COS70008 – Technology Innovation Project and Research

Developing a Web Based System for predicting and analysing Malicious Attacks using a Hybrid Machine Learning Model

Assignment – 1

Research Paper Review and Ethics Practices

Student Name: Arun Ragavendhar Arunachalam Palaniyappan

Student ID: 104837257

Date: 20 / 03 / 2025

Contents

Word Count (excluding Table of Contents, References, etc.): 2297

Abbreviations

CPS: Cyber Physical System
AI: Artificial Intelligence
ML: Machine Learning
SVM: Support Vector Machines
DHS: Department of Homeland Security
ICT: Information and Communication Technology
ACS: Australian Computer Society
CISA: Cybersecurity and Infrastructure Security Agency

List of Figures

1.Research Paper Review

1.1 Introduction

The rapid growth of digital technology has transformed society and industry, provided numerous benefits, but, has also introduced significant cybersecurity threats (Alenezi et al., 2020). Increasing dependence on digital platforms has led to more frequent and sophisticated cyberattacks, especially involving malware. Malware is malicious software designed to harm, disrupt, or gain unauthorized access to computer systems, threatening personal data, business operations, and critical infrastructure (Gandhi et al., 2021).

Cyber-physical systems (CPS), which merge digital and physical parts, are critical in industries such as healthcare, manufacturing, transportation, and energy (Chowdhury et al., 2023). A successful attack on these systems can have serious consequences. Malware attacks worldwide have spiked by around 358%, and ransomware alone has climbed by 435% since 2020, highlighting the urgent need for new detection strategies (Cybersecurity Ventures, 2023).

This preliminary literature review examines different malware types, detection methods (traditional and machine learning), and approaches to integrate these methods into a web application (Gandhi et al., 2023; Sharma et al., 2023). The goal is to identify effective techniques, uncover knowledge gaps, blend established methods with innovative improvements and establish a foundation for selecting a suitable cybersecurity solution (Nataraj et al., 2023).

1.2 Literature Review and Analysis

Malware comes in many types, each spreading and causing harm in its own way. According to Alenezi et al. (2020), the most common examples include:

Viruses: Attach themselves to good programs and copy themselves when those programs run.

Worms: Make copies of themselves automatically, without needing user help.

Trojans: Pretend to be safe software so that users install them by mistake.

Ransomware: Locks or encrypts files and demands money to unlock them, with a big example being the Colonial Pipeline attack explained by Beerman et al. (2021).

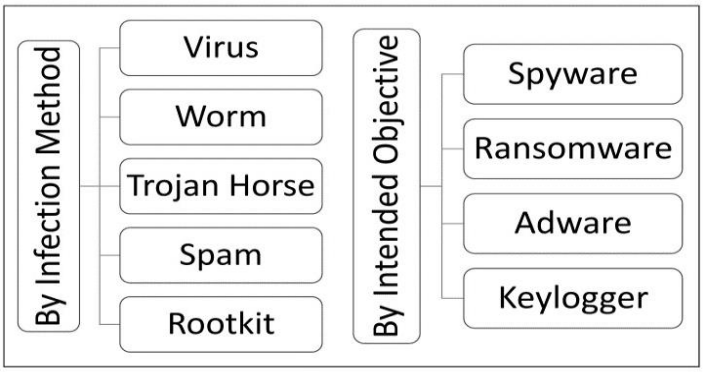

Figure 1: Classification of Malware (Alenezi et al., 2020)

Over the years, different **traditional detection** methods have been used to spot these threats. **Signature-based detection** is quick and accurate for threats that are already known but fails if the malware is totally new (often called a zero-day attack). Nataraj et al. (2023) explain that this approach involves looking for distinct patterns—or "signatures"—in harmful files. Another approach is **sandboxing,** where suspicious files run inside a safe environment so experts can watch for harmful behaviour (Nataraj et al., 2023). While sandboxing is very thorough, it can be time-intensive and use a lot of computing power. Meanwhile, **behavioural analysis** looks at live software activity to catch odd actions that might hint at malware. Although this method finds newly emerging threats, Sharma et al. (2023) note that it can accidentally flag safe programs as harmful.

As a result, usage of **Artificial Intelligence (AI)** and **Machine Learning (ML)** are becoming more common for smarter detection. **Supervised learning** (using Decision Trees, Random Forests, and Support Vector Machines, etc.) is very accurate if it has large, well-labelled datasets to learn from (Chowdhury et al., 2023). In contrast, **unsupervised methods** (such as autoencoders or clustering) excel at spotting strange patterns that might signal a hidden, unknown threat but may have more false positives as well (Lee et al., 2023).

An increasingly popular solution is to use **hybrid ML** models, which combine supervised and unsupervised techniques. These models give high accuracy for known malware and can also detect new or hidden types with reasonable success.

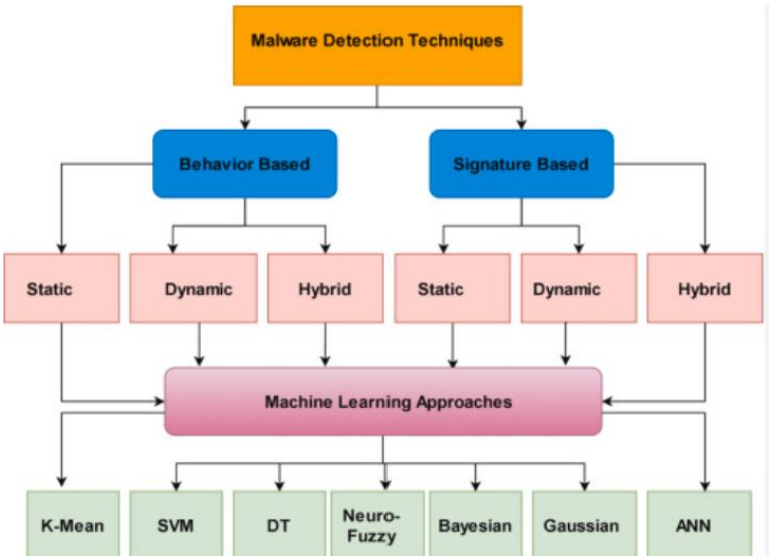

Figure 2: Different Techniques for Malware Detection (Lee et al., 2023)

Turning these detection ideas into a **web-based application** adds another layer of complexity. Roberts et al. (2023) suggest using **Flask (a lightweight Python framework)** because it is simple to set up and works easily with popular ML libraries. On the other hand, Mitchell et al. (2023) highlight the **MERN stack (MongoDB, Express.js, React.js, Node.js)** for its speed and ability to handle real-time data, but note that mixing Python-based AI tools with a JavaScript-driven setup can be tricky.

Finally, integrating classic antivirus or signature-based scanners directly into an app can be fast for spotting known malware but has limited power against new threats (Nataraj et al., 2023). Despite

growing evidence that hybrid ML solutions work well, there are still open questions about how these methods perform in real-time and on a large scale, especially when many users are involved.

1.3 Research Methodology

In line with research books focusing on key design principles (Creswell, 2014; Kothari, 2004) and standard machine learning guidelines (Bishop, 2006), this study follows a clear, step-by-step plan to investigate malware, review detection tools, and propose an effective method for a real-world web application. Initially, works by Alenezi et al. (2020), Gandhi et al. (2023), Sharma et al. (2023), Nataraj et al. (2023), and Chowdhury et al. (2023) were closely examined to understand how different approaches manage both well-known and emerging malware.

This method review synthesizes and narrows down insights from the literature review by contrasting several non-AI techniques with AI methods for malware detection and examines integration into web applications via Flask-built internal APIs.

A review of traditional (non AI) approaches reveals both strengths and trade-offs: signature-based detection quickly pinpoints recognized threats but lacks the flexibility to handle unfamiliar malware; sandboxing offers rich insights by testing suspicious files in a secure environment, though it demands considerable computing power; and behavioural analysis can spot emerging risks by tracking unusual actions in real time, but often raises false positives by flagging harmless software as malicious.

In contrast, AI-driven methods adapt better to changing threat patterns, particularly when supported by ample labelled data in a supervised learning scenario. Unsupervised methods detect odd behaviours without labelled data but tend to raise more false alarms. Gandhi et al. (2023) and Chowdhury et al. (2023) recommend combining these supervised and unsupervised methods into hybrid ML models to improve accuracy, adapt to unseen malware, and reduce needless alerts. Also, they stress that pairing strong feature extraction techniques with powerful classification algorithms can help to solve typical issues faced by traditional detection methods such as lower and inconsistent detection rates, false alarms, and failure to recognise evolved or brand-new malware.

Putting this plan into action starts with gathering a wide set of malware samples from open repositories, then carefully cleaning and preparing those files. Nataraj et al. (2023) explain that both static (analysing file properties without execution) and dynamic analysis (observing software behaviour in controlled environments) approaches are useful for feature extraction, ensuring a complete view of how the malware operates. After that, Sharma et al. (2023) propose testing these methods in a simulation of real-world malware attack scenarios to confirm they are reliable and effective.

From these steps, the inference is that hybrid AI models deliver strong results by blending supervised classification (for known threats) and unsupervised anomaly detection (for unknown threats). However, one main obstacle might be the diversity of malware datasets, because using too few or very similar samples can limit the model's overall accuracy and flexibility. Still, the expected finding is that the hybrid ML approach can outperform others in spotting both familiar and never-before-seen attacks, while also limiting false alarms.

2.Ethics Practices

2.1 Case Study Scenario

In 2021, the **DarkSide** hacking group carried out one of the most disruptive cyberattacks in recent memory by targeting American oil pipeline giant, **Colonial Pipelines**. They broke into the company's network using an old VPN account that did not have multi-factor authentication, allowing them to encrypt critical files and force a shutdown of the main fuel supply system (Beerman et al., 2021). This led to major fuel shortages in the southeastern United States, causing public panic, economic turmoil, and long queues at gas stations. Feeling the pressure, Colonial Pipelines paid a **$4.4 million** ransom in Bitcoin. Unfortunately, the decryption tool they received was slow and only partly helpful, so the company had to rely on its own backups to fully restore operations (Hall, 2021). This incident revealed big security weaknesses in vital infrastructure and showed how a single oversight can spark a nationwide crisis.

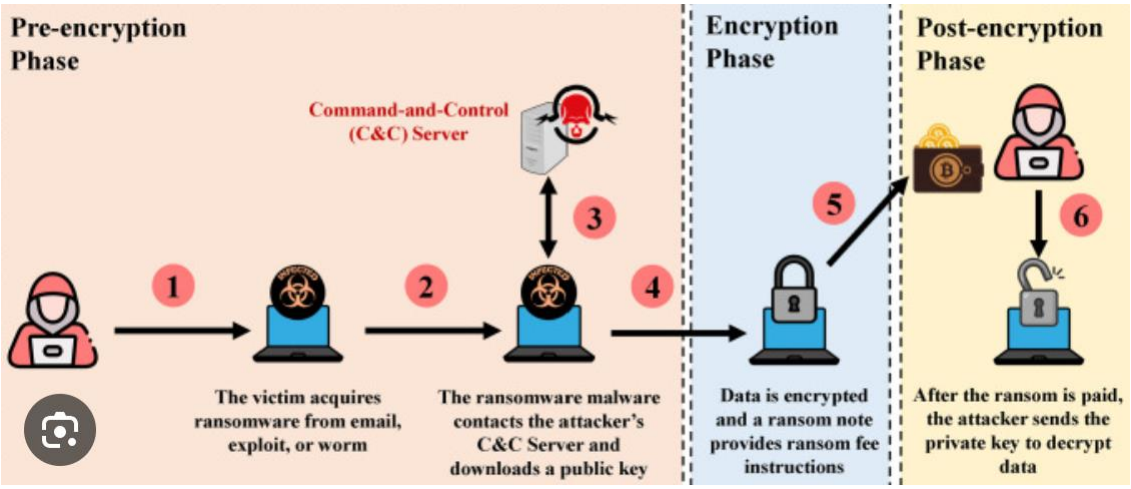

Figure 3: The effect of the Darkside Ransomware on the Colonial Pipelines CPS (Beerman et al., 2021)

2.2 Ethical Dilemma

A key ethical question emerged from the Colonial Pipeline breach: should organizations pay cybercriminals to quickly resume services, or should they refuse, knowing it may lead to prolonged shutdowns? Although paying the ransom helped the pipeline get back to work faster, it also rewarded criminal behaviour and possibly encouraged future attacks (Reeder, 2021). On top of that, there is no guarantee that paying will bring systems back online, as DarkSide's faulty decryption tool proved in this case.

Another dilemma was Colonial Pipeline's slow disclosure of the attack. Withholding details from government agencies and the public allowed rumours to spread and fuelled unnecessary panic buying. This raises concerns about **corporate responsibility**: do companies managing critical infrastructure have a duty to immediately share information on cyber incidents to protect the public?

Finally, the incident sparked a debate over whether governments should completely ban ransom payments. Some believe this would remove the financial motive behind ransomware, while others

argue it might lead to even longer disruptions of essential services. Balancing national security, public welfare, and discouraging cyber extortion remains a challenging policy issue (Hall et al., 2021).

2.3 ICT Involvement

Information and Communication Technology (ICT) experts and cybersecurity professionals played critical roles in fighting and investigating the Colonial Pipeline attack:

Federal Agencies: The FBI led the investigation and retrieved part of the ransom by tracing Bitcoin transactions (Beerman et al., 2021). They worked with the Department of Homeland Security (DHS) to help other key infrastructure operators guard against similar attacks.

Cybersecurity Firms and Ethical Hackers: These groups examined the systems that were breached, identified weaknesses, and studied DarkSide's ransomware methods. They shared what they learned with the wider security community.

IT and Incident Response Teams: Colonial Pipeline's internal staff and outside consultants teamed up to isolate infected systems, restore backups, and improve security measures, curbing further damage.

Regulatory and Government Agencies: The Cybersecurity and Infrastructure Security Agency (CISA) and others reviewed existing rules and recommended new policies that require prompt incident reporting and stricter cybersecurity safeguards (Reeder, 2021).

2.4 Application of the ACS Code of Ethics

Following the **Australian Computer Society (ACS) Code of Ethics** is vital for fair and responsible work in malware research and AI-based security: Each one of the ACS ethical codes were breached in the Colonial Pipelines ransomware incident, from the company's side.

1.2.1 Public Interest: Cybersecurity solutions must always protect the public and critical services. The shutdown of Colonial Pipelines caused major fuel shortages and economic problems, showing why early threat detection is so important to avoid large-scale harm.

1.2.2 Quality of Life: Cybersecurity solutions should also help keep everyday life running smoothly. The attack disrupted daily routines, caused uncertainty, and made people lose confidence.

1.2.4 Competence: Security professionals need to stay updated on the latest threats and solutions. This breach shows that even small mistakes, like an unused VPN account, can lead to big problems.

1.2.3 Honesty & 1.2.6 Professionalism: Open and timely communication about cyber incidents is crucial for public trust. The delay in reporting the Colonial Pipeline attack caused panic and worsened its impact. Following ethical standards means sharing true information quickly, even if it might hurt a company's reputation in the short term, so that everyone stays informed and lessons are learned.

The learnings from the mistakes made in this issue can be a wake-up call for all Cyber Physical Systems across the globe.

2.5 Adopting and maintaining Equity and Accessibility

Inclusive design is vital to ensure systems are fair and accessible for everyone. For example, **ACS Case No. 25** shows Ilnaz facing cultural and religious challenges when forced to share an office with a male colleague, highlighting the need for systems that respect diverse cultural norms. **ACS Case No. 31** illustrates Peter's difficulty in observing his prayers at work, emphasizing the importance of accommodating religious practices. **ACS Case No. 32** demonstrates how Anna designed a user interface for remote Aboriginal communities by using culturally sensitive images, ensuring the system is user-friendly for disadvantaged groups. Additionally, **ACS Case No. 28** features Katherina's voluntary support for disability groups, underscoring ICT's role in enhancing quality of life for the disadvantaged. Finally, **ACS Case No. 24** exposes how inconsistent disability coding in legacy systems can lead to poor policy decisions, stressing the need for accurate and inclusive data systems. Together, these case studies reinforce the importance of designing research and systems that promote equity and accessibility for all users (ACS Code of Professional Conduct Case Studies, 2014).

2.5 Conclusion

This research underscores the urgent need for robust cybersecurity measures, spotlighting the broad range of malware types and exploring both time-tested and next-generation detection strategies. After weighing traditional, supervised, unsupervised, and hybrid machine learning methods, it becomes clear that hybrid solutions deliver the strongest combination of accuracy, adaptability, and real-time responsiveness. Events like the Colonial Pipeline ransomware attack demonstrate the high stakes of inadequate defense mechanisms and reinforce the demand for dependable, ethically sound detection tools.

Despite these advantages, several challenges remain. Achieving instant detection without overloading system resources is a considerable hurdle. Additionally, limited dataset diversity can hinder a model's ability to detect novel threats, and scaling small-scale proofs of concept into fully operational developments is often more complex than anticipated. To address these gaps, this study advocates hybrid machine learning as a key strategy, blending supervised and unsupervised approaches within a web-based framework. Technologies such as Flask enable quick deployment and seamless AI integration, tackling many of the pitfalls found in single-method or traditional solutions. At the same time, adhering to the ACS Code of Ethics ensures transparent, responsible data handling, plus alignment with professional standards.

Building on these insights, upcoming research will delve more deeply into project specifics like malware dataset collection, cleaning, and preparation, as well as the specific algorithms and AI model combinations best suited for hybrid approaches. Detailed steps for integrating these solutions into a web application will also be outlined, giving practical guidance on how to move from prototype to full-scale development.

4.References

1. Alenezi, M. N., Alabdulrazzaq, H., Alshaher, A. A., & Alkharang, M. M. (2020). Evolution of malware threats and techniques: A review. *International Journal of Communication Networks and Information Security, 12*(3), 326–334.
2. Beerman, J., Berent, D., Falter, Z., & Bhunia, S. (2021). A review of the Colonial Pipeline ransomware attack. *Cybersecurity Journal, 12*(3), 245–261.
3. Chowdhury, D., Stevens, L., & Grant, P. (2023). Cyber-physical systems anomaly detection using machine learning. *IEEE Transactions on Cybersecurity, 38*(4), 523–541.
4. Cybersecurity Ventures. (2023). *2023 official cybercrime report.* Retrieved from https://cybersecurityventures.com
5. Gandhi, V., Kumar, S., & Kumar, S. (2023). Detection and classification of malware using machine learning techniques.
6. Hall, T. (2021). Examining the Colonial Pipeline ransomware incident and its impact on national security. *International Journal of Cyber Threat Intelligence, 9*(2), 87–105.
7. Lee, C., Wang, H., & Kim, S. (2023). Anomaly-based malware detection using autoencoders and decision trees. *International Conference on Cyber Threats, 28*(1), 215–232.
8. Mitchell, S., Brown, L., & Carter, P. (2023). Evaluating MERN stack for AI integration. *Web Systems and Security Journal, 17*(4), 89–106.
9. Nataraj, L., Yegneswaran, V., & Porras, P. (2023). Dynamic pattern recognition using signature analysis.
10. Reeder, J. R. (2021). Cybersecurity's Pearl Harbor moment: Lessons learned from the Colonial Pipeline ransomware attack. *U.S. Cyber Defense Review, 7*(4), 112–130.
11. Roberts, T., Lee, J., & Adams, R. (2023). Efficiency of Flask in AI model deployment. *International Journal of Web Applications, 34*(2), 101–119.
12. Sharma, P., Kaur, J., & Singh, H. (2023). Malware detection using behaviour analysis.
13. Creswell, J. W. (2014). *Research design: Qualitative, quantitative, and mixed methods approaches* (4th ed.). SAGE Publications.
14. Kothari, C. (2004). *Research methodology: Methods and techniques* (2nd ed.). New Age International.
15. Bishop, C. M. (2006). *Pattern recognition and machine learning*. Springer.
16. Australian Computer Society. (2014). *ACS Code of Professional Conduct case studies (Version 2.1)*. Australian Computer Society.