Malware Analysis on Hardware

Akshay Bitra
Applied Information Technology
George Mason University
Fairfax, VA, USA
abitra@gmu.edu

Meena Rapaka
Applied Information Technology
George Mason University
Fairfax, VA, USA
mrapaka@gmu.edu

Siva Naga Lakshmi K
Applied Information Technology
George Mason University
Fairfax, VA, USA
skaramse@gmu.edu

Abstract - As the world is technologically advancing, the growth of malware has grown increasingly. Malware has become a significant topic of research these days with new innovations taking place. Traditional malware detection software systems have ended up being insufficient whereas behavioral based malware detection frameworks which use data and control flow graphs have turned out to be refining yet are inadequate as they are based on resources available and still inclined to threats. Hardware based malware detection systems has turned out to be efficient response to decrease the exploitability as they are less available to access for misuse. Data visualization helps us better understand which components are being affected by the malware and their trends. Utilizing supervised machine learning classifiers on Hardware Performance counters detects malware projects and the applications can be further characterized with high precision.

Keywords

Malware, Hardware Performance Counters, Principal Component Analysis, Classifiers, Sandbox, Polymorphic Malware

I. Introduction

As of late, the fast development of data innovation has emphatically upgraded PC frameworks both as far as computational and organizing capacities. In any case, these innovative discrete computing scenarios, other than empowering the improvement of proficient PC applications, have raised imperative issues about the security and wellbeing of the interconnected equipment frameworks.[13]: As a result of the fast improvement of data innovation lately, the computing ability of computers has progressed extraordinarily, which brings in a few malicious infections and considerable measures of dangers such as viruses and backdoor programs. In reality, threats like viruses and backdoor programs have made unique technological and innovative disasters globally. There are notable efforts in generating effective and treacherous attacks by contriving new and inventive techniques to attack a processing framework. [12]: Such kind of software implementing those techniques are termed as malware. Malware comprises of malicious and harmful software which are utilized to take control of the framework or leak information outside of a protected framework.

Development of malware has been a significant issue with new innovations taking place day to day. McAfee – a global computer security software company reports that there are 75 million malware cases in 2017 and quite a few add up by every minute. As the world is becoming more technology built, anti-malware systems are to be implemented in abundance to secure the systems from these threats.

Anti-malware softwares are developed to sense and fix the system by erasing destructive programs. Traditional anti-virus systems are based on signatures to detect the malware, the drawback of this system was that the attacker can betray the AV by generating programs which will help the signature to be benign software. [14]: AV programs are inclined to breaches when the security is compromised and misused. Traditional malware detection software systems have ended up being insufficient whereas behavioral based malware detection frameworks which use data and control flow graphs have turned out to be refining yet are inadequate as they are based on resources available and still inclined to threats. Hardware based malware detection systems has turned out to be efficient response to decrease the exploitability as they are less available to access for misuse. [15]: Utilizing supervised machine learning classifiers on Hardware Performance counters detects malware projects and the applications can be further characterized with high precision. Hardware based frameworks provide effective resource implementation,

sooner detection of threats. Even when they are prudent, they come with certain drawbacks which incur low false positives, design constraints in capability to monitor the HPC, lesser detection latency in analyzing hardware performance counters and running machine learning classifiers.

In this paper, Section IV explains about malware, different kinds of malware represented in the data and Hardware Performance Counters (HPC) are discussed. Data visualization is implemented using Tableau, python, radar graphs and surface plots, multiple plots are visualized, and their trends are observed among different attributes in section V. In further sections, Similarity measure between various Hardware Performance Counters is calculated and a correlation matrix is built which helps us determine the correlation factor. Principal Component Analysis approach is implemented to factor analysis and different classifiers are modelled to find the accuracy measure.

II. Related Work

Many findings are made in relevance to the malware and their behavior on the Hardware Performance Counters. In [1] their paper has derived outputs for the Hardware Performance Counters based on the number of Counters that allows changes to be made on the Performance. They have figured out a way in which the standard machine classifiers are needed to run multiple times on the system to achieve the desired system. They have proposed a system in which their own kind of HPCs differs from the Standard ones previously built in the system. Ensemble Learning techniques in their model enable the counters to be in less number yet enhance the results, what exactly is being done here is the changes made to the Hardware of the performance counters rather than the software as hardware yields more accurate and faster results.

They [1]: have used 2-4 HPCs instead of the 8-16 HPCs and have boosted their Hardware systems to get better results and provide more accurate results. It also has shown that there are very low overheads and the detection of the malware is faster. They have proven another fact alongside their performance is that the increase in the number of HPCs does not result in the performance upgrade in the detection system. Two techniques stated in this paper [1] are the AdaBoost and Bootstrap aggregation that enhance the performance. The results got shows an enhancement in the Performance of detection by 17%.

The work in [3]: implemented for Android Malware, applied KNN classifier and Artificial Neural Network but have not provided results about Hardware Overheads. The work in [4, 5]: tells that malwares at runtime are undetected due to High Latency. The work in [6]: has suggested changes in Microprocessors Pipeline for Malware detection, but changes made directly to the HPCs can result in distorted results and irreversible changes.

Another Paper [2]: tells us about the HPCs that can be used to get access to various information. They have provided us with the Analysis of the software at Runtime. They can only be accessed from the Restricted Mode from the Operating System. HPCs are noisy by nature, the Cache events for HPCs. HPCs are Tested on numerous samples, until the right key is found. If there are cache hit more precise, samples to be worked on decreases. They have concluded that HPCs were introduced as various side-channels for multiple CPUs also explored its power for cache-misses and tells HPCs have great potential alongside tested on a time-based cache attack. Some of the challenges that were faced by the authors of the same are Loss of events in Large Event Counts, which happens rarely though. This Paper deals with small event counts, which are naturally noisy. Count is different for various Hardware Platforms. A malicious attacker can hack into, and have some preloaded libraries run before the actual program running.

III. Dataset

Dataset was already constructed, and it has 11848 records and it has 17 different columns out of which 16 are hardware performance counters, one column is class which describes about the 6 different malwares in the dataset. How malware have been affected the counters and the counter duration/time consumed is recorded in the dataset. The hardware performance counters belong to different hardware parts in the CPU like cache loads, bus. Cycle or memory. They are different from one another. Dataset is represented visually in Fig. 1 which shows that Benign is the highly affected malware.

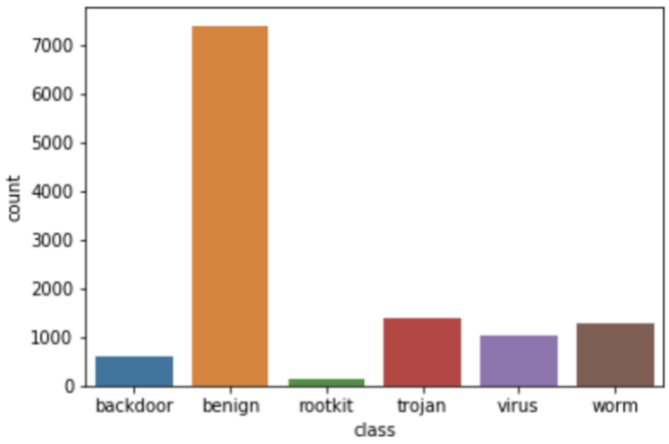

Figure 1: Class Vs Count

IV. Exploratory Data Analysis

A. Malware Analysis

Malware is a general term used to describe various kinds of malicious software. It is defined as a software which infects individual computers or a network and causes damage to them by exploiting target system vulnerabilities. A malware invasion can lead to disastrous affects, data theft and breaching the network frameworks. It is divided into different categories, in our dataset we have considered 6 different types of malwares. The most dangerous malware affects right now is Virus, then followed by different types of worm malware and trojan malware. Our dataset shows that benign malware is highly affected throughout which is followed by other different kinds of malware. There are six different malwares used for analysis and classification. [13]

The following malwares are defined below:

Virus: Virus is something similar to the biology term, which we don't need. It is a small set of instructions or a script which affects the systems health or the computer. It will spread from one computer to other computer and leaves malicious files as it spreads. [22] The virus can consume whole lot of memory in the system where it reads, writes and deletes the files of the consumer. Viruses are always present on the system and comes with executable files, it gets active once you run or open the host files of it. [25] It can spread easily from one system to other system and it will affect the system components using the disk, email attachments and the files. For Virus will need a host file to affect, without the host file it is difficult. [21]

Worm: Worm is similar to the virus malware where it replicates the malware files and will cause the same damage as virus. Contrast to virus, worm will be the standalone software and doesn't require host program or no human intervention to spread. Worm is one of the dangerous malwares compared to all the types and it will enter through the weakness of the system and it takes advantage of the files exchange feature, whereas superior worms will control the encryption, wipers to harm the system.

Trojans: Trojans malware looks as a legitimate software and it tricks the user to load and to execute the files, the malware name has been derived from the wooden horse that Greeks used to destroy the troy. Once the trojan malware is activated it can achieve many numbers of attacks on the host system, from popping up windows multiple times to damaging the host by deleting files, stealing the data or activating the malware. These are also known to provide access to the backdoor malware and the trojans do not replicate or reproduce the files by infecting other files. The trojans files will be activated using the human intervention or opening an attachment or downloading the attachment.

Rootkit: It is a program or a collection of software tools which will give access and control of the computer to the threat actor. There have been many uses for this type of software to give remote end user support, often rootkits open a backdoor for the victim systems and introduces malicious software. Rootkits are frequently attempting to prevent detect malicious software by antivirus software. These can be installed in many numbers of ways, including the email attachments or social tactics and it tricks the users by giving permission and to install on the system and cybercrimes occur by having remote access of the system. [24]

Backdoor: It is a method of avoiding the authentication, providing a secure remote access to the computer and access to the plaintext while the attempts to be undetected. It will be in the form of installing program or can be notification to the existing program. [23]

Benign: It is a prank virus which will not cause damage to the system. This kind of virus will randomly display the messages on the screen or cause the computer to make a clicking sound. Most of the viruses are benign type.

B. Hardware Performance counters:

Hardware Performance Counters also referred to as Hardware Counters is a combination of special purpose registers constructed into the day to day microprocessors that keeps a track of the number of various activities with reference to hardware that go through the computer systems. Sophisticated users usually count on the counters for performing tuning. Implementations of the HPC's in a particular processor varies from how the developer wants to deal with the measuring of various events. [2]:

The HPC's outperforms the Software based techniques based on the overheads that the HPC's provide. The Hardware counters overcomes Software techniques by another benefit of not needing to make any changes to the source code for the performance measurements. There are difficulties that do arise upon the processing as there are limited registers that are required to store the counters, due to which the process needs to be run multiple times to achieve the desired performance metrics. Not only the performance is enhanced by different methods that are implemented on the HPC's but there is a cost reduction that plays a major role, that is because developers trying to increase the number of Hardware counters will not resolve the problem, instead boosted methods applied to the

Hardware counters are needed for the successful, faster and reliable processing.

V. Data Visualization

A. *Horizontal Bar Chat:*

The graph depicts relation between the class and sum of the hardware counters. It shows how malware has been affected the counters in different manner. On the X-axis we can see the different counters are considered like instructions, Branch-loads, Branch-Misses and Bus cycles and the scale is varying for all the variables from Millions to Billions as the values are using aggregation. On the Y-axis the malware class being considered. From visualizing it we can consider benign is the malware which is being affected more in every counter. Second malware which is affected more was trojan in all the counters except the Branch-Misses. But Brach-Misses second affected malware was Virus. And the least affected malware was backdoor in all the cases leaving the Branch-Misses where it was trojan is affecting less to the counter.

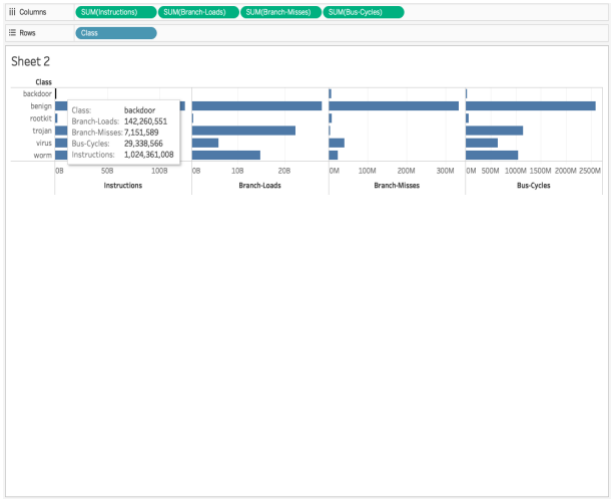

Figure 2: Class Vs Branch Loads, Branch Misses, Bus Cycles, Instructions

B. *Bar Graph:*

This graph depicts relation between the Class versus branch instructions and bus cycles but using distinct function and from the bar graph we can observe which counter is being affected by which malware. On the X-axis it considers the different malwares in branch instructions and bus cycles and on the Y-axis considers count of the branch instructions and bus cycles and the scale is not varied. From visualizing it we can say that count of benign malware is most affected one in both the counters, Trojan is the second most affected malware in the bus cycles and respectively Virus is the second most affected malware in branch instructions. The least affected malware was the rootkit in both the counters. The orange depicts the bus cycles and the blue depicts the branch instructions.

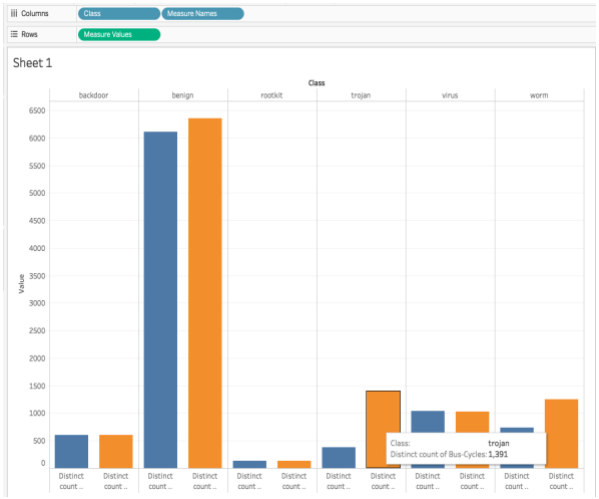

Figure 3: Class Vs Bus Cycles, Branch Instructions

C. *Bubble chart:*

This chart is similar to the bubble visualization, but the bubbles are packed tightly whereas in bubble visualization the bubbles are spread over a grid. The above visualization depicts the large amount of bubble area in the small space. This chart doesn't contain any axis, but here we are considering the average of instruction counter and the Malware classes. From the chart we can say that trojan is affecting more in the instructions counter and the second highest is the worm malware and the least malware affected was backdoor.

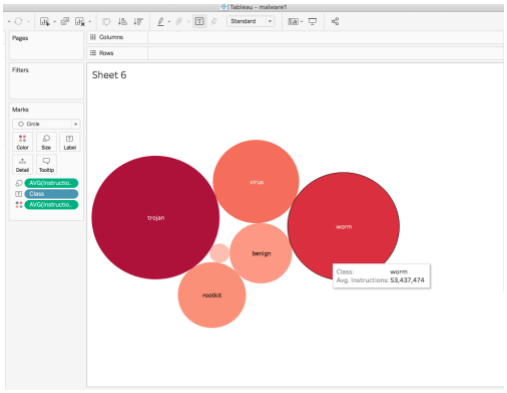

Figure 4: Instructions Vs Malware

D. *Scatter Plot*

The scatter plot is visualized to compare the LLC loads and iTLB load Misses with respective to the Malware class. From graph we can depict the scale is varied for both counters, and Median of the counters so the rootkit is affected in both the malware and the second highest malware affected was different for both the counters trojan for the LLC loads and worm for the iTLB load misses. The least affected malware was virus in both the cases.

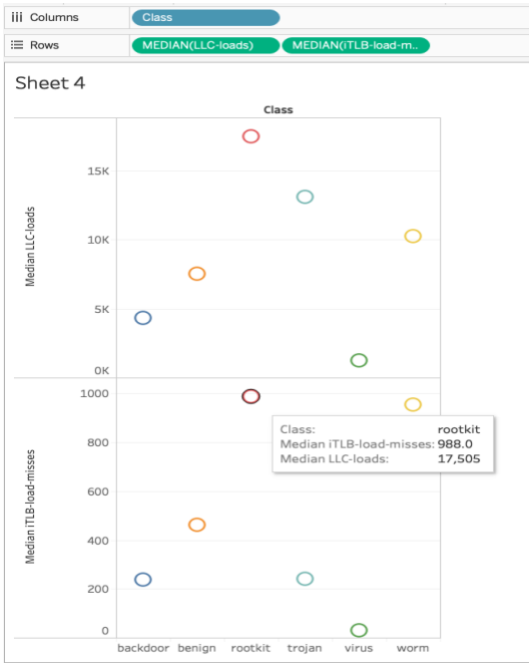

Figure 5: LLC-Loads Vs TLB-Load Misses

E. *Surface Plot*

The above figure depicts a three-dimensional data. It shows how different Hardware's in a system are affected by the various malwares in a system. Surface plots demonstrate a utilitarian connection between an assigned ward variable (Y), and two free factors (X and Z). The figure explains about the Backdoor Malware that disturbs the various Hardware's, instructions hardware is the most affected by the backdoor as we can see that the peaks are highest in case of the instructions.

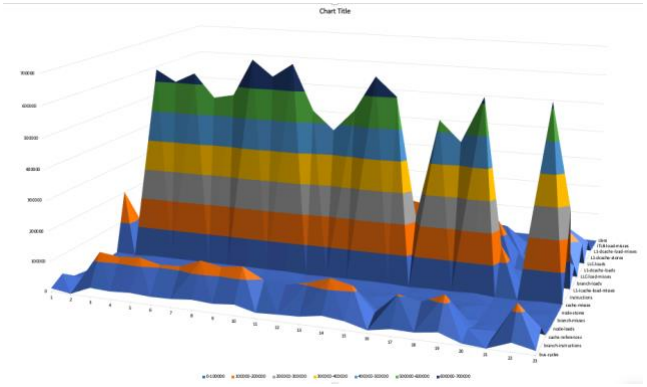

Figure 6: HPC Counters Vs Class

F. *Radar Plot*

This plot is a graphical representation of more than two variables, usual cases considers many variables for the visualization. These start from a same point and extends to a particular length based on their value. The above is the Benign Malware's radar plot that shows the length that extends from the center point to their respective value. The different colors depict the different types of the malware's that exist in the dataset provided. Brown color being the most dominant is the instruction hardware performance counter being affected by the benign malware. There are several other colors that are in less amount and visibility due to the less effect on the hardware performance counters by the malwares.

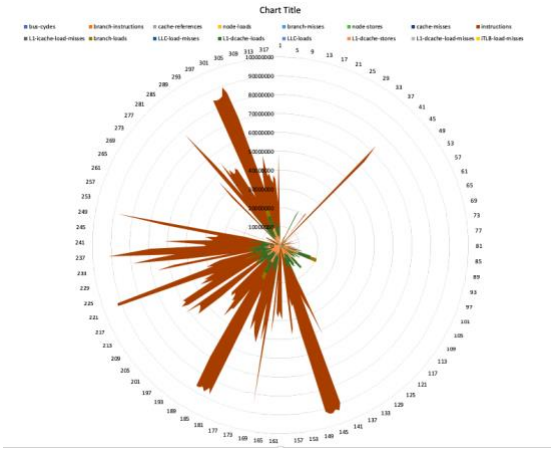

Figure 7: HPC Counters Vs Class

G. *Box plot*

Box plots are a measure of how well the data is distributed in the dataset. Using Python, Boxplot can be drawn calling Series.box.plot() and DataFrame.box.plot() or DataFrame.boxplot() to visualize the distribution of values

within each column. Here in this graph we compare the ITLB-load-misses with the class of malware, we can depict that rootkit has a greater number of ITLB-load misses when compared with other malware classes, followed by benign and backdoor being the least affected.

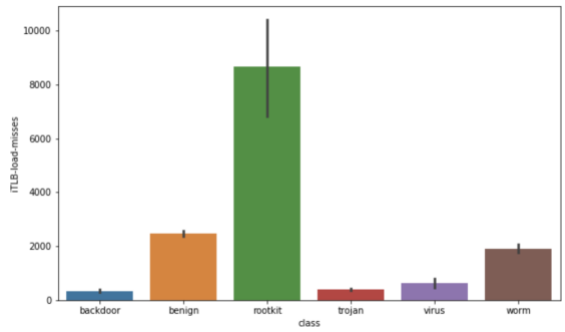

Figure 8: Class Vs ITLB-Load-Misses

H. *Box and Whisker plot*

Box and whisker plot are data exploratory graphs which visualizes the distribution of the dataset. A Box and whisker plot is also called as Boxplot and displays the data as a five-number summary. In this plot, we have first quartile, median, third quartile, maximum and minimum which is represented as shown below in Fig. 8. [9]: Here in Figure 9, the distribution between Node-Loads with the Class is depicted, Benign falls in the upper quartile and rootkit in the lower quartile. We can observe that worm has a count of 1263 Node-Loads and is the median.

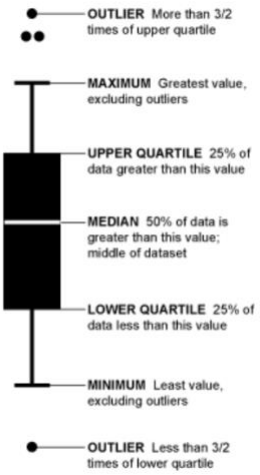

Figure 9: Box and Whisker plot [10]:

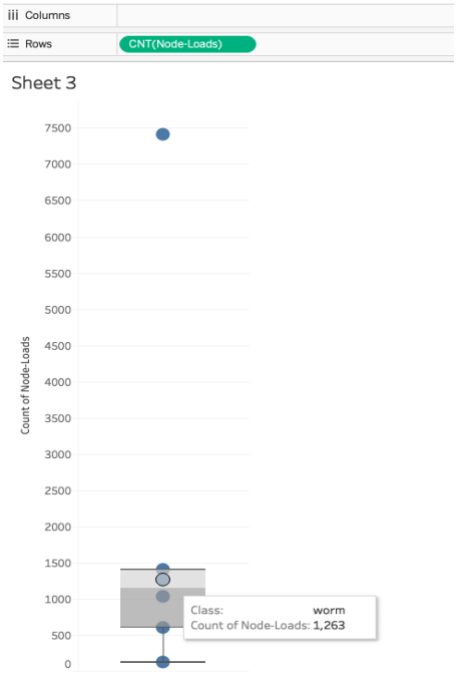

Figure 10: Class Vs Count of Node-Loads

VI. Correlation:

After exploring the dataset using different visualization techniques, we found that benign is being most affected, but to find out that when malware is affected which counter perform similar operation. To find the similarity between the counters how they are related to each other we perform correlation and will find which are positively correlated and negatively correlated.

Correlation is basically a statistical measure how two or more variables are fluctuating together. Positive correlation where two variables will see the parallelly effects and changes and negative correlation indicates one value is increased and other is decreased. The correlation matrix is visualized using heatmap. Heatmap is a way of graphical representation of the matrix values where the graph will be depicted using color and it has range value from positive to negative. Large values will be represented in the lighter shade and smaller values will be using the light darker shade in our graph. In the correlation matrix only either upper diagonal matrix or lower diagonal matrix to be considered, due to both the half will depict same information. only one part of the matrix considered to determine the correlation factor.

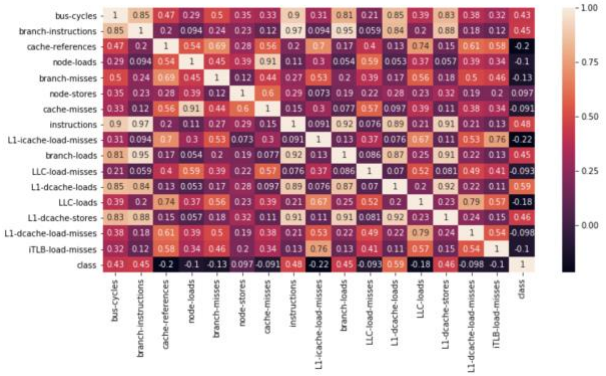

Figure 11: Correlation Matrix

From the map we can determine the positively correlated counters are branch-instructions and instructions being the top most correlated one and rest are branch-loads and branch instructions, branch-loads vs instructions. The negatively correlated counters are LLC Loads vs classes being the top one, class vs branch misses being the second one.

VII. Principal Component Analysis

In Statistics and Machine Learning, a decision is made based on a different number of aspects. We call these aspects as Features in Machine Learning which are some of the judging factors dependent on which the choice is made with respect to the classifier. If the features are in high quantity, the classifier can get more unpredictable which is complex. Likewise, correlation exists between these features and are sometimes redundant which is the basic ideology behind feature reduction. Reducing the dimension of the feature space which has great impact on machine learning classifier is defined as Feature reduction. To achieve this, two techniques – Feature Elimination and Feature Extraction are used. [8]:

Principal Component Analysis (PCA) is a Feature Extraction process which segregates the input variables [11]: in such a way that we can select best features in the dataset by discarding the least valuable features and decides the behavior of the application. We performed PCA using WEKA and python which provides us with ranked features and define the relationship between them.

On our dataset, using WEKA we first model the data using clustering techniques and perform PCA which helps us rank the attributes. Search method is chosen to be Ranker with attribute evaluator as PCA and is performed on the class of malwares. Now, the features are ranked accordingly based on their eigen vectors. Different variance and ranked features can be set. We have set multiple trails to check the performance of HPC, firstly we set the variance to be 0.85 and 0.93 and found the top 5 attributes to be bus cycles, L1.dcache loads, cache misses, node stores and LLC load misses with their respective eigen vectors.

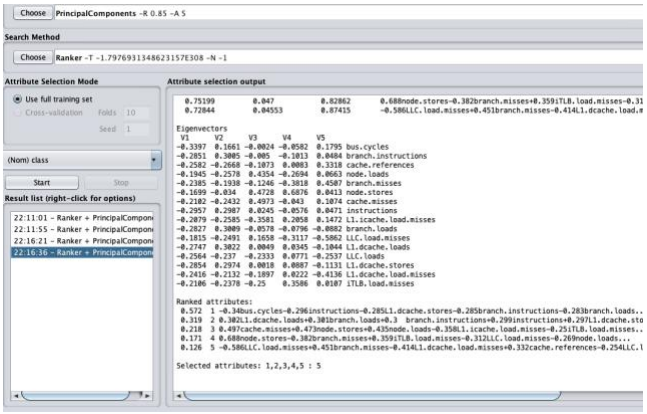

Figure 12: PCA with 0.85 variance

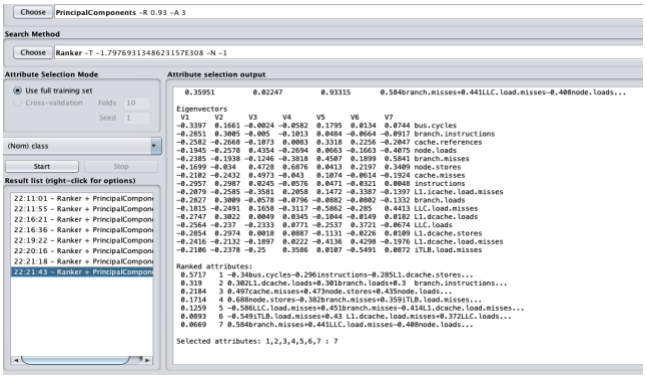

Figure 13: PCA with 0.93 variance

PCA for Data Visualization: This technique will help us to visualize the data, visualizing the 2D or 3D is not challenging, the data used in this more than 4 dimensional. The data will be reduced to two dimensions by standardizing the input values and fitting them into two components. After reducing the input variables to two principal components, they will not have any meaning assigned to them. The two dimensions are plotted in the graph using python, for all the malware classes and the highest malware was benign and the second one is backdoor for the two components.

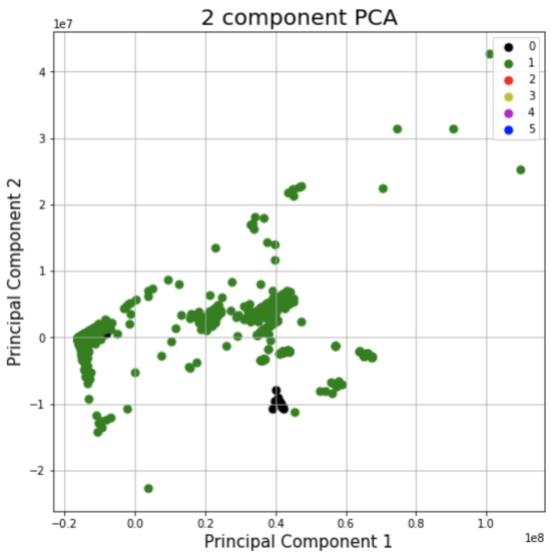

Figure 14: PCA for class (malware)

VIII. Machine Learning Classifiers:

After the PCA is performed the data will be classified using machine learning techniques how the data is trained to see how the hardware counters performance is evaluated. Here we are using the Random forest and decision tree classifier to evaluate the performance.

Random Forest: Random forests are an ensemble learning method for classification and regression. It operates by constructing a multitude of decision trees and outputs the mean prediction of individual trees. Before running the classifier, we are considering testing and training data as the ratio of 70% and 30% respectively. Once we get the training data classifier is used from the sklearn library random forest classifier is called and the accuracy gained was 91% using the whole dataset. [17]

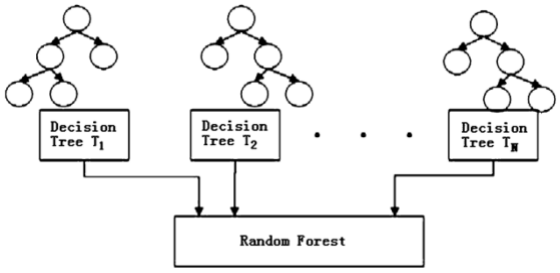

Figure 15: Random Forest Algorithm

Decision Tree: A decision tree is a graph that uses a branching method to illustrate every possible outcome of a decision. It considers the whole dataset and we are using the same ratio of training and testing data is considered. The accuracy gained was 88% on the whole dataset. [18]

Collinearity: it is an occurrence where a predictor variable in multiple regression model and can be linearly predicted from others with a significant degree of accuracy. [19]

To find collinearity between the variables from the dataset we use variance inflation factor and after finding the collinearity the similar variables which are predicting will be deleted and only one variable will be considered and then again classifier is going to run. By taking the features from the collinearity and again data is split into training and testing ration, the random forest and decision tree classifiers are performed on them. The accuracy gained after performing the classifier is 86.7% for the random forest and 86.2% for the decision tree. The accuracy has been decreased for the features selected. [20]

Variance inflation factor: it is ratio factor of the variance in a model with multiple term and divided by the variance of a model with a single term alone. It measures the collinearity in a regression analysis and provides an index it measures the variance how much estimated regression coefficient will be increased with collinearity. [16]

```
[[ 143   26    0    9    2    1]
 [   9 1253    0    1    1    0]
 [  11    2    0    3    1    1]
 [  46    6    0   57    2    0]
 [  32    8    0    1  217    1]
 [  17    4    0    7    3  296]]
Accuracy0.9101851851851852
```

Figure 16: Random Forest without Feature Reduction

```
[[ 114    9   33   23    2    0]
 [  36 1187   32    6    3    0]
 [   5    1    9    1    1    1]
 [  21    0   31   59    0    0]
 [  22    0   20    6  211    0]
 [  12    3   10    8    0  294]]
Accuracy0.8675925925925926
```

Figure 17: Random Forest with Feature Reduction

```
[[ 107   24    6   21   13   10]
 [  22 1203    2   10   17   10]
 [   5    3    1    4    3    2]
 [  22    6    2   63    8   10]
 [  20    5    1    4  227    2]
 [   8    5    2    5    4  303]]
Accuracy0.8814814814814815
```

Figure 18: Decision Tree without Feature reduction

```
[[ 101   16   11   36    6   11]
 [  44 1170   12   20   12    6]
 [   4    1    2    8    2    1]
 [  20    2    7   64    9    9]
 [  10    3    6   12  226    2]
 [   7    4    6    7    2  301]]
Accuracy0.8629629629629629
```

Figure 19: Decision Tree with Feature Reduction

IX. Future Work

Hundreds and thousands of Malwares are found out each day. It usually takes around 54 days for the Antivirus companies to detect them. The malwares are of massive size, it takes a lot of time to process all the data and to delete them. 15% of the data is not even taken care of at most times. In the work [7]: All these drawbacks must be dealt with various procedures. We should be considering wider datasets so that no malwares are missed in the cleaning.

With the implementation of the Machine Learning Techniques there will be certain results that are achieved, but these results are not entirely enough for the better understanding of the Dataset and the malwares associated with it. There [2] should be more indulgence and improvements in the implementations of artificial intelligence along-side the data mining techniques [1], this way it will be helpful for the data to be checked out entirely for the malwares that harms the hardware's. We know that the feature extraction [6] occurs on the data after they have been run on the Sandbox, what we would be willing to change is that feature extracts happens at the time of processing by the sandbox. Moreover, changes shall be made to the detection and cleaning of the polymorphic malwares so that they don't get slipped by Antivirus Software's.

X. Conclusion

We can conclude by saying that there are many malwares that arise because of the Hackers every minute. The antivirus companies take a lot of time and sometimes they are unable to examine the entire data and figure out the various malwares. It is suggested that the widespread consideration of the datasets and involvement in the Artificial intelligence along with machine learning techniques yields way better results as compared to the case in which only the machine learning concepts. We have shown how the Malwares effect the various hardware components and which all components are being affected the most in the entire dataset of the Hardware Performance Counters. Benign is the Malware that has the large wide spread in the entire dataset.

Acknowledgments

The authors would like to thank Ms. Setareh Rafatirad, who provided us with an opportunity and expertise to work on this project – Malware Analysis on Hardware, which helped us in doing a lot of research and analysis and we learnt new concepts.

Role of Team Members

Akshay Bitra, Meena Rapaka, Siva Naga Lakshmi K worked together as a team. In this project, we had to work on data analysis, data visualization and data exploration. We have equally divided every step and collaboratively worked towards achieving the results.

For the paper, Akshay wrote HPC, Future work, Related work, Surface plot, Radar Graphs and conclusion. Meena wrote regarding Abstract, Introduction, Box plot, Box and Whisker plot, PCA, Acknowledgment and composed the paper according to IEEE format. Lakshmi penned on Malware and its types, Dataset, Horizontal graph, Bar graph, Scatter plot, Bubble graph, Correlation, PCA and ML Classifiers.

References

[1] N. P. S. M. P. D. A. S. S. R. H. H. Hossein Sayadi, "Ensemble Learning for Effective Run-Time Hardware-Based Malware Detection: A Comprehensive Analysis and Classification," in *55th ACM/IEEE Annual Design Automation Conference*, San Francisco, 2018.

[2] A. G. Leif Uhsadel and I. Verbauwhede, "Exploiting Hardware Performance Counters," in *2008 5th Workshop on Fault Diagnosis and Tolerance in Cryptography*, Belgium , 2008.

[3] J. D. M. M. J. S. A. T. A. W. S. S. S. Stolfo, "On the Feasibility of Online Malware Detection with Performance Counters," in *ACM SIGARCH Computer Architecture News*, 2013.

[4] A. Garcia-Serrano, "Anomaly Detection for malware identification using Hardware Performance Counters".

[5] L. S. T. Stolfo, "Unsupervised anomaly-based malware detection using hardware features," in *International Workshop on Recent Advances in Intrusion Detection*, 2014.

[6] C. D. ,. I. G. ,. N. A.-G. ,. a. D. P. Meltem Ozsoy1, "Malware-Aware Processors: A Framework for Efficient Online Malware Detection," 2015.

[7] A. K. M. C. T. S. A. R. B. R. F. L. H. V. S. T. W. G. Y. A. D. J. J. D. T. Brad Miller, "Reviewer Integration and Performance Measurement for Malware Detection," 2015.

[8] "Wikipedia," [Online]. Available: https://en.wikipedia.org/wiki/Principal_component_analysis.

[9] "Khanacademy," [Online]. Available: https://www.khanacademy.org/math/statistics-probability/summarizing-quantitative-data/box-whisker-plots/a/box-plot-review.

[10] "FlowingData," 15 February 2015. [Online]. Available: https://flowingdata.com/2008/02/15/how-to-read-and-use-a-box-and-whisker-plot/.

[11] B. Matt, "TowardsDataScience," 18 April 2017. [Online]. Available: https://towardsdatascience.com/a-one-stop-shop-for-principal-component-analysis-5582fb7e0a9c.

[12] H.-D. Huang, T.-Y. Chuang, Y.-L. Tsai and C.-S. Lee, "Ontology-based intelligent system for malware behavioral analysis," in *International Conference on Fuzzy Systems*, Barcelona, Spain, 2010.

[13] O. Randive, "Analyzing Hardware Based Malware Detectors Using Machine Learning Techniques," 2018.

[14] S. K. L. a. A. O. M. a. W. Lu, "MalwareTextDB: A Database for Annotated Malware Articles," in *Association for Computational Linguistics(pp. 1557-1567)*, Vancouver, Canada, 2017.

[15] Y. A. Mohamed Nassar and Haidar Safa, "Modeling Malware as a Language".

[16] "Wikipedia," [Online]. Available: https://en.wikipedia.org/wiki/Variance_inflation_factor.

[17] "Wikipedia," [Online]. Available: https://en.wikipedia.org/wiki/Random_forest.

[18] "Wikipedia," [Online]. Available: https://en.wikipedia.org/wiki/Decision_tree.

[19] "Wikipedia," [Online]. Available: https://en.wikipedia.org/wiki/Multicollinearity.

[20] M. Rouse, "Whatls.TechTarget," [Online]. Available: https://whatis.techtarget.com/definition/correlation.

[21] "Cisco," [Online]. Available: https://www.cisco.com/c/en/us/about/security-center/virus-differences.html.

[22] "Techterms," 16 August 2011. [Online]. Available: https://techterms.com/definition/virus.

[23] "malware.wikia," [Online]. Available: http://malware.wikia.com/wiki/Backdoor.

[24] M. Rouse, "Whatls.techtarget," [Online]. Available: https://searchsecurity.techtarget.com/definition/rootkit.

[25] "PCmag," [Online]. Available: https://www.pcmag.com/encyclopedia/term/38542/benign-virus.

Survey Paper Question

In the paper *Ensemble Learning for Effective Run-Time Hardware-Based Malware Detection* has helped our paper in the comprehensive analysis and classification part and how different malwares are affected in the various platforms. It has made us consider the 16 HPCs and are not the 2-4 Boosted Versions of the HPCs.It helped us in such a way that, considering the non-boosted versions of the HPCs made the results more appropriate as it more appropriate for our research. It has given us how the consideration is proceeded in the case of the count of the HPCs and the effect of the Malwares on the system.

Another Paper, *Exploiting Hardware Performance Counters* has helped us in getting to know how the hardware performance counters can have various kinds of information in them that is really helpful for our research and the extraction of different data. The paper tells how the Analysis of the software is important as well for the extraction at runtime. It shows how they can be accessed from the restricted mode from the Operating System which has helped us know more about the security they possess. It tells why the HPCs are important, as they provide us with great results in the Performance. The *Figure 1* is in reference to the work we have done in case of the effect on the Hardware performance Counters by the Malwares.

Cache eviction method	RDTSC	TSC	Cycles	L1	L2
L2 evict on PIII	14,8	13,6	13,2	13,6	n/a
L1 evict on PIII	45,7	58,7	57,3	33,7	n/a
L1 evict on PIV	45,3	60,0	60,4	34,1	n/a
WBINVD	n/a	156,4	244,2	35,6	76,1
L2 evict on PIV	not possible	not possible	not possible	not possible	not possible

Figure 1: HPCs in a Real Attack.

Based on the paper - *MalwareTextDB: A Database for Annotated Malware*, I have learnt how models are constructed for data collection and analysis using NLP. The database created, is said to be the first of its kind which consists annotated malware reports. Framework for annotating malware using Malware Attribute Enumeration and Characterization (MAEC) vocabulary is initiated. The Bart Rapid Annotation tool was used to annotate the reports, their main aim was to map the words which describe malware actions and their behaviors. In out project, we have six different malware categories. Based on the aforementioned paper helped to gain more insights about the malware categories, their behaviors and how a malware database was built. Natural language techniques were implemented on the feature set using unigrams, bigrams, POS etc., where as we performed PCA to rank the features. Recall, precision and accuracy were considered to be important when creating the database as they evaluate the performance. Annotating the database was classified into different tasks where the words are annotated at different levels such as token labels, relational labels, attribute labels. Challenges faced by annotating them manually and using machine algorithms and their functionality is measured and a sentence is classified using above mentioned labels, their F1 scores, accuracy and precision are calculated.

The paper - *Modeling Malware as a Language*, talks about Feature extraction and how some approaches focus on static analysis i.e., the traditional approach while the rest focus on dynamin analysis. In this research paper, a step-wise analysis is discussed to classify a malware executable file using static and dynamic analysis approaches collaborated with machine learning algorithms. A language is built from the scratch, vocabulary is defined, malware-words and malware-documents are defined and document - distance like instance, are calculated using KNN and word2vec classifiers. In our project to calculate the accuracy and performance counter, we have used Random Forest and Decision Trees algorithms to calculate the accuracy. It was interesting to learn about how a language and an ontology is built as a part of malware analysis.

In the paper *Large-scale Malware Classification Using Random Projections and Neural Network* has helped our project in the model training and how to apply classifiers on the dataset. In the survey paper they have considered the random projections and different neural network layers has been used to get the better accuracy, but we have used the feature reduction technique- where it will reduce the features using collinearity. When it removes the parallel ones then the accuracy will be decreased but the precision and recall values will be changed. In our project we have used random forest classifier for our dataset that was best classifier, in future we would like to consider the neural network classifier and to have large amount of data as our dataset was small.

Another paper, it is a chapter about *Historical Overview* it talks about the history how malwares has come into existence and how the trojan malware was evolved and what techniques were used to remove them. Though this paper hasn't helped us in our project, but we learnt more about the hardware trojans concept in different time period. Like the research of hardware trojans and approaches they are following in current times. Detection of trojan malware with different techniques and how it used for the hardware components.