National College of Ireland

Enhancing Detection of Encrypted Fileless Malware using Memory Forensics and AI Techniques

MSc Research Project
MSc Cyber Security

Sanket Mahajan
Student ID: x23272970

School of Computing
National College of Ireland

Supervisor: Niall Heffernan

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Sanket Shrikrishna Mahajan		
Student ID:	x23272970		
Programme:	MSc. Cyber Security	**Year:**	2024-25
Module:	MSc Research Project		
Supervisor:	Niall Heffernan		
Submission Due Date:	11 Aug 2025		
Project Title:	Enhancing Detection of Encrypted Fileless Malware using Memory Forensics and AI Techniques		
Word Count:	615	**Page Count:**	8

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: Sanket Mahajan

Date: 11 Aug 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):	

Configuration Manual

Sanket Shrikrishna Mahajan

Student ID: x23272970

1. Introduction:

The Hybrid Fileless Malware Detection Tool is an aspect developed under the MSc in Cybersecurity in the National College of Ireland. It is a forensic system using a Python framework aiming to recognize volatile memory driven fileless malware using encrypted characteristics by adopting a hybrid approach intermingling rule-based heuristics and clustering driven at an AI level.

The tool presents the ability to deal with structured data supplied by the Volatility memory forensics plugins, profile entropies, map to the MITRE ATT&CK, and run unsupervised clustering to detect suspicious processes and patterns of infections.

This configuration manual provides:

- System requirements for running the tool
- Setup and installation steps
- Module configuration
- Functional features
- Deployment & testing instructions

2. System Requirement:

Whatever system you set up needs to meet at least the following requirements to execute both, its memory forensics pipeline as well as its AI clustering tool:

2.1. **Hardware Requirements:**

Component	Requirement
Operating System	Ubuntu, Kali Linux, Windows 10
Processor	Dual Core Intel i5 or AMD
RAM	Minimum 8 GB
Storage	Minimum 30GB of free disc space for installation of VM and Linux and to run the tool
Graphics	NVDIA/ AMD/ Intel

2.2.**Software requirements:**

- Python 3.8
- Jupyter Notebook
- matplotlib
- sckit-learn
- seaborn
- Pandas
- NumPy

2.3. **Dataset:**

The tool requires pre-extracted structured datasets from parsing memory dumps with Volatility3 and merging outputs. The CIC-MalMem-2022 dataset was used for experimental purposes as it covers fileless Memory attack techniques.

3. Environment Setup:

3.1. Initial Installation:

The project is built in a virtual environment to avoid contact with direct malware. For this, download and install the VM VirtualBox and Kali Linux 2025.2.

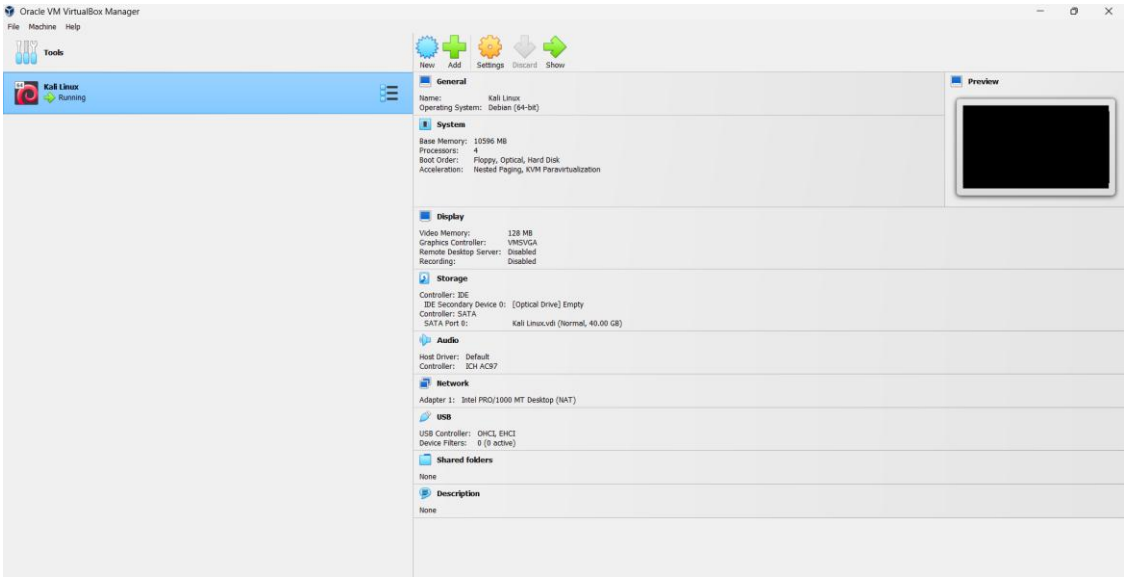

Fig 1: VM VirtualBox Setup

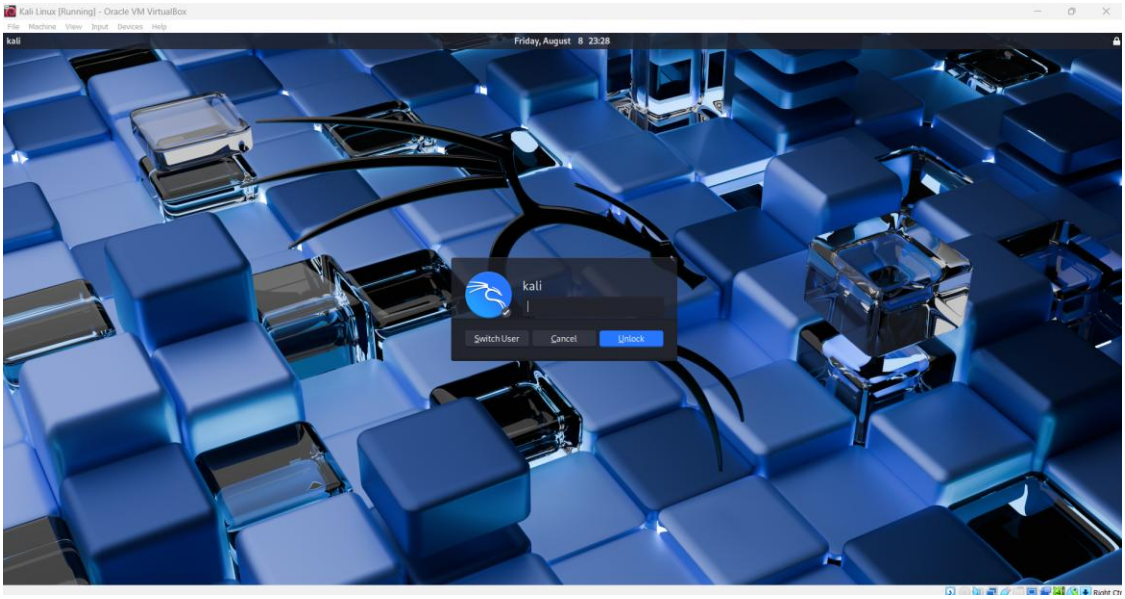

Fig 2: Kali Linux Setup

Download and install Python 3.13.3 and Jupyter notebook Once downloaded run the command:

jupyter notebook --no-browser

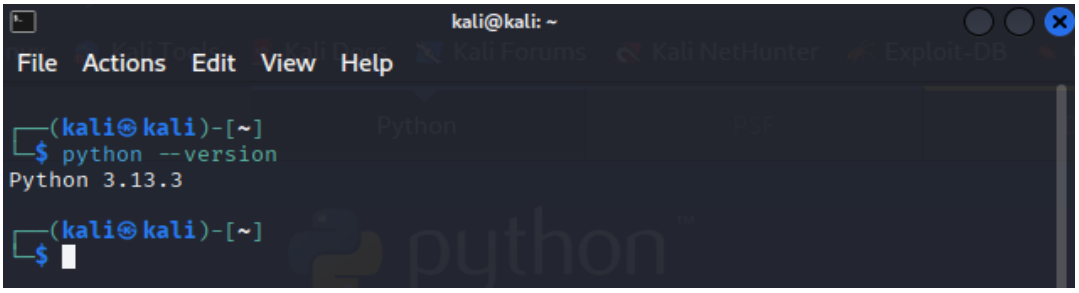

Fig 3: Checking Python version

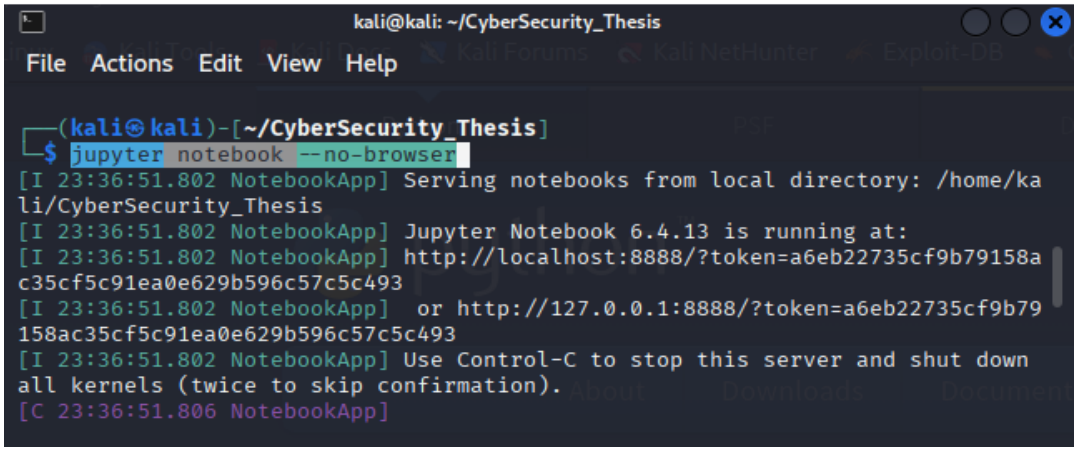

Fig 4: Starting Jupyter Notebook

Below is the project directory with the key components:

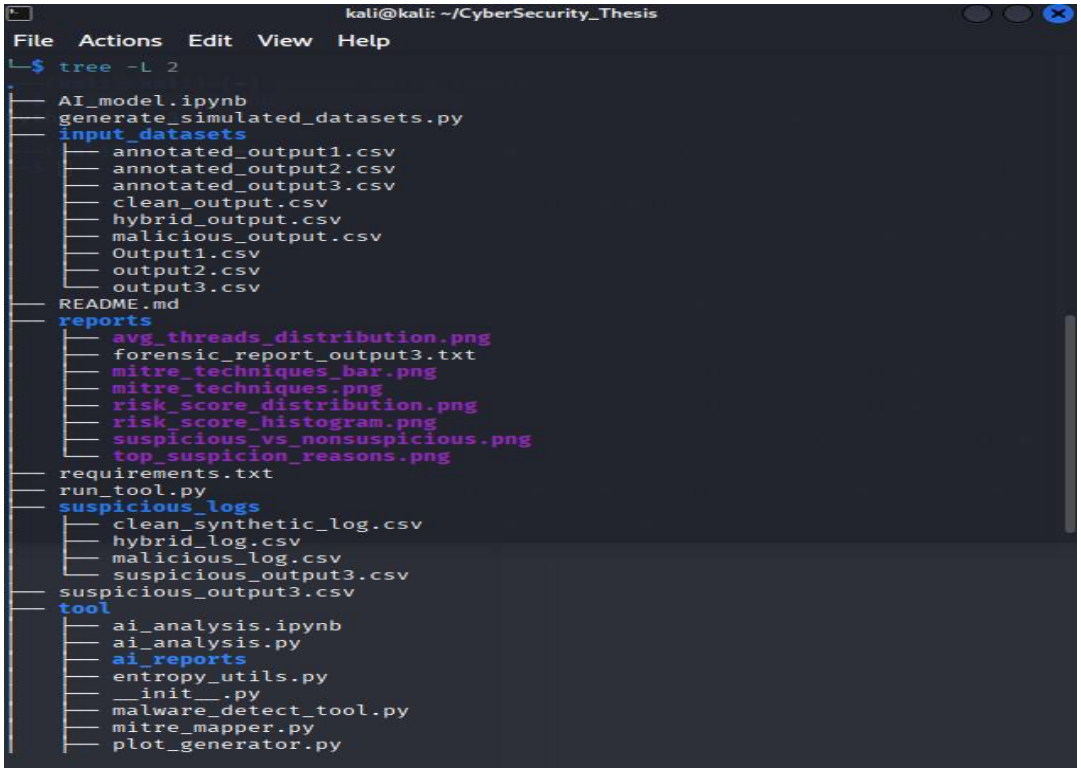

Fig 5: Project Directory

3.2. **Setup Instructions:**

1. Clone and enter the Project directory:

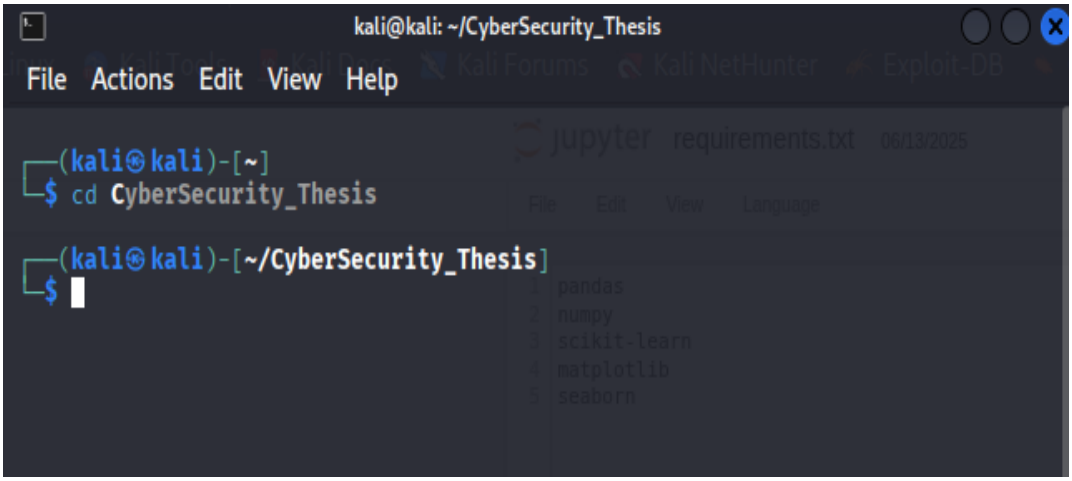

Fig 6: Enter in the Project directory

2. Create a virtual environment:
   python3 -m venv venv
   source venv/bin/activate
   venv\Scripts\activate

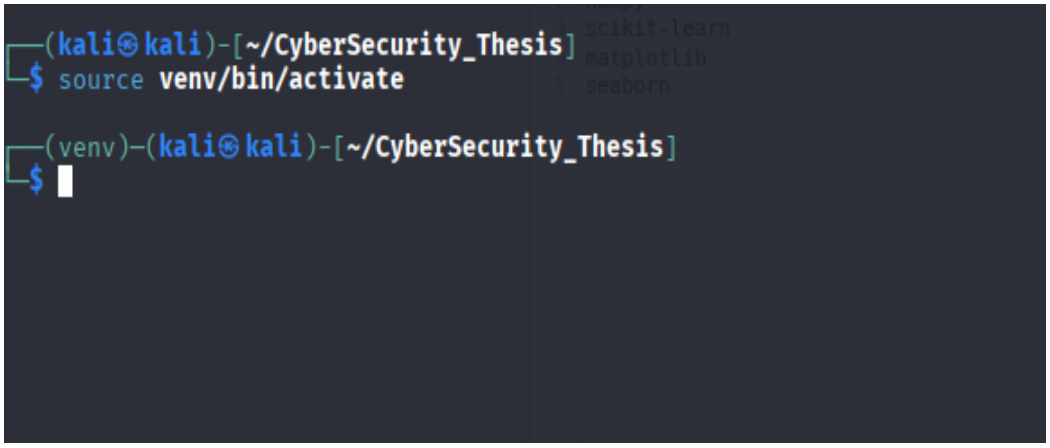

Fig 7: enter in the Virtual Environment

3. Then install the dependencies:
   pip install --upgrade pip
   pip install -r requirements.txt
4. Now to run the CLI tool:
   python tool/rule_engine.py –input input_datasets/hybrid_output.csv –output suspicious_logs/hybrid_log.csv

```
┌──(venv)─(kali㉿kali)-[~/CyberSecurity_Thesis]
└─$ python3 run_tool.py \
  --input input_datasets/hybrid_output.csv \
  --output suspicious_logs/hybrid_log.csv
[DEBUG] Result shape: (350, 5)
[DEBUG] Example row output: 0
 True
1    Code injection detected; High number of callba ...
2                                  T1106; T1055; T1060
3                                                    7
4                                                 High
Name: 0, dtype: object
Analysis complete: 350 suspicious rows saved to suspicious_logs/hybrid_log.cs
v

┌──(venv)─(kali㉿kali)-[~/CyberSecurity_Thesis]
└─$
```

Fig 8: run the tool

5. AI clustering analysis:
   Type a command:
   python tool/ai_analysis.py

```
┌──(venv)─(kali㉿kali)-[~/CyberSecurity_Thesis]
└─$ python3 tool/ai_analysis.py
Loading dataset: ./input_datasets/hybrid_output.csv
Shape: (350, 61)
PCA plot saved.
t-SNE plot saved.
Anomaly scores saved to ai_reports/hybrid_output_analyzed.csv
[Anomaly-highlighted t-SNE plot saved.
AI analysis complete.

┌──(venv)─(kali㉿kali)-[~/CyberSecurity_Thesis]
└─$
```

Fig 9: Command to run AI analysis tool

6. To generate Charts, type the command:
   python tool/plot_generator.py
7. Comparing AI vs Rule-Based model:
   Use rf_hybrid_detection_pipeline.ipynb file to generate results.

4. Result:

After the successful execution, fileless malware detection tool produces the following results.

4.1. Suspicious Logs:

CSV files listing all processes flagged by the rule-based engine and AI clustering that includes risk score, suspicious reasons, and mapped MITRE techniques.

4.2. Summary reports:

The text-based reports summarizing detection counts, top suspicious reasons and high-risk processes which can be included in the incident response report.

4.3. Visualization:

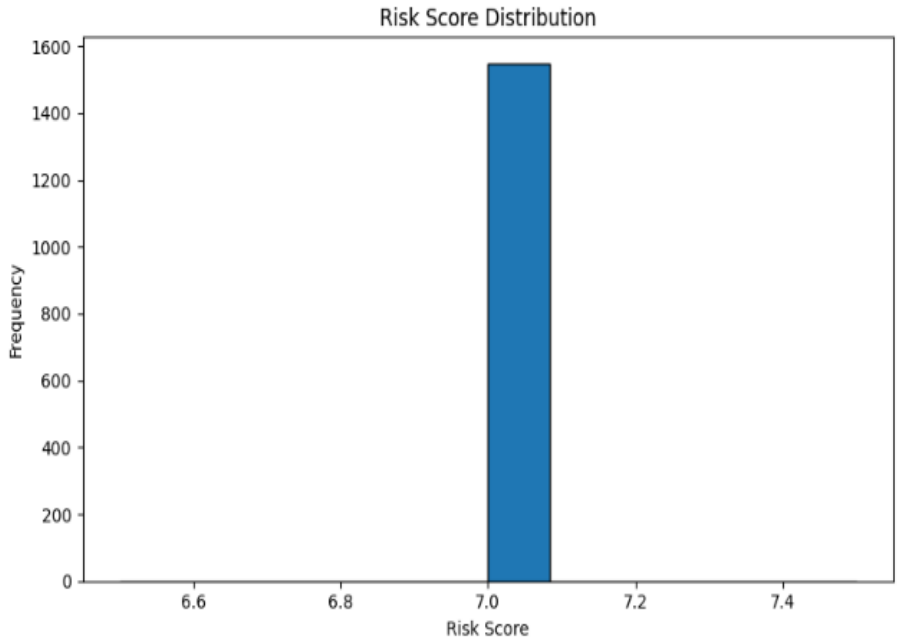

Fig 10: Risk Distribution Analysis

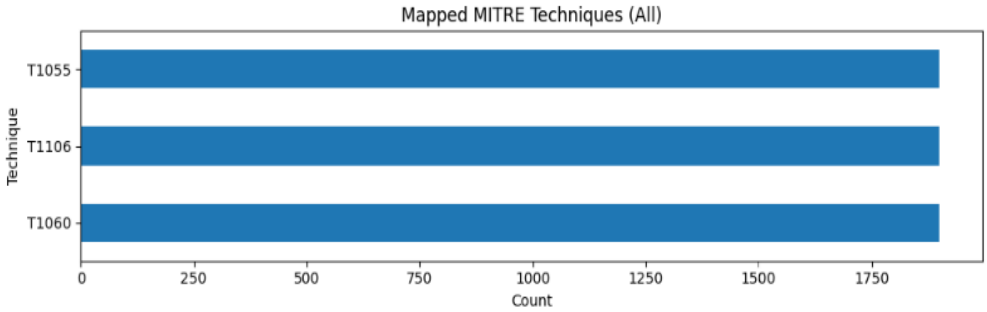

Fig 11: MITRE ATT&CK techniques frequency

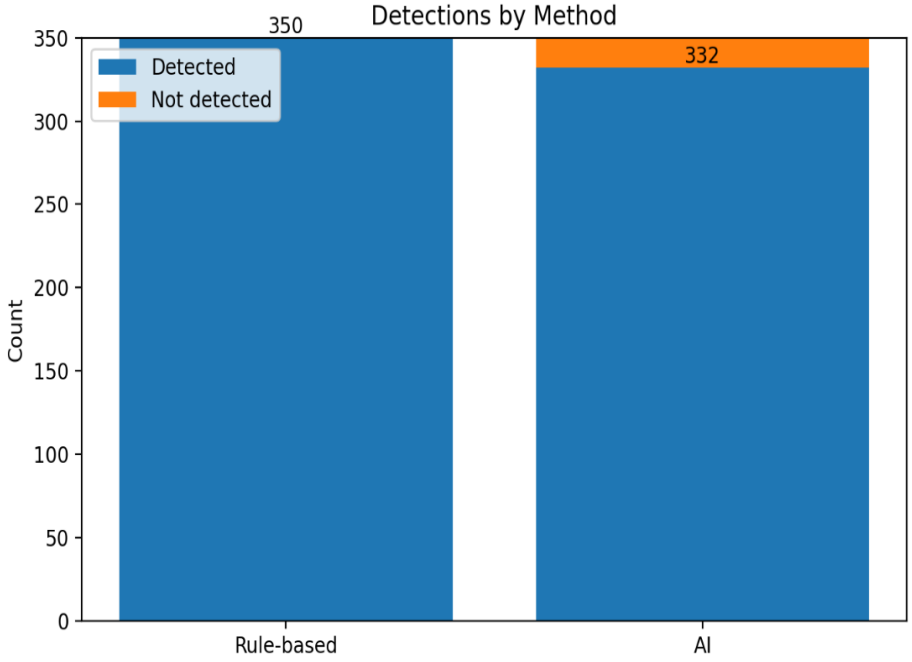

Fig 12: Rule vs AI comparison plot

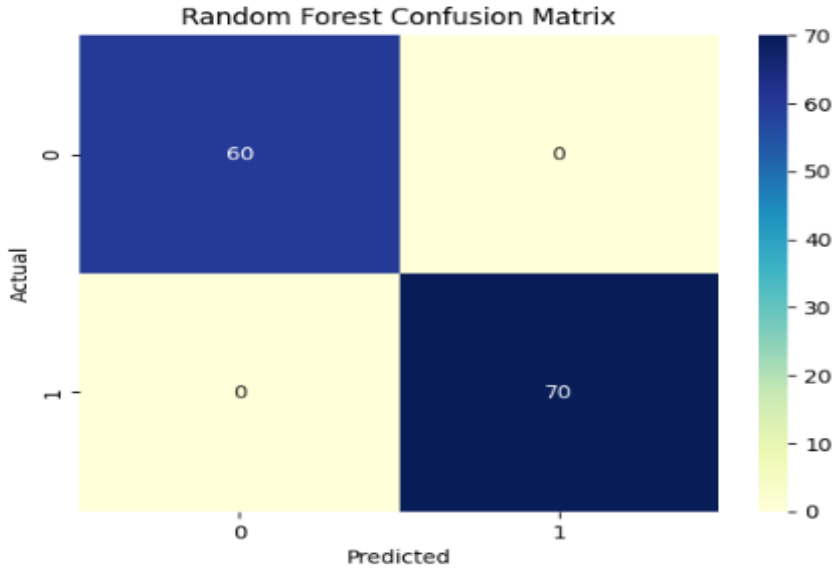

Fig 13: Confusion matrix

5. Security Considerations:

The tool is designed to ensure secure management of forensic data as well as to avoid any accidental running of malicious code.

- Safe data formats: To perform all the analysis, CSV-formatted outputs, which are produced by Volatility, are used instead of raw memory pictures. The payloads that are malware are absent from the datasets.
- Offline execution: The pipeline is fast whether with or without the internet, if dependencies are pre-installed.
- Dataset Integrity: Original evidence files or forensic data should be maintained on read only medium so that originality of evidence could not be tampered. All processing or generated files are isolated in a report of suspicious logs.
- Environment Isolation: It has been advised to install it on virtual environment (venv) to circumvent dependency problems. Because the meeting is a voluntary activity, optional use of Kali Linux virtual machine to provide additional security.

References

docs.jupyter.org. (n.d.). *Jupyter Project Documentation — Jupyter Documentation 4.1.1 alpha documentation*. [online] Available at: https://docs.jupyter.org/en/latest/.

docs.python.org. (n.d.). *3.8.17 Documentation*. [online] Available at: https://docs.python.org/3.8/.

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

MITRE (2025). *MITRE ATT&CK*. [online] Mitre.org. Available at: https://attack.mitre.org/.

NumPy (2022). *Overview — NumPy v1.19 Manual*. [online] numpy.org. Available at: https://numpy.org/doc/stable/.

Pandas (2024). *pandas documentation — pandas 1.0.1 documentation*. [online] pandas.pydata.org. Available at: https://pandas.pydata.org/docs/.

scikit-learn.org. (n.d.). *Documentation scikit-learn: machine learning in Python — scikit-learn 0.21.3 documentation*. [online] Available at: https://scikit-learn.org/0.21/documentation.html.

volatility3.readthedocs.io. (n.d.). *Volatility 3 — Volatility 3 1.1.1 documentation*. [online] Available at: https://volatility3.readthedocs.io/en/latest/index.html.

W3Schools (n.d.). *Matplotlib Pyplot*. [online] www.w3schools.com. Available at: https://www.w3schools.com/python/matplotlib_pyplot.asp.

www.unb.ca. (n.d.). *Malware Memory Analysis | Datasets | Canadian Institute for Cybersecurity | UNB*. [online] Available at: https://www.unb.ca/cic/datasets/malmem-2022.html.