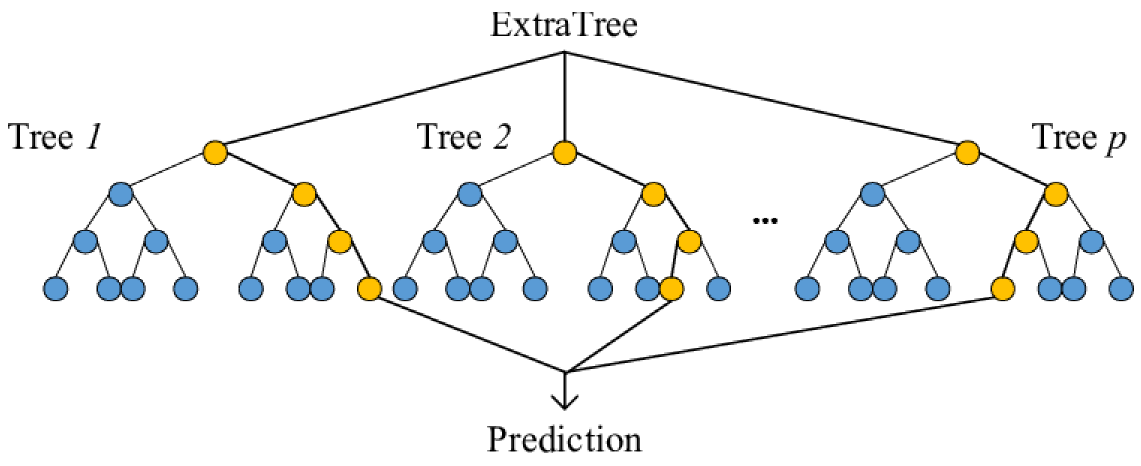
We will also be needing **feature extraction and model selection** modules to import the **train_test_split** model.

Model_selection is a method for setting a **blueprint** to analyze data and then using it to measure new data. Selecting a proper model allows you to generate **accurate results** when making a prediction.

4. Fitting the models and training the dataset

We shall be using two algorithms on our dataset –

I. **ExtraTreesClassifier** – ExtraTrees fits a number **of randomized decision trees on various sub-samples of the dataset and uses averaging to improve the predictive accuracy** and control over-fitting, to optimise our dataset, which means that this will help us to reduce the number of columns by **choosing some of the most important columns**(ie features) to be used for further training of model, so that the efficiency and accuracy of model is increased.



After fitting this model on our dataset the output was-

(138047, 54) (138047, 13)

This shows that the columns or features got reduced from 54 to 13.

Now, we shall also be importing the **NumPy library.**

We will be using the **feature_importances_ technique** to assign scores to the most important features of our dataset.

- ✓ **<u>Feature Importance</u>**- refers to techniques that **calculates a score for all the input features for a given model** and the scores simply represent the **importance of each feature**. A higher score means that the specific feature will have a larger effect or influence on the model that is being used to predict a certain variable.

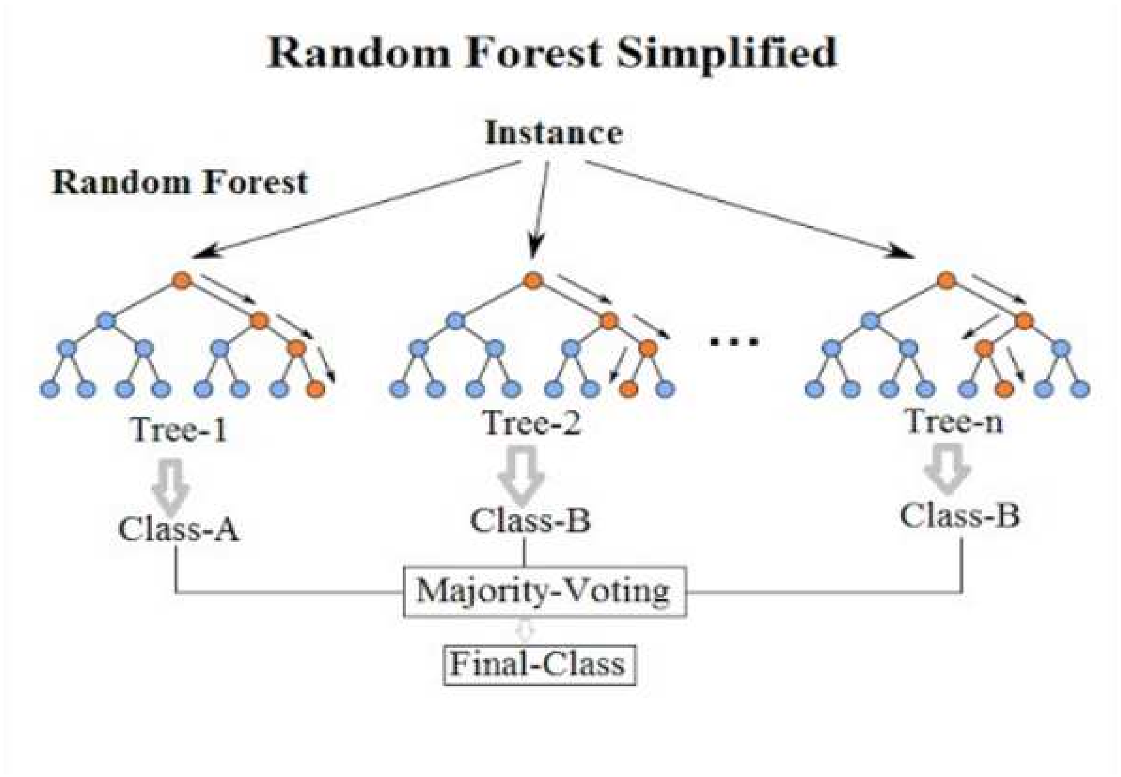
Applying this and printing the most important features, we get the output as -

```
1 DllCharacteristics 0.1954565726703676
2 Machine 0.11234546056273001
3 Characteristics 0.08794687037710715
4 VersionInformationSize 0.06350051762547866
5 Subsystem 0.06131216491810897
6 MajorSubsystemVersion 0.054384894768517114
7 ImageBase 0.052874652882128365
8 SectionsMaxEntropy 0.04352956871800817
9 ResourcesMinEntropy 0.040083198742687676
10 ResourcesMaxEntropy 0.039262705860991384
11 SizeOfOptionalHeader 0.026602287469025328
12 SectionsMinEntropy 0.020049351744441633
13 ResourcesMinSize 0.020034344565688
```

This shows that the most important feature or column of our dataset is DllCharacteristics , then Machine, and so on.

II. **<u>RandomForestClassifier</u>** – This is an **ensemble learning method for classification, regression and other tasks**, in which we grow multiple forest trees such that each tree comprises of the square root of the total number of features that are present.

For classification tasks, the output of the random forest is the **class selected by most trees.** For regression tasks, the **mean or average prediction** of the individual trees is returned.



We shall be importing the **RandomForestClassifier from sklearn.ensemble** and then apply the random Forest classifier on our data set using the **train_test_split model** and train the data on Legit and malware training set.

- ✓ **train_test_split** – It is a function in Sklearn model selection for **splitting data arrays into** two subsets**: for training data and for testing data.** With this function, you don't need to divide the dataset manually.

By default, Sklearn train_test_split will make random partitions for the two subsets.

5. Checking the accuracy of our model

On using the score() function we calculated the accuracy or the score of this algorithm to be 99.44585295182905.

We also calculated the the confusion Matrix on our data set, which came out to be-

```
array([[19355,    86],
[   73,  8096]])
```

- ✓ **A confusion matrix**- is a tabular representation of the number of **correct and incorrect predictions** made by a model/classifier. It can be used to evaluate the **performance of a classification model** through the calculation of accuracy, precision, recall, etc.

Using confusion matrix, we can find the number of false negatives and false positives.

The **False Negatives** means the number of times that the model predicted a negative ie non legitimate/malicious, and the actual was a positive ie safe.

The **False Positives** means the number of times that the model predicted a positive ie safe but the actual was a negative ie malicious.

And finally the false positives and false negatives on our dataset are as –

```
False positives :  0.4423640759220204
False negatives :  0.8936222303831558
```

➢ SUMMARY AND CONCLUSIONS-

In this project, I tried creating a **Malware(Intrusion) Detection System** on Machine Learning, using the **Extra Trees and Random Forest Classifier.** Here, the dataset was read first, some preprocessing took place, models were fitted and trained on to the dataset and finally we estimated the score and accuracy of our model in predicting whether the set of files were malicious or not. Also, I achieved the **accuracy 99.446** , which means the model did a good prediction and worked well.

➢ APPLICATIONS OF MALWARE DETECTION SYSTEM-

Malware is one of the most serious security threats and in the digital era, it is spreading more due to vulnerabilities in the system or carelessness of users. In order **to protect a computer from damage or remove malware** from a compromised computer system, it is essential to **accurately detect malware.** But determining whether a given program contains malware is generally undecidable and complex at times.

Thus, malware detection system is very important nowadays and finds wide application in **commercial anti-virus products installed on end-user clients.**

They are used in different sectors **of information technology (IT) systems and individual computers from malicious software, or malware.**

➢ REFERENCES :

https://www.researchgate.net/figure/Architecture-of-the-Malware-Detection-System_fig1_255787076

https://towardsdatascience.com/5-feature-selection-method-from-scikit-learn-you-should-know-ed4d116e4172

https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html

https://www.educative.io/edpresso/how-to-create-a-confusion-matrix-in-python-using-scikit-learn

https://www.bitdegree.org/learn/train-test-split

https://en.wikipedia.org/wiki/Random_forest

https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.ExtraTreesClassifier.html

https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

https://www.analyticsvidhya.com/blog/2017/09/common-machine-learning-algorithms/