Classifications Strategies of Malicious Software

Tate McGeary
CECS 478 – Fall 2019

Michael Scheid
CECS 478 – Fall 2019

***Abstract* —** In recent years there has been a spur in the growth of real-time detection of malicious software using analysis of hardware performance counters of active program threads on processors. The concept proposed by C. Malone., et al. in his paper [1]. The analysis originally done using Eurequa which is an A.I. modeling software to create a linear relationship. C. Malone, et al. expanded the concept of detecting software and software types to utilizing A.I./M.L. for real-time detection. Currently, the field has expanded to utilize tree algorithms, neural networks, rules based, or any other type of A.I. or M.L..

I. INTRODUCTION

There are two ways to categorize software, binary and multi-class. Both instances will characterize what is and is not a benign software, however in the case of multi-class the types of malicious software can be categorized based off of a probability depending on the algorithms chosen. The granularity in multi-class lends itself to suitable to detect and resolve attacks however it requires a larger data set then binary classification to work reliably. On the other hand, binary classification allows the detections of non-benign software but lacks the ability to help determine what the attack or threat so that the issue can be resolved. But in the instance of binary classification, a smaller dataset can be used since the classification is benign versus malware. It should be noted however in all instances of using A.I./M.L. the larger the dataset used, the more accurate the classification. Additionally, both types of classification the number of hardware performance counters used, which provide hardware events, can change the accuracy rating.

II. FEATURE SELECTION

When determining the number of attributes to use, which are retrieved by hardware performance counters (HPCs), a feature reduction method must be utilized. Whether it is feature selection, feature extraction, or any other method some quantifiable dimensionality reduction must take place as there are more hardware events then HPCs. At this point it should be understood that when the number of events exceeds the number of available HPCs multiple sampling points will be made by the classification algorithms and in this case when the determination is real-time the extra time used to retrieve more data can compromise the security further as it is time that may be necessary to stop the threat. One such method of dimensionality reduction is feature selection using GainRatioAttribute Evaluation in WEKA.

$$GainRatio(Class, Attribute) = \frac{H(Class) - H(Class \mid Attribute)}{H(Attribute)}$$

Figure 1. GainRatioAttribute [3]

The use of the GainRatioAttibute feature selection provided the following data for binary and multi-class:

Attribute	Description
Attributes used for dimensions 2, 4, and 8	
L1-dcache-stores	Number of level 1 data cache store instructions that successfully wrote to the level 1 cache.
L1-dcache-loads	Number of level 1 data cache load instructions where the level 1 cache was accessed successfully.
Attributes used for data set dimensions 4 and 8	
branch-misses	Total number of branching instruction set paths not executed or taken by the CPU.
branch-loads	Number of instruction sets that can have unconditional or conditional branching paths to code execution that were successfully initiated.
Attributes used for data set dimensions 8	
iTLB-load-misses	Number of page walk requests due to an instruction that failed to load in the instruction translation lookaside buffer due to not being in the iTLB at the time of access.
instructions	Amount of instructions executed for the program in a measured time interval.
branch-instructions	Total number of instruction sets that can have unconditional or conditional branching paths to code execution.

Attribute	Description
bus-cycles	Number of cycles required to read or write a single transaction between CPU or external memory on the bus.

Figure 2. Dataset Attributes and Descriptors [4]

In this case, GainRatioAttribute was over InfoGainAttribute. While both are similar InfoGain favors classes with a large number of samples due to the non-normalization of data, while GainRatioAttrbiute normalizes data leaving it less influenced by imbalanced sets. For the data set used in training and testing, an imbalanced set was utilized.

III. TESTING AND VALIDATION

The testing method used for the data set provided is cross validation with K folds. The purpose of cross validation is use a subset of the data to check the training, while the remaining subsets are to be used for training a model. The method will divide the subset into K sets, 10 in this case, and will train on K-1 subsets and test on the Kth subset. In this manner, the data set is randomized and split into 10 data subsets so that when training the model it will have 9 trainable data sets with 1 testing set [4].

IV. CLASSIFICATION ALGORITHMS

A. Logistic Regression

A statistical model used to predict the probability of a certain class or event based off an existing data set. Each output modeled as 0 or 1 which can then be compiled into a set of data used to predict the overall class [5]. It is an alternative implementation to building rules using multinomial logistic models with an estimator. The output will be in tabular form where the output is the probability of given inputs m attributes class j is the output. All outputs are computed with an odds ratio in Weka, where the odds ratio represents the constant effect of the predictor value which replaces the probability of the likelihood of each classification [6].

B. JRIP

JRip is an algorithm proposed by William W. Cohen as an optimized version of IREP. JRIP implements propositional rule learning and Repeated Incremental Pruning to Produce Error Reduction. It works in multiple stages. There is a building phase that is growing and pruning of the rules. In the growing phase is a phase were rules are grown, the rules grown are selected with the condition that gives the greatest information. After growing the rules, the algorithm will prune each rule. Once generation and pruning have been completed the next stage is to generate and prune two variants of the rules. Once the rules have been generated, the variants are computed for the smallest DL. The most minimal DL variant is selected, all others are deleted [7].

C. OneR

OneR stands for “One Rule” is a type of classification algorithm that uses one rule for each predictor (attribute) in the data set. The rule used is the one with the smallest total error when compared to the other rules. It uses a 1R classifier that has one parameter as the minimum bucket size for discretization [6]. The algorithm functions as follows:

Counts the occurrence of each attribute that appears. Looks at the most frequent class, assigns the rule to that class and calculates the total error of the rules of each predictor. Chooses the predictor with the smallest error.

The OneR algorithm allows for easy human interpretable data, while only being slightly less accurate than the state-of-the-art classification algorithms [8].

D. J48 — Decision Tree

J48 is a decision tree algorithm that implements the Iterative Dichotomiser 3. It implements a version called a C4.5 decision tree which proposed by Ross Quinlan. The algorithm uses information entropy in order to classify [9] each set. The algorithm will select a class of p dimensions that has already been classified during the training phase and select an attribute of x_i from a class of c_i. At each node of the decision tree, C4.5 will then split an attribute into one class or the other; making a binary decision; splitting the data based off a normalized information gain which is based off of a delta of entropy. After the decision has been made it will prune the tree based off of a greedy selection adding a leaf node to the overall decision tree that causes the greatest decrease of entropy [10]. For J48 the default setting is a pruning threshold of 0.25. [10]

E. NAïVE BAYES

It is a classifier that is built around Bayes’ rule and a naïve assumption about the feature set. The assumption is that for each feature the probability of that feature is independent of all other features.

$$P(Class \mid Datum) = \frac{P(Datum \mid Class) * P(Class)}{P(Datum)}$$

Naïve Bayes works through the concept of joint and conditional probabilities. It can be advantageous to use because it uses a naïve approach in weighing and evaluating attributes. On small sets of data, the algorithm can yield better results than some others due to having a lower chance to overfit in it's categorization. It can be a quick and simple ML model to build comparatively to others used and its resource usage is not too high. However, on continuous real-valued data it can fail since the binning of data and assignment of classes can be sub-optimal as data is lost. Additionally it's performance can be skewed as it is very sensitive to biased data [11].

F. SUPPORT VECTOR MACHINE — SMO

Support Vector Machines are trained to determine a hyperplane that segments the data in a vector space into clusters corresponding to the target classes. In this case the hyperplane should segment the data vector space into regions corresponding to malware categories. SMO or Sequential Minimal Optimization is an efficient algorithm to train the SVM [12].

Sequential minimal optimization is used for quadratic problems. Problematically, support vector machines can be slow to build and train, to reconcile the speed it can be optimized with SMO. The implication is that the running time ($O(N^3)$) which requires memory of size N, the conclusion of this algorithm is the larger the dataset the slower the training. SMO aids SVMs by changing the chunking method to a return of alphas that help in satisfying the constraint optimization. The alphas will be used to identify support vectors in deciding the data class [12].

G. MULTILAYER PERCEPTRON

The multi-layer perceptron model is a fully connected neural network with one hidden layer. Each node is connected to all nodes in the following layer with a weight that multiplies value that is passed to the following layer. The nodes have an sigma activation function that triggers activates the neuron when the input is greater than the threshold. During the back propagation phase, the weights are updated by using the calculated error and default learning rate. The learning rate specifies how much the weights are corrected during each pass.

H. CONVOLUTIONAL NEURAL NETWORK

Convolutional neural networks (CNNs) are a type of deep neural networks that consist of fully connected layers that reduce the dimensionally of an input image using pooling techniques that mimic the visual processing system of mammals.[16] Given a 2-D image as input the pooling layer reduces the value passed on in the network. The stride value and kernel size are hyperparmaters used to create a selection boundary that iteratively steps across and down then image. As the selection boundary moves across the image, it reduces the dimensionally of the input by selecting maximum value of the surrounding pixels.

V. DATASET ANALYSIS

The data was used in two ways. First was through a binary classification of malware and the second was a multi-classification. The difference between the two data sets is how the data is distributed. In both cases the data was used on all classification algorithms, however the labels of the malicious software differ. In the binary classification all malicious software was generalized to a class called malware and non-malicious software generalized to benign. However, for multi-classification the malicious software was given labels: backdoor, rootkit, trojan, worm, and virus: but the benign software was still generalized. In both cases the data set remained the same except for labels, this can be problematic because when training ML and AI type models the more data present the more accurate the models can be. In the case of binary, the models are easier to train as all resolution is lost and a generalized class is given, while for multi-class the data present for malicious software is greatly reduced as the malware is now broken down into types. Yet this is helpful because a multi-classification will allow for the system to not only detect an attack but to utilize strategies to mitigate or stop the attack.

Additionally, the data was transformed to into 16px-by-16px grayscale images for use with the CNN. For each attribute, the maximum observed value was used to fit that attribute's values on a range from 0 to 255 which corresponds to the pixel value in the grayscale image. Each data sample was repeated until it would fill a 16px row. Then each row was duplicated until we obtained a 16x16 matrix. The 16x16 matrix was written to disk as a .png This transformation was performed with the 4-attribute

dataset and 8-attribute dataset.

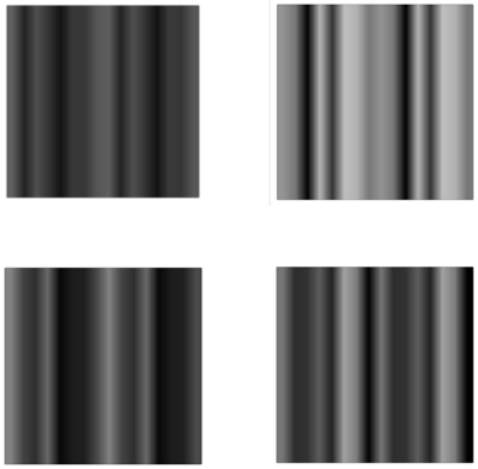

Figure 3. Gray scale images samples. Virus (top left), Trojan (top right), Rootkit (bottom left), and Benign (bottom right)

A. Binary Classification

Binary classification was tested with classification methods such as, logistic regression, JRip, J48, and OneR. These algorithms were tested against a reduced data set of varying dimensions; 2, 4, and 8. The variance of attributes will reflect the accuracy and flexibility of properly classifying the data, however it is important that all aspect observed and assessed as hardware can be a limiting factor because each processor can have a varying amount of HPCs, 2, 4, 8, or more.

The accuracy elements that will be used to create a decision on the better model will be F-measure, Accuracy: correct classification instances; and ROC area graphs. The F-measure and correct classifications will be the most necessary attributes to look at in the decision making process, however looking at the ROC areas and their graphs can indicate the performance of the algorithms. The F-measure is important since is combines the precision and recall to observe the combined metric. ROC indicates how well a model will classify the dataset in its decision making process.

1. 2 Attribute

With 2 attributes accuracy will be greatly diminished in most respects depending on the model chosen. This means that not all models are good for limited data sets, it indicates that while some models might be more accurate on a large set of data; given a limited amount to classify it can be inaccurate.

After running the training and testing the models on the dataset the below accuracy elements are given based off of the positive truth of correct classification of malware.

	Accuracy	F-Measure	ROC Area
J48	0.90	0.88	0.95
JRIP	0.89	0.87	0.91
LR	0.54	0.31	0.55
OneR	0.86	0.84	0.85

Figure 4. 2 Attribute results.

With the given dataset, a decision can be made. Looking at the accuracy and F-measure. In the presented information the inaccuracy can immediately be seen regarding the logistic regression (LR). Due to how the logistic regression assesses information it lacks data on correctly being trained by quantity. Since accuracy is not only determined by the number of attributes available to the model but the quantity of the training set. The larger the training set, the more accurate. In this case the most desirable would be J48 as it correctly classified 90% of the data and maintained the highest F-measure at 0.88 indicating it most likely had the best precision and recall overall when looked at together. The ROC as seen as a numeric value is the highest as well which means that the model most likely has the highest correctness in deciding if a program is malicious. Further examination of ROC can be done through figure 5.

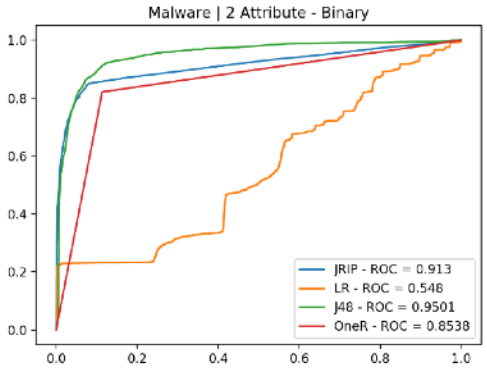

Figure 5. ROC

The graphs above indicate the performance of the models. Immediately, logistic regression can be seen as the worst when mapping the false positive rate versus the true positive rate. The best happens to be J48 as it reaches it's maximum the quickest and maintains it through out the decision process.

Considering all aspects when operating a 2 HPC process, J48 would most likely be the best model. On the other hand, if more attributes are added that decision may change.

2. 4 Attribute

Adding two more attributes will always increase performance of any model, however not all processors are capable. When given hardware that supports more event monitoring it is always worthwhile to utilize it in order to increase the accuracy measurements.

	Accuracy	F-Measure	ROC Area
J48	0.96	0.95	0.97
JRIP	0.94	0.93	0.96
OneR	0.86	0.84	0.85
LR	0.73	0.66	0.73
MLP	0.90	0.89	0.96
CNN	0.98	-	-

Figure 6. 4 Attribute results

With the additional dimensions to the dataset used for testing and training, individually the performance of the accuracy of the models has already increase by at least 10%. In the case of the above data, logistic regression is still the worst performing model however J48 is becoming marginally better then JRIP. Depending on the running time it may be more worthwhile to use JRIP over J48. By pure metrics without considering decision time, J48 maintains the highest accuracy, F-measure and ROC area. Which can be further seen when looking at figure 7.

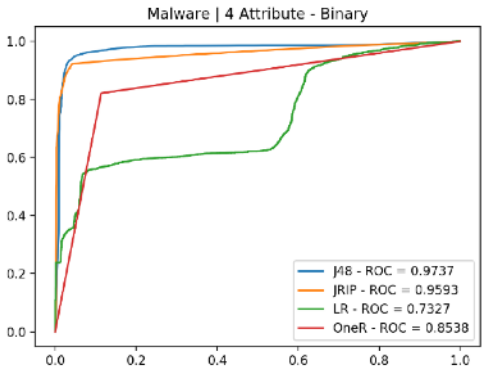

Figure 7. ROC

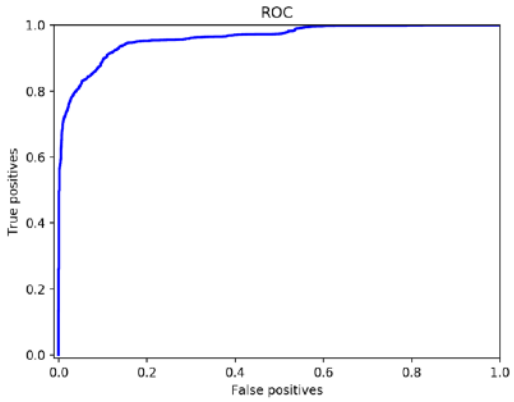

Figure 8. ROC MLP

Considering that J48 still is a better option when compared to its contemporaries the graph itself shows a fast growth to 1.0 on the true positive rate while JRIP has a bit slower growth and this can imply that J48 is still the better model in terms of response and decision during real-time evaluation.

The CNN obtained the highest accuracy; however, due to shortcomings in the CNN testing phase the F-measure and ROC were not available for comparison.

3. 8 Attribute

In 8 dimensions the accuracy measurements of the data set of all values reaches it's "maximum" as most hardware components do not exceed 8 HPCs. In figure 9 the best model is J48 with a marginal improvement from 4 attributes and over JRIP. When the decision is close enough that it can be considered marginal at best the ROC graph will aid, additionally so would run-time. It should be notes however that neither J48 or JRIP has exceedingly long runtimes.

	Accuracy	F-Measure	ROC Area
J48	0.96	0.96	0.98
JRIP	0.95	0.95	0.97
LR	0.75	0.68	0.75
OneR	0.86	0.84	0.85
MLP	0.93	0.92	0.97
CNN	0.91	-	-

Figure 9. 8 Attribute results

Observing figure 10 the growth of the better models, J48 and JRIP following similar growth rates until reaching both reach a saturation point. The saturation

point essentially is when the model reaches it's slowest true positive rate versus a false positive rate.

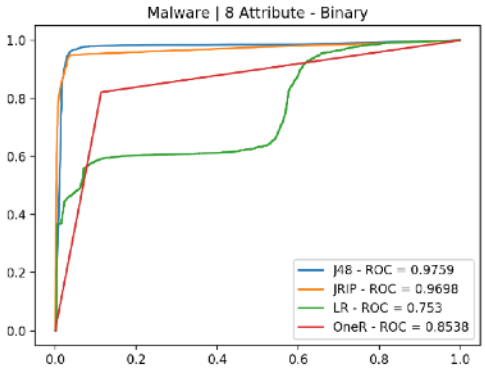

Figure 10. ROC

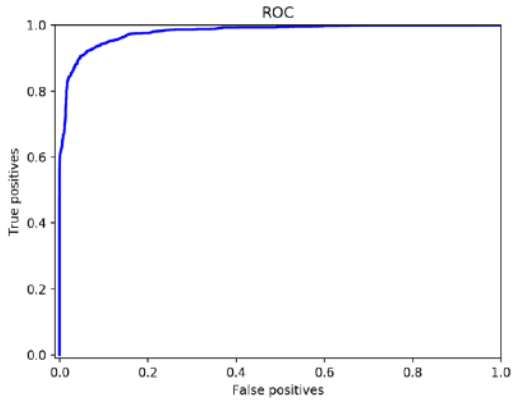

Figure 11. MLP ROC

The delta between the two is smallest enough to be considered negligible and does not clearly indicate if one is true better. However, looking at only metrics and not external factors J48 is the better option. The neural network models performed well, but were not better than the rule based models for this trial.

3. Cross Comparison of dimensions

Typically, when adding in more dimensions an algorithm will behave better yet there is always a question. The question is at what point is the return minimal enough to were the process or training cost outweighs the usage of the extra HPCs. The determination can be done by taking the best models across dimensions and comparing their values. Since all versions of the binary classification indicated J48 as the better option J48 will be compared against itself.

Dim	Accuracy	F-Measure	ROC Area
8	0.96	0.96	0.98
4	0.96	0.95	0.97
2	0.90	0.88	0.95

Figure 12. J48 Results

Observing the above accuracy measurements there is a clear improvement in use 4 and 8 attributes over 2; however between 4 and 8 the difference is small. With a small delta between the 4 and 8 attributes it would most likely make sense to use the 4 dimensions as it would be faster to train models on a reduced dataset over one that requires twice as much information. An added benefit is the reduced tax on access the HPCs from the processor or in calculations as the there are not as many nuance branches to determine the classification. Overall, given a choice 4 attributes work fine. If the architecture does not allow for 4 and 8, then J48 at 2 attributes suffices to accomplish correct classification.

A. Multi-Classification

Unlike binary classification, multi-classification allows for greater resolution of the data set as it no longer indicating malicious or benign but now shows the type of malicious software. The resolution becomes important since it allows the detection system more insight so that it can use a strategy to stop the attack, which would not be possible in binary without extra processing. The classes that will be determined are benign, virus, worm, trojan, backdoor, and rootkit. However, unlike binary classification the data set needs to be a magnitude larger otherwise classes may lack enough data to train a model to properly identify leaving for a large uncertainty when deciding classes. When observing the data for 4 and 8 attributes there will be some classifications that are unknown requiring an assessment into the individual categories to determine the problem areas. The problem areas can be resolved by increasing the amount of data present in each class by either adding dimensions, which can be problematic for sampling due to limited HPCs, or increase the amount of software in each category until the model is able to correctly identify the classes.

1. 4 Attributes

With 4 attributes used for classification the models can suffer in the decision process on the tested data set. Looking at figure 13, there lacks a large unknown for the F-measure implying that the precision and recall of the model could be improved by increasing the number of dimensions or the data set could be expanded by adding more malicious software of the uncertain classes.

	Accuracy	F-Measure	ROC Area
LR	0.73	?	0.82
NB	0.47	?	0.77
SMO	0.71	?	0.73

MLP	0.96	0.85	0.98
CNN	0.81	-	-

Figure 13. 4 Attribute Results

MLP is the clearly the best performing model. It is hard to judge the second best performing model due to the missing F-measures. Yet Naïve Bayes can be ruled out immediately as its accuracy in correctly identifying classes is less then 45%. The average ROC areas are close but it looks like Logistic Regression is the better option. To correctly asses it however the ROC graphs need to be assessed and the individual attributes. Refer to appendix A for the graphs of the ROC for each class. In the breakdown of the data in figure 14, the classes with uncertainty vary between models.

	F-Measure	ROC Area	Class
LR	?	0.731	backdoor
	0.804	0.796	benign
	0.063	0.895	rootkit
	0.756	0.844	trojan
	0.493	0.838	virus
	0.878	0.929	worm
NB	0.177	0.87	backdoor
	0.565	0.722	benign
	?	0.631	rootkit
	0.551	0.784	trojan
	0.485	0.849	virus
	0.787	0.942	worm
SMO	?	0.757	backdoor
	0.79	0.675	benign
	?	0.585	rootkit
	0.551	0.83	trojan
	0.502	0.81	virus
	0.884	0.882	worm

Figure 14. Breakdown of Results

Immediately a conclusion can be given that each model requires a different number of minimum data elements of p dimensions to remove the uncertainty in prediction as each model has a different category or set of categories unknown. Overall however, logistic regression has the highest F-measures on average across all the classes with only backdoor be unknown. This could be resolved by increasing the attributes or expanding the data. However with the current data logistic regression will be the more obvious option to select if resource constraints rule out MLP.

2. 8 Attributes

MLP again performs surprisingly well. Adding 4 more attributes drastically reduces the uncertainty in some models used for classification. Seen in figure 15 for example logistic regression no longer has uncertain values in the total weighted F-measure.

	Accuracy	F-Measure	ROC Area
LR	0.81	0.78	0.87
NB	0.46	0.52	0.76
SMO	0.80	?	0.81
MLP	0.96	0.86	0.98
CNN	0.83	-	-

Figure 15. 8 Attribute Results

The only f-measure that is uncertain in this case is SMO. Overall from a general standpoint, logistic regression would be the better option to choose as it has a 1% on correct classification and a F-measure for all classes. In figure 14, the individual class breakdown shows that only logistic regression and naïve bayes defines all classes. It should be noted that while on average the logistic regression is better across most aspects, depending on the malicious software it may not be as accurate, for example backdoor between the two shows naïve bayes as the better choice but lacks accuracy for other classes in comparison.

	F-Measure	ROC Area	Class
LR	0.005	0.849	backdoor
	0.851	0.832	benign
	0.34	0.893	rootkit
	0.781	0.88	trojan
	0.808	0.924	virus
	0.893	0.943	worm
NB	0.179	0.879	backdoor
	0.553	0.727	benign
	0.3	0.636	rootkit
	0.54	0.75	trojan
	0.499	0.83	virus

	0.545	0.923	worm
SMO	?	0.796	backdoor
	0.845	0.778	benign
	0	0.635	rootkit
	0.778	0.834	trojan
	0.803	0.9	virus
	0.885	0.882	worm

Figure 16. Breakdown of Results

Further looking at Appendix A for the 8 attribute ROC graphs shows that the logistic regression has some of the highest true positive rates against the false positive indicating it as a better option.

For decisions, it would be better to use logistic regression as it classify every software but at a cost. If the goal is to also serve mitigation and prevention measures the it may be worthwhile to expand the data set used to train to ensure higher accuracy measures.

3. Cross Comparison of dimensions

Looking at the two best models that were developed based off of the dataset, logistic regression for both, the primary difference occurs in the classification. The 4 dimension attribute has uncertainty in the classification of backdoor while the 8 dimension does not. To resolve the 4 dimension the dataset would need to be expanded, however the larger the dataset the larger the cost. It will take time and effort to gather the data, classify the data, and train the model. Depending on the timeframe this may not be a feasible option. As such the 8 attribute would be better if the necessary HPCs are available.

VI. CONCLUSION AND DISCUSSION

In the end, the best algorithm or model for classifying programs or any task is determined based off of current parameters and goals. In the case of Binary, there was marginal differences in the 4 and 8 attribute models but in the multi-class attribute there was a large different due to uncertainty. A question might be, to simply use the binary classification to classify data, however there is an issue. Binary classification will only indicate if an attack is occurring and the resolution of the threat will be delayed as additional work needs to be done to determine the threat. With a multi-classification method the strategies can be pre-planned then served as the attack is identified.

Furthermore the limiting factor in any model and training decision is based off of hardware constraints. Certain models will require more resources slowing down or reducing performance of the device. Additionally, the number of available HPCs vary between processors. Between these two requirements, the preference of models can vary widely as the most accurate model may not be the best model for the application.

We are skeptical of the performance record of the MLP model for Multi-Classification because the accuracy and f-measures were far beyond that of the other models. We failed to get the CNN model working using Keras/Python but switched to MatLab to finally get a working classifier.

VII. REFERENCES

[1] e. a. C. Malone, "Are hardware performance counters a cost-effective way for itnegrity cehcking of programs"," in *ACM STC Workshop*, 2011.

[2] M. Hall, "Class GainRatioAttributeEval," 15 October 2019. [Online]. Available: http://weka.sourceforge.net/doc.dev/weka/attributeSelection/GainRatioAttributeEval.html.

[3] Unknown, "PERF_EVENT_OPEN(2) Linux Programmer's Manual PERF_EVENT_OPEN(2)," 8 October 2019. [Online]. Available: http://man7.org/linux/man-pages/man2/perf_event_open.2.html.

[4] J. Brownlee, "A Gentle Introduction to k-fold Cross-Validation," 15 October 2019. [Online]. Available: https://machinelearningmastery.com/k-fold-cross-validation/.

[5] J. Brownlee, "Logistic Regression for Machine Learning," 15 September 2019. [Online]. Available: https://machinelearningmastery.com/logistic-regression-for-machine-learning/.

[6] I. H. Witten, E. Frank, M. A. Hall and C. Pal, "The WEKA Workbench," 16 September 2019. [Online]. Available: https://www.cs.waikato.ac.nz/ml/weka/Witten_et_al_2016_appendix.pdf.

[7] UNKNOWN, "Class JRip," 15 September 2019. [Online]. Available: http://weka.sourceforge.net/doc.dev/weka/classifiers/rules/JRip.html.

[8] Unknown, "OneR," 15 September 2019. [Online]. Available: https://www.saedsayad.com/oner.htm.

[9] S. L. Salzberg, "Book Review: C4.5: Programs for Machine Learning by J. Ross Quinlan. Morgan Kaufmann Publishers Inc., 1993," *Machine Learning,* vol. 16, pp. 235-240, 1994.

[10] P. Kapoor and R. Rani, "Efficent Decision Tree Algorithm Using J48 and Reduced Error Pruning," *Internation Journal of Engineering Research and General Science Volume 3, Issue 3, May-June,* pp. 1613-1621, 2015.

[11] J. Catanzarite, "The Naïve Bayes Classifier," 31 October 2019. [Online]. Available: https://towardsdatascience.com/the-naive-bayes-classifier-e92ea9f47523.

[12] E. Frank, S. Legg and S. Inglis, "Class SMO," WEKA, [Online]. Available: https://weka.sourceforge.io/doc.dev/weka/classifiers/functions/SMO.html. [Accessed 7 December 2019].

[13] BIG-DATA.TIPS, "Sequential Minimal Optimization," 31 October 2019. [Online]. Available: http://www.big-data.tips/sequential-minimal-optimization.

[14] S. D. Anuj Sharma, "Performance Investigation of Feature Selection Methods and Sentiment Lexicons for Sentiment Analysis.," *IJCA Special Issue on Advanced Computing and Communication Technologies for HPC Applications ACCTHPCA(3),* pp. 15-20, July 2012.

[15] J. Brownlee, "Crash Course On Multi-Layer Perceptron Neural Networks," 15 September 2019. [Online]. Available: https://machinelearningmastery.com/neural-networks-crash-course/.

[16] I. Goodfellow, Y. Bengio, A Courville, 2016, MIT Press, [Online]. Available: http://www.deeplearningbook.org

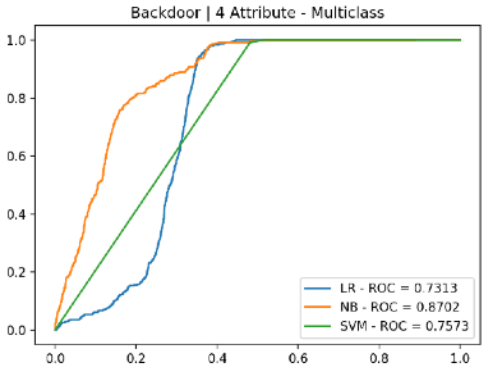

Figure 17

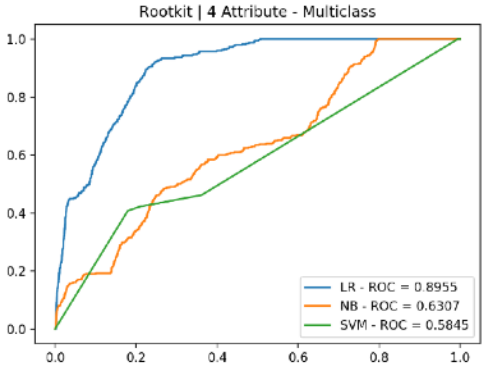

Figure 18

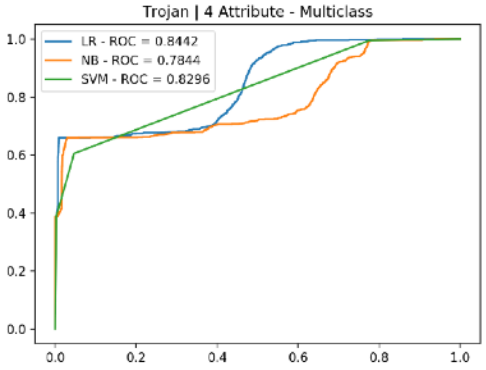

Figure 19

VII. APPENDIX A

1. 4 Attributes ROC Graphs

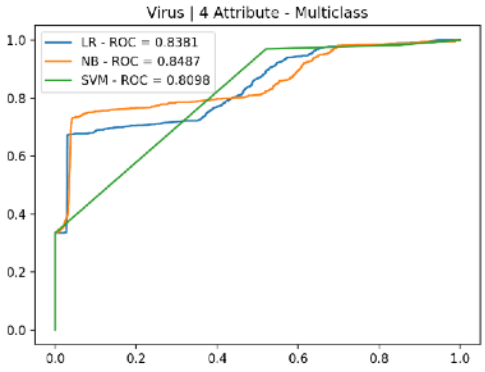

Figure 20

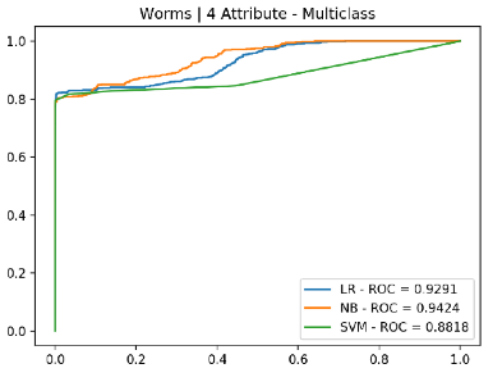

Figure 21

2. 8 Attributes ROC Graphs

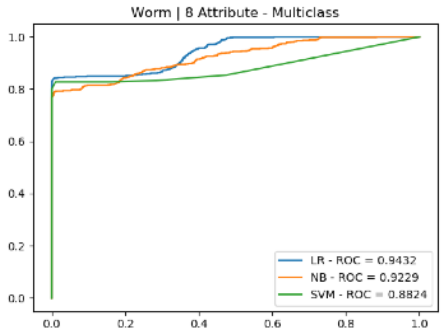

Figure 22

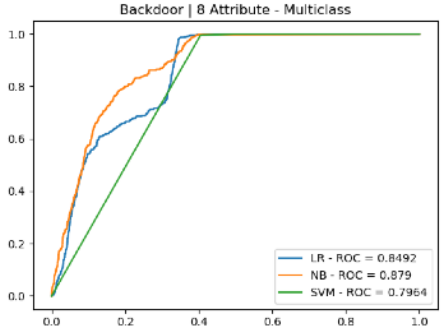

Figure 23

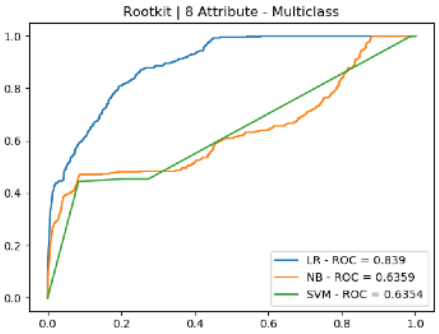

Figure 24

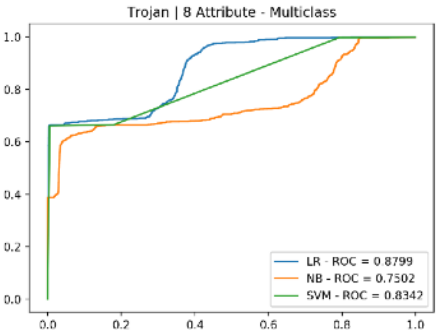

Figure 25

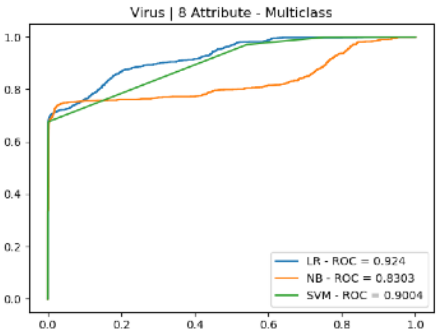

Figure 26

3. 4 Attributes MLP (Multiclass)

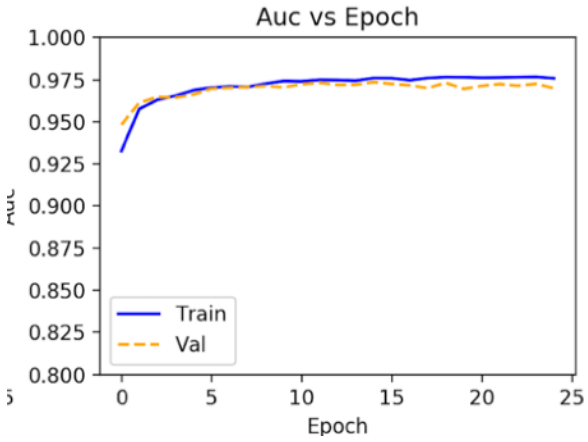

Figure 27

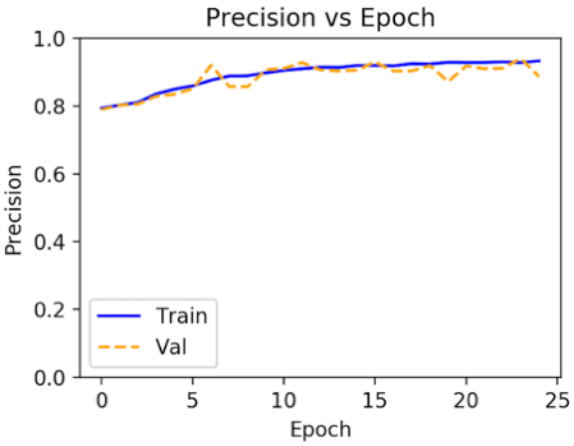

Figure 28

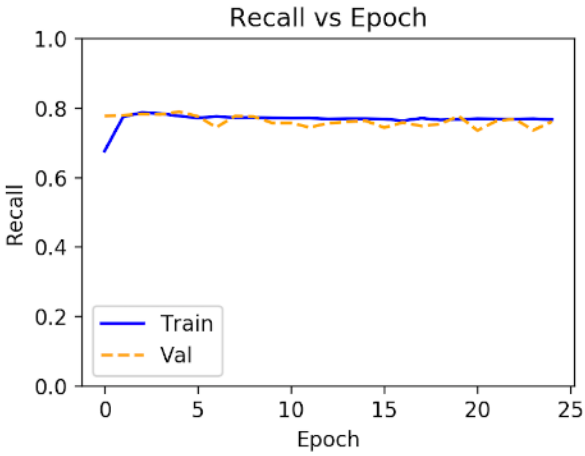

Figure 29

4. 8 Attributes MLP (Multiclass)

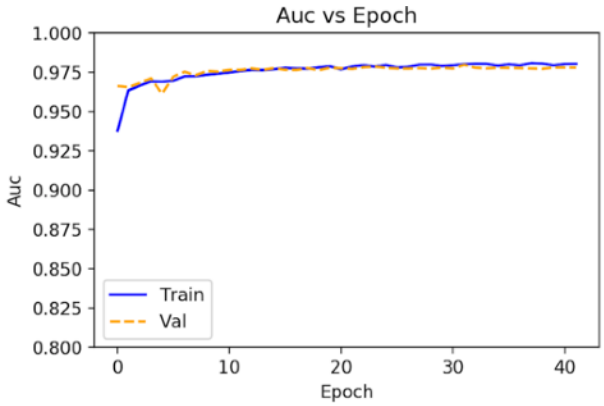

Figure 30

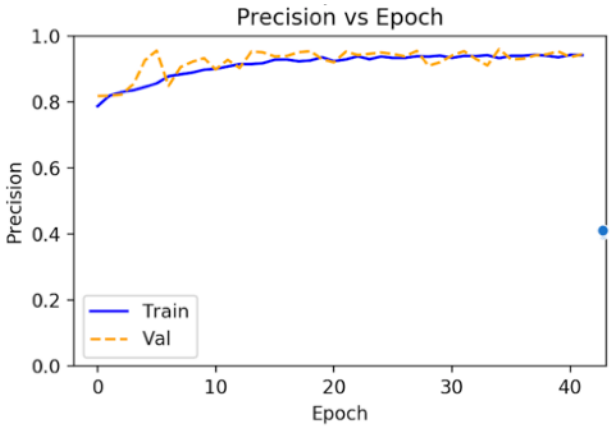

Figure 31

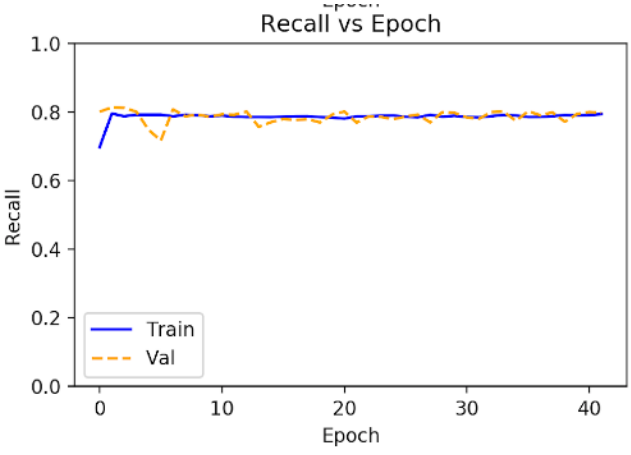

Figure 32

5. 4 Attributes CNN Training

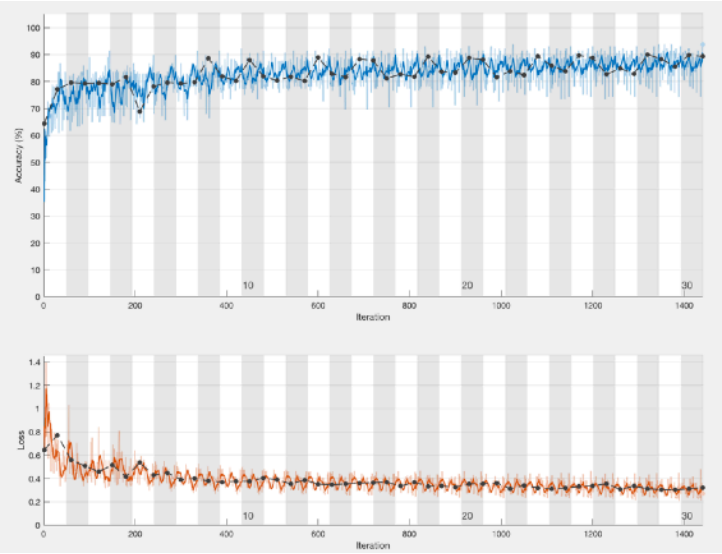

Figure 33

6. 8 Attributes CNN Training

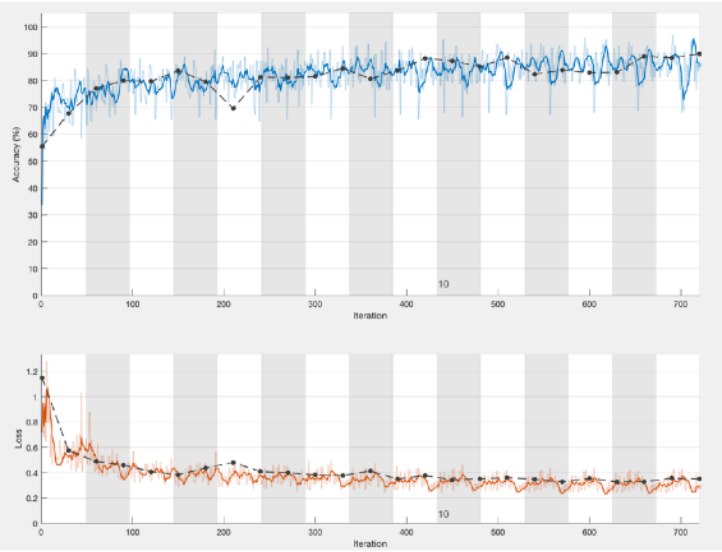

Figure 34

7. 4 Attr. Multiclass CNN Training

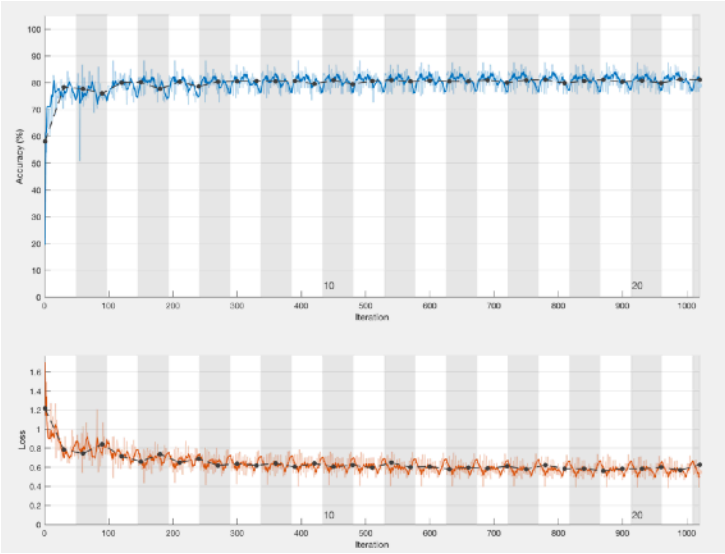

Figure 35

8. 8 Attr. Multiclass CNN Training

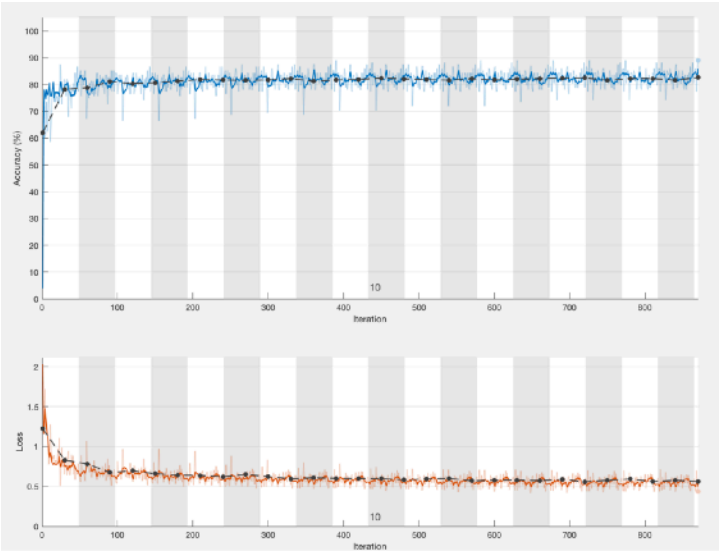

Figure 36