# Evolutionary based Generative Adversarial Learning Approach to Metamorphic Malware Detection by Dr. Kehinde O. Babaagba (k.babaagba@napier.ac.uk)

**Total SICSA contribution received** - £1200

**Spending breakdown** – Summer studentship (Jordan Wylie) - £900 (paid), online conference registration - £300 (to be paid as paper is being prepared for submission on the 26th of September 2022 and conference registration will be for March 2023).

The project details are given below:

1. **Project Overview and Initial Setup**
The goal of this project was to generate evasive metamorphic malware. In order to do this a closed-loop system was designed and implemented in form of a Generative Adversarial Network (GAN). The GAN generated metamorphic Android malware using a single objective evolutionary algorithm (EA) which was previously implemented in our paper in [5]. The sample was passed on to a trained discriminator, which aimed to identify if the generated malware was
either benign or malicious, if it guessed correctly, it got sent back to the single-objective EA.

The project involved the use of malware samples from the Contagio Android malware dump as well as some initial code which contained various functionality including a single objective EA and the Android malware mutation
operators from [2] and [5]. The research work was motivated by the work of [1] who implemented an evolutionary based GAN using an EA as the generator but for image datasets. The methodology of this project involved the implementation and identification of elements of the previous work in [2, 5] that needed to be changed as well as the extension of the previous work to include a GAN.

1. **Implementation**

The implementation involved reusing the existing single-objective EA and mutation operators’ implementation from [5] while making some modifications in the creation of the malware mutants to allow them to morph dynamically. This served as the generator. For the discriminator, 3 different models were chosen as the discriminators, these were Naive Bayes (NB), Long short-term memory (LSTM), and a Deep Neural Network (DNN). The NB and LSTM models were chosen as they were shown to perform well during the experiments carried out in [2]. The DNN
was chosen as this more closely relates to how GANs are implemented. Each of these discriminators were trained to predict a sample’s feature set. At this point, most of the logic for the closed-loop system was also implemented. The next part was to implement the feature extraction and dataset generation and loading. As dis-
cussed in [2] there were two types of features - sequential and non-sequential. In order to generate the non-sequential features, a modified version of the behavioural similarity bash script was created, which just returns the non-sequential features (a count of each system call). The sequential features were generated by reading the system call log generated by strace. The first line in this log was skipped as it never contains a system call. For each line after that the first 12 characters were automatically discarded. The remaining string was then split on the “(” character and the first split was taken. With the functions implemented to extract the sequential features, the functions to obtain the dataset were then implemented. In order to generate the dataset, two functions were created for the respective feature types. These functions looped through two directories, one contained benign samples and the other malware samples. Every sample in those directories were then launched and the features were extracted. Then, the class value of the features was inserted, if the sample was benign a 0 was added, if it was malware a 1 was added. These features were then written to a CSV file which could be loaded using two other functions (depending on the feature type). To evaluate the presented approach, both quantitative and qualitative evaluation methods were used. The quantitative evaluation involved implementing quantitative evaluation functions which calculated the average adversarial accuracy (i.e., the average percentage of anti-virus detection engines that detected the generated samples) as well as a graph showing how many samples bypassed each detection engine.

1. **Testing and Evaluation**

The testing was done which involved testing each component of the GAN. A number of issues that arose with the discriminators were resolved which will be explained fully in the research paper produced from this work. Some research work helped with resolving some of these issues such as [3, 4]. In terms of evaluation an evaluation class was created for the quantitative and qualitative evaluation. The quantitative evaluation has been discussed in section 2, however for the qualitative evaluation, 4 algorithms (k-Nearest Neighbours, Multi-layer Perceptron, Classification and Regression Trees, and Support Vector Machines) were chosen and trained. The set of APKs were then used as testing data so that various evaluation metrics could be calculated, such as accuracy, F1 score, and AUC. The results showed that an EA can be used in the generation and detection of metamorphic malware and the full experimentation details would be provided in the research paper produced from this work.

1. **Potential Impact of the Project**

The potential impact of this project as highlighted in the proposal, and which is in alignment with the SICSA challenge areas are centered around the protection of AI systems. I had previously established that across all the SICSA challenge areas, there is utilization and development of Artificial Intelligence systems such as AI driven healthcare solution among others. It is therefore crucial to protect these systems against malicious attack. Developing AI systems without proper defense mechanism against malware could compromise their utility and functionality, hence the impact of this project is in protecting AI systems from complex families of malware such as metamorphic malware that change their form between generations.

**REFERENCES**

[1] C. Wang, C. Xu, X. Yao, and D. Tao, “Evolutionary generative adversarial networks,” IEEE Transactions on Evolutionary Computation, vol. 23, no. 6, pp. 921–934, 2019.
[2] K. O. Babaagba, Z. Tan, and E. Hart, “Improving classification of metamorphic malware by augmenting training data with a diverse set of evolved mutant samples,” in 2020 IEEE Congress on Evolutionary Computation (CEC), 2020, pp. 1–7.
[3] Z. C. Lipton, J. Berkowitz, and C. Elkan, “A critical review of recurrent neural networks for sequence learning,” 2015. [Online]. Available: https://arxiv.org/abs/1506.00019.
[4] W. Xie, S. Xu, S. Zou, and J. Xi, “A system-call behavior language system for malware
detection using a sensitivity-based lstm model,” in Proceedings of the 2020 3rd International Conference on Computer Science and Software Engineering, ser. CSSE 2020. New York, NY, USA: Association for Computing Machinery, 2020, p. 112–118. [Online]. Available: https://doi.org/10.1145/3403746.3403914.
[5] K. O. Babaagba, Z. Tan, and E. Hart, “Nowhere metamorphic malware can hide - a biological evolution inspired detection scheme,” in Dependability in Sensor, Cloud, and Big Data Systems and Applications, G. Wang, M. Z. A. Bhuiyan, S. De Capitani di Vimercati, and Y. Ren, Eds. Singapore: Springer Singapore, 2019, pp. 369–382.