

Proposal Project OMG

by Orin McGinn

Submission date: 16-Oct-2025 02:39PM (UTC+0100)

Submission ID: 265577126

File name: 49873_Orin_McGinn_Proposal_Project_OMG_695044_1290581219.docx (202.15K)

Word count: 1618

Character count: 9840

Project Proposal: Anomaly-Based Intrusion Detection System

Programme	BSc (Hons) Computing (Computer Networks and Cybersecurity)
Module	Network Programmability
Module Code	ACCB5029
Supervisor	TBA
Student	Orin McGinn
Student Number	2304621
Date Due	12/11/2024

Abstract

Signature IDSs cannot spot zero-day tricks or weird attacks. This plan involves combining a supervised model (XGBoost) [2] with an unsupervised autoencoder [4]. First, we clean the CIC-IDS2017 data [1] – pull out bad rows, make flow features, and fix the class imbalance with SMOTE. Then we let XGBoost catch worst traffic while the auto-encoder learns what normal looks like and flags odd stuff. A weighted vote ties them together. The code will be in Python, aiming for almost real-time alerts. The project will test on a hold-out set and compare recall, precision, F1, false-positive rate and AUC to a default Snort setup [5]. The result should be a repeatable prototype that out-performs rule-based IDSs but still keeps false alarms tolerable, giving a more proactive shield for modern networks.

1 Contents

2	Introduction and Rationale	4
3	Statement of the Problem	5
3.1	Core limitation of signature-based IDSs	5
3.2	Research Context and Contribution	5
4	Aims and Objectives	6
4.1	Project Aim	6
4.2	Project Objectives (Sub-Tasks)	6
5	Methodology	7
5.1	Research Design	7
5.2	Data Acquisition and Pre-processing (Tasks 1 & 2)	7
5.3	Model Selection and Training (Tasks 3 & 4)	7
5.4	Prototype Development (Task 5)	7
5.5	Evaluation Strategy (Tasks 7 & 9)	8
6	Resources and Project Management	9
6.1	resource requirements	9
7	Proposed Timescale	10
8	Conclusion	11
9	Ethical and IP Considerations	12
10	References	13

2 Introduction and Rationale

The rise of IoT gadgets, cloud services, and software-defined networking has left a huge attack surface. Old IDSs were built for static signatures, not for encrypted payloads or fast-changing port hopping. Because they only react to known patterns, they miss brand-new exploits and can't keep up with how fast threats evolve.

Machine learning may fill the gap. By learning what normal traffic looks like, it can point out anything that strays far from that norm – even stuff no one has seen before. That's the "unknown unknowns" idea: catch attacks which were unknown to exist. Combining a supervised classifier (good recall, easy to understand) with an unsupervised detector (good at finding novel patterns) could give a system that's both strong and flexible.

3 Statement of the Problem.

3.1 Core limitation of signature-based IDSs

Tools like Snort [5] match packets against a list of known bad signatures. They work okay for old malware, but when attackers craft zero-day, polymorphic or deliberately hidden code, those signatures fail. The update lag means attackers get a window to slip in. Small firms especially feel the pain of constantly maintaining signature lists.

3.2 Research Context and Contribution

Many papers talk about data-driven security [6], yet most stay theoretical or hard to repeat. This project aims to give a concrete, open-source solution that meets the degree-program requirement “Development of a novel, technically robust solution to a recognised problem” [3]. By delivering full code and docs, developers add something anyone can take, tweak and test against industry baselines.

4 Aims and Objectives

4.1 Project Aim

Build, run and evaluate a hybrid ML-based IDS that beats a standard signature-based system at finding both known and unknown attacks.

4.2 Project Objectives (Sub-Tasks)

- Survey what anomaly-detection methods exist [6].
- Pull and clean the CIC-IDS2017 data [1], add features and balance classes.
- Code XGBoost (supervised) [2] and an auto-encoder (unsupervised) [4] and blend them with weighted voting.
- Make a Python prototype that can read live traffic and raise alerts quickly.
- Compare the prototype to Snort [5] using recall, precision, F1, false-positive rate and AUC.
- Write complete docs and package the experiment so others can repeat it.

5 Methodology

5.1 Research Design

The investigator will run experiments that pit hybrid IDS against a baseline Snort instance [5] running its default rule set. Both will see the same test traffic from CIC-IDS2017 [1], so the comparison stays fair.

5.2 Data Acquisition and Pre-processing (Tasks 1 & 2)

CIC-IDS2017 [1] provides labelled captures of various attacks (DDoS, brute-force, web attacks) as well as normal traffic. We'll turn raw PCAPs into flow records with CICFlowMeter, ending up with around 80 numeric features per flow:

1. **Handle missing values** – drop rows with infinite or NaN, or replace with the median.
2. **Scale features** – use Min-Max scaling so the neural net learns faster.
3. **Trim redundant columns** – drop highly correlated features to keep the model readable.
4. **Balance classes** – apply SMOTE to generate synthetic attack rows, and undersample benign rows so the model does not just learn “benign”.

After cleaning, the set will be split: 70 % training, 15 % validation, 15 % test, keeping class ratios the same.

5.3 Model Selection and Training (Tasks 3 & 4)

Supervised part: XGBoost [2] will be a binary classifier (malicious or not). Settings will be tuned with Bayesian optimisation, trying to maximise recall while not letting precision tank. Early-stopping on validation will keep it from over-fitting.

Unsupervised part: A deep auto-encoder [4] will only see benign traffic. It will learn a compact shape of normal data; the reconstruction error becomes an anomaly score. A threshold will be picked at the 95th percentile of validation errors to try to keep false warnings low.

Ensemble: A weighted-vote (e.g., $0.7 \times \text{XGBoost} + 0.3 \times \text{auto-encoder}$) will blend the two scores. Weights will be adjusted empirically to favour recall but not crush precision.

All steps will be wrapped in scikit-learn's Pipeline so the whole thing is repeatable [3].

5.4 Prototype Development (Task 5)

The detector will be built in Python. scapy grabs packets, pandas handles feature prep, and the models run in separate threads, so capture, feature creation, and inference don't block each other. Goal is < 10 ms per flow. When something is flagged, a JSON alert will be pushed through a tiny REST API, ready for SIEM tools.

5.5 Evaluation Strategy (Tasks 7 & 9)

On the held-out test set, the following will be calculated:

- **Recall** – how many attacks we catch [3].
- **Precision** – how many alerts are real attacks [3].
- **F1-Score** – balance of the two [3].
- **False-Positive Rate** – proportion of benign traffic wrongfully flagged [3].
- **AUC** – overall discrimination power [3].

Paired t-tests will be run on batch results to see if improvements over Snort [5] are statistically meaningful. Then stress-testing of the system with traffic from 10 Kbps up to 1 Gbps will be performed, logging latency and CPU use, to prove near-real-time claims.

6 Resources and Project Management

6.1 resource requirements

Resource Type	Specific Tool/Library	Purpose
Environment	Virtual Machine (Kali Linux, Parrot, Red Hat, etc.)	Isolated, secure development lab.
Data Analysis	Python, Pandas, NumPy, Matplotlib	Data loading, cleaning, and visualisation.
Machine Learning	Scikit-learn, XGBoost, TensorFlow/Keras	Model training, tuning, and ensemble construction.
Network Interface	Scapy	Packet capture and feature extraction.

6.2 Ethical and IP Considerations

The project will only use publicly available, anonymised network flow data (CIC-IDS2017) [1], mitigating privacy concerns. Ethical consideration will focus on securing the development environment (VM isolation) to prevent the unintended release of malicious code or data. The author acknowledges the University's Intellectual Property Policy and confirm that any significant use of University resources will be discussed with my supervisor to ensure compliance.

7 Proposed Timescale

Phase	Key Tasks	Duration (Months)	Deadline
Phase 1	Proposal, Literature Review (Initial)	September – October	Oct 20, 2025
Phase 2	Data Prep, Feature Build, Initial Model Training	November – January	Jan 16, 2026 (Interim Viva)
Phase 3	Final Model Tuning, Prototype Build, Testing	February – April	Apr 30, 2026 (Final Draft)
Phase 4	Final Report, Viva, Exhibition	May	May 1, 2026 (Final Submission)

8 Conclusion

Sticking with static signature rules is not enough in today's fast-changing threat world. By joining a high-recall gradient-boosted tree [2] with an auto-encoder [4] that spots odd traffic, it is hoped to give a detector that catches more attacks while keeping false alarms at a reasonable level. The prototype will be a working, reproducible piece of software [3] that can be copied, improved, and eventually run in real enterprise environments. Future steps could add live-traffic hooks, adaptive thresholds and tighter SIEM integration, turning the research artefact into a production-ready defence tool.

9 Ethical and IP Considerations

The project will utilise publicly available, anonymised network flow data (CIC-IDS2017) [1], thereby mitigating privacy concerns. Ethical consideration will focus on securing the development environment (VM isolation) to prevent the unintended release of malicious code or data. The author acknowledges the University's Intellectual Property Policy and confirm that any significant use of University resources will be discussed with my supervisor to ensure compliance

10 References

[1]

I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization," in *Proceedings of the 4th International Conference on Information Systems Security and Privacy (ICISSP)*, Funchal, Portugal: SCITEPRESS – Science and Technology Publications, Lda., Jan. 2018. Accessed: Oct. 02, 2025. [Online]. Available: https://www.researchgate.net/publication/322870768_Toward_Generating_a_New_Intrusion_Detection_Dataset_and_Intrusion_Traffic_Characterization

[2]

T. Chen and C. Guestrin, "XGBoost: a Scalable Tree Boosting System," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Francisco, CA: Association for Computing Machinery (ACM), Aug. 2016. Accessed: Oct. 02, 2025. [Online]. Available: https://www.researchgate.net/publication/310824798_XGBoost_A_Scalable_Tree_Boosting_System

[3]

M. P. Marder, *Research Methods for Science*. Cambridge ; New York, Ny: Cambridge University Press, 2011. Accessed: Oct. 05, 2025. [Online]. Available: chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://assets.cambridge.org/9780521145848/frontmatter/9780521145848_frontmatter.pdf

[4]

I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, Massachusetts: The MIT Press, 2016. Accessed: Oct. 05, 2025. [Online]. Available: https://www.researchgate.net/publication/320703571_Ian_Goodfellow_Yoshua_Bengio_and_Aaron_Courville_Deep_learning_The_MIT_Press_2016_800_pp_ISBN_0262035618

[5]

R. Anilkumar, K. Akash, K. R Dodiya, and D. Patel, "Snort in Action: Tailoring Intrusion Detection to Combat Network Threats," *International Journal of Education Science Technology and Engineering (IJESTE)*, vol. 13, no. 10, Oct. 2024.

[6]

D. T. Mane *et al.*, "Detection of Anomaly Using Machine Learning: a Comprehensive Survey," *International Journal of Emerging Technology and Advanced Engineering*, vol. 12, no. 11, Nov. 2022, Accessed: Oct. 05, 2025. [Online]. Available: https://www.researchgate.net/publication/365060809_Detection_of_Anomaly_using_Machine_Learning_A_Comprehensive_Survey

Proposal Project OMG

GRADEMARK REPORT

FINAL GRADE

GENERAL COMMENTS

72/100

PROJECT PROPOSAL – ACCA6020 INDEPENDENT PROJECT

Student Orin McGinn
Mark 72

Intro / A&O (400 words)		
Project title; Identification of a specific problem to be investigated; Well-defined aims and SMART objectives; Explaining why this project is worth doing; Discussion of what the project needs to do in order to be a success; Description any deliverables that will be produced at the end of the project	20	14
Clear and specific title, supported with well identified limitations of current process. Rationale is strong but needs a clearer of the success and deliverables of the project.		

Background to the Project / Literature Review (400 words)		
Description of how the work to be done by in the project relates to the work identified in the literature review; Discussion of any ethical issues raised by the project Identifying what has already been done within the field; Topic familiarity; Brief literature review using appropriate sources – 6-8 journal papers and book chapters rather than just websites; Selection of relevant models & concepts Good context of IDS limitations and the potentials rendered by developing machine learning methods. Literature review is brief and needs a clearer link between existing detection techniques and gaps in the current process.	20	13
Project Design (500 words)		
Clear statement of which development methods are to be used and why they are appropriate; Clear statement of which tools are to be used and why they are appropriate; Clear description of how data collected will be analysed Detailed methodology and strategy. Outlining the	30	20

choices made in the proposal, though could have included a diagram for the system architecture to give better visual representation.		
Resources / Planning (200 words)		
List of any deliverables; Clear diagram showing both timing and resource use; Completed ethics form; Limited and relevant appendices Strong list of tools and libraries that will be used as well as the ethical considerations of IP address compliance. Deliverables could have been listed here in detail and though a timeframe table was provided a visual representation with a Gantt charted would have been a good representation of the time for each stage of the project.	15	10

Document Presentation		
Correct spelling, grammar and punctuation; Flow of narrative/fluency; Coherence; Word processed ; Correct use of referencing with bibliography; Strict word limit Well-structured sections with headings and logical flow. References formatted and credible. Minor graphical inconsistencies in terms of the spaces and bold formatting, with a hyphen seen in the first couple of paragraphs this could be seen as use of AI to write them.	15	11
TOTAL	100	68

PAGE 1
PAGE 2
PAGE 3
PAGE 4
PAGE 5
PAGE 6
PAGE 7

