An excellent overview of machine learning techniques used so far for the analysis of portable executables in Windows environment can be found in [2-4, 9]. The survey in [2-4, 9] provides a detailed overview of the way machine learning has been used so far in the context of malware analysis using Portable Executables by systemising the surveyed papers according to the detection approach, their objectives, feature set, the machine learning techniques employed and the accuracy obtained. Authors in [2] additionally introduce the novel concept of malware analysis economics, regarding the study of existing trade-offs among key metrics, such as analysis accuracy and economical costs. The work in [3,4], provides a detailed distinction between signature-based and behaviour-based malware detection and the approaches used for each technique. Authors in [9] state the importance of the intersection of the features extracted and the classification techniques for the detection of malware and provide a comprehensive investigation on both the feature extraction and the classification/clustering techniques.

Current literature, as covered by [2-4,9], explain the use of various machine and deep learning techniques for the application of malware detection and classification, each varying in the machine learning technique employed and features extracted. Authors in [xx1], extract static features of the PE header to perform malware detection using SVM, Logistic Regression, Random Forest, Extreme Gradient Boosting, and custom hybrid classifiers. They achieved an accuracy of 0.915 using a hybrid classifier on a dataset of 427 malicious PE files and 989 benign PE files. Yewale et al. in [xx2], introduce a method to detect malware using the concept of opcode frequency in the portable exeutable format. With the use of Random Forest, an accuracy of 97% was achieved on a dataset of 100 PE files. Additionally, this paper asserts that 20 most frequently used Opcodes are sufficient for malware detection. Kolosnjaji et. al. [10] have used a neural network based on convolutional and recurrent network layers to obtain the best features for classification using system call sequences. This way a hierarchical feature extraction architecture is generated that combines convolution of n-grams with full sequential modelling. Authors in [xx4], exploit machine learning algorithms such as Naive Bayes, SVM and Decision Tree on a large PE file dataset for the application of malware detection. The paper rests on the analysis of the Windows APIs called by PE files using Objective-Oriented Association (OOA) mining based classification.

The authors in [8], present a novel approach to understand the similarity between malware variants using hashing. The use of hashing leads to dimensionality reduction enabling it for large-scale clustering. While, the paper does not directly employ modern day machine learning techniques, it introduces the idea of similarity in malware structure. To further push the concept of similarity in malware, the use of social network analysis has been proposed by Kim et.al [6] for android malware classification. This work, while not on Windows malware, provides an excellent insight to bring together seemingly disparate branches - malware detection, social network analysis and natural language processing. They have proposed a dynamic malware analysis method that uses Natural Language Processing (NLP) concepts on API system calls and has shown that Linear Support Vector Machines (SVM) optimized by Stochastic Gradient Descent and the traditional Coordinate Descent on the Wolfe Dual form of the SVM are effectively achieving an accuracy as high as 96% with 95% recall score. The integration of social network analysis-based malware family classification has been further enriched in [7] by Jang et. al. The use of social network analysis for detection of Android malware has been performed in [11-14]. The use of community-based feature extraction techniques used here has a direct implication on our motivation, as no such technique has been applied to detect Windows malware defined in a PE32 file.

The application of concepts of graph theory in malware detection has been well established in literature. Authors in [xx7] give a detailed review of the affect of use of malware graphs for malware detection. Elhadi et. al. [xx6] transform a malware file into a simplified data dependent graph. Graph matching algorithm is then used to calculate similarity between the input samples. Graph matching algorithm based on Longest Common Subsequence (LCS) algorithm is employed to demonstrate 98% malware detection rate and 0% false positive rate when applied on 85 files. Chau et. al. in [xx5] detects malware through large-scale graph inference with a high true positive rate of 87%. The authors have successfully evaluated their model on a billion-node graph constructed based on a scalable Belief Propagation algorithm. Apart from system call or API graphs, a novel use of file relation graphs has been proposed in [xx3]. In this work, in addition to the use of file content extracted from the file samples, a study of the use of file relation graphs, for malware detection has been done with the aid of a novel Belief Propagation algorithm based on the constructed graphs to detect newly unknown malware.

xx1. Balram, N., Hsieh, G., McFall, C.: Static Malware Analysis Using Machine Learning Algorithms on APT1 Dataset with String and PE Header Features. International Conference on Computational Science and Computational Intelligence(CSCI) 2019, pp. 90-95

xx2. Yewale, A. and Singh, M.: Malware detection based on opcode frequency. International Conference on Advanced Communication Control and Computing Technologies (ICACCCT). 2016. IEEE.

xx3. Chen, L., Li, T., Abdulhayoglu, M., Ye, Y.: Intelligent malware detection based on file relation graphs. Proceedings of the 2015 IEEE 9th International Conference on Semantic Computing (IEEE ICSC 2015). 2015. IEEE.

xx4. Ye, Y., Wang, D., Li, T., Ye, D., Jiang, Q.: An intelligent PE-malware detection system based on association mining. J Comput Virol **4,**323–334 (2008). https://doi.org/10.1007/s11416-008-0082-4

xx5. Chau, D. H., Nachenberg, C., Wilhelm, J., Wright, A., Faloutsos, C.: Polonium: Tera-Scale Graph Mining and Inference for Malware Detection. Proceedings of the 2011 SIAM International Conference on Data Mining. 2011, 131-142

xx6. Elhadi, A., Maarof, M., Barry, B.: Improving the Detection of Malware Behaviour Using Simplified Data Dependent API Call Graph. International Journal of Security and Its Applications Vol.7, No.5 (2013), pp.29-42.

xx7. Sharma, A. and Prakash, A.: Graphs for Malware Detection: The Next Frontier. Proceedings of the 13th International Workshop on Mining and Learning with Graphs (MLG), pp.8–10, 2017.