2025

ML Based IDS

INTRUSION DETECTION USING MACHINE LEARNING

LIAM DOOCEY

SOUTH EAST TECHNOLOGICAL UNIVERSITY | 20094146

Table of Contents

Introduction

Background Research

In the rapidly evolving landscape of cybersecurity and networking, the integration of Artificial Intelligence (AI) and Machine Learning (ML) into Intrusion Detection Systems (IDS) has emerged as a game changing strategy to enhance threat detection and response capabilities. Traditional IDS', primarily signature-based, often struggle to identify sophisticated attacks.

1. Evolution of Intrusion Detection Systems

Intrusion Detection Systems are a key component of a network security system. They are systems that detect and analyse network transmissions to see if an attack is potentially occurring. Traditionally, Intrusion Detection Systems operate as below:

- **Signature-Based IDS**: These systems detect attacks by comparing network traffic against a database of known threat signatures. While effective against known threats, they are limited in identifying new attacks.
- **Anomaly-Based IDS**: These systems establish a baseline of normal network behaviour and flag deviations as potential threats. This approach offers the advantage of detecting unknown attacks but often suffers from higher false positive rates.

The limitations present in these traditional systems have prompted the exploration of AI and ML techniques to enhance detection accuracy and adaptability.

2. Integration of Machine Learning in IDS

The application of ML in IDS aims to improve the detection of both known and unknown threats by learning patterns associated with malicious activities. Various ML algorithms have been employed, including:

- **Supervised Learning Models:** Algorithms such as Support Vector Machines (SVM), Decision Trees, and Random Forests are trained on labelled datasets to classify network activities as benign or malicious. For instance, a study utilized the UNSW-NB15 dataset to evaluate the performance of Naive Bayes, SVM, and K-Nearest Neighbours (KNN) algorithms in intrusion detection, highlighting the efficacy of SVM in achieving high accuracy (Tahri, 2022)
- **Unsupervised Learning Models:** Techniques like clustering and anomaly detection are used to identify patterns in unlabelled data, facilitating the discovery of novel attack vectors.
- **Deep Learning Models:** Architectures such as Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM) networks have been explored for their ability to model complex temporal patterns in network traffic. Research has demonstrated that LSTM-RNN models can effectively classify multiple attack types using the NSL-KDD dataset (K. Azarudeen, 2024)

Despite these advancements, challenges are still there, particularly concerning the balance between detection rates and false positives. Moreover, the dynamic nature of network traffic necessitates continuous model updates and the incorporation of contextual threat intelligence.

3. Role of Threat Intelligence in Enhancing IDS

Threat intelligence involves the collection and analysis of data regarding current and emerging threats, providing context that can enhance IDS performance. The integration of threat intelligence into IDS enables:

- **Enrichment of Detection Capabilities**: By incorporating indicators of compromise (IOCs) from threat intelligence feeds, IDS can more effectively identify and respond to known threat patterns.
- **Proactive Defence Posture**: Access to real-time threat intelligence allows organizations to anticipate and mitigate potential attacks before they manifest.

A practical example is the integration of Cyber Threat Intelligence (CTI) into Security Information and Event Management (SIEM) systems, which facilitates the automated configuration of detection tools in industrial networks (Tim Ackerman, 2023) This approach underscores the importance of seamless integration between threat intelligence platforms and security infrastructure to enhance situational awareness and response capabilities.

4. Development of Real-Time Alert Mechanisms

Timely detection of and response to security incidents are paramount in minimizing potential damage. The implementation of real-time alert mechanisms within AI-powered IDS involves:

- **Automated Notification Systems**: Utilizing APIs and integration capabilities to seamlessly incorporate threat intelligence reporting within existing communication channels, ensuring that security teams receive immediate notifications of potential threats (Heaton, 2024)

The challenge lies in designing alert systems that are both sensitive to genuine threats and resilient against generating excessive false positives, which can lead to alert fatigue.

5. Challenges and Future Directions

While the integration of AI and threat intelligence into IDS offers significant benefits, several challenges remain:

- **Data Quality and Quantity**: Effective ML models require large volumes of high-quality, labelled data, which can be difficult to obtain in the cybersecurity domain.
- **Evasion Techniques**: Adversaries continually develop methods to evade detection, necessitating adaptive and resilient IDS models.

- **Integration Complexity**: Seamlessly integrating diverse threat intelligence sources and ensuring interoperability with existing security infrastructure can be complex and resource intensive.

6. Industry Examples

The two industry examples I came across in my research were **Darktrace** who offer an array of different ML based security solutions such as cloud and email protection from scams as well as my main concern, network monitoring, and **Cortex XDR** operated by Palo Alto Networks who offer a wide spanned protection net with "Laser Point Accuracy" to defend against the most sophisticated attacks.

- **Darktrace**: Darktrace's network analyser acts as if it is a human security analyst using reasoning to determine the source of a network connection and decide if it comes as legitimate communication or otherwise. (Darktrace, N/A)
- **Cortex XDR**: Cortex XDR works slightly different by taking known attacks and using them to identify common actions between them that might occur in newer attacks that have not been signatured yet. This is like the approach I want to take. (jdelio, 2019)

Specification & Requirements

Functional Requirements

1. **Real-Time Traffic Monitoring:**
   Continuously monitor network traffic to detect anomalies and suspicious patterns as they occur.
2. **Advanced Threat Detection Algorithms:**
   Incorporate AI/ML techniques—including supervised (SVM, Decision Trees, etc.), unsupervised (clustering, anomaly detection), and deep learning (RNN/LSTM) models—to identify both known and novel attack vectors.
3. **Threat Intelligence Integration:**
   Seamlessly integrate multiple threat intelligence feeds to enrich detection capabilities and update the IDS with the latest indicators of compromise.
4. **Automated Alert Generation:**
   Provide immediate, automated alerts via configurable channels (e.g., email, SMS, APIs) when potential threats are identified.
5. **Dynamic Model Updating:**
   Enable continuous model training and updates using new data and feedback to adapt to evolving network behaviours and adversarial tactics.

6. **Comprehensive Logging and Reporting:**
   Generate detailed logs and incident reports to facilitate forensic analysis and compliance audits.

7. **API and System Integration:**
   Offer robust APIs for seamless integration with other security tools (e.g., SIEM platforms) and existing network infrastructure for unified threat management.

Non-Functional Requirements

1. **High Performance and Low Latency:**
   Ensure the system processes large volumes of data in real-time without causing network delays.

2. **Scalability:**
   Design the IDS to scale efficiently as network traffic and data volume increase, accommodating both small and large enterprise environments.

3. **Reliability and Availability:**
   Maintain continuous operation with high uptime, including robust failover and redundancy mechanisms to handle system failures.

4. **Data Security:**
   Protect sensitive information within detection logs and threat intelligence feeds, ensuring data integrity and confidentiality through secure storage and transmission protocols.

5. **User-Friendly Interface:**
   Provide an intuitive dashboard and management interface that facilitates easy configuration, monitoring, and reporting for security teams.

6. **Adaptability:**
   Ensure the system can quickly adapt to emerging threats and integrate new detection algorithms with minimal manual intervention.

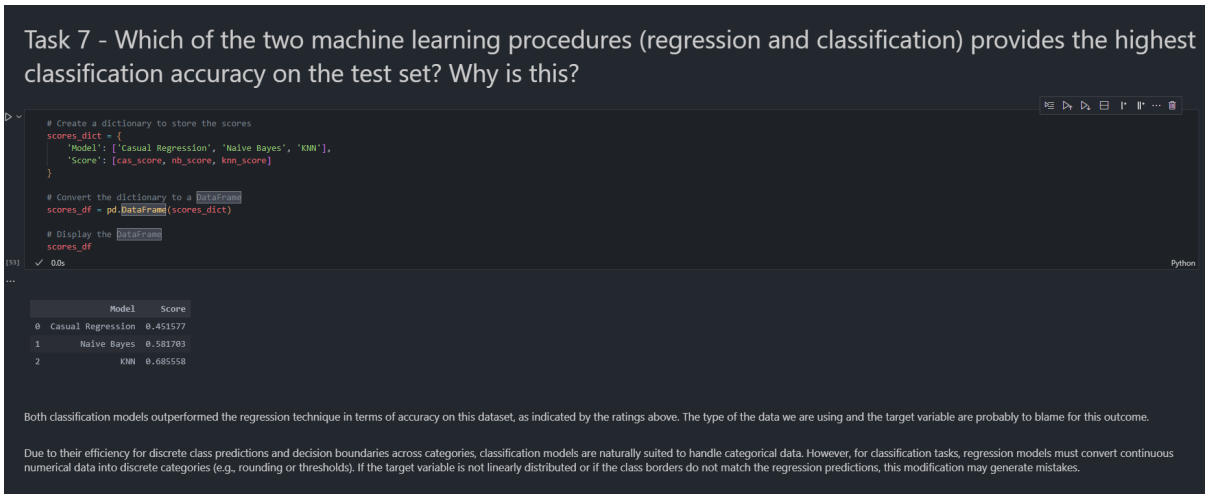
7. **Interoperability:**
   Guarantee seamless compatibility with existing security tools, network infrastructures, and industry-standard protocols for holistic threat management.

8. **Maintainability:**
   Design the system with modular components and clear documentation to simplify ongoing maintenance, troubleshooting, and updates.

Tech Feasibility

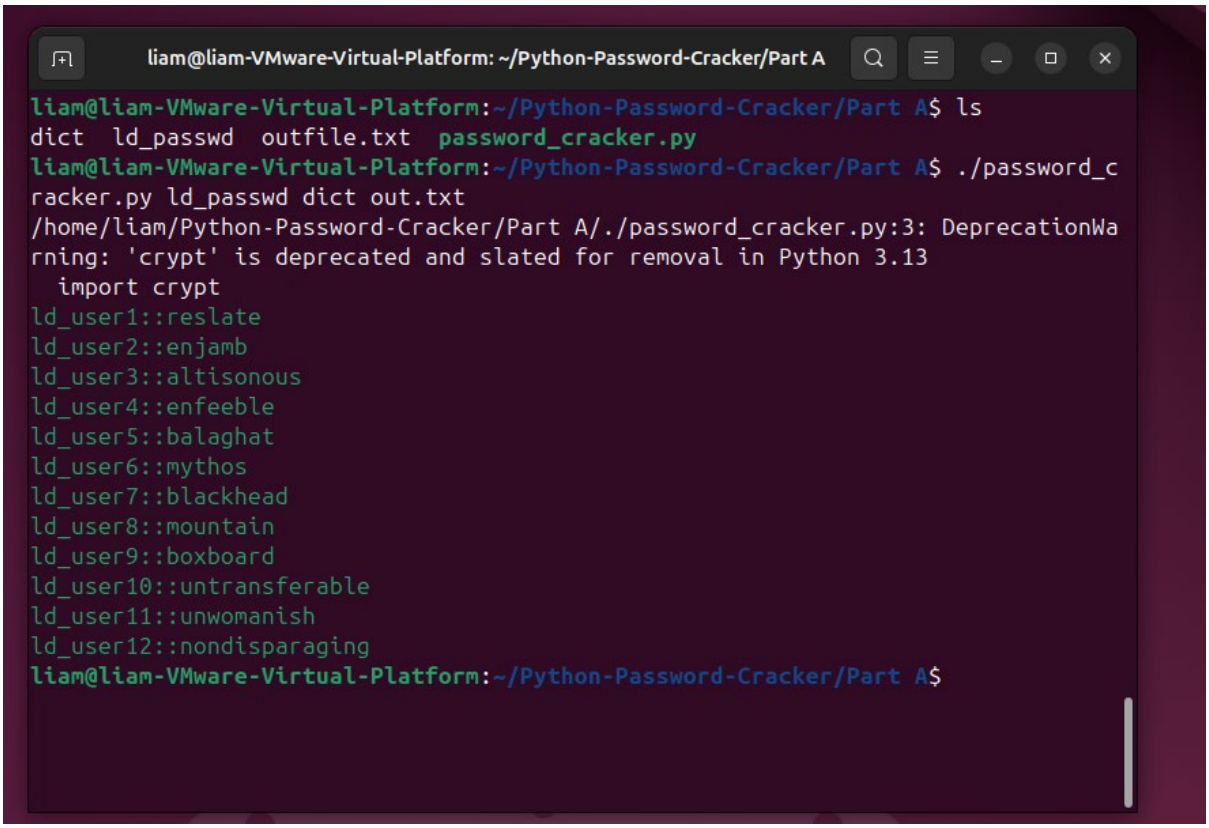
Sci-Kit

To build my Machine Learning model I will be using sci-kit as I have previously used it before within my Datamining modules.



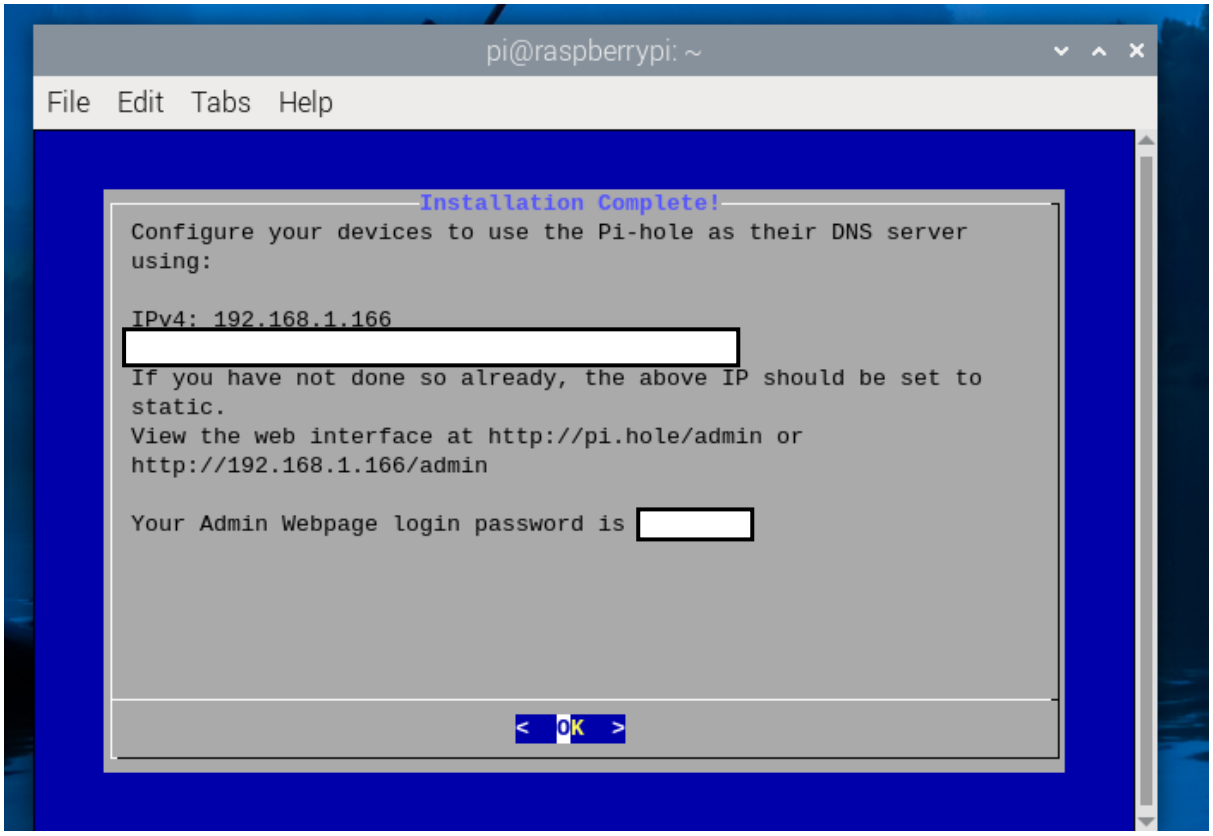
Python

Python will be the core component of this project due to its light-weight nature and flexibility with system interaction alongside the fact I have used it quite a lot in the past.



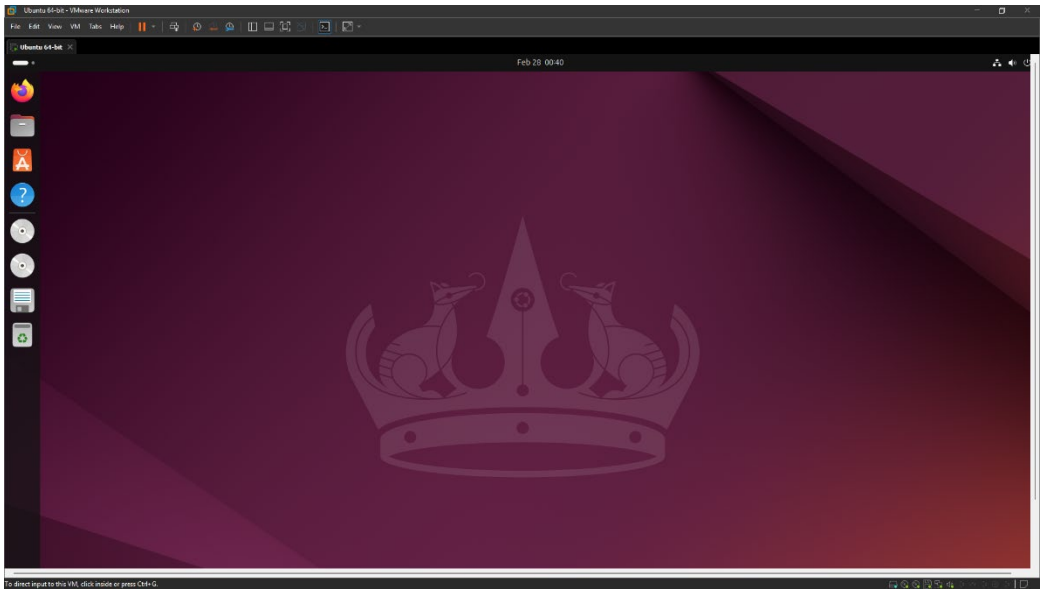
Raspberry-Pi

In terms of deployment, I have chosen to use my raspberry pi 4 as a cheap and accessible deployment mechanism for testing purposes, the advantage of being able to connect via SSH or VNC will help during the development stage and the small compact form factor makes it ideal for home deployment.



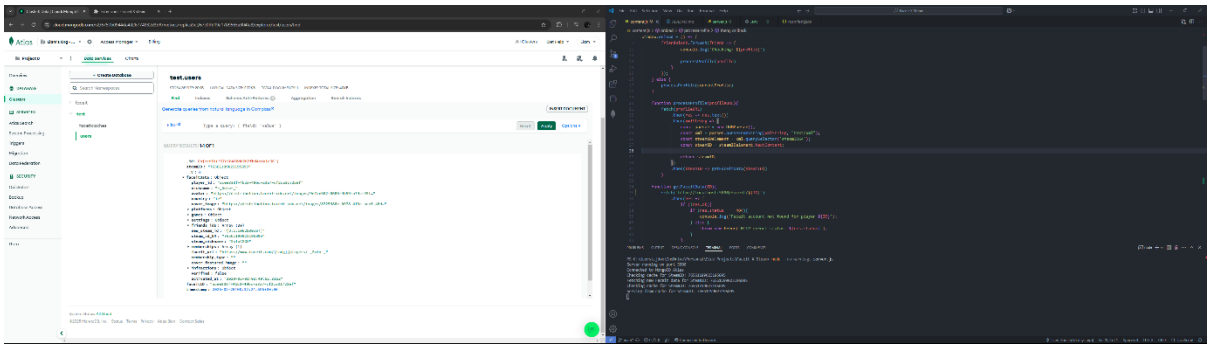
VMware

For development I will be mostly working from a Linux platform, rather than wiping my current windows machine or duel booting I have opted to create an Ubuntu VM using VMware giving me the option to switch from windows to a Unix based system at the click of an application while also having the ability to store system files on the cloud while switching from my desktop to my laptop on the go.



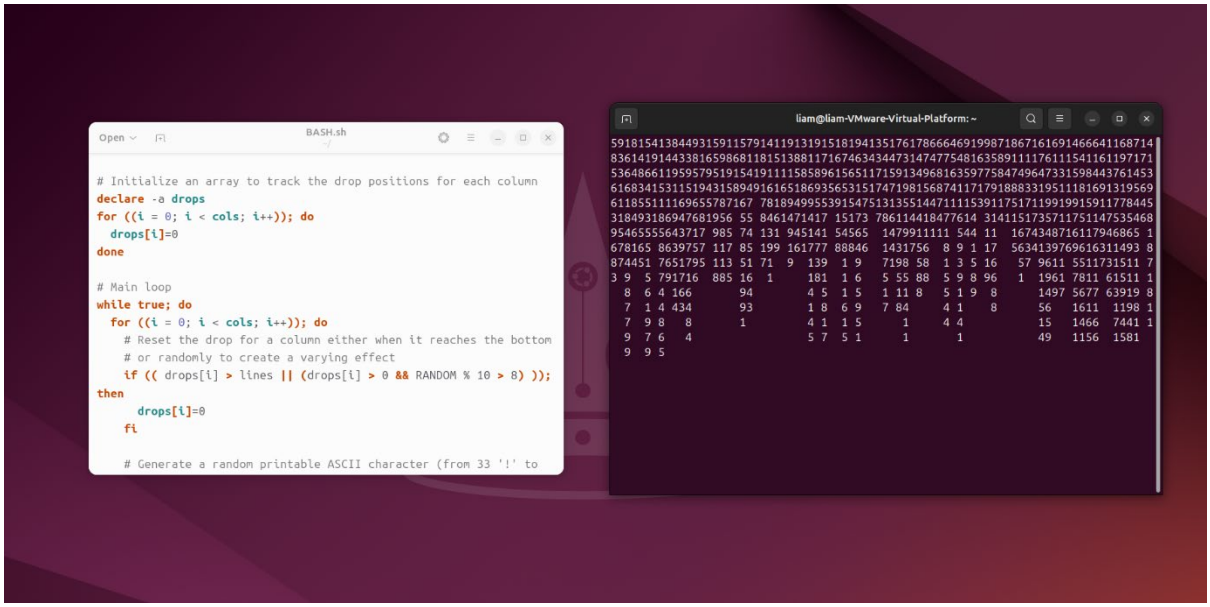
MongoDB

For the backend database I have chosen to use MongoDB due to its flexibility and ease of use for large datasets while also having experience in using it in previous universities modules.



Bash Scripts

Simple bash scripts will also be used to aid in the set up / maintenance of the system while accessing from remote etc.



Methodology

Overview of Software Development Methodologies

Software development projects require a structured approach to ensure efficient execution and delivery. The two most used methodologies are **Agile** and **Waterfall**. Each methodology has its unique strengths, and the choice depends on project requirements, team structure, and business goals.

Agile vs. Waterfall:

Waterfall Methodology

The **Waterfall** model follows a linear, sequential approach, where each phase (e.g., requirements, design, implementation, testing, deployment) is completed before moving to the next. It is best suited for projects with well-defined requirements and minimal expected changes.

Pros:

- Clear project structure and documentation.
- Predictable timelines and budget.
- Easier to manage for fixed-scope projects.

Cons:

- Inflexible to changes after development starts.
- Late-stage testing increases the risk of major issues.
- Limited stakeholder involvement after the initial phase.

Agile Methodology

The **Agile** methodology is an iterative and incremental approach that promotes flexibility, collaboration, and continuous feedback. Work is divided into small iterations, called **sprints**, allowing teams to adjust to changing requirements.

Pros:

- Adaptability to changing requirements.
- Continuous stakeholder feedback improves quality.
- Faster delivery with incremental releases.

Cons:

- Requires active team collaboration and customer involvement.
- Less predictable scope, timeline, and cost.
- Documentation may be less comprehensive.

Choice of Methodology & Justification

For this project, **Agile** has been chosen due to the following reasons:

1. **Flexibility & Iterative Development:** Agile allows incremental improvements based on feedback, making it ideal for evolving project requirements.
2. **Collaboration:** Regular interactions with supervisor ensure alignment with project needs.
3. **Risk Mitigation:** Continuous testing and feedback reduce the risk of major defects at the end of the development cycle.
4. **Faster Delivery:** Frequent iterations ensure early delivery of functional components.

Introduction to Scrum Framework

Scrum is a widely adopted Agile framework that organizes work into sprints and emphasizes team collaboration. It consists of the following key components:

- **Scrum Team:** Includes a Product Owner, Scrum Master, and Development Team.
- **Sprints:** Short development cycles (typically 2-4 weeks) with clear goals.
- **Daily Standups:** Brief team meetings to discuss progress and roadblocks.
- **Sprint Planning & Retrospective:** Meetings to define goals before the sprint and reflect on improvements after completion.

Burndown Charts in Agile

A **burndown chart** is a visual representation of work progress within a sprint. It helps teams track the remaining work versus the planned timeline.

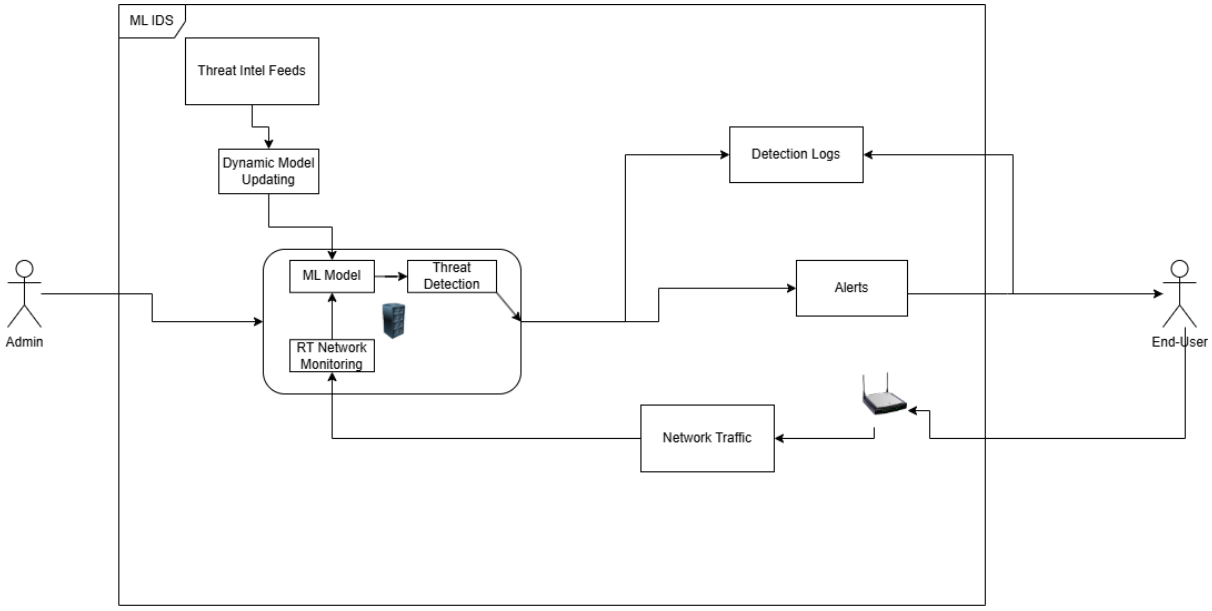
- **X-axis:** Represents time (sprint duration).
- **Y-axis:** Represents remaining work (tasks, story points, or hours).
- **Usage:** Teams use burndown charts to monitor sprint progress and identify potential delays.

Conclusion

The Agile process will be implemented in the form of Scrum as it provides the necessary flexibility and continuous feedback while also being one of the most effective methods for delivering incremental results. Sprints and burndown charts will ensure transparency and progress while also enabling me to course correct as requirements evolve.

Design

Use Case Diagram



User Stories

Traffic Analysis

- *As a security analyst, I want to monitor real-time network traffic so that I can detect intrusions as they happen.*
- *As an end-user, I want the IDS to categorize attacks (e.g., DoS, phishing, malware) so that I can prioritize my responses.*
- *As a network engineer, I want the IDS to automatically block malicious IPs so that I can prevent further attacks.*

Alert System

- *As an end-user, I want to receive alerts when suspicious activity is detected so that I can take immediate action.*

Dashboard

- *As a security analyst, I want a GUI dashboard to visualize IDS activity so that I can analyse* trends easily.

Model Training

- *As a developer, I want to train my ML model on labelled intrusion datasets so that it can improve its detection accuracy.*

Tasks

Traffic Analysis	- **Network packet capturing** - **Extract needed info (IP etc.)** - **Block suspicious sources** - **Store data for ML**
Alerts	- **Determine alert thresholds** - **Add real-time alert function (SMS etc.)**
Model Training	- **Choose ML algorithm** - **Train model on a suitable dataset (KDD99)** - **Test model score**
Dashboard	- **Design interface (Figma)** - **Add real-time activity chart** - **Add logs page**

References

Darktrace, N/A. *Darktrace: Network.* [Online]
Available at: https://darktrace.com/products/network
[Accessed 26 February 2025].

Heaton, M., 2024. *Microsoft Security Copilot - Threat Intelligence Use Cases - Part 1.* [Online]
Available at: https://www.bluevoyant.com/blog/microsoft-copilot-for-threat-intelligence
[Accessed 13 February 2025].

jdelio, 2019. *What is Cortex XDR?.* [Online]
Available at: https://live.paloaltonetworks.com/t5/community-blogs/what-is-cortex-xdr/ba-p/251610
[Accessed 26 February 2025].

K. Azarudeen, D. G. G. R. B. S. R. V., 2024. Intrusion Detection System Using Machine Learning by RNN Method. *International Conference on Environmental Development Using Computer Science (ICECS'24),* 491(E3S Web Conf.), p. 10.

Tahri, R. &. B. Y. &. J. A. &. A. L., 2022. *Intrusion Detection System Using machine learning Algorithms.* [Online]
Available at:
https://www.researchgate.net/publication/361112461_Intrusion_Detection_System_Using_machine_learning_Algorithms

Tim Ackerman, M. K. J. K., 2023. *Integration of Cyber Threat Intelligence into Security Onion and Malcolm for the use case of industrial networks,* s.l.: ResearchGate.