***Malware Detection Using Memory Dump Analysis with Deep Learning (CIC-MalMem-2022 Dataset)***

**1. PROJECT OVERVIEW**

Malware is still a big problem in cybersecurity. Attackers keep coming up with clever ways to slip past the usual file-based detection methods. This project looks into detecting malware by checking out memory dump features, which hasn't been explored much but could be a good way to spot malware in action (Ayan et al., 2021; Bozkir et al., 2021; Shah et al., 2022; Zhang et al., 2023). We’re using the CIC-MalMem-2022 dataset, which has detailed memory recordings from many types of malwares and regular software, allowing us to use machine learning to find harmful patterns in how memory behaves during runtime.

Static analysis can be easily avoided with techniques like packing or encryption, but checking memory allows us to see what’s happening in real time. We’ll be using and comparing both traditional machine learning methods (like Random Forest and SVM) and deep learning models (like CNN and LSTM) to pull out and assess important features. This project also aims to meet the growing need for quick detection and to deal with tricky malware types like MAZE and Emotet.

**Research Questions:**

1. Can deep learning models (like CNN and LSTM) trained on memory dump features find malware more accurately than traditional machine learning models (like Random Forest and SVM)?
2. How do techniques like feature selection, scaling, and transformation impact the success of machine learning and deep learning models in detecting malware based on memory?
3. What kind of performance boosts can we expect by fine-tuning deep learning models for detecting malware in memory?

**Project Objectives:**

1. To process and pull-out relevant features from the CIC-MalMem-2022 dataset.
2. To train and assess traditional machine learning models (like Random Forest and SVM) using those features.
3. To train and improve deep learning models (like CNN and LSTM) on the same dataset.
4. To compare how traditional and deep learning models perform against different malware types.
5. To pick the top model based on accuracy, precision, recall, and F1-score.

**References**

Ayan, E., Karabulut, B., Ünver, H.M., 2021. Diagnosis of Pediatric Pneumonia with Ensemble of Deep Convolutional Neural Networks in Chest X-Ray Images. Arab J Sci Eng 47, 2123. https://doi.org/10.1007/S13369-021-06127-Z

Bozkir, A.S., Tahillioglu, E., Aydos, M., Kara, I., 2021. Catch them alive: A malware detection approach through memory forensics, manifold learning and computer vision. Comput Secur 103, 102166. https://doi.org/10.1016/J.COSE.2020.102166

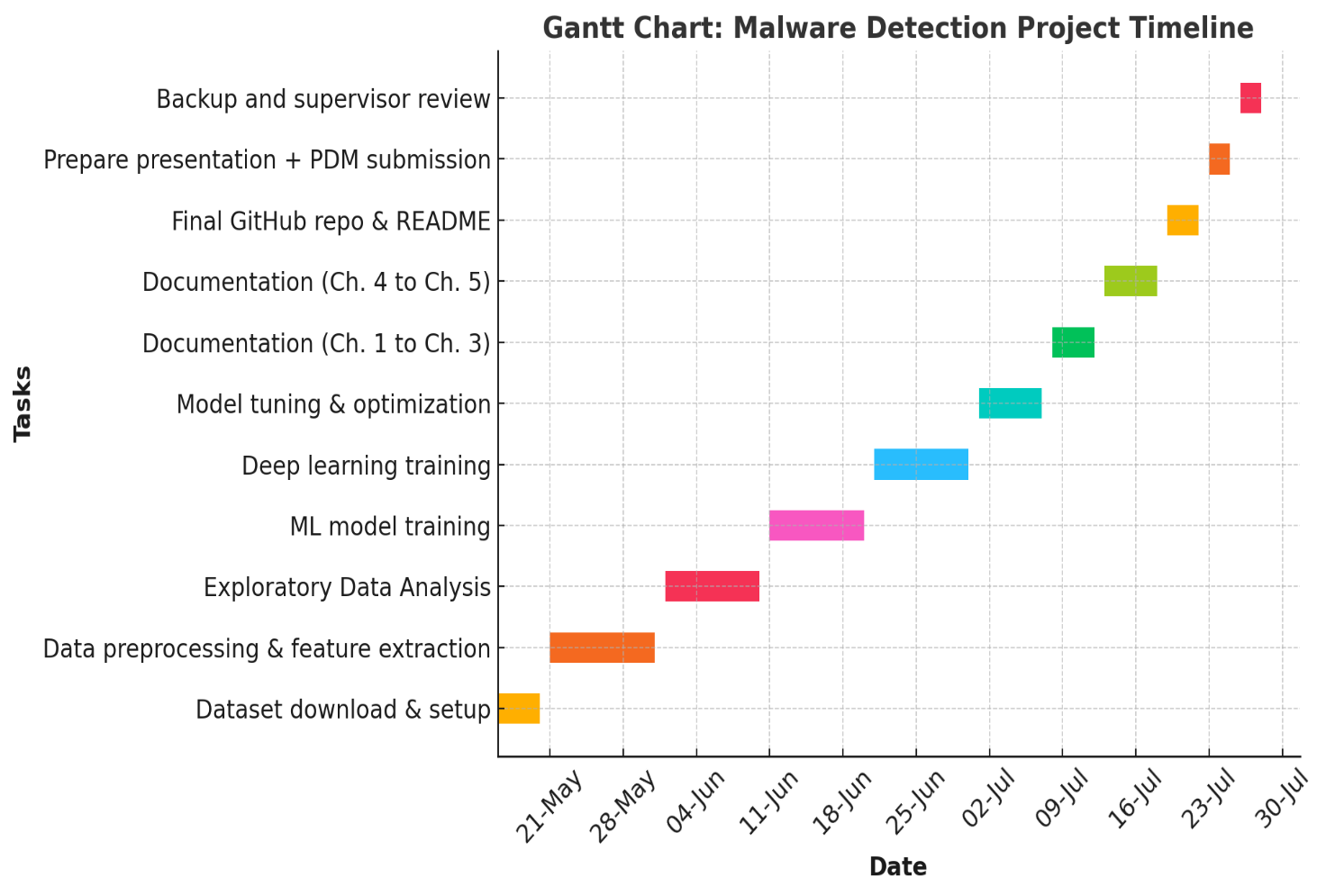
Shah, S.S.H., Ahmad, A.R., Jamil, N., Khan, A. ur R., 2022. Memory Forensics-Based Malware Detection Using Computer Vision and Machine Learning. Electronics 2022, Vol. 11, Page 2579 11, 2579. https://doi.org/10.3390/ELECTRONICS11162579

Zhang, S., Hu, C., Wang, L., Mihaljevic, M.J., Xu, S., Lan, T., 2023. A Malware Detection Approach Based on Deep Learning and Memory Forensics. Symmetry 2023, Vol. 15, Page 758 15, 758. https://doi.org/10.3390/SYM15030758

**2. PROJECT PLAN: TASK LIST AND TIMELINE**

|  |  |  |
| --- | --- | --- |
| **Date Range** | **Task Name** | **Description** |
| 16 May – 20 May | Dataset download & setup | Download CIC-MalMem-2022 dataset and prepare Google Colab + GitHub |
| 21 May – 31 May | Data preprocessing & feature extraction | Parse memory dumps, clean data, and engineer ML-ready features |
| 01 Jun – 10 Jun | Exploratory Data Analysis | Analyze feature distributions, correlations, and class imbalances |
| 11 Jun – 20 Jun | ML model training | Train and validate baseline models (SVM, RF, XGB) |
| 21 Jun – 30 Jun | Deep learning training | Train and evaluate CNN and LSTM models |
| 01 Jul – 07 Jul | Explainability and model tuning | Apply SHAP or LIME for interpretability, tune best model |
| 08 Jul – 12 Jul | Draft documentation Ch.1 to Ch.3 | Write Introduction, Literature Review, Methodology |
| 13 Jul – 18 Jul | Draft documentation Ch.4 to Ch.5 | Write Results, Conclusion, Ethical Discussion |
| 19 Jul – 22 Jul | Finalize GitHub code + README | Polish GitHub repo, add instructions, diagrams |
| 23 Jul – 25 Jul | Prepare Presentation + PDM PDF | Finalize slides, submit PDM with proposal and GitHub repo |
| 26 Jul – 28 Jul | Backup, reflection, supervisor review | Save backups, reflect on feedback, conduct supervisor Q&A |

**GANTT CHART**

****

**3. DATA MANAGEMENT PLAN**

**Dataset Overview**

The CIC-MalMem-2022 dataset is a solid memory forensic dataset put together by the Canadian Institute for Cybersecurity (CIC). It includes memory data from six types of malware (like Emotet, Conti, and AgentTesla) alongside benign software, all recorded on 8GB RAM Windows VMs. Each sample features over 200 behavioral traits taken from memory dumps, making it great for studying malware detection during runtime.

**Data Collection**

We downloaded the dataset directly from the CIC website: <https://www.unb.ca/cic/datasets/malmem-2022.html>. We didn’t use any third-party versions, so the data is original and in line with academic licensing.

**Metadata**

The dataset is in .csv format with about 80,000 labeled records and around 210 features, totaling an estimated size of ~1.5 GB after parsing. The code will be in .ipynb and .py formats, taking up about ~50 MB.

**Document Control**

GitHub Repository: https://github.com/yourusername/malware-memory-detection Commits will happen weekly, with clear messages (like feature: added CNN model or fix: resolved missing labels). Files will be named logically and follow versioning like v1.0, v1.1, etc.

**ReadMe File**

This GitHub repo covers a malware detection project that looks at memory dumps using the CIC-MalMem-2022 dataset. It walks you through everything from prepping the data to training and testing the models. You’ll find both traditional machine learning methods like Random Forest and SVM, as well as deep learning models such as CNN and LSTM, with a comparison of their performance. The dataset comes straight from the Canadian Institute for Cybersecurity. There are easy-to-follow instructions for running the code, and you'll also see outputs like accuracy stats and graphs. If you have any questions or want to collaborate, you can find the contact info in the repo.

**Security and Storage**

All code and data will be stored in Google Drive (with auto-sync) and GitHub (as a public repository). We’ll make weekly local backups and share all submission links with supervisors and examiners.

**Ethical Requirements**

1. **Does the data meet GDPR standards?**

No, the dataset doesn’t have personal or identifiable information; it's all synthetic and anonymized.

1. **Does the project follow UH ethical guidelines?**

Yes, it uses open research data purely for academic purposes, in line with UH policies.

1. **Do we have permission to use this data for our research?**

Yes, the dataset is available under a research-friendly license with no commercial use restrictions.

1. **Was the data collected ethically?**

Yes, it was created by cybersecurity experts at CIC for academic analysis.