



Green University of Bangladesh

*Department of Computer Science and Engineering (CSE)
Semester: (Fall, Year: 2025), B.Sc. in CSE (Day)*

AI-Driven Zero-Day Cyber Threat Prediction and Anomaly Detection System

*Course Title: Artificial Intelligence Lab
Course Code: CSE 316
Section: 231-D3*

Students Details

Name	ID
Promod Chandra Das	231002005
Chinmoy Debnath	231902029

*Submission Date: 03.12.25
Course Teacher's Name: Mr.Mozdaher Abdul Quader*

[For teachers use only: **Don't write anything inside this box**]

<u>Lab Project Status</u>	
Marks:	Signature:
Comments:	Date:

Contents

1	Introduction	3
1.1	Overview	3
1.2	Motivation	3
1.3	Problem Definition	4
1.3.1	Problem Statement	4
2	Design/Development/Implementation of the Project	7
2.1	Introduction	7
2.2	Project Details	7
2.2.1	Subsection_name	7
2.3	Implementation	8
2.3.1	Subsection_name	8
2.4	Algorithms	8
3	Performance Evaluation	10
3.1	Simulation Environment/ Simulation Procedure	10
3.1.1	Subsection	10
3.1.2	Subsection	10
3.2	Results Analysis/Testing	10
3.2.1	Result_portion_1	10
3.2.2	Result_portion_2	10
3.2.3	Result_portion_3	10
3.3	Results Overall Discussion	11
3.3.1	Complex Engineering Problem Discussion	11
4	Conclusion	12
4.1	Discussion	12
4.2	Limitations	12

4.3	Scope of Future Work	12
-----	--------------------------------	----

Chapter 1

Introduction

1.1 Overview

Zero-day cyber attacks are one of the most dangerous and unpredictable forms of security threats, as they exploit software or system vulnerabilities that are completely unknown to developers, security teams, or vendors. Because no patch, signature, or predefined rule exists for these vulnerabilities, organizations remain exposed until the flaw is discovered and fixed. Traditional signature-based intrusion detection systems (IDS) are ineffective against zero-day attacks since they depend solely on pre-identified threat signatures and known behavior patterns. As a result, modern networks require intelligent, adaptive, and self-learning security mechanisms capable of identifying abnormal activity even when no prior information about the attack exists. This project addresses that challenge by proposing an AI-driven anomaly detection system powered by deep learning models such as Autoencoders and Graph Neural Networks (GNN). The goal is to learn the normal behavior of network traffic at a granular level and automatically identify deviations that may signal zero-day intrusions. By continuously monitoring large-scale enterprise network data, the system can produce real-time threat scores, visual alerts, and early indicators of compromise (IoC). The solution is designed to integrate with enterprise SOC environments, reduce false positives, and offer high scalability for modern distributed infrastructures such as cloud platforms, IoT networks, and data centers. Ultimately, this approach aims to enhance the proactive defense capability of cybersecurity systems and provide organizations with a reliable shield against emerging, unpredictable, and high-impact zero-day threats.

1.2 Motivation

Modern cyber attacks are evolving rapidly, becoming more sophisticated, stealthy, and highly adaptive. Attackers now use advanced techniques to bypass traditional security mechanisms, making it extremely challenging for organizations to defend their networks. Every year, businesses lose millions of dollars due to breaches that their existing IDS/IPS systems fail to detect—mainly because these systems rely on known signatures, predefined rules, or historical attack patterns.

However, zero-day attacks pose an even greater threat. They exploit vulnerabilities that security teams are completely unaware of, leaving no time for patching or conventional defense strategies. The rising frequency of zero-day exploits highlights the urgent need for a smarter and more proactive detection mechanism.

Artificial Intelligence offers a promising solution. Instead of depending on signatures, AI can learn the normal behavior of network traffic and identify deviations that may indicate malicious activity—even if the attack has never occurred before. This behavior-based detection approach significantly improves the chances of catching hidden, emerging, or unknown threats.

The primary motivation behind this project is to develop an intelligent cybersecurity system capable of detecting anomalies in real time, reducing false positives, and providing early warning signals before any major damage occurs. By leveraging deep learning and advanced analytics, the aim is to strengthen the overall security posture of modern enterprises and prepare them for the next generation of cyber threats.

1.3 Problem Definition

The current cybersecurity systems rely heavily on: Known attack signatures Static rule-based mechanisms Manual security monitoring This makes them ineffective against zero-day attacks, which follow no known signature or specific pattern. We need a system that learns normal network behavior and identifies malicious deviations automatically.

1.3.1 Problem Statement

Develop an AI-driven intrusion detection system capable of detecting zero-day cyber threats by using behavioral anomaly detection on network traffic. The system must analyze real-time traffic patterns, compute anomaly scores, and notify security analysts of suspicious activities with minimal false positives.

Complex Engineering Problem (CEP)

This project qualifies as a Complex Engineering Problem because it involves:

1. Large-scale network data processing (millions of packets).
2. Non-linear patterns requiring advanced ML/DL techniques.
3. Uncertainty and unpredictability associated with zero-day attacks.
4. Conflicting requirements – low false positives vs. high detection accuracy.
5. Security compliance standards (NIST, ISO27001).
6. Real-time detection constraints and system reliability.

7. Stakeholders: SOC teams, network engineers, management, users.

It demands deep technical knowledge, high-level analysis, multidisciplinary integration, and algorithmic complexity.

Design Goals / Objectives

1. Build a deep learning model (Autoencoder/GNN) trained on normal network behavior.
2. Detect anomalous traffic patterns that may indicate zero-day attacks.
3. Provide real-time threat scoring through a dashboard.
4. Reduce false positives using optimized threshold selection.
5. Ensure scalability for enterprise-level networks.
6. Generate alerts instantly for SOC analysts.
7. Maintain data security, encryption, and compliance with cybersecurity standards.

Applications

1. Enterprise network security
2. SOC (Security Operations Center) threat monitoring
3. Government cybersecurity agencies
4. Banking & financial institutions
5. Cloud infrastructure monitoring
6. Industrial IoT security
7. Detecting malware, DDoS, data exfiltration, ransomware
8. Zero-day threat hunting

Table 1.1: Summary of the attributes touched by the mentioned project

Name of the P Attributes	Explain how to address
P1: Depth of knowledge required	The project requires advanced understanding of AI, deep learning (Autoencoder/GNN), network packet analysis, cybersecurity concepts, threat modelling, anomaly detection, and system integration. This demands multidisciplinary knowledge at a high technical level.
P2: Range of conflicting requirements	The system must maintain high detection accuracy while reducing false positives, ensure real-time processing with minimal latency, and balance resource usage with model performance. These requirements conflict, requiring optimization and trade-offs.
P3: Depth of analysis required	The project involves analyzing large-scale network traffic, extracting behavioral patterns, identifying anomalies, tuning deep learning models, and validating system outputs. Deep computational and logical analysis is essential.
P4: Familiarity of issues	Zero-day attacks are unfamiliar and unpredictable by nature. The project must handle unknown behaviors, previously unseen attack vectors, and non-signature-based threats. This requires robust anomaly modeling.
P5: Extent of applicable codes	The system must align with cybersecurity standards such as NIST SP 800-53, ISO/IEC 27001, data protection regulations, and secure network handling protocols. These codes influence system design and compliance.
P6: Extent of stakeholder involvement and conflicting requirements	Multiple stakeholders are involved: SOC analysts, network administrators, cybersecurity managers, and organizational decision-makers. Their needs differ (detailed logs vs. summarized insights vs. performance efficiency), resulting in conflicting requirements.
P7: Interdependence	The system components (AI model, feature extractor, network traffic monitor, dashboard, alert engine) are tightly interdependent. Failure in one part compromises the entire detection pipeline, requiring coordinated system design.

Chapter 2

Design/Development/Implementation of the Project

2.1 Introduction

Start the section with a general discussion of the project [1] [2] [3].

2.2 Project Details

In this section, you will elaborate on all the details of your project, using subsections if necessary.

2.2.1 Subsection_name



Figure 2.1: Figure name

You can fix the height, width, position, etc., of the figure accordingly.

2.3 Implementation

All the implementation details of your project should be included in this section, along with many subsections.

2.3.1 Subsection_name

This is just a sample subsection. Subsections should be written in detail. Subsections may include the following, in addition to others from your own project.

The workflow

Tools and libraries

Implementation details (with screenshots and programming codes)

Each subsection may also include subsubsections.

2.4 Algorithms

The algorithms and the programming codes in detail should be included . Pseudo-codes are also encouraged very much to be included in this chapter for your project.

- Bullet points can also be included anywhere in this project report.

Algorithm 1: Sample Algorithm

Input: Your Input

Output: Your output

Data: Testing set x

```
1  $\sum_{i=1}^{\infty} := 0$  // this is a comment
  /* Now this is an if...else conditional loop */
2 if Condition 1 then
3   | Do something // this is another comment
4   | if sub-Condition then
5   | | Do a lot
6 else if Condition 2 then
7   | Do Otherwise
  /* Now this is a for loop */
8   | for sequence do
9   | | loop instructions
10 else
11 | Do the rest
  /* Now this is a While loop */
12 while Condition do
13 | Do something
```

Chapter 3

Performance Evaluation

3.1 Simulation Environment/ Simulation Procedure

Discuss the experimental setup and environment installation needed for the simulation of your outcomes.

3.1.1 Subsection

3.1.2 Subsection

3.2 Results Analysis/Testing

Discussion about your various results should be included in this chapter in detail.

3.2.1 Result_portion_1

The results of any specific part of your project can be included using subsections.

3.2.2 Result_portion_2

Each result must include screenshots from your project. In addition to screenshots, graphs should be added accordingly to your project.

3.2.3 Result_portion_3

Each result must have a single paragraph describing your result screenshots or graphs or others. This is a simple discussion of that particular portion/part of your result.

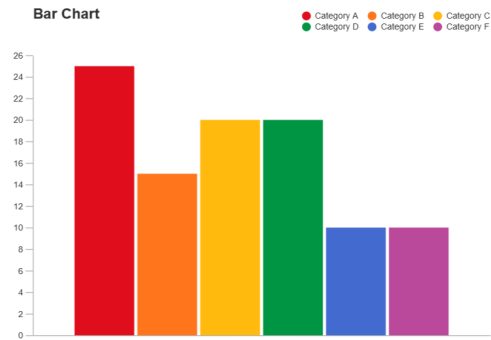


Figure 3.1: A graphical result of your project

3.3 Results Overall Discussion

A general discussion about how your result has arrived should be included in this chapter. Where the problems detected from your results should be included as well.

3.3.1 Complex Engineering Problem Discussion

[OPTIONAL] In this subsection, if you want, you can discuss in details the attributes that have been touched by your project problem in details. This has already been mentioned in the Table 1.1.

Chapter 4

Conclusion

4.1 Discussion

Discuss the contents of this chapter and summarized the description of the work and the results and observation. Generally, it should be in one paragraph.

4.2 Limitations

Discuss the limitations of the project. Limitations must be discussed, with the help of some critical analysis.

4.3 Scope of Future Work

Discuss the future work of the project, that is your plans for more work and extension of your project.

References

- [1] Uthayasankar Sivarajah, Muhammad Mustafa Kamal, Zahir Irani, and Vishanth Weerakkody. Critical analysis of big data challenges and analytical methods. *Journal of Business Research*, 70:263–286, 2017.
- [2] Douglas Laney. 3d data management: controlling data volume, velocity and variety. gartner, 2001.
- [3] MS Windows NT kernel description. <http://web.archive.org/web/20080207010024/http://www.808multimedia.com/winnt/kernel.htm>. Accessed Date: 2010-09-30.