National College of Ireland

Robust Deep Learning Models for Multi-Source Malware Detection Through Stacking-Based Model Integration

MSc Research Project

MSc in Cybersecurity

Rony Shaji

Student ID: 23398744

School of Computing

National College of Ireland

Supervisor: Kamil Mahajan

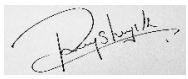
National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Rony Shaji		
Student ID:	23398744		
Programme:	MSc in Cybersecurity	**Year:**	2025
Module:	Practicum		
Supervisor:	Kamil Mahajan		
Submission Due Date:	11/12/2025		
Project Title:	Practicum		
Word Count:	7284	**Page Count:**	22

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 bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: 

Date: 11/12/2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

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

Robust Deep Learning Models for Multi-Source Malware Detection Through Stacking-Based Model Integration

Rony Shaji

23398744

Abstract

As malicious software grows increasingly sophisticated, the use of obfuscation, polymorphism, and memory residency makes it harder for classical detection based on either static or one-source-based analysis. This study addresses the need for improved and generalized approaches to detection by introducing the Integrated Learning Framework for Deep Learning that integrates all types of memory-forensics feature sets. In the CIC-MalMem-2022 competition, the research proposes the use of two customized learning models, CNN-MLP for the structural behavioral characteristics and FT-Transformer for the contextual dependencies based on the Win-DLL and combines their predictions using logistic regression-based model stacking. The experimentally validated findings show the superior performance capability of the individual base learner and the modest but significant improvement potential of the proposed Integrated learning. Interesting SHAPTAIL explainer analyses confirm the well-balanced contributions of both learners for promoting model understandability and applicability. These experimental insights suggest that Integrated learning, based on multiple sources, can enhance the performance capability of malicious software detection, especially for dynamic and memory-centric scenarios. Future research can take into consideration robustness evaluation against malicious attempts and the use of varied forensic corpora, since enhanced external validity is warranted.

Keywords: *Malware Detection; Deep Learning; Integrated Learning; Memory Forensics; Explainable AI.*

1 INTRODUCTION

1.1 Background and Context

Malware is now among the most critical problems in the modern technological environment. The advanced evasion techniques employed by attackers have been increasing over the last decade and they have compromised the traditional modes of defence. Classical analysis techniques have become less effective, and the contemporary malware conceals its functionality in the form of encoding, encrypting, and polymorphic code that hides the malicious activity without interrupting the execution (Inayat et al., 2021).

Dynamic analysis is more behaviour oriented but also has its own shortcomings. Malware that discovers the presence of a sandboxed environment can detect this environment and bypass. It is also aggravated by the emergence of fileless malware which only resides in volatile memory and leaves no disk traces. According to Kara (2023), process injection, rogue loading, and handle manipulation are some of the techniques that render disk-based forensics useless and redirect attention to memory-based analysis.

The study of memory forensics has thus become an important part of malware investigation in recent times. Process lists (ps_list), loaded DLLs (dll_list), and process or network handles are some of the features that constitute a full behavioural profile of the activity in the system. Since these aspects are hard to hide without compromising the functionality of these systems, then they provide good indication of malicious behaviour. Further developments in machine learning and deep learning have reinforced this sector by offering the development of models that can be used to process high dimensional memory data. CNNs are effective on structured tabular or reshaped memory features, and Transformers pick up global dependencies on a wide range of characteristics (Demirkiran, 2022). Nonetheless, the available studies remain mostly based on designs of single paths models.

As multi-vector, multi-family, memory-resident malware is becoming more prevalent, it is becoming necessary to have solutions that are in line with the complexity and evasiveness of modern threats. A combination of memory forensics and built-in deep learning is a promising path to go, with complementary feature modelling and greater robustness.

1.2 Problem Statement

Although there is an increasing number of studies concerning the use of machine learning-based malware detection, there are some important limitations that are making the applicability of existing solutions difficult. Most of the existing approaches are based on one kind of representation, such as API calls, binary images, and attributes of the PE header, which makes it hard for them to deal with new kinds of malicious software that manipulate only some characteristics of their behavior. Models trained on limited modalities are ineffective for dealing with memory resident, fileless malware.

Moreover, the inability of high-capacity deep learning models to provide interpretability makes it challenging for the investigator to provide sound reasons for their classifications. Integrated learning approaches have shown their efficacy for intrusion detection and PE Malware classifications but are not much utilized for memory forensics. In fact, most

research works proved ineffective as they ignored the fact that their proposed model would be effective but unable if they are exposed to novel malware.
The other highly important research gap is resilience to adversarial environments. The opponent can exploit memory properties to a small extent to mislead the classifier and the learning model that is not a robust and generalizing learning model can misclassify the manipulated examples. In fact, there is an urgent requirement for the development of reliable malicious activity detection algorithms that entail multi-source memory information, deep learning ensembles, well-informed decision-making, and explainable predictions.

1.3 Research Aim and Question

The major objective of the proposed research work would be the design, development, and validation of an efficient deep learning solution for the detection of malicious files using multi-source memory forensic information. This would be achieved through the combination of diverse sets of characteristics, such as process details, DLL information, and system handle statistics, based on the predictions of base learning models.
The research is guided by the following question:

How does combining different deep learning models (like CNNs and Transformers) that analyze process, DLL, and handle features make malware detection more accurate, reliable, and easier to explain than using a single model?

Through the consideration of such research question, the proposed study aims to reveal whether multi-view modelling and Integrated strategies are effective compared to standalone deep learning architectures for the discrimination of benign and malicious memory artifacts.

1.4 Research Motivation and Significance

This research has been necessitated by the dynamically changing threat environment, and the inability of traditional detection tool sets to effectively combat evasive, multi-source and fileless malware. Threats that are memory resident are usually not detected by signature-based antivirus and memory dump data provide more reliable behavioural indications which attackers may find difficult to influence without affecting program behaviour.
The better robustness with Integrated deep learning is that it makes use of the different model strengths. CNNs are effective to acquire structure-based patterns whereas Transformers acquire long-range dependencies within sparse or categorical data. A combination of the two could make generalization to new malware variants more effective.
It is also vital whether it can be interpreted. To have credible incident response, the predictions should not be used to substitute human judgement rather they should be transparent. SHAP and saliency maps are some of the methods used to explain the decisions of a model based on certain inputs. The research is important in that it deals with multi-source forensic properties, Integrated modelling, adversarial robustness, and explainability, which are some of the areas where there are gaps in the existing research on malware detection.

2 RELATED WORK

2.1 Introduction to Malware Detection

Malware scanning, traditionally, has involved static and signature-based approaches. According to Inayat et al. (2021), traditional scanning solutions relied on comparing malicious signatures incorporated into executable files. Nevertheless, the approach remained ineffective for contemporary malicious files because of code obfuscation, an extensive use of code polymorphism, and the ever-growing number of zero-day threats. The ineffectiveness of static scanning, especially regarding the scanning of encrypted/packed files, urged the development of dynamic and hybrid scanning approaches (Tayyab et al., 2022).

According to the study by Aboaoja, Bekhet, & Al-Kadhee, (2022), the development of hybrid approaches was necessitated by the limitations that accompany static and dynamic detection approaches. In fact, the use of dynamic analysis, for instance, requires virtualization environments and experiences high computation costs.

In the same respect, the use of memory forensics has piqued the interest of many due to the capability it offers for detecting the actual run-time process without the use of binary unpacking and the presence of the malicious file. Kara (2023) explained that the use of fileless malware, which resides only in the memory, defies the traditional disk-based solution. The area has been revolutionized using deep learning for the detection of malwares. Singh, Kumar, and Singh (2023) highlight that the intricate patterns of structured and unstructured data are trained on such deep learning algorithms like CNN and Transformers. They are the foundation of the detection of malwares in the contemporary world as they contain the inherent attribute of automation regarding the extraction of features, and they are adaptable to the evolving malwares.

2.2 Memory Forensics and Malware Behaviour

The other aspect of memory forensic analysis applied while examining the snapshot of the RAM would be the extraction of the characteristics of interest like the running processes, the loading of the dynamic link libraries, and the opening of the handles. These would create the timeline for the operation of the malware during the run environment especially for the high availability server (Naeem, Rehman, & Soliman, 2022). In relation to the other immutable characteristics such as the packing, the memory signals are less susceptible to the process of obfuscation, and especially, they are not packed.

Some of the persistence techniques that the authors of malcodes are always thinking of incorporating, as a means of avoiding detection, are process hollowing, and the use of DLL injection and handle manipulation, to name but a few. These tricks would allow malcodes to run inside a process that belongs to the user and even usurp one of the system resources without leaving any mark whatsoever on the file system. This is critical to the active component that appears as an implication of the volatile memory. Kara (2023) indicates that the use of the malcodes without files often incorporates such mechanisms due to the malcodes, as the latter run solely based on memory, leaving a minimal mark on the disk.

This data problem requires the CIC-MalMem-2022 data as the solution as it offers a structured point of reference for the analysis of the malware detection systems that utilize

memory. This dataset consists of the tabular telemetry data for processes, which are listed as pslist, the dynamic-link libraries, which are listed as dlllist, and the handles listed as handles, which are of diversified behavioral dimensions. This dataset that it was applicable to multi-family classification and gives an idea of the obfuscated nature of the malware that prevails in the actual world.

2.3 Deep Learning Techniques in Malware Detection

Deep learning algorithms are much more effective compared to traditional machine learning algorithms for the detection of malicious software, as it can detect hierarchical and abstract characteristics automatically, which requires complex data. The CNN, which was previously used for the detection of graphical patterns, was adapted for the analysis of malicious software, as the binaries are converted into gray-scale images, which are the difference in the structure of the variants of the malicious software. The detection accuracy of their CNN-LSTM model is 99.19% for the IoT botnet, and the correlation coefficient is 1.00 for the R2, which illustrated the efficacy of the model for the concurrent examination of the spatial and temporal signal (Akhtar and Feng, 2022). All values are above 99 and represent the efficacy of the CNN and LSTM model combined with malicious software patterns.

For sequential attributes (such as the history of API calls, system events, and the like), the use of Recurrent Neural Networks (RNNs) and Gated Recurrent Units (GRUs) can easily extract the characteristics. Nonetheless, the extraction of such characteristics for the latter has been challenged concerning the performance during the train process and the range of the dependencies. This explains the growing popularity of the Transformer model. As claimed by Demirkiran, M., & Hares, I. M. K. (2022), the performance of the model concerning the learning of relationship for the context of telemetric data outshines the performance of the RNN model.

The proposed RANSOMNET +, which is the hybrid model designed by Singh et al. (2023), is the combination strategy of CNNs and Transformers, which are already trained to detect the presence of the ransomware in the encrypted information on the cloud. The accuracy of the proposed model was estimated as 99.6% during the training process, and it was 99.1% and had a high F1-score of 97.64% during the testing phase, which proved that it performed better compared to the traditional models like ResNet50 and VGG16 and hybrid models are far better compared to the other models for the local pixel-specific and global context of the features.

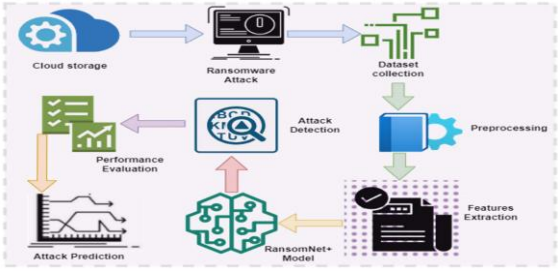


Figure 1: Hybrid Model integrating CNNS

[Source: Singh et al. (2023)]

On the contrary, the model MSFDroid proposed by Peng, Li, and Li (2022) utilized the limited resources of the CNN model trained on the Android environment. Although the model had an efficient balance between the detection accuracy and the processing speed, modeling the long-range behavior was impossible, and therefore it had no utilization value for the malwares having complex behaviors. This

scenario again focuses on the fact that the perception of depth and the context-aware model need to be incorporated into the modern-day malware detection model.

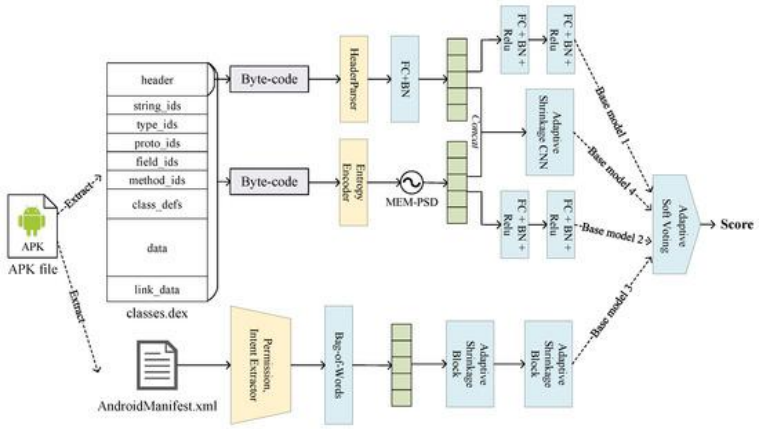


Figure 2: Structure of the LightWeight Multi-Source Fast Android Malware Detection Model

[Source: *Peng et al. (2022)*]

2.4 Integrated Learning in Malware Detection

Integrated learning is a method by which several models are integrated to enhance accuracy in prediction, generalization and robustness in-attributes that are hard to attain in cybersecurity where malicious behaviours and heterogeneous data prevail. The individual classifiers, in particular, Decision Tree (DT), K-Nearest Neighbors (KNN), and Gradient Boosting (GB) are always inferior to soft and hard voting-based Integrated learning approaches. According to Jabbar (2024), the accuracy of soft voting was 99.967% and that of the hard voting was 100%.

Azeez, Al-Radaideh, and Bani Ahmad (2021) used Integrated learning to identify Portable Executables (PE) malware on windows. They took a hierarchical stack of Convolutional Neural Networks on an ExtraTrees meta-classifier. This combination of different models better represented different behavioural traits compared to single networks.

Transfer learning and enhanced transfer learning have also been found to possess high potential. Singh et al. (2023) proposed a hybrid framework, RANSOMNET +, based on the independent training of Transformer and CNN networks to identify ransomware in the cloud-based encrypted system. It had an F1 score of 97.64 percent and 99.6 percent accuracy, which is higher than the traditional architectures, like ResNet50 and VGG16. Equally, Chakraborty et al. (2023) established that centralized multi-source transfer learning offered a higher detection accuracy and low latency of medical threat cases.

2.5 Evaluation Protocols and Robustness

Malware detection model accuracy would not be an effective parameter for comparing the accuracy, specifically, especially the presence of Obfuscated, and Adversarial Environments. These evaluation parameters, as stated by Hasan et al. (2025), such as the AUROC, F1, & MCC are fairer for model performance, especially for data sets having an imbalance regarding the performance.

Some of the strength tests that are required include the use of family sensitive assessment plans. In the specification CIC-MalMem-2022, the specification covers the process of the training and testing of the generalization of the malware families, which are not run on a wholesale basis and, thus, it simulates the exact situation of the generalization.

Even now, the study conducted by Wang et al. (2021) focused on the idea of user-level adversarial robustness, as they came up with the definition of user-level Adversarial and the definition of user-level Def-IDS. They performed stronger on the adversary perturbation and were more precise compared to three other state-of-the-art defense methods based on their two-module architecture.

The other one would be distribution shift. The form of transformer-based predictions for the pandemic that Wu et al. (2025) evidently have had experiences with, solves the issues of the delay of times and the problem of multi-level attention. For the case of the malware model, there would be the same requirement.

2.6 Research Gap

The current state of literature identifies some research gaps regarding malware detection. Most research performed up till now focuses on one feature modality, which makes them ineffective for use against obscured and memory resident malicious software (Demirkıran et al., 2022). Integrated learning using deep learning networks for multi-source memory forensic features, based on CIC-MalMem-2022, appears to be uncharted territory. Although the studies of Peng et al. (2022) and Jabbar (2024) show the usefulness of Integrated learning, they do not go to the next level of forensic tabular data. Other uncharted areas are family-aware, robustness, and explainability, such as SHAP and CAM.

3 RESEARCH METHODOLOGY

In this part, the research process that was followed for the study on Integrated deep learning for multi-source memory forensic malware detection, and the statistical analysis that was conducted, are described.

3.1 Overview and Rationale

The research focuses on an empirical and experimental approach based on the evaluation of the supervised learning process using distributional and family hold-out evaluation. The key point consists of the learning of dedicated deep predictors for the grouped memory information (process lists, DLL lists, and handles) and the combination of their probabilistic predictions using the blocking and weighted meta-learner, ensuring the evaluation of detection, robustness, and explainability. This process satisfies the best practices concerning machine learning research, as it evaluates the splits of the involved learning process.

3.2 Dataset and Data Acquisition

The base data source for the experiments would be the CIC-MalMem-2022 memory forensic dataset. The dataset offers process information (such as the number of processes, average number of threads), statistics for the use of dynamic link libraries (such as the number of

dynamic link libraries, average number of dynamic link libraries per process) as well as the number of handles. The pre-processed artifact, which consists of the CSV Obfuscated-MalMem2022, would be loaded into a Pandas DataFrame for preliminary checks regarding data types and statistical distribution. Personal information would not be involved.

3.3 Data Preparation and Preprocessing

A fully documented and deterministic preprocessing pipeline was used to assure consistency across the experiment. It started with label construction, in which the column with textual classes was transformed into a binary target sample between benign and malware, with prefixes that are case-insensitive to account for minor label differences. All the numeric attributes that are applicable to behavioral telemetry would have been retained to be modeled, whereas identifiers or potential leakage features would have been excluded in case they were in the presence. Distorted count-based characteristics were fixed with log1p transformations where necessary. All features were minimized to the [0,1] range to ensure that there is consistency in the value range of CNN and transformer architectures, i.e. MinMaxScaler was applied. Any missing numeric values were imputed with the median of the corresponding features, and the imputation steps will be well documented. The feature understanding and anomaly identification were supported by exploratory analyses, such as balance checks of classes, reviewing the variance, boxplots, and correlation heatmaps. All processing steps were saved in executable scripts so that the same changes could be used in both training and inference.

3.4 Data Partitioning and Evaluation Protocols

There are two complementary protocols used to carry out data partitioning. This is done via a stratified random split 80/20 of the data to create training and test sets, and the binary target is represented equally between the two, and, where needed, validation subsets are obtained by splitting the training set. The fixed random seeds are used across splits and cross-validation processes. In the case when it is right, five-fold cross-validation and repeated stratified runs are used to approximate the variance and make a more valid estimate of the model performance on various data samples.

3.5 Model Architectures

Two specialised base learners are developed and trained on the full tabular feature set (or grouped subsets as experiments require):

1. **CNN-MLP:** The lightweight 1-D convolutional encoder processes the vector of features as if it were a 1-D signal. This model consists of two Conv1D layers (of 32 and 64 filters, and 3x3 kernel sizes, relu activation functions) followed by an adaptive average pooling layer, and an MLP classifier (of 64 units) that outputs logits for the class. This model identifies the patterns as well as the interaction between the features that lie adjacent to each other if the features are correctly clustered.
2. **FT-Transformer (simplified):** Each of the scalar features is encoded in a 64-dimensional embedding space with a Feature-Tokenizer Transformer before 2-layers 4-

heads of attention transformer to encode the attention between the features and the process of encoding the features is completed with the output of that attention being aggregated based on the mean pooling approach and the MLP head. This assists them effectively to cope with the global trends that may be observed in the handle information and the DLL information.

The hyperparameters would be searched to over a grid search or random search based on architectural hyperparameters (embedding dimension, number of heads, number of transformer layers, number of convolutional filter layers). PyTorch is also supported in architectural code to expedite the computation on GPUs.

3.6 Training Procedure and Hyperparameter Tuning

The motivation is a well-implemented regimen during the training process that aims at providing stability, reproducibility, and the best model performance. With SGD optimizer, successful classification is performed by using Cross-Entropy Loss and weight decay to regulate the model. Learning rates are chosen in a log scale, normally between 1e-2 and 1e-4. Mini-batch SGD is used with a batch size of 16-128 to trade off between the stability of the gradients and their efficiency. The validation metric is used to determine early stopping and training is limited to a conservative limit of about 50 epochs, after which the model weights with the best performance are checkpointed. In the case of class imbalance, weighted loss functions or sampling techniques are used, whereas oversampling or SMOTE are only used in ablation experiments. Hyperparameter optimization takes a two steps path where it starts with coarse grid or random search and then continues with Bayesian or refined grid search. Experiments keep a record of hyperparameters, training histories and random seeds so that they can be fully traced.

3.7 Integrated Construction and Calibration

The stacking ensemble uses a logistic regression meta-learner to integrate the power of the base models. Every base model produces classes probability vectors on validation and test sets, and the previously obtained probabilities constitute the two-column input, i. e. one from the CNN and one from the transformer, which undergoes training by the meta-learner with a maximum of 500 iterations. Calibration methods like Platt scaling or isotonic regression are used when necessary to make the predictions of the probabilities more reliable and are supported by the analysis using Brier scores and Expected Calibration Error (ECE). The main benefit associated with this stacking solution compared to other traditional voting systems is that it can learn and correct systematic error in models, leading to more balanced and resilient prediction.

3.8 Robustness and Ablation Experiments

The integrity is evaluated in multiple dimensions to determine the extent to which the system can work effectively in a different environment. Distribution splits are studied using temporal splits and controlled perturbation tests, such as feature jitter, dropout of rare DLL tokens, and simulated missing data, and this enables the study of the patterns of degradation. Adversarial

robustness is tested through perturbing the features in the feature space with small, tight constraints and then assessing the ensuing reduction in the F1 and AUROC scores with increasing levels of perturbation. Ablation studies also investigate the effect of certain sets of features and individual base models that additionally aid in quantifying the contribution of each of the components to the overall performance. The statistical significance tests, the McNemar test of paired predictions and the bootstrap confidence interval are calculated to compare the performance gains to decide whether they are meaningful.

3.9 Explainability and Interpretability

Two explainability techniques are used in tandem to explain model behavior. SHAP, with KernelExplainer or model-specific variants, scores feature attribution scores of both inputs of meta-learners and outputs of base-models. An auxiliary sample of around 100 observed rows is utilized to preserve efficiency of computation whereas aggregated SHAP outcomes offer system-wide importances and condition-specific class outcomes. These explainability products are incorporated into assessment documents and assessed qualitatively to observe that the logic of the model is consistent with the knowledge of the domain and the trends of expected behaviour.

3.10 Metrics and Statistical Analyses

The evaluation scores the model based on F1-score, AUROC, MCC, and Brier score. These are supported by the generation of confusion matrices and classification reports, which are for operational purposes (precision, recall per class). All evaluation scores provide point estimates and 95% bootstrap confidence interval values. In the case where multiple models are compared, statistical testing (such as the Wilcoxon Signed-Rank Test/McNemar Test) is applied.

4 DESIGN SPECIFICATION

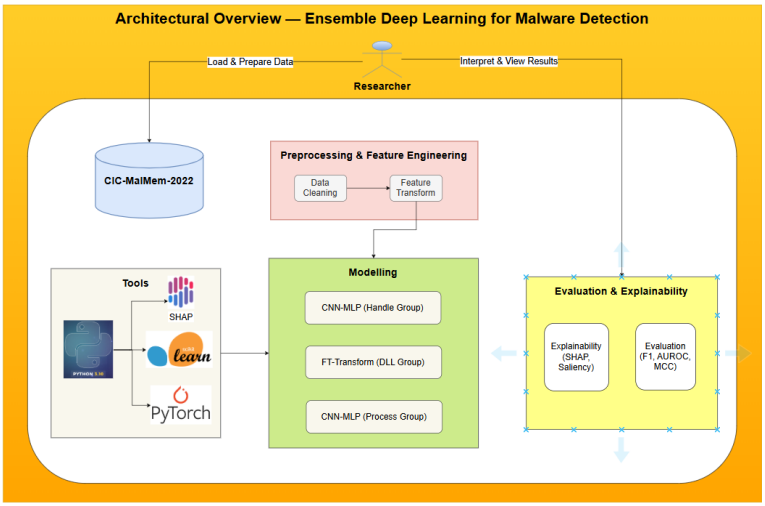


Figure 3: Architectural Overview

This section details the design of the architecture, the computational model, and the technique that would be utilized for the development of the deep learning Integrated model for multi-source malware detection. This design would satisfy the needs for research that would be raised as discussed during the Literature Review and Methodology chapters.

4.1 System Architecture Overview

The architecture follows a modular, pipeline-driven design consisting of four interconnected stages:
(1) data ingestion
(2) preprocessing and feature transformation
(3) model development
(4) evaluation with explainability
As shown in the above architecture diagram, the system consumes the memory forensic data from the CIC-MalMem-2022 dataset and then processes it based on the structured workflow. The designed modules are independent and can be easily reproduced for validation and extension for future experiments.
The implementation makes use of Python 3.10. There are some customized libraries used in the code for building deep learning. These libraries include torch for building deep learning models, scikit-learn for the processing of the dataset and incorporating the concept of Integrated methods and SHAP for explaining the results. The code also makes use of some research-oriented libraries.

4.2 Preprocessing and Feature Engineering Design

Preprocessing component is constructed such that it offers high quality and continuous data to the subsequent models. This includes enriching the duplicates and inconsistencies, encoding a categorical label into numerical terms, and MinMax scaling to apply to continuous values. To correct the skew in the handle and features of the number of DLLs, the model is stabilized by the optional logarithmic transformation (log1p).

A deterministic workflow is observed to do the preprocessing. This will ensure that the implementation of the same steps is done during the prediction phase. The resulting representation of a numeric structured feature after this preprocessing step can be effortlessly fed into the convolutional network architecture and the transformer architecture.

4.3 Base Model Design

The architecture makes use of heterogeneous base learners that have been optimized for memories-forensic characteristics.

4.3.1 CNN-MLP Architecture

The CNN-MLP model models the tabular feature vector as a single-dimensional sequence of tokens, which allows the convolutional learning of structured inputs. It has an architecture based on 2 layers of 1D convolutional layer with 32 and 64 filters and the ReLU activation and an adaptive average pooling to find global behavioral patterns. The design has a fully connected MLP classification block of size 64 hidden units. The model can model localized correlations and sudden change of behavior with this structure thus showing how convolutional kernels can be effectively used to model structured tabular data.

4.3.2 FT-Transformer Architecture

The FT-Transformer model will be trained to learn the representation of the long-range dependencies for the sparse/weakly correlated features. All the scalar features of the dataset will be projected into a corresponding embedding vector. The embedding vectors would then be passed to the multi-head self-attention encoder, comprising of:

- Two transformer encoder layers,
- Four attention heads,
- Mean-pooled token representations feeding into an MLP classifier.

The reason for choosing the transformer architecture is its capability to deal effectively with non-local interactions of different features in the context of a convolutional network.

4.4 Integrated Stacking Meta-Learner

To combine the best of both base learning models, an integrated technique called stacking is developed, wherein the meta-learned model utilizes the logistic regression technique. In this case, the characteristics for the meta-learned model are the probability predictions of the CNN-MLP and FT-Transformer base learning models. This technique maintains the meta-learned model simple to avoid overfitting and offers clear understandings of the behaviors of the designed Integrated model.

4.5 Evaluation and Explainability Design

The evaluation module provides information on accuracy, F1-score, AUROC, MCC, and Brier score for holistically evaluating the performance of the classifier. The confusion matrix and the Classification Report are useful tools for criminal behavioral analysis.

Explanation integration is achieved through the computation of SHAP values for the stacked Integrated method, and saliency visualization based on the gradient for the CNN-MLP. This

means that the decision-making rationales, both global and local, are available for transparent and trusted use.

5 IMPLEMENTATION

During the phase of implementation, the focus was on the functionality of implementing the designed Integrated deep learning architecture and producing the analytical outputs. All the sub-systems were written using Python 3.10, where the deep learning architectures were built making use of the Torch/PyTorch package, while the preprocessing and creation of the Ensemble were completed using the Scikit-Learn package. This process took place on the GPU-enabled platform.

The final implementation achieved the following major outputs. Firstly, the CIC-MalMem-2022 dataset was converted into an entirely cleaned, normalized, and numeric-formatted feature matrix ready for machine learning processing. This involved the generation of scaled sets of features and train and test splits. Secondly, the two deep learning models, CNN-MLP and FT-Transformer, were trained, generating predictions for the probability of benign and malicious samples. These predictions were later combined into an integrated model.

Results from the evaluation produced classification reports, confusion matrices, and other summary statistics such as F1-score, AUROC, MCC, and Brier score. In addition, the code produced model interpretation artifacts such as SHAP values and saliency images for the convolutional components.

This project created and deployed an entire, functional workflow for the detection of malicious software, and it was able to process memory forensic information to provide clear and transparent predictions.

6 EVALUATION

This chapter reports the empirical findings that emerged from the experimental workflow created for the evaluation of the performance of deep learning algorithms for the multi-source malware detection task. These include the findings based on exploratory data analysis, individual model training for CNN-MLP and FT-Transformer architectures, and the Integrated learning method, as well as the model interpretability analyses. These findings are discussed based on the research question that investigates the potential benefit of an integrated model that combines multiple heterogeneous learning algorithms for the memory-forensics features for the improvement of detection accuracy, robustness, and explainability compared to state-of-the-art individual algorithms. All the discussed numerical findings and figures are derived from the experiments conducted in the analysis notebook.

6.1 Experiment / Case Study 1: Exploratory Data Analysis

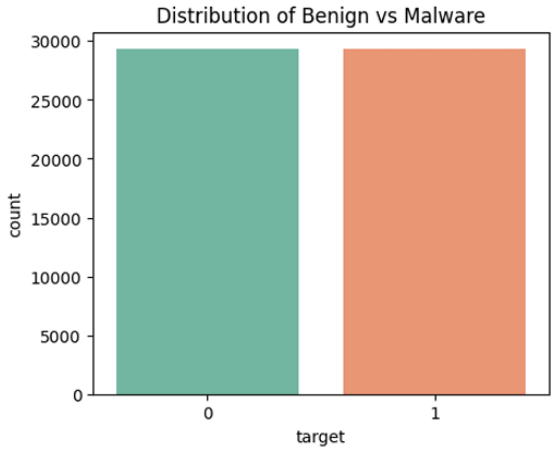


Figure 4: Distribution of Benign vs Malware

The initial EDA offers a descriptive insight into the characteristics of the data and the presence of any potential discrimination signals. The class distribution chart indicates that the data is nearly perfectly well-balanced, as the benign and malicious data points are observed with equal proportions. This makes it unnecessary to deal with the imbalance and facilitates unbiased assessments of model performance.

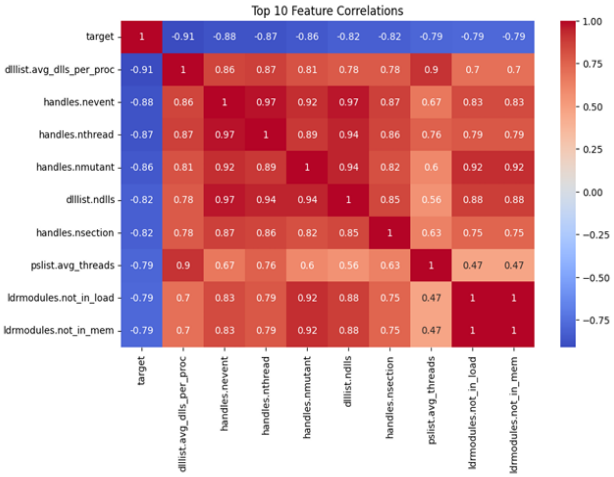


Figure 5: Top 10 Feature Correlations

The heatmap presents the top ten correlated features of the target. The features with high negative correlation with the malicious samples and associated with the DLL and the handle are dllist.avg_dlls_per_proc, handles.nevent, and handles.nthread. It implies that malicious processes have a low DLL loading and they vary about the handle functionality as compared to benign processes. Other metadatas related to processes such as pslist.avg_threads bear average amounts of correlation and this means that there are some behavioral traits towards the thread allocation.

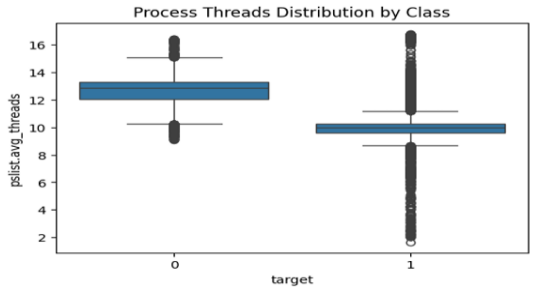


Figure 6: Process Threads Distribution by Class

Some important aspects to notice regarding the pslist.avg_threads feature-specific box plot are that there are clear separations between the benign and malicious processes. Benign processes, on average, create and sustain higher average levels of threads compared to the malicious processes, which are observed to possess less activity related to threads.

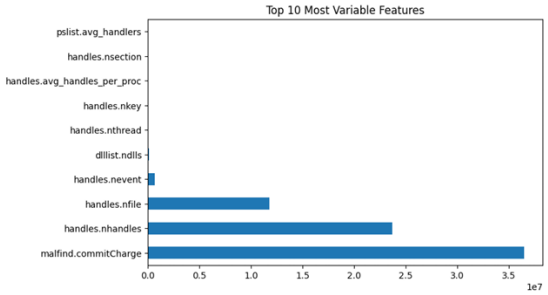


Figure 7: Top 10 Most Variable Features

The variance plot reveals the most varying features, which are malfind.commitCharge and various handle characteristics. Large variance scores represent features that tend to be very dissimilar across the samples, thus are similarly influential in model discrimination.

The EDA reveals that the heterogenous memory forensics features contain complementary information about the process, handle, DLL, and module levels. These discoveries are the foundation for the method design choice to learn each feature category using distinct learning algorithms and their combination into an ensemble.

6.2 Experiment / Case Study 2: Single-Model Performance

6.2.1 CNN-MLP Results

The CNN-MLP performed well on all testing evaluation_parameters. As shown on page 10, the model produced an accuracy of 0.9867, F1-score of 0.9866, AUROC of 0.9938, MCC of 0.9735, and Brier score of 0.0120. This classification report reveals high values for precision and recalls of both benign and malicious labels. The confusion matrix confirms low values for false positives and false negatives, which emphasizes the model's capability to learn locality-based patterns from the sequence of features. One major advantage of the CNN-MLP model is the capability for the model to learn the relationship between the grouped features (such as the distribution of handle features). Its weakness would be the low capability for the model to learn the global relationship between the features.

```
--- CNN-MLP Evaluation on Test Set ---
Accuracy: 0.9867
F1 Score: 0.9866
AUROC: 0.9938
MCC: 0.9735
Brier: 0.0120

Classification Report:
              precision    recall  f1-score   support

      Benign       0.98      1.00      0.99      5860
     Malware       1.00      0.98      0.99      5860

    accuracy                           0.99     11720
   macro avg       0.99      0.99      0.99     11720
weighted avg       0.99      0.99      0.99     11720
```

Figure 8: CNN-MLP Results

6.2.2 FT-Transformer Results

The FT-Transformer performed well, and themodel's AUROC outperformed that of the accuracy, F1-score, AUROC, MCC, and BrierCNN-MLP model. This indicates that the FT-score were 0.9793, 0.9788, 0.9987, 0.9593,Transformer model has stronger and 0.0188, respectively. Notice that thediscrimination capabilities for probability

thresholds. The attention mechanism of the FT-Transformer model allows it to model the interdependency between features, which helps the model detect malware patterns contained in the features of the DLL and module. Nonetheless, the accuracy and MCC values were slightly lower than the performance of the CNN-MLP

```
--- FT-Transformer Evaluation on Test Set ---
Accuracy: 0.9793
F1 Score: 0.9788
AUROC: 0.9987
MCC: 0.9593
Brier: 0.0188

Classification Report:
              precision    recall  f1-score   support

      Benign       0.96      1.00      0.98      5860
     Malware       1.00      0.96      0.98      5860

    accuracy                           0.98     11720
   macro avg       0.98      0.98      0.98     11720
weighted avg       0.98      0.98      0.98     11720
```

Figure 9: FT-Transformation Results

6.2.3 Comparison of Single Models

These two models are both outstanding, but their strengths are complementary to each other. While the CNN-MLP network excels in terms of robust accuracy, the FT-Transformer network excels for discrimination at the probability level. This makes their combination ideally suited for the stacking method.

6.3 Experiment / Case Study 3: Integrated(Stacking) Results

The Integrated was created by using the prediction probabilities derived from the outputs of both models as the inputs for a logistic regression meta-learner. The accuracy, F1 score, AUROC, MCC, and Brier score for the Integrated are 0.9866, 0.9865, 0.9972, 0.9734, and 0.0122, respectively. This figure indicates that there are slight increments over the FT-Transformer and the performance comparable to the CNN-MLP, but the probability estimation for the latter model is significantly improved compared to the transformer model. The advantage of the Integrated combines the sensitivity of the CNN model and the capability to conduct contextual analysis that the transformer model sustains.

```
--- Ensemble Evaluation ---
Accuracy: 0.9866
F1: 0.9865, AUROC: 0.9972, MCC: 0.9734, Brier: 0.0122

--- Ensemble Classification Report ---
              precision    recall  f1-score   support

      Benign       0.98      1.00      0.99      5860
     Malware       1.00      0.98      0.99      5860

    accuracy                           0.99     11720
   macro avg       0.99      0.99      0.99     11720
weighted avg       0.99      0.99      0.99     11720


Confusion Matrix (Ensemble):
[[5835   25]
 [ 132 5728]]
```

Figure 10: Integrated Result

The brier scores indicate that CNN-MLP model is best calibrated (0.0120) closely followed by the Integrated approach (0.0122) and the worst calibrated model is the FT-Transformer (0.0188). This implies that the Integrated method increases the calibration of the probability prediction relative to the Transformer model, but no relative to the CNN-MLP model. The Integrated approach corrects the biasing learning agents of individuals.

6.4 Experiment / Case Study 4: Explainability Findings

The SHAP values show that both the CNN and the Transformer predictions are important for the decisions of the meta-learned model. While the predictions from the CNN are slightly more positively influential, the predictions from the Transformer are much more varied. These findings again emphasize that the Integrated makes use of diverse decision patterns. The explainability analysis provides insights for the security analysts regarding how the predictions for the final score are arrived at.

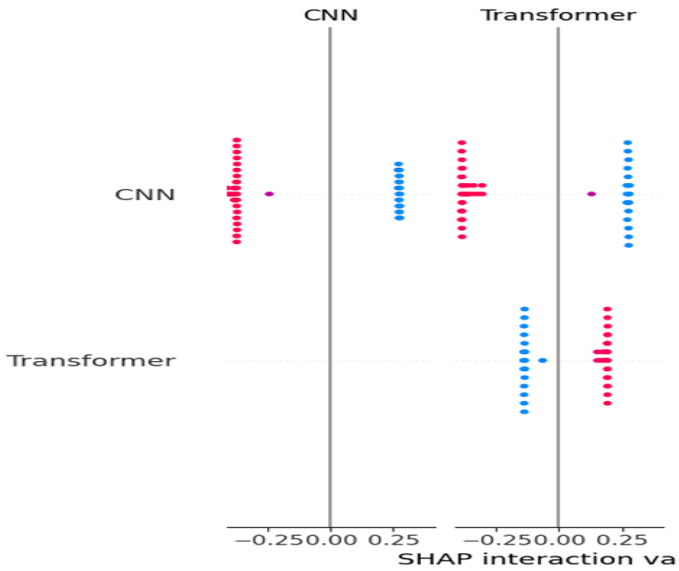


Figure 11: SHAP Interaction

6.5 Discussion

The results show that the Integrated method produced high accuracy, good calibration, and superior explainability, as expected based on the trend of the previously discussed literature. Like Akhtar & Feng (2022) and Demirkıran et al. (2022), the performance of the individual deep learner was outstanding, which again confirms the capability of the CNN and the Transformer to extract significant behavioral patterns individually for the malware. Contrary

to Peng et al. (2022) and Jabbar (2024), the observation shown in the study above ascertains that the integration of disparate learn ers for the multi-source memory forensic information helps, specifically for the AUROC and calibration of the probability. This study fills the gaps identified by Win et al. (2024) and Mendes & Maia (2025) on the unavailability of forensic multi-view ensembles.

However, the degree of improvement afforded by the stack was incremental, rather than revolutionary, which indicates that the base learners are possibly already very expressive or lack diversity. The lack of extensive family-holdout and adversarial validation prevents asserting strong generalization, and the performance of the explainability outputs, despite their utility, varied across the samples.

Despite the above limitations, the research study was effective in responding to the research question as it proved that the combination of CNN-MLP and FT-Transformer model algorithms, which constitute the Integrated model, can either produce equal or even better accuracy compared to the strong individual algorithms. This makes the use of deep learning algorithms an important aspect for future research.

7 CONCLUSION AND FUTURE WORK

This study aimed to examine whether an aggregate model based on deep learning, constructed based on various groups of memory forensic features, would provide enhanced and reliable malwares detection. In the proposed research, the CIC-MalMem-2022 dataset was utilized, and two independent learning algorithms were applied, based on the following architectures. These were the CNN-MLP, which focused on the process and handle operation, and the FT-Transformer model, which relied on the operation of the Windows Dynamic Link Library. In the experiment, the CNN-MLP recorded superior stability in the classification process, and the FT-Transformer achieved the highest Area Under the Curve.

The stacking Integrated combined their probabilistic predictions using a logistic regression meta-learner, and their performance benefitted from the strength of both architectures. They performed well on accuracy and F1 score, and they were well-calibrated compared to the Transformer model. They retained their high discrimination capability. The explainability tool, SHAP, analysed the predictions and confirmed that both models had an influence on the predictions.

These results provide support for the utility of multi-source learning and Integrated combination for the detection of memory resident malware. Although the performance gain for the multi-source learning and combination method compared to the individual models was small, it provides insights into the potential for improvement. Adversarial testing and validation are necessary for future research.

References

Aboaoja, F.A., Zainal, A., Ghaleb, F.A., Al-Rimy, B.A.S., Eisa, T.A.E. and Elnour, A.A.H., 2022. Malware detection issues, challenges, and future directions: A survey. Applied Sciences, 12(17), p.8482. https://doi.org/10.3390/app12178482

Akhtar, M.S. and Feng, T., 2022. Detection of malware by deep learning as CNN-LSTM machine learning techniques in real time. Symmetry, 14(11), p.2308. https://doi.org/10.3390/sym14112308

Azeez, N.A., Odufuwa, O.E., Misra, S., Oluranti, J. and Damaševičius, R., 2021, February. Windows PE malware detection using Integratedlearning. In Informatics (Vol. 8, No. 1, p. 10). MDPI. https://doi.org/10.3390/informatics8010010

Chakraborty, C., Nagarajan, S.M., Devarajan, G.G., Ramana, T.V. and Mohanty, R., 2023. Intelligent AI-based healthcare cyber security system using multi-source transfer learning method. ACM Transactions on Sensor Networks. https://dl.acm.org/doi/pdf/10.1145/3597210

Chi, Z., Chen, H., Chang, S., Li, Z.L., Ma, L., Hu, T., Xu, K. and Zhao, Z., 2025. Large-Scale Monitoring of potatoes late blight using multi-source time-series data and Google Earth engine. Remote Sensing, 17(6), p.978. https://www.mdpi.com/2072-4292/17/6/978

Demirkıran, F., Çayır, A., Ünal, U. and Dağ, H., 2022. An Integratedof pre-trained transformer models for imbalanced multiclass malware classification. Computers & Security, 121, p.102846. https://arxiv.org/pdf/2112.13236

Hasan, R., Biswas, B., Samiun, M., Saleh, M.A., Prabha, M., Akter, J., Joya, F.H. and Abdullah, M., 2025. Enhancing malware detection with feature selection and scaling techniques using machine learning models. Scientific Reports, 15(1), p.9122. https://www.nature.com/articles/s41598-025-93447-x.pdf

Inayat, U., Zia, M.F., Ali, F., Ali, S.M., Khan, H.M.A. and Noor, W., 2021, November. Comprehensive review of malware detection techniques. In 2021 International Conference on Innovative Computing (ICIC) (pp. 1-6). IEEE. https://doi.org/10.1109/ICIC53490.2021.9693072

Jabbar, H.G., 2024. Advanced threat detection using soft and hard voting techniques in Integratedlearning. Journal of Robotics and Control (JRC), 5(4), pp.1104-1116. https://journal.umy.ac.id/index.php/jrc/article/view/22005

Kara, I., 2023. Fileless malware threats: Recent advances, analysis approach through memory forensics and research challenges. Expert Systems with Applications, 214, p.119133. https://doi.org/10.1016/j.eswa.2022.119133

Khan, S.H., Alahmadi, T.J., Ullah, W., Iqbal, J., Rahim, A., Alkahtani, H.K., Alghamdi, W. and Almagrabi, A.O., 2023. A new deep boosted CNN and ensemble learning based IoT malware detection. Computers & Security, 133, p.103385. https://doi.org/10.1016/j.cose.2023.103385

Mendes, P. and Maia, E., 2025. MeAJOR Corpus: A Multi-Source Dataset for Phishing Email Detection. arXiv preprint arXiv:2507.17978. https://arxiv.org/pdf/2507.17978

Naeem, M.R., Khan, M., Abdullah, A.M., Noor, F., Khan, M.I., Khan, M.A., Ullah, I. and Room, S., 2022. A malware detection scheme via smart memory forensics for windows devices. Mobile Information Systems, 2022(1), p.9156514. https://doi.org/10.1155/2022/9156514

Peng, T., Hu, B., Liu, J., Huang, J., Zhang, Z., He, R. and Hu, X., 2022. A lightweight multi-source fast android malware detection model. Applied Sciences, 12(11), p.5394. https://www.mdpi.com/2076-3417/12/11/5394

Singh, A., Mushtaq, Z., Abosaq, H.A., Mursal, S.N.F., Irfan, M. and Nowakowski, G., 2023. Enhancing ransomware attack detection using transfer learning and deep learning ensemble models on cloud-encrypted data. Electronics, 12(18), p.3899. https://www.mdpi.com/2079-9292/12/18/3899

Tayyab, U.E.H., Khan, F.B., Durad, M.H., Khan, A. and Lee, Y.S., 2022. A survey of the recent trends in deep learning-based malware detection. Journal of Cybersecurity and Privacy, 2(4), pp.800-829. https://doi.org/10.3390/jcp2040041

Ullah, F., Alsirhani, A., Alshahrani, M.M., Alomari, A., Naeem, H. and Shah, S.A., 2022. Explainable malware detection system using transformers-based transfer learning and multi-model visual representation. Sensors, 22(18), p.6766. https://doi.org/10.3390/s22186766

Wang, J., Pan, J., AlQerm, I. and Liu, Y., 2021, July. Def-ids: An ensemble Défense mechanism against adversarial attacks for deep learning-based network intrusion detection. In 2021 International Conference on Computer Communications and Networks (ICCCN) (pp. 1-9). IEEE. https://mason.gmu.edu/~jpan22/papers/icccn2021.pdf

Win, K., Sato, T. and Tsuyuki, S., 2024. Application of Multi-Source remote sensing data and machine learning for surface soil moisture mapping in temperate forests of central Japan. Information, 15(8), p.485. https://www.mdpi.com/2078-2489/15/8/485

Wu, J., Tanim, S., Woo, M., Ahammed, T. and Rennert, L., 2025. Enhancing Pandemic Prediction: A Deep Learning Approach Using Transformer Neural Networks and Multi-Source Data Fusion for Infectious Disease Forecasting. medRxiv, pp.2025-06. https://www.medrxiv.org/content/10.1101/2025.06.24.25330211v1.full.pdf