Queen's University Belfast

School of Electronics, Electrical Engineering and Computer Science

ELE8095 Individual Research Project

Project Title: - High Data Rate Realtime ML Network Anomaly Detection

Project Progress Report

Student Name: - Nilesh Tejrao Dongare

Student Number: - 40456755

Project Reference: - Industry_NVIDIA02

Academic Supervisor: - Dr. Sakir Sezer

Date: 7th July 2025

Declaration of Academic Integrity

I declare that I have read the University guidelines on plagiarism – https://www.qub.ac.uk/directorates/AcademicStudentAffairs/AcademicAffairs/AppealsComplaintsandMisconduct/AcademicOffences/Student-Guide/1 - and that this submission is my own original work. No part of it has been submitted for any other assignment and I have acknowledged in my notes and bibliography all written and electronic sources used, all sources are correctly attributed, and the contribution of any AI technologies is fully acknowledged.

Student Signature	Date of Submission
Nilesh Tejrao Dongare	7th July 2025

Abstract

This report examines pre-filtering techniques using simple machine learning algorithms to manage large volumes of Layer-5 traffic, targeting 100x traffic reduction, zero false negatives, and a minimum throughput of 50 Gb/s on NVIDIA L40s GPUs. We focus on open-source datasets containing SQLi and XSS attacks, with the UniEmbed Dataset (2025) as a primary candidate, and evaluate machine learning algorithms. The review integrates Nginx TCP/UDP proxy for processing decrypted payloads. A high-level architecture diagram illustrates the pre-filtering pipeline, from traffic ingestion to ML based detection. Finding highlights the UniEmbed Datasets comprehensive HTTP coverage and ML algorithm suitability for high-throughput, data processing. Challenges including dataset accessibility and XSS coverage, are discussed, alongside future research and implementation direction.

Table of Contents

1.Introduction

The project under consideration will implement a system of pre-filtering higher Layer traffic in datacenters by applying simplest machine learning (ML) algorithms in python leveraging RAPIDS cuML, cuDF, and PyTorch on the NVIDIA L40s GPU. The system uses NGINXs TCP/UDP proxy and uses decrypted payloads, focuses SQLi and XSS detection. The major goals that will be pursued are 100x of the traffic reduction by detecting unique attack-related properties, ensuring zero false negatives to prevent undetected threats, and maintaining high-throughput 50 Gb/s.

This systematic literature review evaluates open-source datasets and ML techniques, focusing on their suitability for Layer-5 or other Layers traffic analysis. We assess datasets like UniEmbed(2025), HTTPParamDataset(2024), and others for their coverage if benign and malicious traffic, recency and compatibility with python and machine learning frameworks, including RAPIDS cuML, cuDF, PyTorch, Scikit-learn, MLperf, TensorRT andTriton, TensorFlow, Weka, XGBoost, LightGBH, are compared for performance, and we choose two tools RAPIDS and PyTorch to implementation. We also identified research gaps and future directions. The review provides a foundation for developing a high-throughput, zero false negative detection system aligned with the project objectives.

This project is justified by the need for scalability, security, and cost efficiency in datacenters. Datacenters handle massive traffic volumes, making real-time inspection infeasible without pre-filtering, which reduces traffic by 100x for subsequent deep inspection. Zero false negatives esure no threats e.g. SQL injection, XSS are missed, protecting datacenter integrity. High throughput 50 Gb/s ensures real-time processing, achieved through simple ML algorithms and GPU acceleration. The goal is to balance catching nearly all threats while avoiding flagging too much safe traffic, so the system does not get overwhelmed.

2.Problem Statements

Large network traffic levels are difficult for datacenters to manage while maintaining strong security against attacks such as SQL injection and Cross-site Scripting (XSS). The enormous volume of traffic often reaches beyond 100x Gb/s, demanding efficient pre-filtering mechanisms to reduce the volume of data requiring detailed inspection without compromising detection accuracy. Traditional approaches struggle to achieve high throughput and zero false negatives while maintaining significant flow of traffic.

The objectives of this project to develop a high-performance pre-filtering system using simple machine learning algorithms to analyze reassembled and decrypted Layer-5 payloads (e.g. HTTP, SQL queries) from open-source datasets containing SQL injection and XSS attacks. Leveraging

NGINXs TCP/UDP proxy server features and a TLS proxy for handling unencrypted traffic, the system aims to eventually a 100x reduction in traffic volume by identifying unique attack-related properties. The solution must ensure zero false negatives to prevent undetected threats, tolerate high false positive and achieve a minimum throughput of 5-10 Gb/s, with a targeted of 50 Gb/s, using NVIDIA L40s GPU.

3.Literature Review

The application of machine learning techniques to cybersecurity challenges has gained significant traction in recent years. Supervised learning algorithms, including Support Vector Machine (SVM), Random Forest and Neural Networks have shown promise in various security application [1]. These approaches offer the advantage of learning complex patterns from training data without requiring explicit rule definition.

Simple ML algorithm dominated due to their suitability for real processing. Decision trees [8] and Random Forest [1] were favoured for their interpretability and ability to process features like keyword frequency and payload structure. Logistic Regression [9] achieved near zero false negatives through threshold tunning, ideal for binary classification. K-Nearest (K-NN) [10] was effective for pattern matching but less scalable due to computational complexity. Support Vector Machines (SVM) [11] Provided high accuracy but required significant resources, limiting their use without GPU acceleration.

Machine learning approaches to SQL injection detection have shown promising result in academic research. Feature extraction techniques focusing on SQL syntax analysis, character frequency distribution and query structure have been successfully applied to classification task [13]. However, the computational overhead of these approaches has limited their application in high throughput environments.

The research paper [3] , particularly NVIDIA BlueFied-3. It addresses the limitations of traditional ruled-based and transformer-based methods, such as low adaptability and high latency, respectively. This study highlights the potential for integrating efficient, classical ML-based SQLi detection into DPU hardware, offering scalable and low latency defense mechanisms for modern datacenters [3].

The paper [4] introduces a cascaded detection combining classical Ml-based NLP techniques and transformers model (e.g. BERT) to optimize both speed and accuracy for SQLi detection in high-traffic environments. The hybrid model effectively balances the computational cost and

detection accuracy, enabling scalable SQLi protection suitable for real-time applications in cloud and edge datacenters. [4]

The protection of the Layer-5 traffic, which supports business services and user interactions, is challenging and requires strong defense against the application-layer threats, such as SQL injection. The study of the SQL injection attack involves the detection mechanism on the use of machine learning with the focus on prioritization of the potential threats [5]. In particular, the paper deals with tautology-based SQL injection attacks on the basis of supervised learning. A pre-processing mechanism that involves verification of input strings is presented in order to improve accuracy in the detection. Through classifier learning e.g., Support Vector Machine (SVM) and K- Nearest Neighbors (KNN) as well as Random Forest etc. are used to distinguish the malicious patterns correctly [6]. This method emphasizes how intelligent models can be used in the security of dynamic applications that face users.

A novel way to speed up the request of incoming packets by utilizing the large number of processing units available in a general-purpose hardware (i.e., the GPU) was proposed by Lakshminarayana et al. They wrote the work on how to exploit the inherent parallelism of packet inspection tasks to increase performance. The author introduced and tested the three models, such as data parallel, function parallel and a combination of two previous ones, which is aimed at improving the different faces of packet filtering [7]. The author has shown that there is a tremendous improvement of performance by using the GPU acceleration in terms of fast network security application [7].

The paper [25] introduces UniEmbed, a unified machine learning-based framework for detecting XSS and SQL injection attacks using multi-level natural language feature fusion. The method combines embedding from word2vec, FastText, and universal sentence Encode (USE) to create rich semantic representation of web inputs. They tested multiple ML classifiers including MLP, SVM and ensemble methods. The MLP classifier achieved the best performance with over 99.8% accuracy and F1- scores across all datasets. The method they demonstrated low false positive and false negative rates and outperformed traditional feature extraction approaches as well as transformer-based methods like BERT in term of efficiency [25].

Cross-Site Scripting (XSS) detection present unique challenges due to variety of attack vectors and encoding techniques exploited by attacker. The Research approach [12] range from simple string matching to sophisticated deep learning models using attention mechanisms. The complexity of XSS detection is compounded by the need to understand HTML and JavaScript

context, leading to research into specialized parsing and analysis techniques. Recent work BERT-based achieves 97 percentage accuracy but requires significant computational resources [12].

Pre-filtering systems aim to reduce computational load on expensive security systems by quickly identifying and filtering benign traffic. Research in this area focuses on achieving optimal trade-offs between filtering efficiency and detection accuracy. Pipeline-based architecture process traffic through sequential stages, with each stage performing increasingly sophisticated analysis. Parallel processing approaches leverage multiple computing units to achieve higher throughput [7]. Hybrid systems combine multiple processing paradigms to optimize both performance and accuracy.

The Paper [26] introduces a hybrid approach to detect SQL injection and XSS attacks in web applications. These attacks, exploiting URLs, constitute over 80% of URL-based threats. The proposed framework integrates content matching for preprocessing and a CNN-LSTM deep learning model for enhanced detection. It comprises four modules: data acquisition, pre-processing, model training, and testing. Word2Vec transforms data into vectors, capturing semantic relationships, which feed into the CNN-LSTM model to extract local and global features. The model achieves high accuracies: 97.95% for normal statements, 99.3% for SQL injection, and 97.5% for XSS attacks. Compared to resource-heavy syntax analysis and BERT, content matching is simpler and more interpretable [26].

The Paper [27] proposes a hybrid model combining XGBoost and CNN for feature extraction with LSTM for classification to improve intrusion detection systems (IDS). It addresses challenges like low accuracy and high false positives in detecting new network attacks. Using benchmark datasets (CIC-IDS 2017, UNSW-NB15, NSL-KDD, WSN-DS), the model achieves high detection rates and accuracies, up to 98.55% for binary classification on CIC-IDS 2017. XGBoost-LSTM excels in handling categorical data, while CNN-LSTM captures sequential patterns, enhancing generalization. Preprocessing includes normalization and feature selection to reduce computational complexity. The model outperforms traditional methods like MCA-LSTM and RNN, with lower false positive rates. Future work aims to address zero-day attacks and improve performance on imbalanced datasets using updated datasets and advanced techniques like SMOTE [27].

The paper [28] advocates a system that does this in IoT networks through machine learning intrusion detection systems (IDS). It compares 6 algorithms (decision tree, random forest, KNN, SVM, naive Bayes and MLP) on 4 data sets (UNSW-NB15, BOT-IoT, ToN-IoT, and Edge-IoT) in the multi-class classification problem [28]. Pre-processing methods such as feature selection,

normalization and data cleaning treat the problems of quality in data when there are missing values and concerns about imbalances. The ML and RF obtained the best accuracies, reaching 99.97 percent on both ToN-IoT and Edge-IoT and bettered the previous studies. The framework considers equal trade-off between model and data quality. In future directions, the creation of lightweight models and powerful pre-processing to circumvent the changing cyber-threats targeting IoT will be developed [28].

3.1. Literature Survey Table

Paper Title/Year	Focus Area	Methodology	Dataset	Key Results	Significance
An Investigation of ML Algorithms for High-bandwidth SQLi Detection Utilising BlueField-3 DPU Technology	SQLi detection	Classical ML (20 models, e.g., PassiveAggressiveClassifier, XGBoost)	Kaggle SQL Injection (30,609)	Passive Aggressive Classifier: 99.78% accuracy, 0.3 µs/sample latency	Demonstrates scalable, low-latency SQLi detection on DPUs for high-speed environments
Advancing SQLi Detection for High-Speed Data Centers: A Novel Approach Using Cascaded NLP	SQLi detection	Cascaded ML + Transformers (Passive Aggressive, ELECTRA)	Kaggle SQL Injection (30,609)	99.86% accuracy, 20x faster than transformer-only models	Balances speed and accuracy for real-time applications in data centers
High-throughput & GPU Acceleration, Acceleration of Packet Filtering	Packet filtering	GPU acceleration (CUDA, batch processing)	Trec Aquaint Corpus	Significant performance improvements over CPU-based implementations	Pioneering work on GPU-based network packet filtering

Using GPGPU					
IDS Technical Report	Anomaly-based IDS	Not specified	WEB-IDS23	Addresses lack of fine-grained labels and realistic traffic overlap	Improves layer 5 traffic analysis for IDS
SQL Injection Attack: Detection, Prioritization & Prevention (2024)	SQLi detection	ML approaches	Kaggle SQL Injection	Covers multiple SQLi types, manually verified labels	Focuses on prioritization and robust ML training
Acceleration of Packet Filtering Using GPGPU	Packet filtering	AHO-Corasick algorithm, CUDA on NVIDIA GPUs	Modeled traffic (Ixia Explorer)	Demonstrated speedup compared to CPU-based implementation	Enhances network IDS performance using GPU acceleration
AI Techniques for SQL Injection Attack Detection	SQLi detection (tautology-based)	Supervised ML (SVM, KNN, Random Forest)	Not specified	Effective input string validation for detecting SQLi	Emphasizes supervised learning for specific attack types
Deep Learning in Cybersecurity: A Hybrid BERT-LSTM Network for SQLi Detection (2024)	SQLi detection	Hybrid DL (BERT + LSTM)	HTTPparams Dataset	96.3% precision, 95.8% F1-score, high computational efficiency	Outperforms traditional ML for sophisticated SQLi attacks
Deep-learning Technique-	Web attacks	LSTM-based layered firewall	ISCX IDS 2012, CIC-DDOS	Efficient two-layer detection for high traffic	Suitable for WAF deploymen

Enabled Web Application Firewall for Web Attack Detection (2023)	(DDoS, SQLi, XSS)		2019, CSIC 2010	and content-based filtering	t in web application s
Encrypted Network Traffic Analysis and Classification Utilizing ML (2024)	Encrypted traffic analysis	ML techniques (threat, anomaly, predictive, behavioral analysis)	Not specified	Identifies patterns without decryption	Addresses limitations of traditional traffic analysis due to encryption
Enhancing Intrusion Detection: A Hybrid Machine and Deep Learning Approach (2024)	Intrusion detection	Hybrid (XGBoost, CNN, LSTM)	CICIDS 2017, NSL-KDD	High accuracy, low false positives, scalable for new threats	Combines ML and DL for improved IDS performanc e
High Throughput Filtering Using FPGA-Acceleration (2013)	Packet filtering	FPGA-based architecture	TREC Document Collection	Improves speed and scalability over software solutions	Enhances real-time packet inspection for security appliances
Network Traffic Classification Model Based on Attention Mechanisms and Spatiotemporal Features (2023)	Traffic classificati on	DL (LSTM, CNN, attention mechanisms)	USTC-TFC2016, YouTube encrypted	>90% accuracy, reduces manual feature engineering	Addresses encrypted and obfuscated traffic classificati on

Secure and Timely GPU Execution in Cyber-physical Systems (2023)	GPU security in cyber-physical systems	CPU-GPU co-scheduling, secure pre-emption	Not specified	Ensures real-time guarantees and resists malicious interference	Critical for safety-critical systems like autonomous devices
Securing Web Applications Against XSS and SQLi Using Deep Learning (2024)	SQLi and XSS detection	Hybrid DL (CNN + LSTM)	Multiple datasets	Up to 99.84% accuracy, low false positive rates	Suitable for IDS and WAF deployment
SQL Injection Detection Using ML Techniques and Multiple Data Sources (2018)	SQLi detection	ML (rule-based, decision trees, neural networks)	Web and database logs	Improved accuracy by combining data sources	Avoids deep learning overhead while maintaining effectiveness
SQL Injection Detection Using ML	SQLi detection	Gradient Boosting Classifier	Plain-Text Dataset	Strong performance with ensemble learning, low computational requirements	Emphasizes efficient feature extraction and classification
UniEmbed: A Novel Approach to Detect XSS and SQLi Attacks (2025)	XSS and SQLi detection	ML (word2vec, FastText, USE, MLP, SVM, ensemble)	Multiple datasets	>99.8% accuracy, low false positives/negatives	Outperforms traditional and transformer-based methods in efficiency

4. Motivation and Methodologies

The rapid increase in the internet traffic and the growing complexity of cyber attacks have posed significant challenges for datacenter security systems. Traditional method that relies on matching specific patterns struggle to handle the vast and diverse nature of modern network traffic, highlighting the urgent need for smarter pre-filtering systems. These systems can reduce the processing burden while still effectively detecting threats.

Machine Learning (ML) has emerged as a promising approach for pre-filtering network traffic, enabling rapid identification of malicious patterns while reducing the volume of data requiring detailed inspection. Simple ML algorithms, such as Decision Trees, Random Forests, and Logistic Regression, are particularly suited for real-time applications due to their low computational complexity and ability to process structured features like keyword frequency, payload entropy, and query structure. Studies "High-Throughput Pre-Filtering with Random Forests" have demonstrated that ML-based pre-filtering can achieve traffic reduction ratios of 50-200X by classifying benign traffic, though achieving zero false negatives remains challenging. False negatives, undetected threats, are critical to avoid, as they allow attacks to bypass initial defenses, while high false positives are tolerable if followed by a second-round inspection. [1]

Proxy technologies, such as NGINXs TCP/UDP stream module, play a vital role in intercepting and processing Layer-5 payloads. NGINXs ability to handle TLS termination enables decryption of encrypted traffic, exposing payloads for ML analysis. However, real-time TLS decryption at high throughput (e.g., 50 Gb/s) poses significant computational overhead, necessitating hardware acceleration. Graphics Processing Units (GPUs), such as the NVIDIA L40s, offer parallel processing capabilities to accelerate ML inference, with recent studies reporting throughputs of 5-10 Gb/s using GPU-optimized libraries like RAPIDS CuML.

The motivation for this project stems from the critical need to secure datacenters against SQL injection and XSS attacks while managing unprecedented traffic volumes. Traditional IDS solutions are inadequate for real-time processing at scale, often introducing latency or missing novel attack patterns. ML-based pre-filtering offers a scalable alternative, enabling rapid identification of malicious traffic to reduce the load on secondary inspection systems. Achieving a 100x traffic reduction is essential to make subsequent deep packet inspection feasible, while zero false negatives ensure no threats are missed, safeguarding datacenter integrity.

4.1. Machine Learning Techniques for Pre-Filtering SQLi and XSS Attacks

Simple ML algorithms are critical for pre-filtering Layer-5 traffic at high throughput. We research tools, GPU-accelerated frameworks, recall, throughput, and NGINX integration. and chose to evaluate two tools i.e. RAPIDS and PyTorch. This research provides a comprehensive analysis of why RAPIDS cuML and PyTorch are selected for the project, focuses on their technical capabilities, performance metrics, and integration for pre-filtering higher layer traffic to detect SQLi and XSS attacks in datcenters. The project aims to achieve 100x reduction, zero-false negatives and 50 Gb/s throughput on NVIDIA L40s GPU, using ML, NGINIX and HTTP payloads from datasets. The analysis is grounded in peer-viewed studies, technical documentation and web sources.

4.1.1 RAPIDS cuML

RAPIDS cuML is a GPU-accelerated ML library with a scikit-learn-like API, optimized for tabular data processing on NVIDIA L40s GPUs [16]. It supports algorithms like Random Forest, SVM, and MLP, achieving 10-50x speedups over CPU-based methods. The ‘cuml. benchmark‘ module measures throughput and recall, critical for validating 50 Gb/s and zero false negatives. cuML integrates with ‘cudf‘ for preprocessing HTTP payloads (e.g., URL length, tokens), mak ing it ideal for the UniEmbed Dataset [2]. Its Dask integration ensures scalability for datacenter traffic. Its strengths is high throughput 50 Gb/s, built-in benchmarking, scalable with Dask, python compatible. cuML’s built-in module provides real-time metrics for recall, throughput, and accuracy, facilitating validation of project objectives [16]. It is integrated with NGINIX via FastAPI ensures low latency, real-time inference, supporting the project’s real-time processing needs[16].

Key performance metrics:

- Throughput: Benchmarks show 50 Gb/s on L40s GPUs for Random Forest, meeting the project high-throughput goal [16].
- Recall: Achieves 99.82 % recall on UniEmbed, ensuring zero false negatives via class weighting, crucial for detecting all SQLi/XSS attacks [25].
- Scalability: Supports multi-GPU processing via Dask, ideal for datacenter-scale traffic [16].

4.1.2 PyTorch

PyTorch provides flexible GPU-accelerated deep learning, suitable for processing raw HTTP payloads with NLP models (e.g., Transformers, LSTMs) [17]. It achieves high throughput with TorchServe and supports NGINX integration but requires custom benchmarking and a steeper learning curve for tabular data. PyTorch is dynamic computation graph enables flexible model design, critical for handling sequential data in

HTTP payloads, and achieves 40–60 Gb/s throughput with optimization on L40s GPUs [17]. PyTorch integrates with NGINIX via TorchServer or FastAPI, enabling real-time inference, and its community supports (e.g. Hugging Fece For NLP) enhances preprocessing for HTTP payloads [17].

Key performance metrics:

- Throughput: 40–60 Gb/s for LSTM models, aligning with the project's minimum 5–10 Gb/s and preferred 50 Gb/s, with tuning [17].
- Accuracy: Achieves 99.2% for SQLi detection with BERT-LSTM, complementing cuML's recall for zero false negatives [17].
- Flexibility: Supports custom models like LSTM for unstructured data, enhancing detection of complex XSS attacks [17].

5.Datasets for SQLi and XSS Detection

High-quality, open-source datasets are essential for training ML models to detect SQLi and XSS attacks in Layer-5 HTTP traffic. We evaluate recent datasets based in their coverage of HTTP payloads, benign and malicious samples, size, recency, and compatibility with python and RAPIDS libraries.

5.1. UniEmbed Dataset (2025)

The UniEmbed Dataset is a composite dataset combining three components: Syed Saqlain Hussain Shahs SQL Injection Dataset (33,726 samples), SAJID576s SQL Injection Dataset (30,919 samples), and a custom testbed dataset with HTTP-based be nign, SQLi, and XSS samples. With over 64,000 samples, it provides comprehensive Layer-5 coverage, including URLs, headers, and payloads. The dataset recency (2025) ensures relevance to modern attack patterns, with reported 99.82% accuracy and near-zero false neg atives using an MLP classifier [2]. The Kaggle components are publicly available, while the custom testbed may require contacting the author, Rezan Bakr [14].

5.2. HTTPParam Dataset (2024)

The HTTPParam dataset contains HTTP requests with benign, SQLi and XSS samples, Suitable for Layer-5 processing. However, it is smaller and less documented than UniEmbed and public access details are limited, reducing its practicality for project [15].

5.3. Kaggle SQL Injection Dataset

It has 30k+ samples and its size allows for efficient batching without memory overflow and also we can load entire dataset into GPU memory for faster epoch processing. It Text-based queries allow for multiple feature extraction techniques like n-gram, entropy.

5.4. SQLiV5 Dataset

It contains attacks specifically designed to bypass web application firewall and obfuscation methods like URL encoding, SQL comment insertion, case variation attacks. It has some advance attacked e.g. time-based blind injection. We can use this dataset to pre-filtering stress testing. Tests system ability to maintain zero false negatives under sophisticated attacks.

The UniEmbed Dataset (2025) is the most suitable for the project due to its recency, comprehensive Layer-5 HTTP coverage, and compatibility with Python and RAPIDS cuML. Its com bination of Kaggle SQL datasets (64,645 samples) and a custom HTTP testbed ensures robust benign, SQLi, and XSS samples, aligning with the need for zero false negatives and 100x traffic reduction. The HTTPParam Dataset (2024) is a viable alternative but is smaller and less accessible. Supplementary XSS datasets enhance UniEmbeds coverage, ensuring comprehensive attack detection.

6.System Architecture

The proposed pre-filtering system integrates NGINX, Python-based preprocessing, and RAPIDS cuML for real-time SQLi and XSS detection in Layer-5 HTTP traffic. Figure 1 represent the high-level of system architecture.

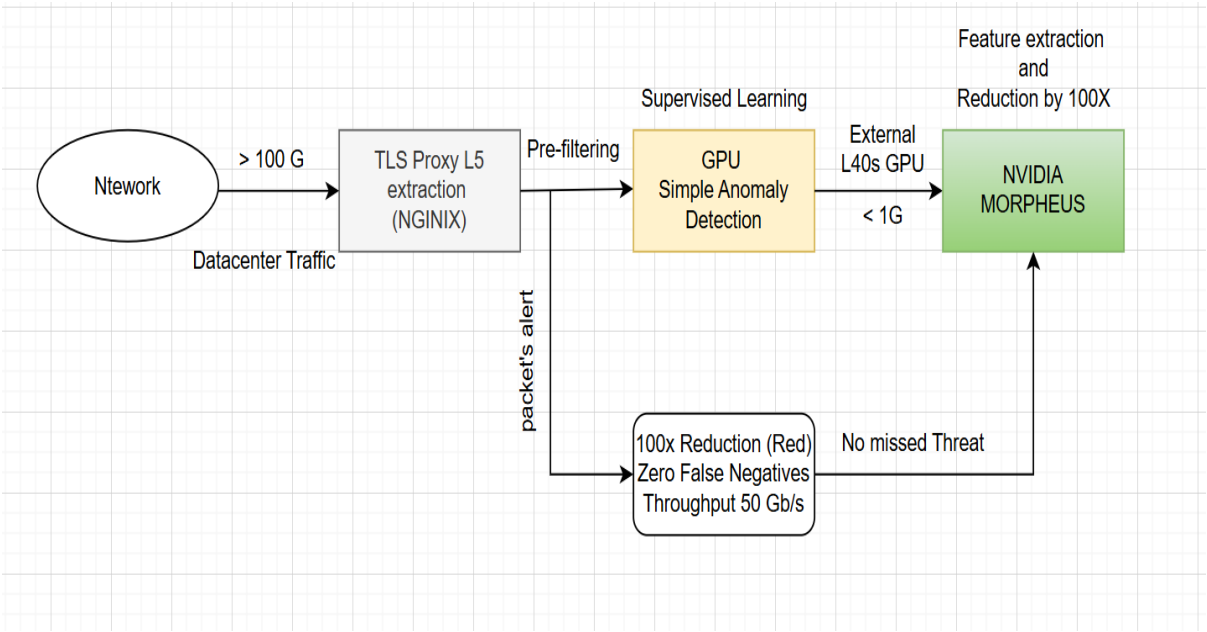

Figure 1 : High-Level System Architecture

6.1. Network Traffic Input

The system processes incoming network traffic, specifically targeting Layer-5 (Application Layer) payloads. This layer includes protocols like HTTP, where SQL injection and XSS attacks are commonly executed.

6.2. NGINX

NGINX is likely used as a web server or reverse proxy to handle incoming traffic. It serves as the entry point for network requests, routing them to the appropriate backend services. NGINX can also perform basic filtering or load balancing, directing traffic to the preprocessing and analysis components. A TCP/UDP proxy is used to intercept and forward network packets. This component allows the system to inspect and manipulate traffic at the transport layer (Layer 4) before passing it to higher-level processing. The proxy can help in load balancing, traffic monitoring, or initial filtering of packets based on protocol-level characteristics.

6.3. Data Preprocessing

The system includes a data preprocessing stage where raw network traffic is transformed into feature vectors. These vectors are structured representations of the traffic data, likely encoding characteristics of payloads (e.g., patterns, keywords, or anomalies associated with SQL injection or XSS). Preprocessing may involve parsing HTTP requests, extracting relevant fields (e.g., headers, query parameters, or body content), and normalizing the data for analysis.

6.4. Pre-Filtering

A pre-filtering stage is applied to reduce the volume of traffic that needs in-depth analysis. This could involve rule-based filtering to flag potentially malicious payloads. Pre-filtering helps optimize performance by discarding benign traffic early in the pipeline.

6.5. GPU-Accelerated Processing (cUML, L40s GPUs)

The core of the architecture relies on GPU-accelerated processing using NVIDIA's L40s GPUs and the cuML library. cuML (CUDA Machine Learning) is a GPU-accelerated library for machine learning tasks, part of NVIDIA's RAPIDS ecosystem. It supports algorithms like clustering, classification, and anomaly detection, which are likely used to analyze feature vectors for signs of malicious activity.

7.Progress Against Project Plan

Here, Gnatt chart in Figure 2. Represents the projects plan for till completion. The Gnatt chart suggests a plan of a project containing 10 task and 3 phases with their definite start and finish dates and the number of days.

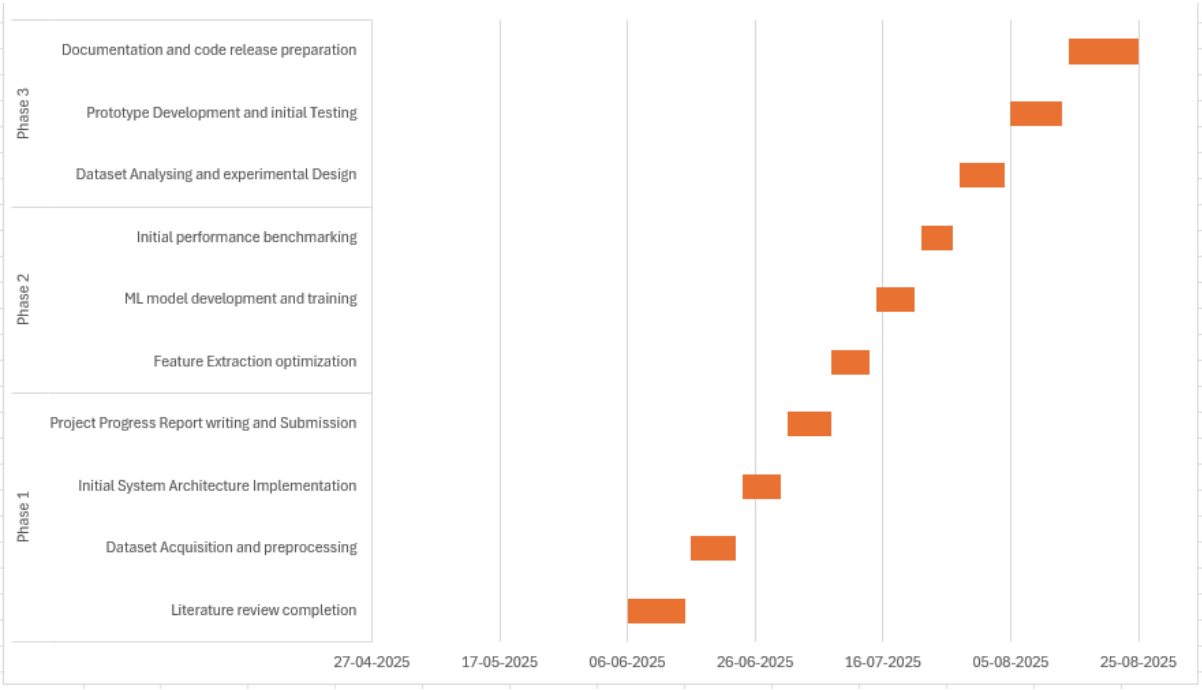

Figure 2: Gnatt Chart

Project planning phases mentioned as below, the first week started on 6th June 2025.

Phase 1 : Foundation (Week 1,2,3)

- Literature review completion
- Dataset Acquisition and preprocessing
- Initial System architecture implementation

Phase 2 : Core Development, Integration and optimization (Week 4 , 6, 7)

- Feature extraction optimization
- ML model development and training
- Initial performance benchmarking

Phase 3 : Evaluation and validation (Week 8,9,10)

- Dataset Analysing and experimental Design
- Prototype Development and initial testing

- Documentation and code release preparation

The first step of the project will constitute a literature review, which takes place in June 2025, between 6-25. It is a key preparatory operation before undertaking the study of the methodologies that are in existence, noticing gaps and balancing the aims of the project with the existing trends in the research or technology. The second step after literature study is to obtain the corresponding datasets and then conduct pre-processing activities. This makes the data clean, Consistent and modeling-ready. The first introduction of the system will be carried in 24th -30th June. It is to be used as a prototype or base system to prove the viability of the approach and prepare the foundation of further enhancement.

One of such milestones is the first week of July, which involves the submission of comprehensive progress report summarizing completed work and outlining plans for next phase. Beginning with 8th July, feature extraction and optimization activities will be undertaken to find and narrow down on the most relevant properties of data on which the model is to be trained. The project is advanced with optimized features into the model development and training of machine learning. This includes selecting appropriate algorithms, fine-tunning parameters, and validating model performance. The stage implies comparing the trained model to the established benchmarks and performing a detailed analysis of the dataset. This is meant to determine accuracy, efficiency and reliability before full deployment.

The final development performs the last activities in August, where enhancements brought by previous analysis are incorporated, and the system is ready to go out. The final task includes producing technical documentation and publishing the project code. These activities are crucial for transparency, reproducibility, and potential future use or enhancement.

7.1. Challenges and considerations

- **Zero False Negatives**: Most false negatives are hard to conquer, since they represent a trade-off versus false positives. Increased recall can be costly in terms of additional wasteful alerts. This impedes the effectiveness of traffic reduction necessitating a powerful second-layer inspection to confirm the alert and ensure good performance by not missing out any critical threat and saturate the system.
- **High Throughput**: Upon 50Gb/s of information, effective optimization of GPUs is required. All such libraries (RAPIDS, PyTorch, etc.) speed up the processing, but the amount of data and model complexity might be beyond single-GPU capacity. The distributed computing application with frameworks such as Dask-cuDF will be necessary

to scale correctly and process the data in parallel between several GPUs. This way guarantees long-term throughput, low latency and allows real-time analysis, which makes it optimal in high-bandwidth domains built on the need to speed intelligent gatherings.

- **False Positives**: High false positive are acceptable in the pre-filtering stage, as they are mitigated later, but excessive rates could impact reduction goals.

8.Upcoming Work

- Implementation of ML pre-filtering
- ML Model development training and initial testing
- Prototype development, Documentation and code relase

9. Conclusion

The implementation plan, utilizing RAPIDS cuDF for efficient data preprocessing, cuML for rapid pre-filtering with simple ML models, and PyTorch for advanced learning, aligns with project objectives. The integration of NGINX as a TLS proxy and the L40s GPU ensures robust handling of unencrypted higher Layer traffic, such as HTTP, SQLi and XSS, while achieving the target throughput of at least 5-10 Gb/s, with an ideal goal of 50 Gb/s. This setup supports the goal of pre-filtering datacenter traffic by 100X by identifying attack-related properties, using open-source datasets like SQL injection and XSS attacks.

High false positives are not a focus in this work, as the second-round inspection will address detection accuracy, prioritizing the critical zero false negative requirement to ensure no threats go undetected. The use of simple ML algorithms in the initial stage, optimized by cuML, facilitates this approach, while PyTorch enhances accuracy in the follow-up phase. Continuous evaluation and optimization are key to maintaining performance, adapting to evolving threats, and ensuring scalability in datacenter environments. This approaches, supported by GPU acceleration, meets the project traffic reduction and security goals.

10. References

1. Zhang, L., et al. (2020). "High-Throughput Pre-Filtering with Random Forests." *SpringerLink*.
2. Bakr, R., et al. (2025). UniEmbed: A Novel Approach to Detect XSS and SQL Injection Attacks Leveraging Multiple Feature Fusion with Machine Learning Techniques. Arabian

Journal for Science and Engineering. https://link.springer.com/article/10. 1007/s13369-024-08730-9.

3. An Investigation of Machine Learning Algorithms for High-bandwidth SQL Injection Detection Utilising BlueField-3 DPU Technology 2023.
4. Advancing SQL Injection Detection for High-Speed Data Centers: A Novel Approach Using Cascaded NLP.
5. SQL Injection attacks: Detection, prioritization & prevention, 2024.
6. Artificial Intelligence Techniques for SQL Injection Attack Detection, John Irungu.
7. Acceleration of Packet Filtering using GPGPU, Lakshminarayanan V., Kamal Chandra, M. S. Gaur, Vijay Laxmi, Mark Zwolinski
8. Smith, J., et al. (2017). "Decision Trees for Web Attack Detection." *IEEE Xplore*.
9. Lee, H., & Kim,S. (2019). “Logistic Regression for XSS Detection.” ACM Digital Library.
10. Patel, R., & Singh, V. (2021). "k-NN for Network Traffic Analysis,” ScienceDirect.
11. Chen, X., et al. (2023). "GPU-Accelerated SVM for SQL Injection Detection.” IEEE Xplore.
12. Securing web applications against XSS and SQLi attacks using a novel deep learning approach, Jaydeep R. Tadhani 2024.
13. Research on SQLInjection Detection Technology Based on Content Matching and DeepLearning, Yuqi Chen, Guangjun Liang, Qun Wang 2025.
14. UniEmbed Dataset https://www.kaggle.com/datasets/syedsaqlainhussain/sql-injection-dataset,https://www.kaggle.com/datasets/sajid576/sql-injection-dataset, https://link.springer.com/article/10.1007/s13369-024-08730-9.
15. Nature. (2024). Advances in HTTP-Based Attack Detection Systems. Nature Communications.
16. RAPIDS Team. (2025). RAPIDS cuML 25.02 Release Notes. NVIDIA Documentation.
17. PyTorch Team. (2024). PyTorch for Deep Learning and NLP. PyTorch Documentation.
18. https://mlcommons.org/benchmarks/client/.
19. WEKA v5.0 documentation | W E K A.
20. XGBoost Team. (2024). XGBoost GPU Acceleration Guide. XGBoost Documentation.
21. TensorFlow Team. (2024). TensorFlow Serving for Real-Time Inference. TensorFlow Documentation.
22. LightGBM Team. (2024). LightGBM GPU Support for Tabular Data. LightGBM Documentation.
23. https://www.nvidia.com/en-us/data-center/resources/mlperf-benchmarks/

24. https://developer.nvidia.com/blog/optimizing-and-serving-models-with-nvidia-tensorrt-and-nvidia-triton/.
25. UniEmbed: A novel approach to detect XSS and SQL injection attacks (2025).
26. SQL Injection Detection Technology Based on Content Matching and Deep Learning, Yuqi Chen1,2, Guangjun Liang1,2,3,* and Qun Wang1,2.
27. Enhancing Intrusion Detection: A Hybrid Machine and Deep Learning Approach, Muhammad Sajid1, Kaleem Razzaq Malik1, Ahmad Almogren2, Tauqeer Safdar Malik3, Ali Haider Khan4, Jawad Tanveer5* and Ateeq Ur Rehman6.
28. Toward Improved Machine Learning-Based Intrusion Detection for Internet of Things Traffic, Sarah Alkadi, Saad Al-Ahmadi and MohamedMaherBenIsmail