***RA181102801099***

***Atharv shah***

*Microsoft Malware Prediciton*

*liTERATURE SURVEY*

1. <https://arxiv.org/pdf/1802.10135.pdf>

It’s a research paper of Winner of Microsoft Malware Classification Challenge. Here he discusses about the all the major process involved in the attaining 1st position in the completion. Dataset is almost a half terabyte. It consists of 9 types of Malware Ramnit, Lollipop, Kehilos\_ver3, Vundo, Simbda, Tracur, Kehilos\_ver1, Obfuscater.ACY, Gatak. Raw data contains the hexadecimal representation of the file’s binary content, without the header. The dataset also includes a metadata manifest, which is a log containing various metadata information extracted from the binary, such as function calls, strings, etc. This was generated using the IDA disassembler tool. The original question posed to participants was to classify malware to one of the 9 classes. The dataset can be downloaded from the competition website.

In this paper, they have provided a short description of the characteristics of the Microsoft Malware Classification Challenge dataset. This dataset is becoming a standard dataset with more than 50 papers citing it. We enumerated these references as much as possible and compared their main contributions with respect to the dataset. The comparison helps the understanding of what the existing contributions are, and what the potential research directions can be. The authors aim to keep the reference table updated. We encourage the community to cite this paper when using the dataset, and update us about such work soit can be added to this paper.

1. .<https://arxiv.org/pdf/1603.02754.pdf>

This paper discusses very famous and important Ensemble Machine learning algorithm which is XGBoost. In this paper, we describe a scalable end-to-end tree boosting system called XGBoost, which is used widely by data scientists to achieve state-of-the-art results on many machine learning challenges. We propose a novel sparsity-aware algorithm for sparse data and weighted quantile sketch for approximate tree learning. More importantly, we provide insights on cache access patterns, data compression and sharing to build a scalable tree boosting system. By combining these insights, XGBoost scales beyond billions of examples using far fewer resources than existing system. In this paper they have described a scalable tree boosting system which is widely used by data scientists and provides state-of-the-art results on many problems. We proposed a novel sparsity aware algorithm for handling sparse data and a theoretically justified weighted quantile sketch for approximate learning. Our experience shows that cache access patterns, data com-press ion and sharing are essential elements for building scalable end-to-end system for tree boosting. These lessons can be applied to other machine learning systems as well. By combining these insights, XGBoost is able to solve real-world scale problems using a minimal amount of resources.

1. <https://www.zdnet.com/article/what-is-malware-everything-you-need-to-know-about-viruses-trojans-and-malicious-software/>

It’s a task of Malware Classification, so for that first we need understand the problem statement very well. So this blogs in detail discusses everything about the malware like history of malware, Morris worm etc.

Malware includes Computer virus, Trojan Malware, Spyware, Ransomware, wiper Malware, Computer worm, adware, botnet, cryptocurrency miner malware, lifeless malware, Internet of things Malware.

Some of the most basic cybersecurity practices can go a long way to protecting systems -- and their users -- from falling victim to malware. Simply ensuring software is patched and up to date, and all operating system updates are applied as quickly as possible after they're released, will help protect users from falling victim to attacks using known exploits.

Time and again, delays in patching have led to organizations falling victims to cyberattacks, which could've been prevented if patches had been applied as soon as they were released.

1. <https://www.analyticsvidhya.com/blog/2018/03/introduction->

k-neighbors-algorithm-clustering/

This article in details discusses about the classic machine learning algorithm which is K-nearest Neighbor (KNN)

K-nearest Neighbor is a Supervised machine learning algorithms. It can be used for both Classification and regression based tasks,

However, it is more widely used in classification problems in the industry. To evaluate any technique, we generally look at 3 important aspects Ease to interpret output, Calculation time, Predictive Power.

KNN works by finding the distances between a query and all the examples in the data, selecting the specified number examples (K) closest to the query, then votes for the most frequent label (in the case of classification) or averages the labels (in the case of regression).

We can find K by Hyper-parameter tuning, As the values of K increases model start to Over-fit and vice versa

In the case of classification and regression, we saw that choosing the right K for our data is done by trying several Ks and picking the one that works best.

1. <https://towardsdatascience.com/an-introduction-to-t-sne-with-python-example-5a3a293108d1>

It’s an article about the very famous dimensionality reduction technique t-SNE. After all the data processing, feature engineering we need to visualize it on the plane, But technically it’s not possible to visualize more the 3 dimensional, So we reduce all dimensions/total features into a 2D space, using t-SNE.

-Distributed Stochastic Neighbor Embedding (t-SNE) is an non-linear technique primarily used for data exploration and visualizing high-dimensional data. In simpler terms, t-SNE gives you a feel or intuition of how the data is arranged in a high-dimensional space. It was developed by Laurens van der Maatens and Geoffrey Hinton in 2008.

The t-SNE algorithm calculates a similarity measure between pairs of instances in the high dimensional space and in the low dimensional space. It then tries to optimize these two similarity measures using a cost function.

t-SNE is used in different domain such as climate research, computer security, bioinformatics, cancer research, etc. t-SNE could be used on high-dimensional data and then the output of those dimensions then become inputs to some other classification model. t-SNE could be used to investigate, learn, or evaluate segmentation.

1. <https://machinelearningmastery.com/gentle-introduction-bag-words-model/>

This blog discusses about one of the Text-Processing Technique.

We are given with the bytes file which is a Hex Code file, which can be considered as a text and for processing we are using here Bag of Words.

Bag-of-words model, or Bow for short, is a way of extracting features from text for use in modeling, such as with machine learning algorithms.

The approach is very simple and flexible, and can be used in a myriad of ways for extracting features from documents.

A bag-of-words is a representation of text that describes the occurrence of words within a document. It involves two things vocabulary of known words, to measure of the presence of known words.

To represent a word we can represent it 1,2,3,…….. N sequence of words.

So we are going to use the Bi-Gram BOW i.e. Considering two sequence of words as one entity and processing on the same method on all entity.

1. <https://medium.com/opex-analytics/opex-101-random-forest-6a8e4bd4626c>

This Blog discusses about the another Ensemble Machine learning algorithm which is Random Forest.

The random forest is one of many supervised machine learning algorithms that can be used to predict an outcome, whether that outcome is a number (a prediction process we call regression) or represents membership in a group (known as classification). It’s a non-parametric model, and thus does not make any strong assumptions about the data distribution, or specify any parameters (like the slope of a linear regression line).

random forest also uses the ensemble learning method, meaning it combines many underlying models into one, using all the individual models’ predictions together. The models it combines together are called decision trees (hint: where the ‘forest’ name comes from). Consequently, to truly understand a random forest, we must first understand the component models that form its foundation.