Supervised and Unsupervised Learning Techniques Utilizing Malware Datasets

Daryle Smith, Sajad Khorsandroo and Kaushik Roy

February 4, 2023

Supervised and Unsupervised Learning Techniques Utilizing Malware Datasets

Daryle Smith
Department of Computer Science
North Carolina A&T State University
Greensboro, USA
dsmith18@aggies.ncat.edu

Sajad Khorsandroo
Department of Computer Science
North Carolina A&T State University
Greensboro, USA
skhorsandroo@ncat.edu

Kaushik Roy
Department of Computer Science
North Carolina A&T State University
Greensboro, USA
kroy@ncat.edu

***Abstract*—Malware continues to gain momentum as it becomes more sophisticated against detection. Monitoring tools and antivirus software do not have the ability to keep up with the ever-going changes of these malignant variants. Due to these dilemmas, machine learning has gained popularity in classification and detection of malware related data. In this study, two separate datasets, Malware-Exploratory and CIC-MalMem-2022, undergo a series of supervised and unsupervised learning procedures to first gather information for observation. The developed model in this research utilizes three clustering algorithms for analysis, K-Means, DBSCAN, and GMM. The model also uses seven classification algorithms for predicting malware including Decision Tree, Random Forest, Ada Boost, KNeighbors, Stochastic Gradient Descent, Extra Trees, and Gaussian Naïve Bayes. Results have shown that Malware-Exploratory dataset averaged an accuracy score of 90% while CIC-MalMem-2022 dataset averaged a score of 99%. Both datasets also showed consistency across all three clustering algorithms. Besides, correlation between variables do not necessarily need to be highly related for malware detection. Future studies will determine if the results remain stable against feature selection and genetic algorithms.**

Keywords—area under the curve-receiver operating characteristics (AUC-ROC), density-based spatial clustering of applications with noise (DBSCAN), Gaussian Mixture Model (GMM), hierarchical density-based spatial clustering of applications with noise (HDBSCAN), supervised machine learning, unsupervised machine learning

I. INTRODUCTION

Malware detection continues to be a focal point as attacks continue to gain momentum. Reports have indicated that over 270,000 new malware variants have been detected in the first half of 2022 [1]. The traditional methods of detecting malware have been effective but struggle to keep up with newly introduced variations. For example, sandbox systems performing certain functions continue to be problematic and leave the system at risk for infection [2]. Furthermore, monitoring the events within an operating system and using software scanning tools to find malware related signatures becomes an exhaustive approach resulting in considerable performance cost. The inability to detect new malware is a key concern as malware continues to get more sophisticated against detection tools [3]. To alleviate some of these hardships, the introduction of machine learning algorithms against malware datasets and samples are making great contributions within respective areas.

Results are showing evidence that supervised and unsupervised learning techniques are working and proving to be useful against big data [4]. Classification and regression related problems work well when using supervised learning techniques as they use algorithms to create a hypothesis function to correlate input data to anticipated outputs [5]. These models provide parameters during the training phases with labeled data and tell the system what output is related to each specific input value [6]. Though the trained model is presented with test data that has been labeled, the labels are foreign to the algorithm. The end goal is to see how accurately the algorithm will perform on unlabeled data. Unsupervised learning takes a different approach on its data by using algorithms to analyze and cluster unlabeled datasets. These algorithms discover patterns or data groupings without the need for human intervention [7]. Their ability to find relationships and variances in information make them a viable solution for experimental data analysis. Clustering and dimensionality reduction are two cases in point where these algorithms work well.

This paper aims at providing a comparative study on malware-related datasets. To investigate whether low correlation can still predict malware samples with high accuracy, two different datasets have been chosen. The first having high correlation against variables while the other has low correlation against variables. The two datasets will undergo a series of unsupervised learning algorithms for clustering, K-Means, DBSCAN, and GMM. They will also be tested against six supervised learning algorithms in detecting and predicting malware. These algorithms include Decision Tree, Random Forest, Ada Boost, KNeighbors, Stochastic Gradient Descent, Extra Trees, and Gaussian Naïve Bayes.

The rest of the paper is organized as follows. Section II reviews existing related work. Section III discusses our method and overviews the selected datasets along with the data preprocessing and machine learning algorithms used. Section IV presents obtained results. Section V outlines possible future work , while section VI concludes the paper.

II. RELATED WORK

One possible approach to apply an unsupervised learning algorithm against Distributed Denial of Service (DDoS) attacks was conducted in [8] where the research was geared towards detecting Slow Drip. A detection tool using statistical classification was created to formulate a proper dataset. It monitored query responses at the Second Level Domain (SLD) whenever the threshold's limit was reached for the given day. Once reached, the timestamp, the fully qualified domain name, the query type, and response code were all captured for clustering purposes. Other research has been shown to classify and detect malware at the domain name system (DNS) level, but research in [8] focused solely on passive DNS. After collecting data over 7 months and manually performing data preprocessing techniques, the hierarchical density-based spatial clustering of applications with noise (HDBSCAN) algorithm was used on the dataset [8]. Ultimately, the clustering described attacks that have general features in common and represent attacks that are likely coming from the same attacker. Nine groups of attacks were determined on the final dataset.

Manzano et al. [8] analyzed Android network traffic using six supervised learning algorithms. Two statistical methods of feature reduction and feature selection were applied on the dataset, followed by principal component analysis (PCA) and logistic regression (LR). These were used to select the most dominant features related to the bidirectional packets and the time properties of the resulting flows. The features were used to train the algorithms using multiclass and binary classification. For comparison metrics and performance evaluation, precision, recall, F-measure, accuracy, and area under the curve (AUC-ROC) were used. Concluding results show that random forest had the best accuracy at 96% and an AUC-ROC of 0.98 in binary classification. Multiclass classification also had its best accuracy with random forest at 87% and an AUC-ROC over 95%. Evidence showed this was better than the five other machine learning algorithms and both experiments used the thirteen most dominant features selected by LR.

Research in [9] uses feature selection and supervised learning against a malware dataset that consists of 89 benign and 68 malicious samples. The goal of this study is to use unsupervised and supervised learning techniques to determine if feature selection leads to higher accuracy gains. Virtustotal [9] has been used for static analysis because of its ability to scan multiple antivirus engines. For dynamic analysis, the Malwr sandbox [9] was used based on its highly efficient results against malware samples. The data obtained from the analysis processes are presented using XML. The XML data then gets transformed using JDOM [9] to create a comma separated values (CSV) file. The file is used as an input for the employed machine learning algorithms. A plethora of supervised learning algorithms were used, and it was determined that multilayer perceptron (MLP) had the best results with and without feature selection [9]. The only unsupervised learning algorithm used in this study was estimation maximization (EM) for its benefits to analyze latent unobserved data.

Authors in [10] conduct a study on the Android operating system to detect mobile malware. This space is limited in detection methods, and as smartphone users continue to grow, the attacks of mobile malware increase. This research uses supervised learning to detect 10 subtypes of mobile trojans by evaluating dynamic hardware features such as CPU usage, memory usage, and battery usage. The dataset was obtained from 47 different mobile users in real time and was classified using Random Forest, K-Nearest Neighbors, and Ada Boost. The classification results are based on app specific features and global device features concluding that Random Forest performed the best with an F1 score of 0.73, a False Negative Rate (FNR) of 0.380, and a False Positive Rate (FPR) of 0.009 [12].

III. Methodology

This section will introduce the two datasets being analyzed, the data preprocessing steps, the unsupervised learning algorithms, and the supervised learning algorithms. For unsupervised learning, the objective is to use three clustering algorithms against the unlabeled datasets to see if they are producing consistent results. For supervised learning, the objective is to use seven algorithms to see if they can predict malware with high accuracy. It is known that the Malware-Exploratory dataset has very low correlation amongst its features, so the intention is to see if the model can still predict at high accuracy. All tests being conducted are performed in Jupyter Notebook using python version 3. Fig. 1 provides a diagram of these steps.

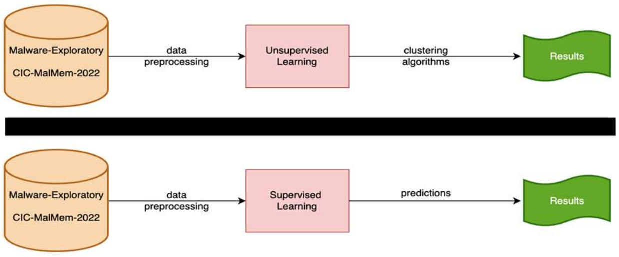

Fig. 1. Methodology - steps and procedures

A. Datasets

The first malware dataset, Malware-Exploratory, was downloaded from [11]. The original file downloaded was already in csv format and contained 216,352 rows and 58 columns. This dataset was labeled meaning the first row included the header. The dataset contained 140,849 benign samples and 75,503 malignant samples. As this dataset is open source, the size will vary over time as more updates are added to the csv file.

The second malware dataset, CIC-MalMem-2022, was downloaded from [12]. The original file downloaded was also in a csv format and contained 58,596 rows and 57 columns. This dataset was also labeled meaning the first row included the header and contained an even split of malignant and benign samples at 29,298 each.

B. Data Preprocessing

Malware-Exploratory dataset was reduced to use 55 features due to improper datatypes. Cells that contained NaN values were converted to 0 so that the machine learning algorithms could function correctly. The main feature in this dataset was called "legitimate". This label let us know if the sample was benign (0) or malignant (1). During unsupervised learning, the first row was removed as it contained the labels.

CIC-MalMem-2022 dataset was reduced to use 56 features due to improper datatypes. The most important feature called "Class" lets us know if the sample was benign or malignant. Due to using string datatypes, benign values were converted to 1 while malignant values were converted to 2. During unsupervised learning, the first row was removed as it contained the labels.

C. Unsupervised Learning

Three clustering algorithms were used to observe what each dataset looked like once the labels were removed. The fist algorithm used was K-means. This algorithm is used to calculate distances from centroids to points and groups points to the closest cluster. As data is not labeled, it is required to find a value for k which is determined by the elbow method. As the value of k increases, less elements are seen in each cluster. The average distortion decreases and the point where this distortion declines the most is the elbow point.

The second algorithm used against each dataset was density-based spatial clustering of applications with noise (DBSCAN). Unlike K-means, the value for k is not needed and this algorithm determines that a point belongs to a cluster if it is close to many points from that cluster. Two important parameters for this algorithm are eps, which is the distance that specifies the neighborhoods, and minPts, the minimum number of data points to define a cluster.

The third algorithm used is Gaussian Mixture Model (GMM). This method is very similar to K-means but classifies data into different categories based on the probability distribution. The best use case for this algorithm is when there is uncertainty about the correct number of clusters, and when clusters have different shapes.

D. Supervised Learning

Each dataset will be tested against seven algorithms. For the first dataset, the algorithms are trying to predict the "legitimate" feature as it tells us if the sample is benign or malignant. The same goes for the second dataset, its feature is labeled as "Class". Both datasets are using a 20% test size. The algorithms are as follows: Decision Tree, Random Forest, Ada Boost, KNeighbors, Stochastic Gradient Descent, Extra Trees, and Gaussian Naïve Bayes.

IV. Results

This section provides figures and tables for both datasets with respect to the unsupervised and supervised learning techniques. Fig. 2 provides a heatmap of both raw datasets after data preprocessing. The objective is to provide insight on how correlated the variables are within each dataset before any tests have been performed. Fig. 3 shows the clustering graphs for all unsupervised learning algorithms for dataset Malware-Exploratory. Table 1 shows all classification results for each supervised learning algorithm for dataset Malware-Exploratory, Fig. 4 shows the clustering graphs for all unsupervised learning algorithms for dataset CIC-MalMem-2022, Table 2 shows all classification results for each supervised learning algorithm for dataset CIC-MalMem-2022. Fig. 5 will provide a wholistic view of both datasets under unsupervised learning, and Table 3 will provide a comparison of accuracy scores against both datasets.

A. Malware-Exploratory

The elbow graph produced 2 clusters for using the K-Means algorithm. For each clustering graph, the points remained consistent with regards to distance from the centroids. For supervised learning the test size was set at 20% and each algorithm used 28,096 benign samples, 15,175 malignant samples, for a total of 43,271 samples. The result break-down for accuracy are as follows: Decision Tree 99.97%, Random Forest 99.88%, Ada Boost 100%, KNeighbors 98.23%, Stochastic Gradient Descent 64.94%, Extra Trees 99.75%, and Gaussian Naïve Bayes 64.94%.

B. CIC-MalMem-2022

The elbow graph did not produce a definitive cluster, therefore a series of tests concluded that using 4 clusters showed the best results for the K-Means algorithm. For each clustering graph, the points remained consistent with regards to distance from the centroids. For supervised learning the test size was set at 20% and each algorithm used 5,889 benign samples, 5,831 malignant samples, for a total of 11,720 samples. The result break-down for accuracy are as follows: Decision Tree 99.99%, Random Forest 99.99%, Ada Boost 99.99%, KNeighbors 99.91%, Stochastic Gradient Descent 98.98%, Extra Trees 99.99%, and Gaussian Naïve Bayes 99.22%.

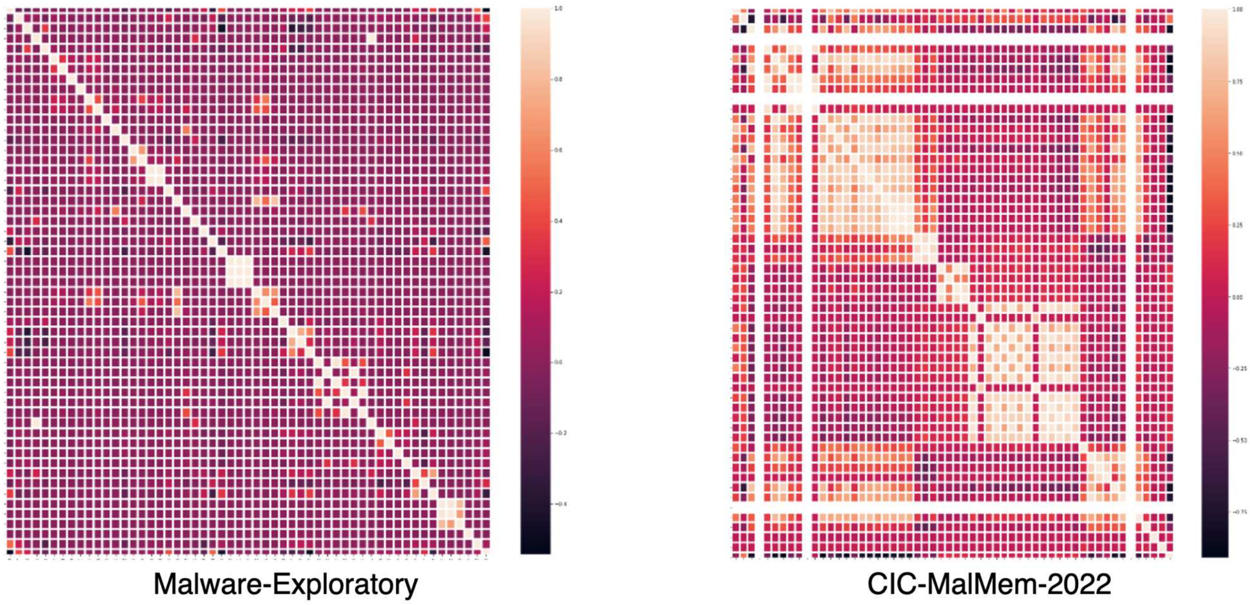

Fig. 2. Dataset heatmaps

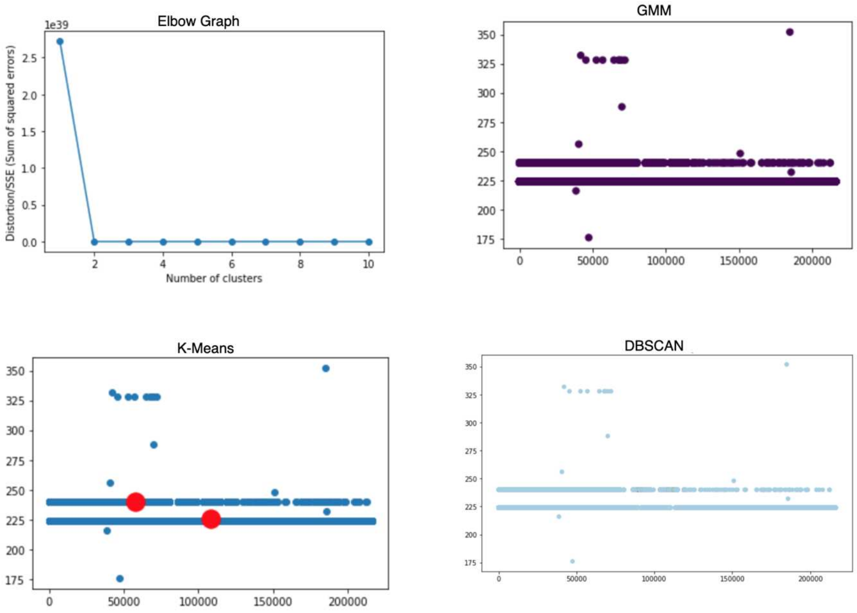

Fig. 3. Malware-Exploratory unsupervised learning results

TABLE I. MALWARE-EXPLORATORY CLASSIFICATION REPORT

Classifiers		***precision***	***recall***	***f1-score***	***support***	***accuracy***
Decision Tree	0	1.00	1.00	1.00	28096	99.97%
	1	1.00	1.00	1.00	15175	
	accuracy			1.00	43271	
	macro avg	1.00	1.00	1.00	43271	
	weighted avg	1.00	1.00	1.00	43271	
Random Forest	0	1.00	1.00	1.00	28096	99.98%
	1	1.00	1.00	1.00	15175	
	accuracy			1.00	43271	
	macro avg	1.00	1.00	1.00	43271	
	weighted avg	1.00	1.00	1.00	43271	
Ada Boost	0	1.00	1.00	1.00	28096	100%
	1	1.00	1.00	1.00	15175	
	accuracy			1.00	43271	
	macro avg	1.00	1.00	1.00	43271	
	weighted avg	1.00	1.00	1.00	43271	
KNeighbors	0	0.99	0.99	0.99	28096	98.23%
	1	0.97	0.98	0.97	15175	
	accuracy			0.98	43271	
	macro avg	0.98	0.98	0.98	43271	
	weighted avg	0.98	0.98	0.98	43271	
Stochastic Gradient Descent	0	0.65	1.00	0.79	28096	64.94%
	1	1.00	0.00	0.00	15175	
	accuracy			0.65	43271	
	macro avg	0.82	0.50	0.39	43271	
	weighted avg	0.77	0.65	0.51	43271	
Extra Trees	0	1.00	1.00	1.00	28096	99.75%
	1	1.00	1.00	1.00	15175	
	accuracy			1.00	43271	
	macro avg	1.00	1.00	1.00	43271	
	weighted avg	1.00	1.00	1.00	43271	
Gaussian Naïve Bayes	0	0.65	1.00	0.79	28096	64.94%
	1	1.00	0.00	0.00	15175	
	accuracy			0.65	43271	
	macro avg	0.82	0.50	0.39	43271	
	weighted avg	0.77	0.65	0.51	43271	

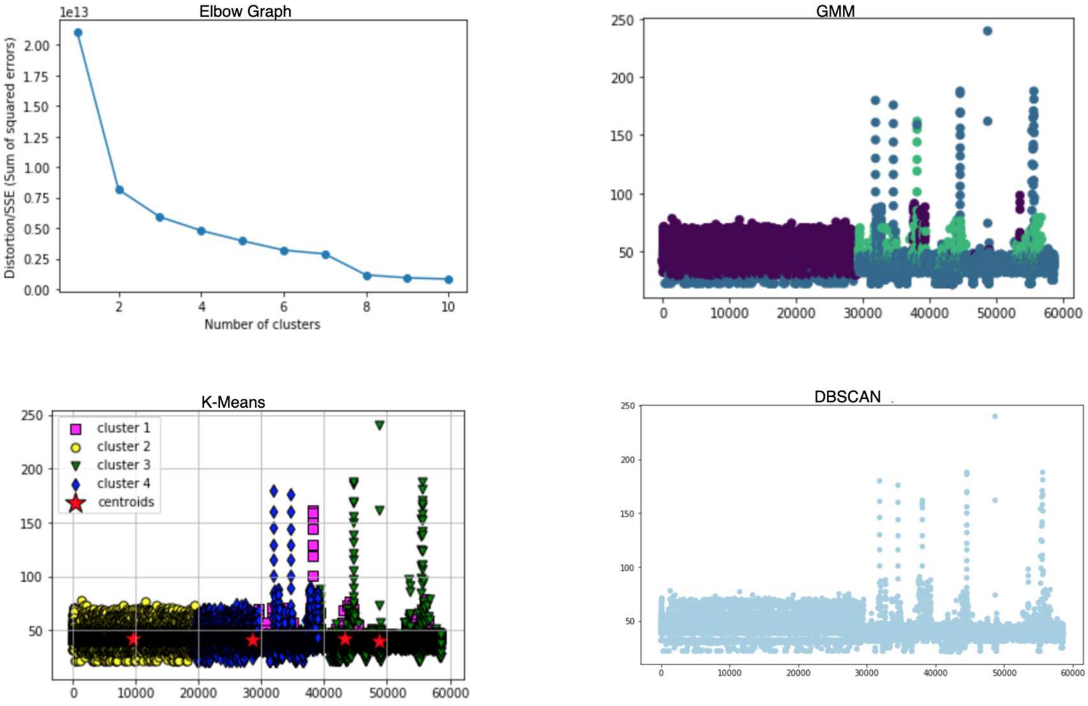

Fig. 4. CIC-MalMem-2022 dataset unsupervised learning results

TABLE II. CIC-MALMEM-2022 CLASSIFICATION REPORT

Classifiers		***precision***	***recall***	***f1-score***	***support***	***accuracy***
Decision Tree	1.0	1.00	1.00	1.00	5889	99.99%
	2.0	1.00	1.00	1.00	5831	
	accuracy			1.00	11720	
	macro avg	1.00	1.00	1.00	11720	
	weighted avg	1.00	1.00	1.00	11720	
Random Forest	1.0	1.00	1.00	1.00	5889	99.99%
	2.0	1.00	1.00	1.00	5831	
	accuracy			1.00	11720	
	macro avg	1.00	1.00	1.00	11720	
	weighted avg	1.00	1.00	1.00	11720	
Ada Boost	1.0	1.00	1.00	1.00	5889	99.99%
	2.0	1.00	1.00	1.00	5831	
	accuracy			1.00	11720	
	macro avg	1.00	1.00	1.00	11720	
	weighted avg	1.00	1.00	1.00	11720	
KNeighbors	1.0	1.00	1.00	1.00	5889	99.91%
	2.0	1.00	1.00	1.00	5831	
	accuracy			1.00	11720	
	macro avg	1.00	1.00	1.00	11720	
	weighted avg	1.00	1.00	1.00	11720	
Stochastic Gradient Descent	1.0	0.99	0.99	0.99	5889	98.98%
	2.0	0.99	0.98	0.99	5831	
	accuracy			0.99	11720	
	macro avg	0.99	0.99	0.99	11720	
	weighted avg	0.99	0.99	0.99	11720	
Extra Trees	1.0	1.00	1.00	1.00	5889	99.99%
	2.0	1.00	1.00	1.00	5831	
	accuracy			1.00	11720	
	macro avg	1.00	1.00	1.00	11720	
	weighted avg	1.00	1.00	1.00	11720	
Gaussian Naïve Bayes	1.0	1.00	0.99	0.99	5889	99.22%
	2.0	0.99	1.00	0.99	5831	
	accuracy			0.99	11720	
	macro avg	0.99	0.99	0.99	11720	
	weighted avg	0.99	0.99	0.99	11720	

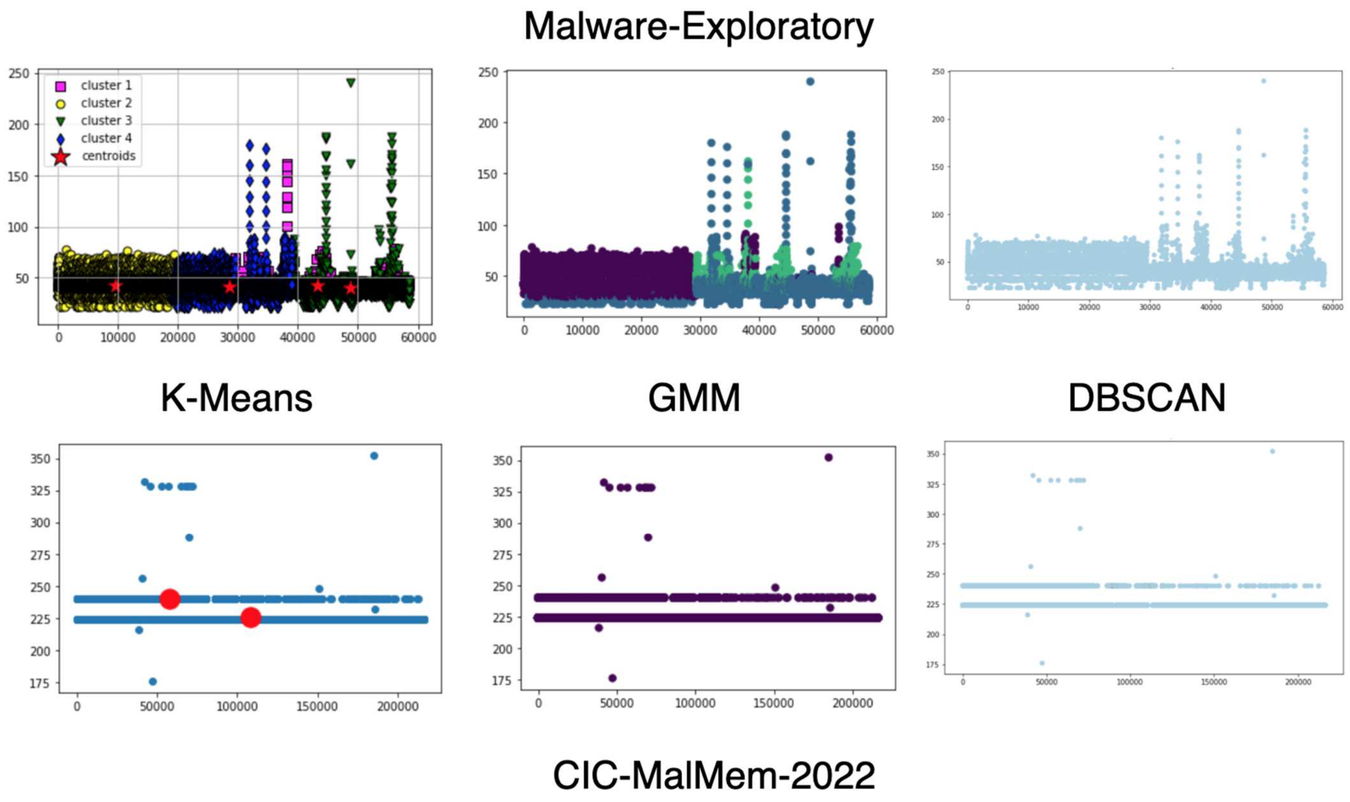

Fig. 5. Unsupervised Learning – wholistic view of clustering graphs

TABLE III. ACCURACY SCORE COMPARISON

Classifiers	***Malware-Exploratory***	***CIC-MalMem-2022***
Decision Tree	99.97%	99.99%
Random Forest	99.88%	99.99%
Ada Boost	100%	99.99%
KNeighbors	98.23%	99.91%
Stochastic Gradient Descent	64.94%	98.98%
Extra Trees	99.75%	99.99%
Gaussian Naïve Bayes	64.94%	99.22%

According to the heatmap, Malware-Exploratory dataset showed a far less correlation between its features, but predictions still obtained high accuracy for detecting malignant samples at an average of 90%. The heatmap for CIC-MalMem-2022 dataset showed extreme correlation between most features and likewise the accuracy scores are high, all being greater than 98%. Both datasets showed consistency across all clustering algorithms, as the graphs are very similar to one another.

V. Future Work

Next steps for each dataset are to perform feature selection against the most prominent labels using Pearson correlation coefficient. This algorithm is chosen due to measuring the association between variables of interest as it is based on the method of covariance. The same exact procedures will be done using the 3 clustering algorithms to test consistency across the graphs, followed by the 7 tests for supervised learning. Finally, these new datasets will be merged to form one complete dataset and again the same steps will be performed. The merged dataset will also be tested against a genetic algorithm. Fig. 6 provides a visual of these steps.

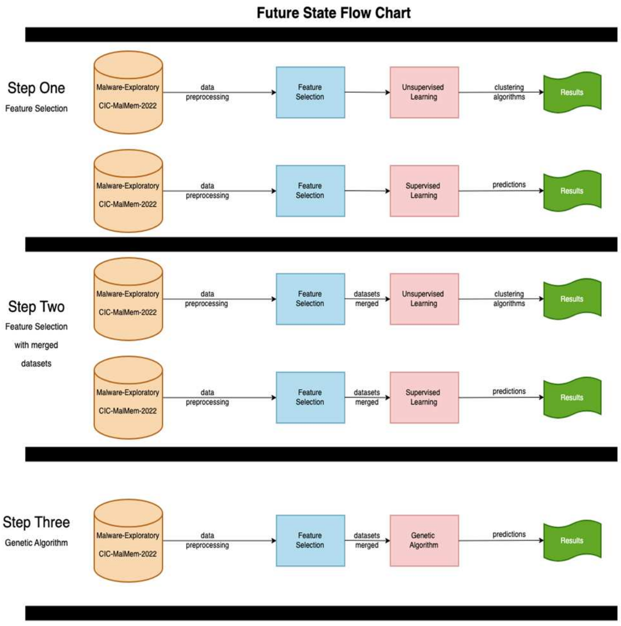

Fig. 6. Future State – steps and procedures

VI. Conclusion

Malware will continue to cause disruption to many different victims in the coming years, without any signs of slowing down. The sophistication of these malicious programs to go undetected against antivirus and monitoring tools are causing the cyber security communities to investigate the use of machine learning algorithms. In this study, a model is developed to cluster and detect malware datasets using unsupervised and supervised learning. Preliminary results have shown great success with accuracy scores well over 90% detection. Future studies will be conducted using feature selection against each dataset along with integrating a genetic algorithm.

References

[1] S. Cook, “Malware statistics in 2022: Frequency, impact, cost &amp; more,” Comparitech, 12-Oct-2022. [Online]. Available: https://www.comparitech.com/antivirus/malware-statistics-facts. [Accessed: 17-Nov-2022]

[2] U. Adamu and I. Awan, "Ransomware Prediction Using Supervised Learning Algorithms," 2019 7th International Conference on Future Internet of Things and Cloud (FiCloud), 2019, pp. 57-63, doi: 10.1109/FiCloud.2019.00016.

[3] F. A. Aboaoja, A. Zainal, F. A. Ghaleb, B. A. S. Al-rimy, T. A. E. Eisa, and A. A. H. Elnour, “Malware Detection Issues, Challenges, and Future Directions: A Survey,” Applied Sciences, vol. 12, no. 17, p. 8482, Aug. 2022, doi: 10.3390/app12178482. [Online]. Available: http://dx.doi.org/10.3390/app12178482

[4] D. Gibert, C. Mateu, and J. Planes, “The rise of machine learning for detection and classification of malware: Research developments, trends and challenges,” Journal of Network and Computer Applications, vol. 153. Elsevier BV, p. 102526, Mar. 2020 [Online]. Available: http://dx.doi.org/10.1016/j.jnca.2019.102526

[5] J. Lehr, J. Philipps, V. N. Hoang, D. von Wrangel, and J. Krüger, “Supervised learning vs. unsupervised learning: A comparison for optical inspection applications in quality control,” IOP Conference Series: Materials Science and Engineering, vol. 1140, no. 1. IOP Publishing, p. 012049, May 01, 2021. doi: 10.1088/1757-899x/1140/1/012049

[6] D. Petersson, “What is supervised learning?,” SearchEnterpriseAI, 26-Mar-2021. [Online]. Available: https://www.techtarget.com/searchenterpriseai/definition/supervised-learning. [Accessed: 17-Nov-2022]

[7] IBM Cloud Education, “What is unsupervised learning?,” IBM, 21-Sep-2020. [Online]. Available: https://www.ibm.com/cloud/learn/unsupervised-learning. [Accessed: 18-Nov-2022]

[8] R. Burton, “Unsupervised Learning Techniques for Malware Characterization,” Digital Threats: Research and Practice, vol. 1, no. 3. Association for Computing Machinery (ACM), pp. 1–26, Sep. 30, 2020 [Online]. Available: http://dx.doi.org/10.1145/3377869

[8] C. Manzano, C. Meneses, P. Leger, and H. Fukuda, “An Empirical Evaluation of Supervised Learning Methods for Network Malware Identification Based on Feature Selection,” Complexity, vol. 2022. Hindawi Limited, pp. 1–18, Apr. 07, 2022 [Online]. Available: http://dx.doi.org/10.1155/2022/6760920

[9] K. O. Babaagba and S. O. Adesanya, “A Study on the Effect of Feature Selection on Malware Analysis using Machine Learning,” Proceedings of the 2019 8th International Conference on Educational and Information Technology. ACM, Mar. 02, 2019 [Online]. Available: http://dx.doi.org/10.1145/3318396.3318448

[10] J. S. P. de Wit, D. Bucur, and J. van der Ham, “Dynamic Detection of Mobile Malware Using Smartphone Data and Machine Learning,” Digital Threats: Research and Practice, vol. 3, no. 2. Association for Computing Machinery (ACM), pp. 1–24, Jun. 30, 2022 [Online]. Available: http://dx.doi.org/10.1145/3484246

[11] L. Borges, “Malware-exploratory,” Kaggle, 01-Aug-2021. [Online]. Available: https://www.kaggle.com/code/lucaslba/malware-exploratory/data. [Accessed: 20-Jul-2022]

[12] T. Carrier, P. Victor, A. Tekeoglu, and A. Lashkari, “Detecting Obfuscated Malware using Memory Feature Engineering,” Proceedings of the 8th International Conference on Information Systems Security and Privacy. SCITEPRESS - Science and Technology Publications, 2022 [Online]. Available: http://dx.doi.org/10.5220/0010908200003120