Title : Windows based malware prediction system using deep learning techniques

Problem Statement:

The project aims at developing a model which accurately predicts the probability that an operating system will be hit by a malware. Its primary goal is to detect malware among many operating systems and building a model which can accurately do the task at hand.

Objectives:

* To understand the working ,of 3 differently abled models that tries to predict which of those operating systems will be hit with any type of malware.
* Predict the vulnerability of the system and the probability that the system will be affected soon.
* An Interface for users to see their system vulnerability in a friendly manner
* Analyze and compare the models and their shortcomings and advantages over the other model

System Architecture:

* There are 5 major components - The database, model development environment, EC2 Instance, EC2 Snapshot and Elastic Internet Gateway.
* The database contains both test and training sets along with over 89,21,483 operating system values.
* The model development environment which does preprocessing followed model training, visualization and accuracy testing and choosing accurate model.
* EC2 Instance houses the flask server, frontend designed with jquery and bootstrap, training model h5 file.
* EC2 snapshot is an incremental backup of the EC2 volume which is managed by the AWS Console

Methodology:

Prior to training of the models chosen under this comparative study, the dataset was feature engineered in order to obtain such high levels of accuracy. The following steps were incorporated in feature engineering:

* Introduction of five new variables from the understanding of the malware and operating systems by using time split validation:

AppVersion2 indicates whether your Windows Defender is up to date.

Lag1 is the difference between AvSigVersion\_Date and Census\_OSVersion\_Date. S

Lag5 is the difference between AvSigVersion\_Date and July 26, 2018.

driveA is the ratio of harddrive partition used for the operating system with the total hard drive.

driveB is the difference between harddrive partition used for the operating system and total hard drive.

* Variables were frequency encoded. There are a lot of time dependent variables that exist like (EngineVersion, AvSigVersion, AppVersion, Census\_OSVersion, Census\_OSBuildRevision) . And the test set would have variables from versions from beyond the time, the train and test variables are frequency encoded separately.
* Among all our category variables, there are a combined 211,562 values. It is tedious to one-hot-encode all. then for each value, we will test the following hypotheses

H0:Prob(HasDetections=1 given value is present)=0.5

HA:Prob(HasDetections=1 given value is present)≠0.5

The test statistic z-value equals p̂ , the observed HasDetections rate given value is present, minus 0.5 divided by the standard deviation of p̂ . The Central Limit Theorem tells us

z-value=p̂ −0.5SD(p̂ )

If the absolute value of z is greater than 2.0, we are 95% confident that Prob(HasDetections=1 given value is present) is not equal 0.5 and we will include a boolean for this value in our model.

The three models used are:

LightGBM- To understand the working of boosting slow learner algorithm

Recurrent Neural Network - To understand the working of the most basic fast learning neural network

XdeepFM-

Result and conclusion:

* Feature engineered models gave more than 70% accuracy while models without feature engineering were only able to predict with 30-45% accuracy

|  | LGBM | RECURRENT NEURAL NET | XDeepFM |
| --- | --- | --- | --- |
| Training accuracy | 0.50719 | 0.6769 (epoch 15) | 0.5897 |
| Validation Accuracy | 0.520162 | 0.6068 | 0.5916 |

* The hassle free GUI was also successfully built.
* Catching a virus before it hits a system will improve efficiency and simplify code base of various anti virus deployments
* Future work could comprise of combining the works of detecting malware from the system codes written by understanding their semantics to help predict.
* We could also classify the malware being detected along with predictions.