Enhancing the Robustness of AI-Based Malware Detection: A Hybrid Approach Combining Static and Dynamic Analysis to Defend Against Adversarial Evasion Attacks

MSc Research Project

MSc in Cybersecurity

Shreya Upasani
X22201068

School of Computing
National College of Ireland

Supervisor: Jawad Salahuddin

National College of Ireland

Project Submission Sheet

Student Name: Shreya Mangesh Upasani

Student ID: X22201068

Programme: Msc in Cybersecurity **Year:** 2024-25

Module: Practicum 2

Lecturer: Jawad Salahuddin

Submission Due Date: 15/09/2025

Project Title: Enhancing the Robustness of AI-Based Malware Detection: A Hybrid Approach Combining Static and Dynamic Analysis to Defend Against Adversarial Evasion Attacks

Word Count: 800

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the references section. Students are encouraged to use the Harvard Referencing Standard supplied by the Library. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action. Students may be required to undergo a viva (oral examination) if there is suspicion about the validity of their submitted work.

Signature: Shreya Mangesh Upasani

Date: 15/09/2025

PLEASE READ THE FOLLOWING INSTRUCTIONS:

1. Please attach a completed copy of this sheet to each project (including multiple copies).
2. Projects should be submitted to your Programme Coordinator.
3. **You must ensure that you retain a HARD COPY of ALL projects**, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer. Please do not bind projects or place in covers unless specifically requested.
4. You must ensure that all projects are submitted to your Programme Coordinator on or before the required submission date. **Late submissions will incur penalties.**
5. All projects must be submitted and passed in order to successfully complete the year. **Any project/assignment not submitted will be marked as a fail.**

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

1. Introduction

This configuration manual provides all required steps to set up, configure and execute the static, dynamic and hybrid malware detection models developed for the MSc Research Project *"Enhancing the Robustness of AI-Based Malware Detection: A Hybrid Approach Combining Static and Dynamic Analysis to Defend Against Adversarial Evasion Attacks"*.

It is intended for use by third parties wishing to reproduce the results or extend the work. The instructions cover environment setup, dataset placement, code execution and output verification.

2. System Requirements

Hardware:

- Minimum 8 GB RAM (16 GB recommended for faster processing)
- At least 10 GB free disk space
- CPU: Quad-core (Intel i5 or higher recommended)
- Optional: GPU for neural network training (NVIDIA CUDA-capable)

Software:

- Operating System: Windows 10/11, macOS, or Linux
- Python 3.9 or higher
- Jupyter Notebook
- Required Python libraries (installed via pip): pip install pandas numpy scikit-learn matplotlib xgboost tensorflow

3. Setup Instructions

1. **Install Python and Jupyter Notebook**
    - Install Anaconda (recommended) or standalone Python 3.9+.
    - If using standalone Python, install Jupyter with: pip install notebook
2. **Install Required Libraries**
   Run: pip install pandas numpy scikit-learn matplotlib xgboost tensorflow
3. **Download & Place Datasets**
    - Place the CIC folder (containing output1.csv, output2.csv, output3.csv) inside the main project folder.
    - Place the ember2018 folder (containing all train_features_*.jsonl and test_features.jsonl) in the main project folder.

4. Running the Models

A. Static Model

1. Open Static code.ipynb in Jupyter Notebook.
2. Execute cells sequentially:
    - **Data Loading:** Loads EMBER JSONL files into DataFrames.
    - **Preprocessing:** Flattens features, removes low-variance columns.
    - **Training:** Runs Logistic Regression, Random Forest, and XGBoost.
    - **Evaluation:** Outputs accuracy, precision, recall, F1-score, ROC-AUC, and confusion matrices.
3. Adversarial perturbations will be applied automatically in the later cells and results will be displayed.

B. Dynamic Model

1. Open Dynamic code.ipynb.
2. Execute cells sequentially:
    - **Data Loading:** Reads output1.csv, output2.csv and output3.csv.
    - **Synthetic Benign Generation:** Creates benign-like samples for balance.
    - **Training:** Runs Logistic Regression, Random Forest and XGBoost.
    - **Evaluation:** Outputs metrics and adversarial robustness results.

C. Hybrid Model

1. Open Hybrid code.ipynb.
2. Execute cells sequentially:
    - **Data Loading:** Loads both static and dynamic datasets.
    - **Early Fusion:** Merges features into combined vectors.
    - **Training:** Runs Logistic Regression, Random Forest, XGBoost and MLP.
    - **Adversarial Testing:** Runs both targeted perturbations and FGSM for MLP.
3. View printed metrics and saved plots.

5. Expected Outputs

- **Performance Metrics:** Accuracy, precision, recall, F1-score, ROC-AUC for each model and configuration.
- **Confusion Matrices:** Displayed in the notebook outputs.
- **ROC Curves:** Saved as PNGs in the working directory (if implemented).
- **Adversarial Results:** Performance drops compared to clean-data baselines.

6. Troubleshooting

- **Memory Errors:** Reduce dataset size by sampling fewer rows in the data loading step.
- **Missing Packages:** Install them via pip install <package_name>.
- **Path Errors:** Ensure dataset file paths match the ones in the code.

References

- Anderson, H.S. and Roth, P., 2018. *EMBER: An open dataset for training static PE malware machine learning models*. arXiv preprint arXiv:1804.04637. Available at: https://arxiv.org/abs/1804.04637 [Accessed 8 August 2025].
- Damodaran, A., Troia, F.D., Visaggio, C.A., Austin, T.H. and Stamp, M., 2022. *A comparison of static, dynamic, and hybrid analysis for malware detection*. arXiv preprint arXiv:2203.09938. Available at: https://arxiv.org/abs/2203.09938 [Accessed 6 August 2025].
- Yen, T.F., Xie, Y., Yu, M. and Yu, Z., 2019. *Integration of static and dynamic analysis for malware family classification with composite neural network*. arXiv preprint arXiv:1912.11249. Available at: https://arxiv.org/abs/1912.11249 [Accessed 10 August 2025].
- Ngo, H.Q., Pham, T.H., Nguyen, V.H., Le, T.D., Nguyen, T.L., Huynh, V.N. and Dang, T.K., 2022. *Fast and efficient malware detection with joint static and dynamic features through transfer learning*. arXiv preprint arXiv:2211.13860. Available at: https://arxiv.org/abs/2211.13860 [Accessed 5 August 2025].
- Sihwail, R., Omar, K. and Ariffin, K.A.Z., 2019. *A survey on malware analysis techniques: Static, dynamic, hybrid and memory analysis*. *International Journal of Advanced Computer Science and Applications*, 10(8), pp.432–445. Available at: https://www.researchgate.net/publication/328760930_A_Survey_on_Malware_Analysis_

Techniques_Static_Dynamic_Hybrid_and_Memory_Analysis [Accessed 10 August 2025].