**An Internship Project Report on**

**Early Warning Signal Generation   
For Intelligent Detection of DDoS Traffic**

Submitted in the partial fulfilment of the requirements for the Summer Internship of

**BACHELOR OF TECHNOLOGY**

**In**

**INFORMATION TECHNOLOGY**

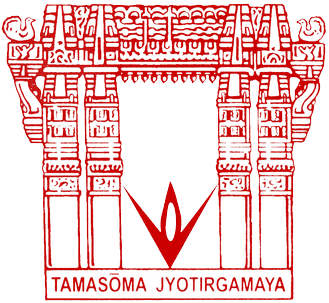
Submitted by

A. ROHITH REDDY 22071A1202

B. RISHI KARTHIKEYA 22071A1210

B. ABHILASH 22071A1211

B. POOJITHA 22071A1212



**DEPARTMENT OF INFORMATION TECHNOLOGY**

**VNR Vignana Jyothi Institute of Engineering & Technology**

(Autonomous Institute, Accredited by NAAC with ‘A++’ grade and NBA)

Bachupally, Nizampet (S.O.) Hyderabad- 500090

May 2025

**An Internship Project Report on**

**Early Warning Signal Generation   
For Intelligent Detection of DDoS Traffic**

Submitted in the partial fulfilment of the requirements for the Summer Internship of

**BACHELOR OF TECHNOLOGY**

**In**

**INFORMATION TECHNOLOGY**

Submitted by

A. ROHITH REDDY 22071A1202

B. RISHI KARTHIKEYA 22071A1210

B. ABHILASH 22071A1211

B. POOJITHA 22071A1212

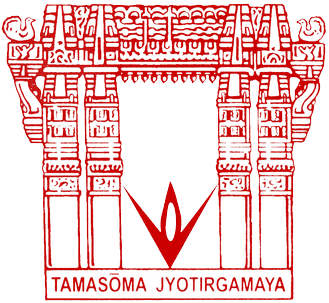
**Under the esteemed guidance of**

Dr. V. Radha Krishna

Associate Professor,

Dept. of Information Technology,

VNRVJIET



**DEPARTMENT OF INFORMATION TECHNOLOGY**

**VNR Vignana Jyothi Institute of Engineering & Technology**

(Autonomous Institute, Accredited by NAAC with ‘A++’ grade and NBA) Bachupally, Nizampet (S.O.) Hyderabad- 500090

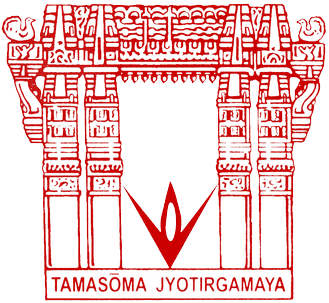
May 2025

**VNR Vignana Jyothi Institute of Engineering & Technology**

(Autonomous Institute, Accredited by NAAC with ‘A++’ grade and NBA) Bachupally, Nizampet (S.O.) Hyderabad- 500090.

**DEPARTMENT OF INFORMATION TECHNOLOGY**

Date:16 May 2025



**CERTIFICATE**

This is to certify that the project work entitled **“Early Warning Signal Generation For Intelligent Detection of DDoS Traffic**” is being submitted by **A. ROHITH REDDY (22071A1202),B. RISHI KARTHIKEYA (22071A1210), B.ABHILASH (22071A1211),B. POOJITHA (22071A1212)** in partial fulfilment for the award of Degree of **BACHELOR OF TECHNOLOGY** in **INFORMATION TECHNOLOGY** to the Jawaharlal Nehru Technological University, Hyderabad during the academic year **2024-25** is a record of bona-fide work carried out by her under our guidance and supervision.

The results embodied in this report have not been submitted by the students to any other

University or Institution for the award of any degree or diploma.

|  |  |
| --- | --- |
| **Under the Guidance of:** | **Head of the Department:** |
| **Dr. V. Radha Krishna,**  **Associate Professor,**  **Dept. of IT,**  **VNRVJIET,**  **Hyderabad.** | **Dr. N. Mangathayaru,**  **Professor,**  **Head of the Department,**  **Dept of IT,**  **VNRVJIET,**  **Hyderabad.** |

External Examiner

**VNR Vignana Jyothi Institute of Engineering & Technology**

Autonomous Institute, Accredited by NAAC with ‘A++’ grade and NBA) Bachupally, Nizampet (S.O.) Hyderabad- 500090.

**Department of Information Technology**

Date:16 May 2025

**DECLARATION**

We declare that the internship project work entitled “**Early Warning Signal Generation For Intelligent Detection of DDoS Traffic**” submitted in the department of Information Technology, Vallurupalli Nageswara Rao Vignana Jyothi Institute of Engineering and Technology, Hyderabad, in partial fulfilment of the requirement for the award of the degree of Bachelor of Technology in Information Technology is a Bonafide record of our own work carried out under the supervision of Dr. V. Radha Krishna, Associate Professor , Department of IT, VNRVJIET. Also, we declare that the matter embodied in this thesis has not been submitted by us in full or in any part thereof for the award of any degree/diploma of any other institution or university previously.

Signature of the Student:

|  |  |  |  |
| --- | --- | --- | --- |
| 1. Rohith Reddy   (22071A1202) | 1. Rishi Karthikeya   (22071A1210) | B. Abhilash  (22071A1211) | B. Poojitha  (22071A1212) |

Place: Hyderabad

Date:16 May, 2025

**ACKNOWLEDGEMENT**

We express our deep sense of gratitude to our beloved **Chairman, Daggubati Suresh Babu, VNR Vignana Jyothi Institute of Engineering &Technology** for the valuable guidance and for permitting us to carry out this project.

With immense pleasure, we record our deep sense of gratitude to our beloved **Principal, Dr. C. D. Naidu** for permitting us to carry out this project.

We express our deep sense of gratitude to our beloved professor **Dr. N. Mangathayaru, Professor and Head, Department of Information Technology, VNR Vignana Jyothi Institute of Engineering & Technology, Hyderabad - 500090** for the valuable guidance and suggestions, keen interest and through encouragement extended throughout period of project work.

We take immense pleasure to express our deep sense of gratitude to our beloved Guide **Dr. V. Radha Krishna, Associate Professor in Information Technology, VNR Vignana Jyothi Institute of Engineering & Technology, Hyderabad,** for his valuable suggestions and rare insights, for constant source of encouragement and inspiration throughout my project work.

We express our thanks to all those who contributed to the successful completion of our project work.

1. A. Rohith Reddy
2. B. Rishi Karthikeya
3. B. Abhilash
4. B. Poojitha

**TABLE OF CONTENTS:**

[ABSTRACT i](#_Toc198329526)

[LIST OF FIGURES ii](#_Toc198329526)

[LIST OF TABLES iii](#_Toc198329526)

[CHAPTER 1 8](#_Toc198338988)

[INTRODUCTION 8](#_Toc198338989)

[1.1 DOS AND DDOS ATTACKS 8](#_Toc198338990)

[1.1.1 DoS Attack 8](#_Toc198338991)

[1.1.2 DDoS Attack 9](#_Toc198338992)

[1.2 REFLECTION-BASED AND EXPLOITATION-BASED DDOS ATTACKS 9](#_Toc198338993)

[1.2.1 Reflection-based DDoS attacks 9](#_Toc198338994)

[1.2.2 Exploitation-Based DDoS attacks 10](#_Toc198338995)

[1.3 LOW-RATE AND HIGH-RATE DDOS ATTACKS 10](#_Toc198338996)

[1.3.1 Low-Rate DDoS Attacks 11](#_Toc198338997)

[1.3.2 High-Rate DDoS attacks 11](#_Toc198338998)

[1.4 INTRUSION DETECTION SYSTEM 11](#_Toc198338999)

[1.5 INTRUSION PREVENTION SYSTEM 12](#_Toc198339000)

[1.6 RESEARCH CHALLENGE 13](#_Toc198339001)

[1.7 PROBLEM STATEMENT 13](#_Toc198339002)

[CHAPTER 2 14](#_Toc198339003)

[LITERATURE SURVEY 14](#_Toc198339004)

[CHAPTER 3 16](#_Toc198339005)

[EVALUATION AND ASSESSMENT OF IDS DATASETS 16](#_Toc198339006)

[3.1 FRAMEWORK 16](#_Toc198339007)

[3.2 EVALUATION 18](#_Toc198339008)

[CHAPTER 4 22](#_Toc198339009)

[METHODOLOGY 22](#_Toc198339010)

[4.1 SYSTEM ARCHITECTURE 22](#_Toc198339011)

[4.2 DATA PREPROCESSING 24](#_Toc198339012)

[4.3 FEATURE EXTRACTION MODULE 24](#_Toc198339013)

[4.3.1 Partial Derivatives Calculation 25](#_Toc198339014)

[4.4 EARLY WARNING DETECTION MODULE 26](#_Toc198339015)

[4.4.1 Computation of Parameters () 27](#_Toc198339016)

[4.4.2 Computation of Residuals 27](#_Toc198339017)

[4.4.3 Computation of Statistical Features 27](#_Toc198339018)

[4.4.4 Setting Thresholds from Training Data 28](#_Toc198339019)

[4.4.5 Applying Thresholds to Test Data 28](#_Toc198339020)

[4.4.6 Determining Alert Levels Using Z Scores 29](#_Toc198339021)

[4.5 EMERGENCY WARNINGS MODULE 30](#_Toc198339022)

[4.5.1 Load Predictions from Early Warning System (EWS) 30](#_Toc198339023)

[4.5.2 Generate First High Alert 30](#_Toc198339024)

[4.5.3 Generate Second High Alert 30](#_Toc198339025)

[4.5.4 Generate Third High Alert 30](#_Toc198339026)

[4.5.5 Generate Continuous Emergency Alerts 30](#_Toc198339027)

[4.6 WORKING EXAMPLE DEMONSTRATING THE PROPOSED METHOD 30](#_Toc198339028)

[CHAPTER 5 42](#_Toc198339029)

[RESULTS 42](#_Toc198339030)

[5.1 DATA PREPARATION 43](#_Toc198339031)

[5.1.1 Data Selection 43](#_Toc198339032)

[5.1.2 Data Preprocessing 44](#_Toc198339033)

[5.1.3 Aggregation of time-based attributes 44](#_Toc198339034)

[5.2 FEATURE EXTRACTION 45](#_Toc198339035)

[5.2.1 Computation of Derivatives 45](#_Toc198339036)

[5.3 EARLY WARNING DETECTION 46](#_Toc198339037)

[5.3.1 Computation of Residuals 46](#_Toc198339038)

[5.3.2 Computation of Statistical Features and Thresholds and Z scores 47](#_Toc198339039)

[5.3.3 Determining Alert Levels Using Z Scores 47](#_Toc198339040)

[5.4 GRAPHS 48](#_Toc198339041)

[5.4.1 Attribute Graphs: 49](#_Toc198339042)

[5.4.2 Early warning graphs: 55](#_Toc198339043)

[CHAPTER 6 61](#_Toc198339044)

[CONCLUSION 61](#_Toc198339045)

[REFERENCES 62](#_Toc198339046)

**ABSTRACT**

Distributed Denial of Service (DDoS) attacks represent one of the most serious threats to network security, leading to significant losses in cybersecurity. This project focuses on developing both an Intrusion Detection System (IDS) and an Intrusion Prevention System (IPS). The detection system leverages machine learning techniques and spectral analysis to identify abnormal activities within the network. Meanwhile, the prevention system employs statistical methods, including Quartile and Z-score distributions, along with Partial Differential Equations (PDEs), to detect variations in network traffic. These tools facilitate the generation of early warning and emergency alerts, allowing for timely intervention before an attack escalates. The proposed system offers a comprehensive approach to DDoS defense, combining data-driven learning with mathematical modeling, resulting in improved accuracy and responsiveness in threat detection and prevention.

**LIST OF FIGURES**

[CHAPTER 1 8](#_Toc198328794)

[INTRODUCTION 8](#_Toc198328795)

[1.1 DOS AND DDOS ATTACKS 8](#_Toc198328796)

[Fig 1.1 DoS attack 8](#_Toc198328797)

[Fig 1.2 DDoS Attack 9](#_Toc198328798)

[1.2 REFLECTION-BASED AND EXPLOITATION-BASED DDOS ATTACKS 9](#_Toc198328799)

[Fig 1.4 Reflection-Based DDoS Attacks 10](#_Toc198328800)

[Fig 1.5 Exploitation-Based DDoS Attacks 10](#_Toc198328801)

[1.3 LOW-RATE AND HIGH-RATE DDOS ATTACKS 10](#_Toc198328802)

[Fig 1.6 Low-Rate and High-Rate DDoS attacks 11](#_Toc198328803)

[1.4 INTRUSION DETECTION SYSTEM 11](#_Toc198328804)

[Fig 1.7 Intrusion Detection System 12](#_Toc198328805)

[1.5 INTRUSION PREVENTION SYSTEM 12](#_Toc198328806)

[Fig 1.8 Intrusion Detection System 12](#_Toc198328807)

[CHAPTER 4 22](#_Toc198328814)

[METHODOLOGY 22](#_Toc198328815)

[4.1 SYSTEM ARCHITECTURE 22](#_Toc198328816)

[Fig 4.1 System Architecture 23](#_Toc198328817)

[4.2 DATA PREPROCESSING 24](#_Toc198328818)

[Fig 4.2 Data Preprocessing Module 24](#_Toc198328819)

[4.3 FEATURE EXTRACTION MODULE 24](#_Toc198328820)

[Fig 4.3 Feature Extraction Module 25](#_Toc198328821)

[4.4 EARLY WARNING DETECTION MODULE 26](#_Toc198328822)

[Fig 4.4 Early Warning Detection Module 27](#_Toc198328823)

[4.5 EMERGENCY WARNINGS MODULE 30](#_Toc198328824)

[Fig 4.5 Emergency Warnings Module 30](#_Toc198328825)

[CHAPTER 5 42](#_Toc198328827)

[RESULTS 42](#_Toc198328828)

[5.4 GRAPHS 48](#_Toc198328832)

[Figure 5.4.1 Flow packets/s-Train Data 49](#_Toc198328833)

[Figure 5.4.2 Flow Bytes/s-Train Data 49](#_Toc198328834)

[Figure 5.4.3 Flow packets/s-Test data 50](#_Toc198328835)

[Figure 5.4.4 Flow Bytes/s-Test Data 50](#_Toc198328836)

[Figure 5.4.5 Change in Flow packets per sec-Train data 51](#_Toc198328837)

[Figure 5.4.6 Change in Flow Bytes per sec-Train data 51](#_Toc198328838)

[Figure 5.4.7 Change in Flow Packets per sec-Test data 52](#_Toc198328839)

[Figure 5.4.8 Change in Flow Bytes per sec-Test data 53](#_Toc198328840)

[Figure 5.4.9 Second derivative of Flow Packets per sec-Train data 53](#_Toc198328841)

[Figure 5.4.10 Second derivative of Flow Bytes per sec-Train data 54](#_Toc198328842)

[Figure 5.4.11 Second derivative of Flow Packets per sec-Test data 54](#_Toc198328843)

[Figure 5.4.12 Second derivative of Flow Bytes per sec-Test data 54](#_Toc198328844)

[Figure-5.4.2.1 Low-Level Alert-Training Data 55](#_Toc198328845)

[Figure-5.4.2.2 Low-Level alert-Testing Data 56](#_Toc198328846)

[Figure-5.4.2.3 Medium-Level-Training 56](#_Toc198328847)

[Figure 5.4.2.4 Medium-Level-Testing 56](#_Toc198328848)

[Figure 5.4.2.4 High-Level-Training 57](#_Toc198328849)

[Figure 5.4.2.5 High -Level-Testing 57](#_Toc198328850)

[Figure 5.4.2.6 Very High level alert-Training data 57](#_Toc198328851)

[Figure-5.4.2.7 Very High level alert-Testing data 58](#_Toc198328852)

[Figure 5.4.2.8 All alerts –Training data 58](#_Toc198328853)

[Figure-5.4.2.9 All alerts –Testing data 58](#_Toc198328854)

[Figure-5.4.2.11 Early Warning Signals –Training data 59](#_Toc198328855)

[Figure-5.4.2.12 Early Warning Signals –Testing data 59](#_Toc198328856)

**LIST OF TABLES**

[CHAPTER 2 14](#_Toc198330957)

[LITERATURE SURVEY 14](#_Toc198330958)

[Table 2.1: Existing patents on DDoS Prevention 14](#_Toc198330959)

[CHAPTER 3 16](#_Toc198330963)

[EVALUATION AND ASSESSMENT OF IDS DATASETS 16](#_Toc198330964)

[3.2 EVALUATION 18](#_Toc198330966)

[Table 3.1 Assigned weights for 12 criterions 18](#_Toc198330967)

[Table 3.2 Based on NETSCOUT Report 19](#_Toc198330968)

[Table 3.3 Final weights 19](#_Toc198330969)

[TABLE 3.4 Comparison between datasets based on proposed evaluation framework 20](#_Toc198330970)

[TABLE 3.5 Available IDS datasets scores 21](#_Toc198330971)

[CHAPTER 4 22](#_Toc198330972)

[METHODOLOGY 22](#_Toc198330973)

[4.4 EARLY WARNING DETECTION MODULE 26](#_Toc198330977)

[Table 4.4.1 Association between Z Scores and Early Warning Signal Alert Level 29](#_Toc198330978)

[4.6 WORKING EXAMPLE DEMONSTRATING THE PROPOSED METHOD 30](#_Toc198330980)

[Table 4.6.1 Sample Network Traffic Flow 31](#_Toc198330981)

[Table 4.6.2 Aggregating Timestamp into Time in Seconds 31](#_Toc198330982)

[Table 4.6.3 Aggregated Flow Packets/s per Second, Flow Bytes/s per Second (After Summation) 32](#_Toc198330983)

[Table 4.6.4 First Partial Derivative , Values Using Central Difference Method 33](#_Toc198330984)

[Table 4.6.5 Second Partial Derivative , Values Using Central Difference Method 34](#_Toc198330985)

[Table 4.6.6 Calculation of Residuals (Res1, Res2) 36](#_Toc198330986)

[Table 4.6.7 Prediction of Label with Threshold 38](#_Toc198330987)

[Table 4.6.8 Calculated Z scores for all t values 39](#_Toc198330988)

[Table 4.6.9 Association between Z Scores and Early Warning Signal Alert Level 39](#_Toc198330989)

[Table 4.6.10 Generated Alert values for all seconds 40](#_Toc198330990)

[CHAPTER 5 42](#_Toc198330991)

[RESULTS 42](#_Toc198330992)

[5.1 DATA PREPARATION 43](#_Toc198330993)

[Table 5.1.1 Selection of data from Training data 43](#_Toc198330994)

[Table 5.1.2 Selection of data from Testing data 43](#_Toc198330995)

[Table 5.1.2 Sample Network Traffic Flow 44](#_Toc198330996)

[Table 5.1.3 Aggregating Timestamp into Time in Seconds 45](#_Toc198330997)

[5.2 FEATURE EXTRACTION 45](#_Toc198330998)

[Table 5.2.1.1 First Derivative , Values Using Central Difference Method 45](#_Toc198330999)

[5.3 EARLY WARNING DETECTION 46](#_Toc198331000)

[Table 5.3.1 Calculation of Residuals (Res1, Res2) 46](#_Toc198331001)

[Table 5.3.2 Calculated Z scores for all t values 47](#_Toc198331002)

[Table 5.3.3.1 Association between Z Scores and Early Warning Signal Alert Level 48](#_Toc198331003)

[Table 5.3.3.2 Generated Alert values for sample data 48](#_Toc198331004)

[5.4 GRAPHS 48](#_Toc198331005)

[Table-5.4.2.1 Alert level 55](#_Toc198331006)

[Table 5.4.2.2 Z-score values at alert and peak stages – Training phase 59](#_Toc198331007)

[Table 5.4.2.3 Z-score values at alert and peak stages – Testing phase 60](#_Toc198331008)

CHAPTER 1

INTRODUCTION

In networking, both hardware and software are structured in layers, each designated for functions related to data transfer between devices. The network layer is responsible for routing data packets from the source to the destination. When an application begins to send data, it is divided into smaller segments known as “packets.” A packet creates a “flow,” and multiple flows together make up the network traffic. However, this traffic can be interrupted by anomalies that disrupt the data flow, leading to what is referred to as an ‘attack.’ The aim of these attacks is to interfere with the services of the application, resulting in data loss or damage. A well-known example of such an attack is the Distributed Denial of Service (DDoS) attack.

1.1 DOS AND DDOS ATTACKS

Denial of Service (DoS) and Distributed Denial of Service (DDoS) are two types of cyberattacks aimed at compromising the availability of online services. In these attacks, perpetrators seek to inundate the targeted system with an overwhelming number of requests, ultimately rendering it incapable of serving legitimate users for a certain duration. While both types of attacks share the same objective of disrupting service, they differ significantly in their execution and scale. A DoS attack is typically simpler, involving a single machine that directs traffic towards the target, yet it can be remarkably damaging. In contrast, DDoS attacks employ a multitude of systems to amplify their impact, resulting in greater complexity and a broader scope of disruption

1.1.1 DoS Attack

A Denial of Service (DoS) attack overwhelms a server, system, or website with excessive traffic, rendering that resource inaccessible to users. In such an attack, an attacker utilizes a single machine to inundate the server, preventing legitimate users from accessing its services. Figure 1.1 illustrates the concept of a Denial of Service (DoS) attack, showing how one system sends traffic to the targeted server.

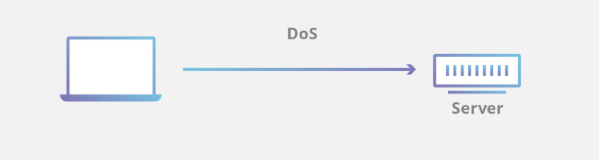


Fig 1.1 DoS attack

1.1.2 DDoS Attack

A Distributed Denial of Service (DDoS) attack overwhelms a server, system, or website with excessive traffic, rendering it inaccessible to users. In this type of attack, an attacker harnesses numerous systems or machines to target a specific server, preventing legitimate users from accessing its services. Figure 1.2 illustrates the concept of a Distributed Denial of Service (DDoS) attack, where multiple systems converge to assault a server.

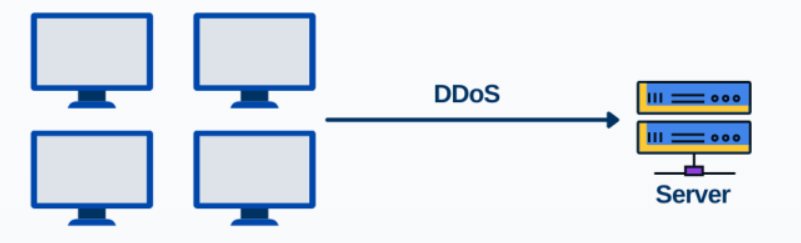


Fig 1.2 DDoS Attack

1.2 REFLECTION-BASED AND EXPLOITATION-BASED DDOS ATTACKS

A Distributed Denial of Service (DDoS) attack overwhelms a server, system, or website with excessive traffic, rendering the resource inaccessible. In this type of attack, the perpetrator utilizes numerous compromised systems or machines to bombard the target server, effectively preventing legitimate users from accessing its services. Figure 1.2 illustrates the concept of a DDoS attack, depicting how multiple systems are employed to target a specific server.



**Fig 1.3 Reflection-Based And Exploitation-Based DDoS Attacks**

1.2.1 Reflection-based DDoS attacks

Reflection-based attacks exploit third-party systems to amplify attacks on a target system. In these attacks, the perpetrator sends a request with a spoofed IP address—specifically, the IP address of the intended victim—directly to the targeted server. As a result, the server responds to the victim's IP address rather than the attacker, inadvertently blocking the legitimate user. Figure 1.4 display attacks TFTP, SNMP, NTP, Web-DDoS, NetBIOS, LDAP, MSSQL, DNS, and SSDP.

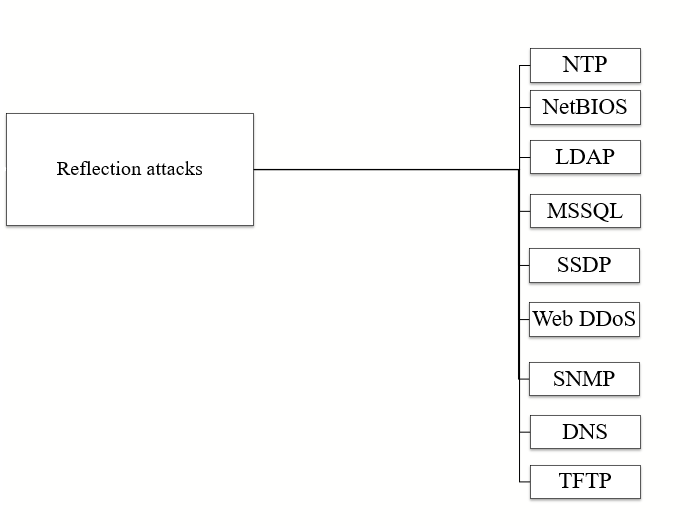


Fig 1.4 Reflection-Based DDoS Attacks

1.2.2 Exploitation-Based DDoS attacks

These assaults occur by taking advantage of weaknesses in networks, applications, protocols, and servers. The attackers focus on the susceptible areas of the server instead of overwhelming the whole server. Figure 1.5 display SYN, UDP-Lag and UDP.

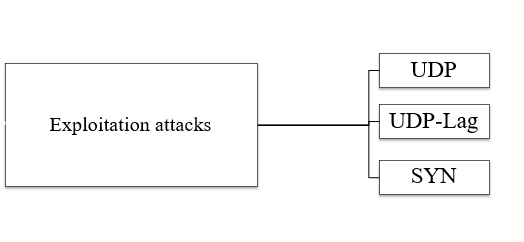


Fig 1.5 Exploitation-Based DDoS Attacks

1.3 LOW-RATE AND HIGH-RATE DDOS ATTACKS

There are two main categories of Distributed Denial of Service (DDoS) attacks, which are based on the speed at which traffic approaches the targeted system or server. DDoS attacks can be further divided into low-rate and high-rate attacks based on the volume and velocity of the incoming traffic. This classification helps identify the attack’s pattern and allows for more effective security measures. Figure 1.6 illustrates the types of DDoS attacks according to their traffic flow, distinguishing between low-rate and high-rate attacks.

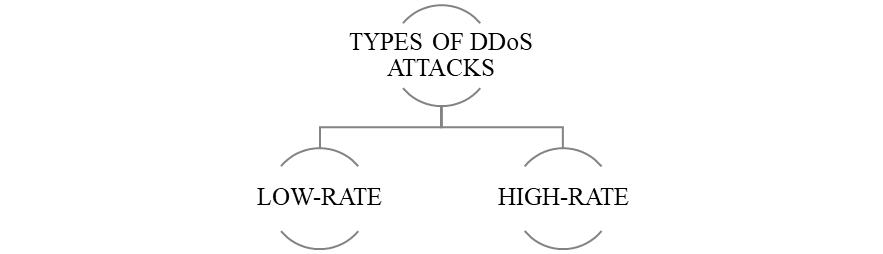


Fig 1.6 Low-Rate and High-Rate DDoS attacks

1.3.1 Low-Rate DDoS Attacks

Low-rate DDoS attacks involve sending relatively small quantities of traffic to the targeted system or server. The objective of these attacks is to evade intrusion detection systems by mimicking normal network traffic. Even though the volume of traffic is lower, they can still create considerable disruptions by exploiting specific vulnerabilities in the target. Since this traffic often resembles legitimate requests, traditional traffic analysis tools may have difficulty distinguishing between genuine and malicious activity, making low-rate DDoS attacks more challenging to detect.

Examples: TFTP, SNMP, NTP, Web-DDoS, SYN and UDP-Lag.

1.3.2 High-Rate DDoS attacks

High-rate DDoS attacks are marked by an enormous influx of malicious traffic. During these attacks, the attacker generates a substantial number of requests in a very brief period. This overwhelming volume can cause delays, crashes, and service interruptions, putting significant pressure on the server’s resources. Because of their speed and effectiveness in disrupting services, high-rate DDoS attacks are regarded as some of the most harmful types of attacks.

Examples: UDP, NetBIOS, LDAP, MSSQL, DNS, and SSDP.

1.4 INTRUSION DETECTION SYSTEM

An Intrusion Detection System (IDS) is a security solution that keeps a close watch on network or system activity to spot signs of malicious actions or policy violations. Think of it as a digital security guard that alerts IT teams in real time whenever something suspicious happens, allowing them to act quickly. Technically, an IDS inspects traffic flowing in and out of a network, checks system logs, and analyses behavior patterns. Fig. 1.7 is an Intrusion Detection System (IDS) that continuously analyzes the incoming traffic of a server and looks for patterns to identify whether an activity is normal or an attack.

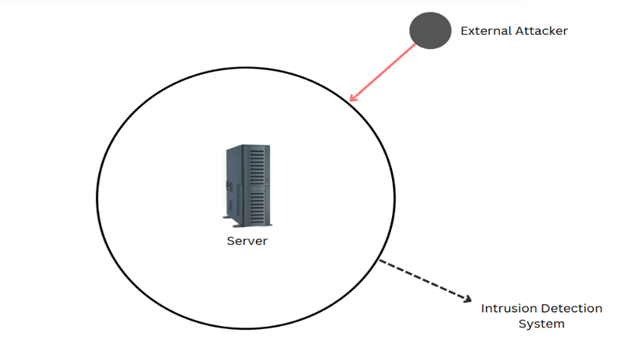


Fig 1.7 Intrusion Detection System

1.5 INTRUSION PREVENTION SYSTEM

An Intrusion Prevention System (IPS) is a crucial cybersecurity tool that continuously monitors network traffic to identify and block harmful activities, such as hacking, malware, and DDoS attacks. It is located directly in the data flow path, inspecting packets as they pass and taking immediate actions such as blocking, dropping, or sending alerts when it detects malicious behavior. Fig 1.8 describes about the traffic from the internet passes through a firewall first, which screens out simple threats.

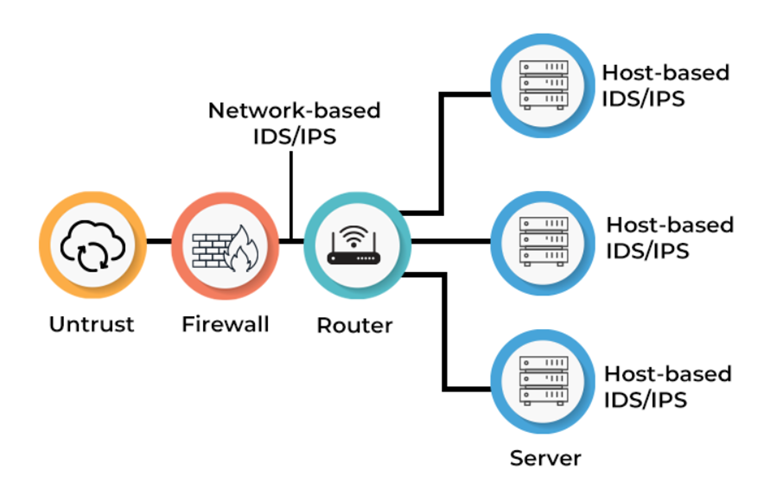


Fig 1.8 Intrusion Detection System

This chapter serves as a solid introduction to how network communication operates and the various vulnerabilities it faces, particularly from attacks like DoS and DDoS. It gives a clear distinction between a DoS attack, which comes from a single source, and a DDoS attack, which is launched from multiple sources, with both types aiming to overwhelm and disrupt services. The chapter dives into two main types of DDoS attacks: reflection-based attacks, which take advantage of unsuspecting third-party systems, and exploitation-based attacks, which specifically target weaknesses in a system. To further illustrate the differences, it classifies these attacks by their speed: low-rate attacks, which can be stealthy and difficult to detect, and high-rate attacks, which tend to be more intense and disruptive. Lastly, the chapter highlights the importance of Intrusion Detection Systems (IDS) and Intrusion Prevention Systems (IPS) as essential tools for monitoring network activity and protecting against these types of threats. By understanding these concepts, readers will be better equipped to recognize and respond to potential risks in network security.

1.6 Research Challenge

**Limited Use of Advanced Statistical Indicators in EWS**

**Gap:** Most existing intrusion prevention systems (IPS) use thresholds or signature-based methods.

**Opportunity:** Incorporate higher-order statistics (e.g., kurtosis, skewness, entropy) as dynamic, early indicators of instability or attack onset.

**Novelty:** Demonstrating how non-traditional signals can provide a lead time advantage before known attacks.

**Lack of Integration Between Physical Models & ML (Physics-Informed Models)**

**Gap:** Existing systems either use pure machine learning or heuristics.

**Opportunity:** Use Physics-Informed Neural Networks (PINNs) or differential equation models to track how packet/byte flows evolve over time and detect deviations early.

**Novelty:** Model the system as a dynamic flow governed by PDEs, then flag residual spikes or unstable behavior as EWS.

**Absence of Standard Metrics for EWS Quality in Intrusion Detection**

**Gap:** Precision, recall, and AUC are not sufficient to evaluate early detection quality.

**Opportunity:** Develop new evaluation metrics (e.g., lead time, false lead rate, early AUC) tailored for EWS.

**Novelty:** Quantify how early and reliable your EWS is relative to attack onset.

1.7 Problem statement

To manage incoming internet traffic to the client network, a system and method are proposed for detecting both low-rate and high-rate DDoS network attacks (Intrusion Detection System, IDS), as well as a strategy for preventing them through early detection (Intrusion Prevention System, IPS). DDoS attacks, which flood network resources with malicious traffic, pose a significant threat to system availability. The challenge is to distinguish between normal traffic and low-rate DDoS attacks. The proposed system aims to tackle this issue by integrating advanced detection capabilities and real-time defense techniques to minimize disruptions and enhance network security.

CHAPTER 2

LITERATURE SURVEY

As cyber threats continue to rapidly evolve, Distributed Denial of Service (DDoS) attacks stand out as some of the most challenging and disruptive types of network attacks. Over the past decade, numerous methodologies have been developed and subsequently patented to detect, mitigate, and prevent these attacks. Patents provide valuable insights into innovative advancements that are relevant both academically and commercially. This chapter aims to offer a thorough overview of the significant patents related to DDoS prevention and defense strategies. By critically examining the technical techniques, outcomes, and limitations of these methods, we hope to highlight current trends and identify the ongoing challenges that need to be addressed for more timely and effective DDoS detection.

Table 2.1: Existing patents on DDoS Prevention

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Patent** | **Title** | **Methodology used** | **Description** | **Grant on** |
| US Patent 10,911,483  [1] | Early detection of dedicated denial of service attacks through metrics correlation | Metrics correlation | The monitoring service uses request data to generate a request frequency value corresponding to the received requests and compares this value to a baseline request frequency value. If the request frequency value exceeds the baseline request frequency value by a maximum threshold value, the monitoring service performs an operation to redirect network traffic originally directed towards the web service. | February 2, 2021 |
| US Patent 9,825,989 B1  [2] | Cyber attack early warning system | Pattern Recognition | The attack alert corresponds to an electrical signal that indicates detection of a malware attack from a remote source. The received data is analyzed using an attack-specific engine that is configured to generate an attack-specific result. An attack value is computed based on the attack-specific result and a consideration of potential attack targets, wherein the attack value is compared to a threshold value to determine whether to generate an early warning alert. | November 2017 |
| US Patent 11,102,240  [3] | Early-warning decision method, node and sub-system | Distributed Structure analysis | The early-warning decision method involves analyzing service request flows directed at a server. It calculates the total flow based on the analysis and the current node's weight. The method compares this flow to an abnormal threshold and decides whether to send instructions for further action on the server based on the comparison. | August 24, 2021 |

In Chapter 2, the focus shifts to a review of various patented techniques designed to detect and mitigate DDoS attacks. It explores important patents that employ methods such as metrics correlation, pattern recognition, and distributed structure analysis for early threat detection. These innovations aim to spot unusual traffic patterns and prompt preventive measures. The chapter highlights the strengths of each approach in enhancing network security, while also acknowledging the challenges related to adaptability and real-time responsiveness. Overall, this section lays the groundwork for future advancements in strategies to defend against DDoS attacks.

CHAPTER 3

EVALUATION AND ASSESSMENT OF IDS DATASETS

While considerable effort has been devoted to the creation of IDS datasets, their systematic evaluation and assessment remain underexplored. This chapter focuses on addressing that gap by introducing a comprehensive evaluation framework, grounded in the existing but limited research in this area.

3.1 Framework

The integrity and effectiveness of machine learning models in cybersecurity are heavily dependent on the quality and comprehensiveness of the datasets employed. To enable a structured, reproducible, and objective evaluation of dataset, we adopted the framework developed by the Canadian Institute for Cybersecurity (CIC)[4], which defines 11 criteria’s for evaluating Intrusion Detection System (IDS) datasets. These 11 core features are:

**1. Complete Network configuration:** A fully configured and operational network setup is essential to mirror the real-world environment accurately. Many cyberattacks reveal their true complexity only in networks that include all critical components like multiple PCs, servers, routers, and firewalls. Without this realism, datasets fail to reflect actual attack behaviours.

**2. Complete Traffic:** Network traffic consists of data packets traveling between sources and destinations—these could be individual hosts, routers, or switches. Depending on how traffic is generated, it can be authentic, near-authentic, or entirely artificial. To build meaningful datasets, traffic should resemble real-world communication patterns as closely as possible.

**3.Labelled dataset:** Datasets are only as valuable as their annotations. Without correct labelling of normal and malicious behaviours, no evaluation or comparison of detection systems holds any credibility. Labels underpin the reliability of every insight drawn from the data.

**4. Complete Interaction:** It’s crucial to include every layer of network interaction, whether internal communication within LANs or traffic between different network zones. This depth of interaction allows for accurate interpretation of anomalous behaviour and improves the evaluation of detection models.

**5. Complete Capture:** Partial data capture undermines the effectiveness of IDS evaluation. Every packet, whether functional or seemingly trivial can hold significance, especially for measuring false-positive rates. Removing non-labelled data, as some datasets do, limits researchers’ visibility into the full traffic spectrum.

**6. Available Protocols:** A complete dataset must reflect a wide variety of network behaviour types. This includes bursty traffic, such as FTP and HTTP, which occurs in spikes, and interactive sessions like web browsing, which involve real-time exchanges. Including both normal and malicious forms of these interactions is vital.

**7. Attack Diversity:** Cyber threats have become increasingly complex, especially with the rise of application-layer and hybrid attacks. A robust dataset must include contemporary attack vectors, categorized here into five groups based on protocol type and attack methodology:

* **TCP Reflective**
* **UDP Reflective**
* **TCP Exploitative**
* **UDP Exploitative**
* **Combined TCP/UDP Attacks**

Reflective attacks misuse third-party devices to unwittingly flood targets, while exploitative attacks directly leverage vulnerabilities in systems.

8. **Anonymity:** Balancing data utility with user privacy is a major challenge. While many datasets strip payloads to anonymize data, doing so severely limits their usefulness—especially for detection techniques like deep packet inspection. A smarter approach is needed that safeguards privacy but retains analytic value.

**9. Heterogeneity:** Datasets can originate from several sources—traffic flows, system logs, or network device outputs. A dataset drawing from only one source might suit specialized use cases, but for comprehensive evaluation, heterogeneous datasets that span multiple origins are far more effective.

**10. Feature set:** The value of a dataset largely depends on its ability to support effective feature extraction. Researchers should be able to derive meaningful characteristics from traffic or logs using standard tools. Poorly structured data limits experimentation and reduces reusability.

**11. Metadata:** Without clear documentation, even well-structured datasets lose their impact. Researchers need transparency on the setup: What operating systems were used? What were the attacker and victim configurations? What scenarios were run? These details are essential for reproducibility and trust in findings.

As DDoS attacks continue to grow in sophistication and variety, it became clear that a broader lens was needed. To address this, we expanded the original CIC framework by adding a new criterion—**DDoS Diversity**. This enhancement helps us better assess how well a dataset captures the full range of DDoS behaviours, ensuring a more realistic and valuable foundation for research and model training.

3.2 Evaluation

Equation (1) is used to measure the proposed framework. In this equation,

* **Wᵢ** represents the weight (or importance) of each feature, defined based on organizational needs or IDS types.
* **Vⱼ** is the weight assigned to each sub-factor, particularly relevant for multi-layered features like "attacks" and "protocols."
* **Fⱼ** indicates the presence or occurrence of a feature or sub-feature in the dataset (can be binary or multi-valued).

There are 12 key features (thus, 12 W values). Two features, “attack types” and “protocol types” contain 5 sub-categories each (hence m = 5 for them), while all other features are single-valued (m = 1). This structured yet flexible evaluation ensures that the dataset's relevance, depth, and applicability to real-world IDS testing are quantitatively measurable.

(1)

To maintain objectivity and avoid inductive bias in the evaluation process, each of the 12 criterions has been assigned an equal weight as shown in table 3.1. This uniform distribution ensures that no single attribute disproportionately influences the overall score, allowing for a balanced and comprehensive assessment of each dataset’s suitability.

Table 3.1 Assigned weights for 12 criterions

|  |  |  |
| --- | --- | --- |
| **S. No** | **Criteria** | **Weights** |
| 1 | Network | 0.05 |
| 2 | Traffic | 0.05 |
| 3 | Labelled Dataset | 0.05 |
| 4 | Interaction | 0.05 |
| 5 | Capture | 0.05 |
| 6 | Available Protocols | 0.25 |
| 7 | Attack Diversity | 0.25 |
| 8 | Anonymity | 0.05 |
| 9 | Heterogeneity: | 0.05 |
| 10 | Feature Set | 0.05 |
| 11 | Metadata | 0.05 |
| 12 | DDoS Diversity | 0.05 |

While all criterions are equally important, Available Protocols and Attack Diversity are each assigned a weight of 0.25 to reflect their internal complexity. These two criteria encompass multiple subcategories where, Available Protocols spans across various communication types like HTTP, HTTPS, SSH, FTP, and Email, while Attack Diversity encompasses five defined divisions-TCP Reflective, UDP Reflective, TCP Exploitative, UDP Exploitative, and TCP/UDP Mixed.

Based on [NETSCOUT 1H 2024 Report](https://www.netscout.com/threatreport/1h2024/ddos-attack-vectors/) [6], we analysed and scored key Distributed Denial of Service (DDoS) attack vectors across five major categories (D1–D5). Each category reflects a distinct set of attack methods used in the wild:

* **D1** contains attacks like MSSQL and SSDP
* **D2** is led by SYN flood attacks
* **D3** captures exploits of older protocols like NTP and TFTP.
* **D4** contains attacks like UDP, UDP\_Lag
* **D5** contains attacks like DNS, LDAP, and SNMP

Table 3.2 Based on NETSCOUT Report

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| D1 | No of Attacks | D2 | No of Attacks | D3 | No of Attacks | D4 | No of  Attacks | D5 | No of Attacks |
| MSSQL | 109328 | SYN | 1742635 | NTP | 527985 | UDP | 0 | DNS | 2340548 |
| SSDP | 153128 |  |  | TFTP | 24558 | UDP\_Lag | 0 | LDAP | 77800 |
|  |  |  |  |  |  |  |  | NETBIOS | 51371 |
|  |  |  |  |  |  |  |  | SNMP | 92059 |
| Total | 262456 |  | 1742635 |  | 552543 |  | 0 |  | 2561778 |
| Proportion | 0.051267 |  | 0.34039 |  | 0.1079 |  | 0 |  | 0.500405 |

To ensure equitable contribution across all divisions, first determined each division's proportion relative to the total, then scaled these proportions so that their collective sum equals 0.25, maintaining balanced representation in the overall allocation as shown in table III.

Table 3.3 Final weights

|  |  |  |
| --- | --- | --- |
| S. No | Diversity | Weight |
| 1 | D1 | 0.012817 |
| 2 | D2 | 0.085099 |
| 3 | D3 | 0.026983 |
| 4 | D4 | 0 |
| 5 | D5 | 0.125101 |
|  | Total | 0.25 |

With this framework, available IDS datasets are evaluated by calculating scores with above equation and weights.

TAB**LE 3.4 Comparison between datasets based on proposed evaluation framework**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Dataset | Network | Traffic | Labelled dataset | Interaction | Capture | Available Protocols | | | | | Attack Diversity | | | | | | Anonymity | | Hetero | | FS | | Metadata | | Diversity | |
| http | https | SSH | FTP | Email | D1 | D2 | D3 | D4 | D5 |  | |  | |  | |  | |  | |
| **DARPA** | ✅ | ❌ | ✅ | ✅ | ✅ | ✅ | ❌ | ✅ | ✅ | ✅ | ❌ | ✅ | ❌ | ❌ | ❌ | ❌ | | ❌ | | ❌ | | ✅ | | ❌ | |
| **KDD** | ✅ | ❌ | ✅ | ✅ | ✅ | ✅ | ❌ | ✅ | ✅ | ✅ | ❌ | ✅ | ❌ | ❌ | ❌ | ❌ | | ❌ | | ✅ | | ✅ | | ❌ | |
| **DEFCON** | ❌ | ❌ | ❌ | ✅ | ✅ | ✅ | ❌ | ✅ | ❌ | ❌ | ❌ | ✅ | ❌ | ✅ | ❌ | ❌ | | ❌ | | ❌ | | ❌ | | ❌ | |
| **CAIDA** | ✅ | ✅ | ❌ | ❌ | ❌ | - | - | - | - | - | ✅ | ✅ | ✅ | ✅ | ✅ | ✅ | | ❌ | | ❌ | | ✅ | | ✅ | |
| **LBNL** | ✅ | ✅ | ❌ | ❌ | ❌ | ✅ | ❌ | ✅ | ❌ | ❌ | ❌ | ❌ | ❌ | ❌ | ❌ | ✅ | | ❌ | | ❌ | | ❌ | | ❌ | |
| **CDX** | ❌ | ❌ | ❌ | ✅ | ✅ | ✅ | ❌ | ✅ | ✅ | ✅ | ❌ | ✅ | ❌ | ✅ | ❌ | - | | ❌ | | ❌ | | ❌ | | ✅ | |
| **KYOTO** | ✅ | ❌ | ✅ | ✅ | ✅ | ✅ | ✅ | ✅ | ✅ | ✅ | ❌ | ✅ | ❌ | ✅ | ❌ | ❌ | | ❌ | | ✅ | | ✅ | | ✅ | |
| **TWENTE** | ✅ | ✅ | ✅ | ✅ | ✅ | ✅ | ❌ | ✅ | ✅ | ❌ | ❌ | ✅ | ❌ | ✅ | ❌ | - | | - | | ❌ | | ✅ | | ✅ | |
| **UMASS** | ✅ | ❌ | ✅ | ❌ | ✅ | ✅ | ❌ | ❌ | ❌ | ❌ | ❌ | ✅ | ❌ | ✅ | ❌ | - | | - | | ❌ | | ❌ | | ✅ | |
| **ISCX**  **2012** | ✅ | ❌ | ✅ | ✅ | ✅ | ✅ | ❌ | ✅ | ✅ | ✅ | ❌ | ❌ | ❌ | ✅ | ✅ | - | | ✅ | | ✅ | | ✅ | | ❌ | |
| **ADFA**  **2013** | ✅ | ✅ | ✅ | ✅ | ✅ | ✅ | ❌ | ✅ | ✅ | ✅ | ❌ | ❌ | ❌ | ❌ | ❌ | ❌ | | - | | ❌ | | ✅ | | ❌ | |
| **CICDDoS 2017** | ✅ | ✅ | ✅ | ✅ | ✅ | ✅ | ✅ | ✅ | ✅ | ✅ | ❌ | ✅ | ❌ | ✅ | ❌ | ✅ | | ✅ | | ✅ | | ✅ | | ✅ | |
| **CICDDoS 2019** | ✅ | ✅ | ✅ | ✅ | ✅ | ✅ | ✅ | ✅ | ✅ | ✅ | ✅ | ✅ | ✅ | ✅ | ✅ | ✅ | | ✅ | | ✅ | | ✅ | | ✅ | |

From table 3.5, the analysis clearly highlights CICDDoS 2019 as the most robust and versatile dataset among the evaluated IDS datasets. Scoring a perfect 1.0, it excels across all critical dimensions—ranging from data diversity and feature richness to availability and DDoS-specific relevance. Its balanced composition makes it exceptionally suited for real-world attack simulations and advanced threat detection research. Although newer datasets like CIC IoV 2024 and CICEV 2023 offer useful perspectives on DDoS attacks within emerging domains such as the Internet of Vehicles and electric vehicle infrastructure, their scope is domain-specific and not fully aligned with traditional computer network traffic.

TABLE 3.5 Available IDS datasets scores

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Dataset | Network | Traffic | Labelled | Interaction | Capture | Availability | Diversity | Anonymity | Heterogeneous | Feature selection | Metadata | DDoS Diversity | Dataset Score |
| **DARPA** | 0.05 | 0 | 0.05 | 0.05 | 0.05 | 0.2 | 0.085 | 0 | 0 | 0 | 0.05 | 0 | 0.53 |
| **KDD’99** | 0.05 | 0 | 0.05 | 0.05 | 0.05 | 0.2 | 0.085 | 0 | 0 | 0.05 | 0.05 | 0 | 0.58 |
| **DEFCON** | 0 | 0 | 0 | 0.05 | 0.05 | 0.1 | 0.085 | 0 | 0 | 0 | 0 | 0 | 0.28 |
| **CAIDAs** | 0.05 | 0.05 | 0 | 0 | 0 | 0 | 0.25 | 0.05 | 0 | 0 | 0.05 | 0.05 | 0.5 |
| **LBNL** | 0.05 | 0.05 | 0 | 0 | 0 | 0.1 | 0 | 0.05 | 0 | 0 | 0 | 0 | 0.25 |
| **CDX** | 0 | 0 | 0 | 0.05 | 0.05 | 0.2 | 0.085 | 0 | 0 | 0 | 0 | 0.05 | 0.43 |
| **KYOTO** | 0.05 | 0 | 0.05 | 0.05 | 0.05 | 0.25 | 0.085 | 0 | 0 | 0.05 | 0.05 | 0.05 | 0.68 |
| **TWENTE** | 0.05 | 0.05 | 0.05 | 0.05 | 0.05 | 0.15 | 0.085 | 0 | 0 | 0 | 0.05 | 0.05 | 0.58 |
| **UMASS** | 0.05 | 0 | 0.05 | 0 | 0.05 | 0.05 | 0.085 | 0 | 0 | 0 | 0 | 0.05 | 0.33 |
| **ISCX**  **2012** | 0.05 | 0 | 0.05 | 0.05 | 0.05 | 0.2 | 0.12 | 0 | 0.05 | 0.05 | 0.05 | 0 | 0.67 |
| **ADFA**  **2013** | 0.05 | 0.05 | 0.05 | 0.05 | 0.05 | 0.2 | 0 | 0 | 0 | 0 | 0.05 | 0 | 0.5 |
| **CICDDoS 2017** | 0.05 | 0.05 | 0.05 | 0.05 | 0.05 | 0.25 | 0.085 | 0.05 | 0.05 | 0.05 | 0.05 | 0.05 | 0.83 |
| **CICDDoS 2019** | 0.05 | 0.05 | 0.05 | 0.05 | 0.05 | 0.25 | 0.25 | 0.05 | 0.05 | 0.05 | 0.05 | 0.05 | 1 |

This chapter introduces a comprehensive framework consisting of 12 criteria designed to evaluate IDS datasets, building upon the original CIC model by incorporating DDoS Diversity. The criteria focus on important aspects such as network realism, the availability of labelled data, the diversity of attacks, and the quality of metadata. Each criterion is given a specific weight, with greater importance placed on protocols and types of attacks. To calibrate the diversity scoring, real-world DDoS attack data from NETSCOUT is utilized. The framework is applied to assess multiple datasets, and in this evaluation, CICDDoS 2019 stands out as the most thorough and balanced dataset, achieving a perfect score of 1.0 for its effectiveness in supporting intrusion detection research.

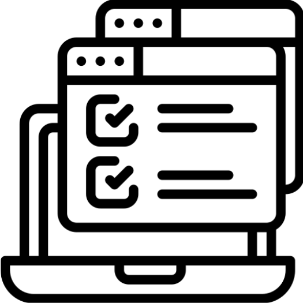
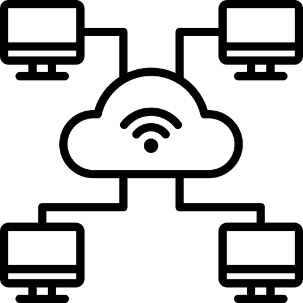
CHAPTER 4

METHODOLOGY

This chapter explores the early warning system designed for detecting DDoS attacks within network environments. Unlike conventional approaches to early warning signal generation in intrusion detection systems, we introduce a novel methodological leap, the application of Partial Differential Equations (PDEs) to extract new, dynamic features from the original dataset. This integration of mathematical modelling with cybersecurity results in a more vibrant and sensitive signal space, allowing us to uncover the subtle and often concealed patterns that precede cyber threats. By transforming static data points into dynamic systems using PDEs, we enhance our model's predictive capabilities while also expanding the way we think about feature engineering in IDS research.

4.1 System Architecture

Figure 4.1 presents the overall flow of the Early Warning and Emergency Alert System for network traffic monitoring. It brings together essential components, data collection, preprocessing, feature extraction, traffic classification, and alert generation into a cohesive pipeline built for real-time DDoS prevention. At the centre of this system is a smart control module that oversees how traffic data is processed and analysed. The following sections take a closer look at each module and how it helps strengthen the system’s ability to spot and respond to threats early.

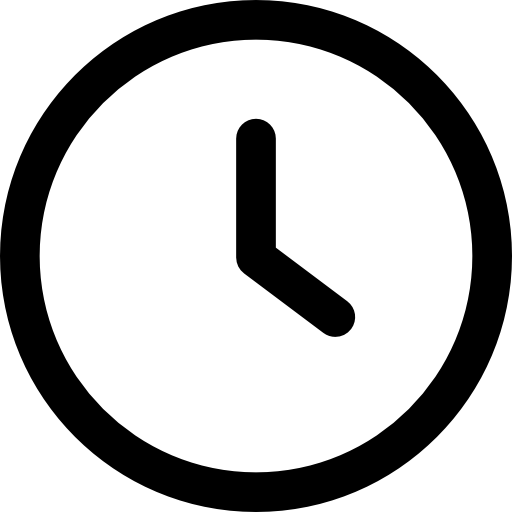


Capture PCAP data and convert to CSV

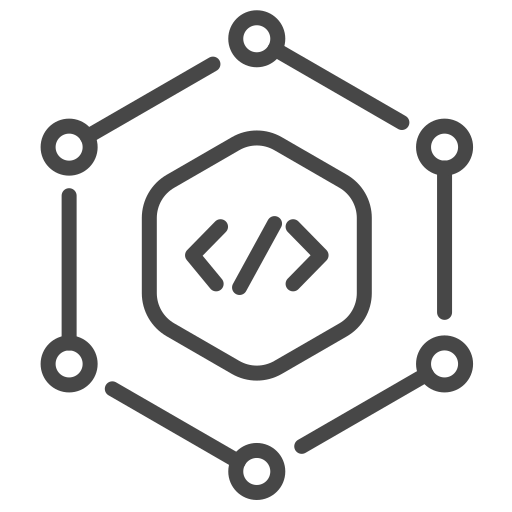
Network

Generate 87 features

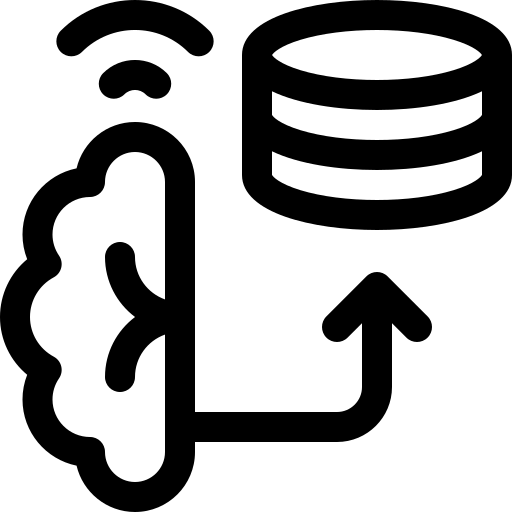
Replace Inf values with NaN and Drop NaN rows. Then group the data into fixed time intervals (Seconds)



Applying Partial Differential equation on train data and test data

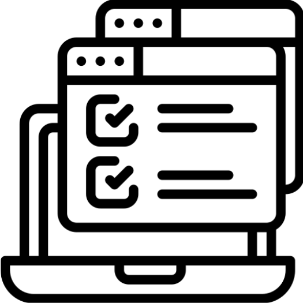
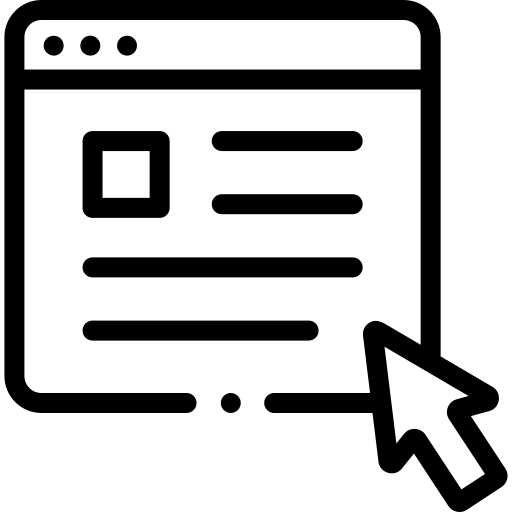
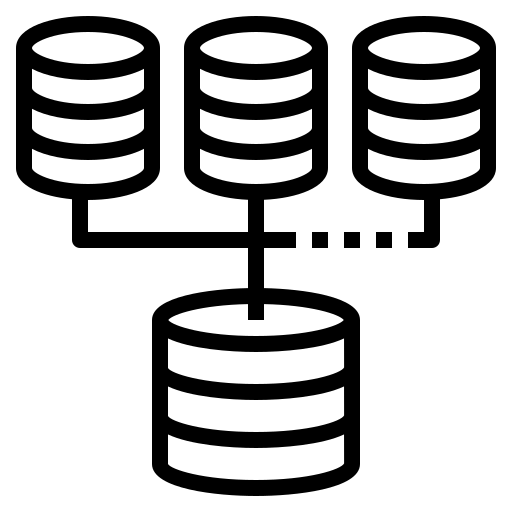


Compute first derivative:



Compute second derivative:

Train dataset is created with , , Seconds and Label

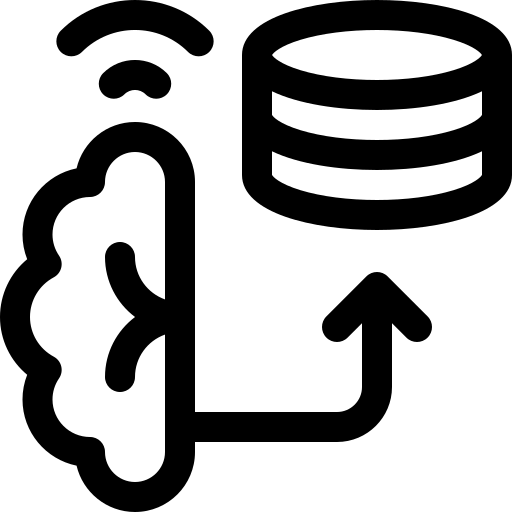


Capture PCAP data and convert to CSV

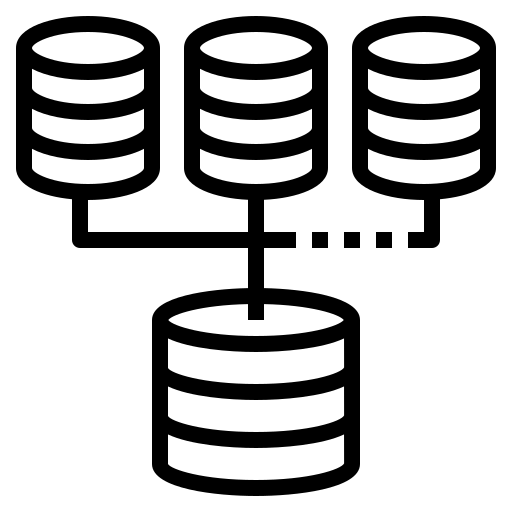
Generate 87 features

Applying Partial Differential equation on train data and test data

Compute first derivative:



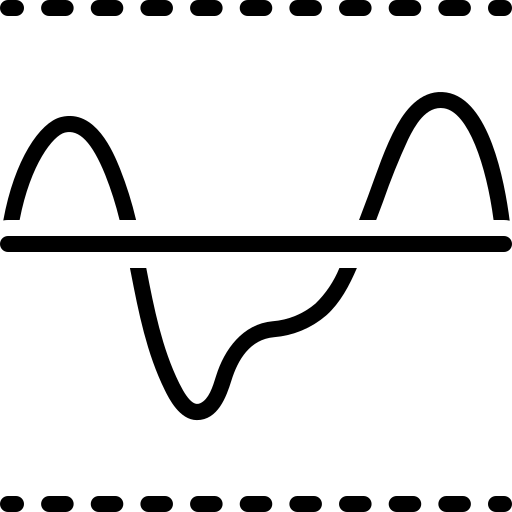
Compute second derivative:



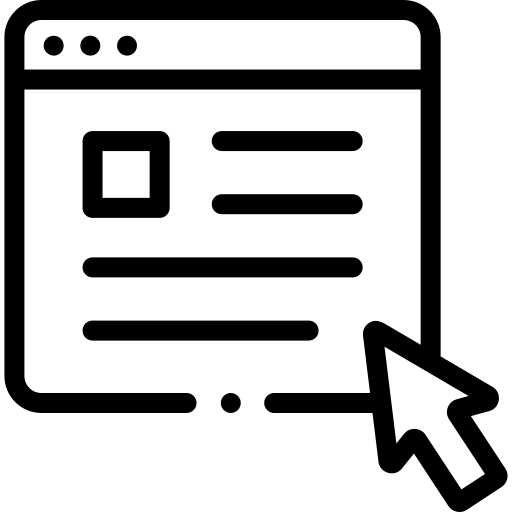
Test dataset is created with , , Seconds and Label

Compute values by fitting above derived values in Linear Regression

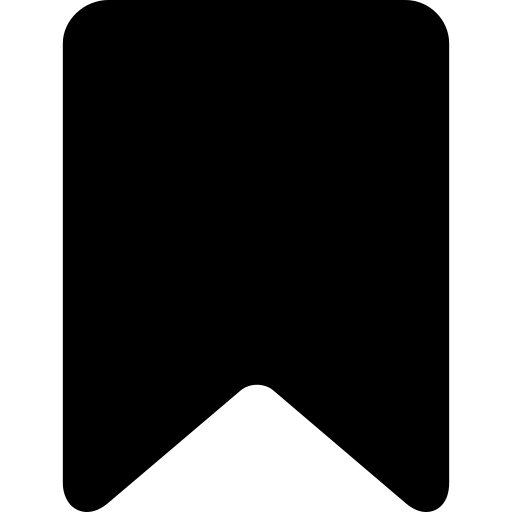
Calculate values of Residual-1,Residual-2,Mean, Standard Deviation and Z-score



Estimate thresholds on training data and store them to use for test data



Calculate values of Residual-1,Residual-2,Mean, Standard Deviation and Z-score



Train  
(Known Data)

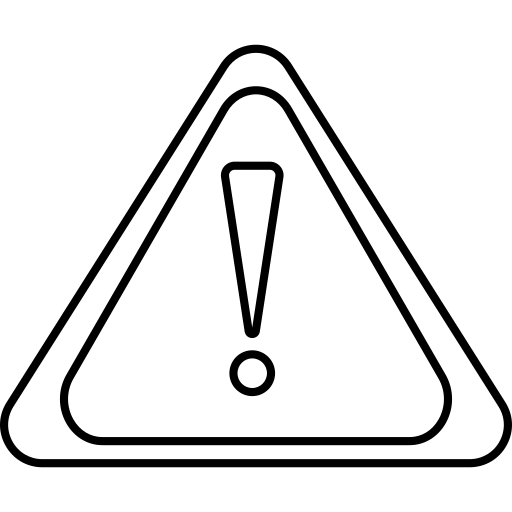
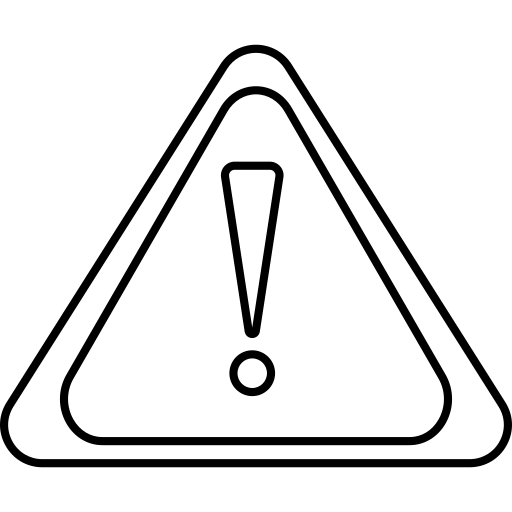
Test  
(unknown Data)



Predict threat via if condition of R1 and R2 and max(Z-score(R1,R2) for level of severity



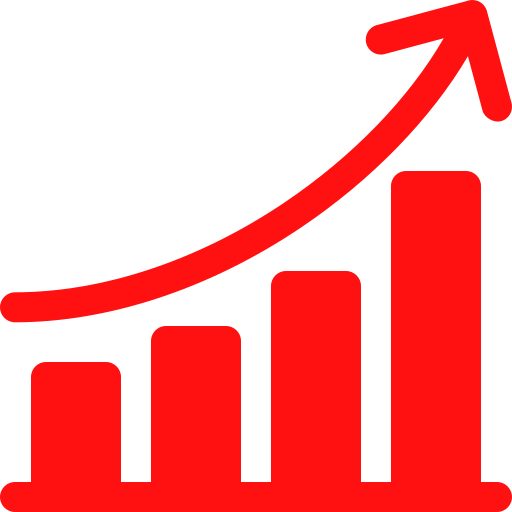
Predict threat via if condition of R1 and R2 and max(Z-score(R1,R2) for level of severity



Generate Early warning signals

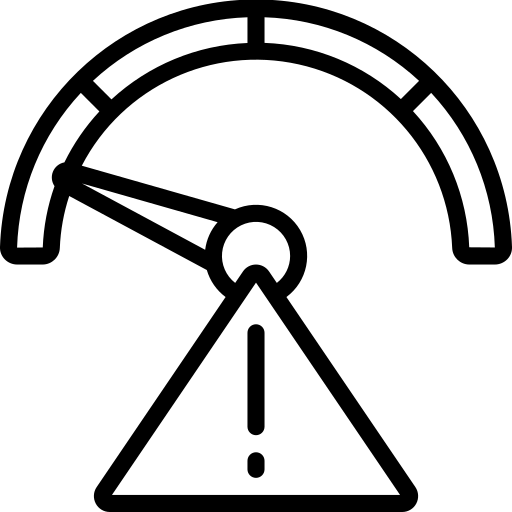
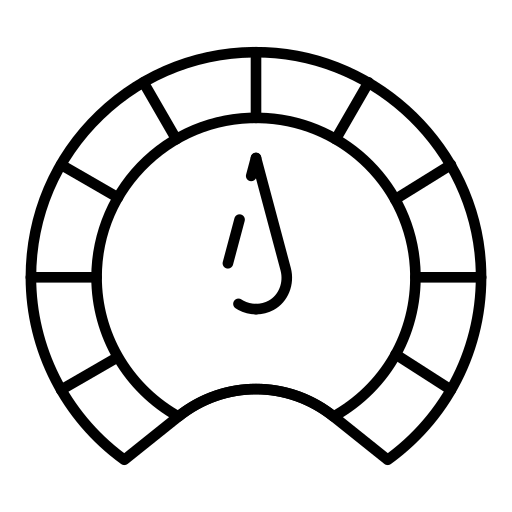
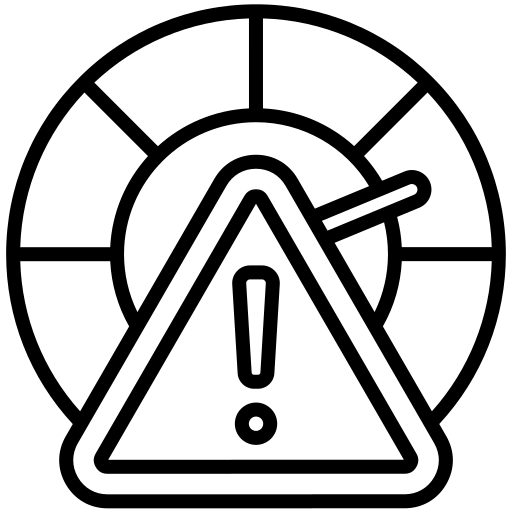
Generate Early warning signals

Load the saved values



Attack

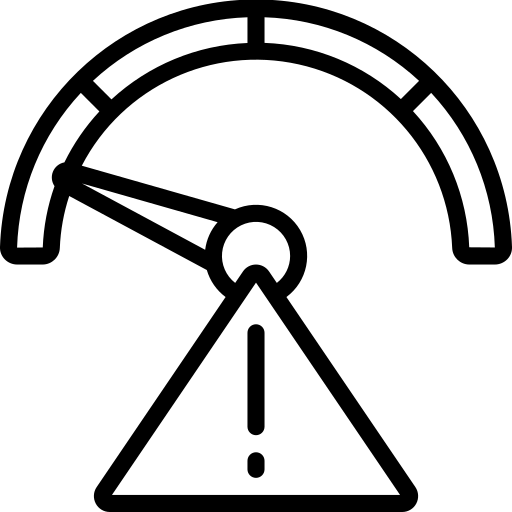
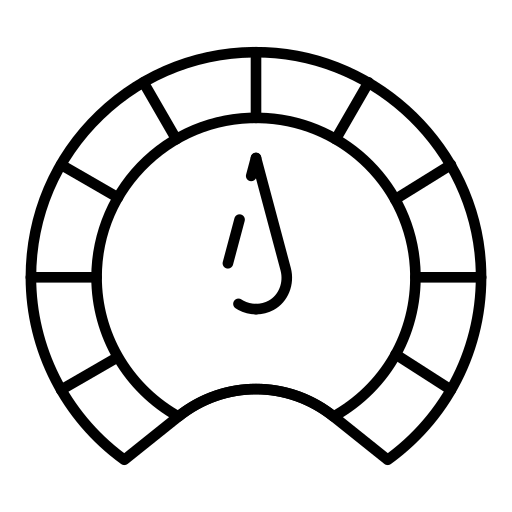
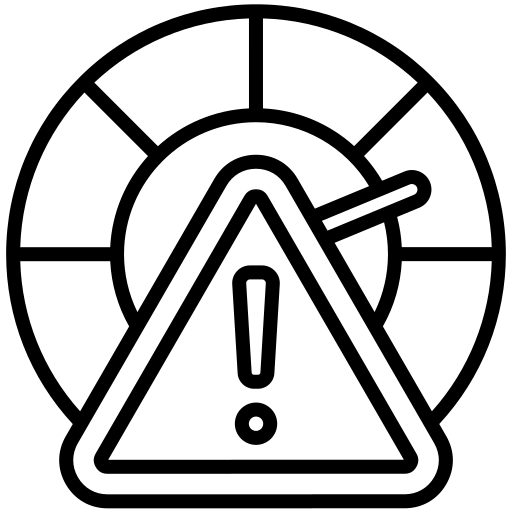
Benign



High alert

Medium alert

Low alert



High alert

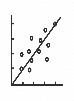
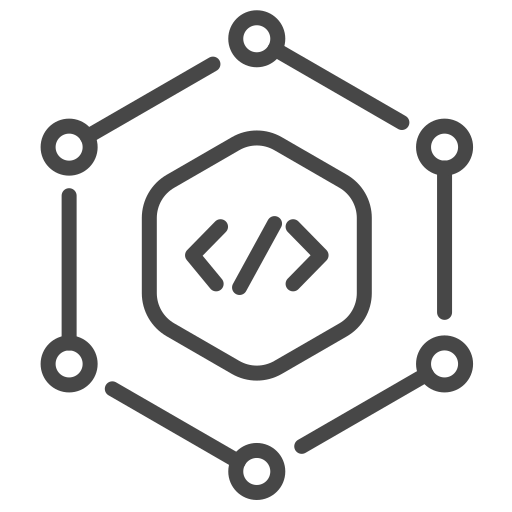
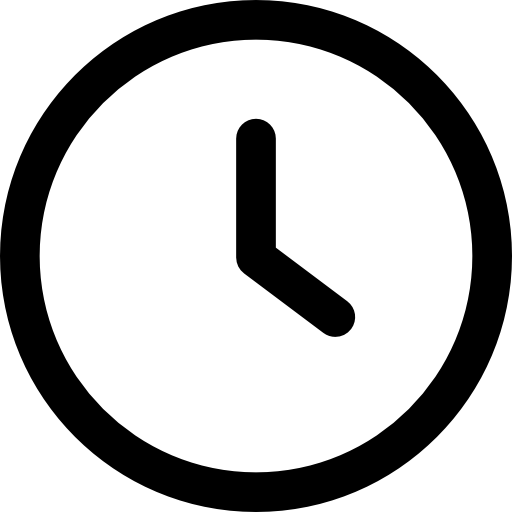
Medium alert

Low alert

Drop all the features except Label, Timestamp, Flow Packets/s (p), Flow Bytes/s(b)

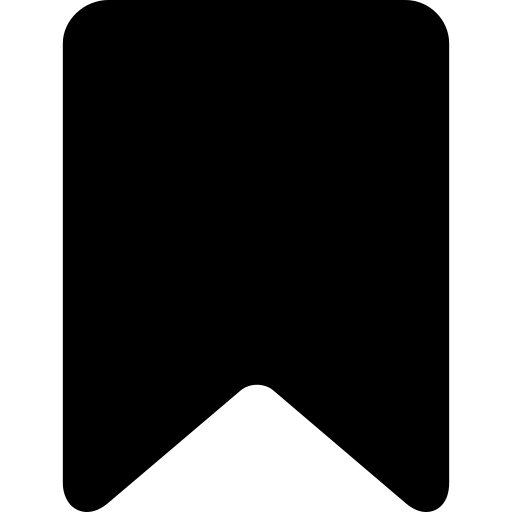
Drop all the features except Label, Timestamp, Flow Packets/s (p), Flow Bytes/s(b)

Replace Inf values with NaN and Drop NaN rows. Then group the data into fixed time intervals (Seconds)



[This Photo](https://machinelearning.glitch.me/) by Unknown Author is licensed under [CC BY-SA-NC](https://creativecommons.org/licenses/by-nc-sa/3.0/)





Load the saved thresholds

Very High alert

Very High alert

Fig 4.1 System Architecture

4.2 Data Preprocessing

Data preprocessing is a vital step in any machine learning workflow. During this phase, we address important tasks such as dealing with missing values, adjusting the scale of features, and encoding categorical data. These steps are essential to make sure our dataset is prepared for analysis. Figure 4.1 provides a straightforward overview of this process;

Load Network Traffic

Drop all features except  
[Timestamp, Flow Packets/s, Flow Bytes/s, Label]

Replace Inf values with NaN and Drop NaN rows

Group the data into fixed time intervals (Seconds) and aggregate Flow Packets/sec, Flow Bytes/sec

Fig 4.2 Data Preprocessing Module

We start by loading the network traffic dataset. For our analysis, we focus on key time-based attributes Flow Packets/s, Timestamp, and Label. After selecting these, we clean the data by replacing any infinite values with NaNs and removing them to maintain quality and consistency. Once cleaned, we break the dataset into equal time intervals (Δt), creating a structured foundation for deeper analysis. The next section walks through how we extract meaningful features from this preprocessed data.

4.3 Feature Extraction Module

Figure 4.3 illustrates how we extract features from Flow Packets/s and Flow Bytes/s to enable early warning signal detection and build an effective emergency alert system. This process involves computing time-based statistical and derivative features, which help the model better understand patterns over time and significantly boost its predictive accuracy.

Compute First Partial Derivates( , **)** using Central difference

Compute Second Partial Derivates ( **)** using Central difference

A Feature set is created with Flow Packets/s, Flow Bytes/s, , **,** and label

Load the aggregated data

Fig 4.3 Feature Extraction Module

4.3.1 Partial Derivatives Calculation

**Partial Differential Equations (PDEs):** A Partial Differential Equation is a type of mathematical expression that shows how something (like network traffic) changes across time and across different directions or dimensions (e.g., bytes/sec, packets/sec). It's like a rule that predicts how a small change in one aspect like (e.g., number of packets) causes changes in the whole system over time. The equations are:

**(1)**

**(2)**

Where:

* + : External input (e.g. Flow Packets/s)
  + : External input (e.g. Flow Bytes/s)
  + **:** First order partial derivative with respect to time – rate of change of
  + **:** First order partial derivative with respect to time – rate of change of
  + **:** Second order partial derivative with respect to time – rate of change of
  + **:** Second order partial derivative with respect to time – rate of change of
  + are parameters

The transformation function is stated as below.

First Partial Derivative of p and b are calculated using Central Difference Method

Central Difference Method:The Central Difference Method is a numerical technique used to approximate derivatives by evaluating the function at points symmetrically located around a given point. For a function f(x), the first derivative is

**(3)**

Where: h is a small step size (e.g., change in time)

The second partial derivative of a time-dependent function f’(x) at a discrete time point,​can be approximated using the second-order central difference method as:

**(4)**

First Partial Derivative:

**(5)**

**(6)**

Second Partial Derivative:

**(7)**

**(8)**

All extracted features Flow Packets/s, Flow Bytes/s, , **,** and label are augmented into a new dataset.

4.4 Early Warning Detection Module

Fig 4.3 shows the process of detecting early warning signals based on the extracted features using Z Scores

Estimate Thresholds on train data to identify attacks and z scores to identify level of attack

Apply Thresholds to test data obtained after feature extraction

Generate EWS based on the computed Z Scores to generate alerts

Categorize EWSs into Low, Medium and High alerts

Compute values by fitting above derived values in Linear Regression only for Train data and Benign Flow data

Compute Residuals, Mean, Standard Deviation, Quantiles and Z score

Fig 4.4 Early Warning Detection Module

4.4.1 Computation of Parameters ()

After calculating derivatives then we compute constants using linear regression

Linear regression is a statistical method used to model the relationship between a dependent variable and one or more independent variables by fitting a linear equation to observed data.

With the help of equations 1 and 2, values are calculated.

4.4.2 Computation of Residuals

From these computed values ***, ,*** we calculate Residual 1 and Residual 2 values, equations used are:

**(9)**

**(10)**

4.4.3 Computation of Statistical Features

We will calculate Mean, Standard Deviation, Q1, Q2, Q3 and IQR of Residual 1 and Residual 2 to calculate Threshold

**Mean x:**

**(11)**

**Standard Deviation σ:**

**(12)**

**Q1 (First Quartile)**

The value at the 25th percentile of the data:

value **(13)**

**Q2 (Second Quartile)**

The value at the 50th percentile of the data:

value **(14)**

**Q3 (Third Quartile)**

The value at the 75th percentile of the data:

value **(15)**

Interquartile Range (IQR)

**(16)**

4.4.4 Setting Thresholds from Training Data

When we calculate the Residuals on the training data set, we determine average and standard deviation values of Residuals, and these are our baseline statistics. Thresholds are established as:

**(17)**

4.4.5 Applying Thresholds to Test Data

For test data we compute the partial derivatives, residuals and with the threshold value that is calculated from training data is used to predict labels using statement,

IF statement:

If (Res1>Threshold1 or Res2>Threshold2) {

label =1

Else {

label=0

where, 1 indicates Attack and 0 indicates Benign

4.4.6 Determining Alert Levels Using Z Scores

We compute z score values for each residual to generate early warning signals

The z-score is a statistical measure that quantifies how many standard deviations a data point is from the mean of a data set. It indicates whether a data point is above or below the mean and by how much. The formula is:

**(18)**

Where:

**=** z score of Rt

**=** R1 or R2 at t sec

**=** Mean ofall values

**=** Standard Deviation of all values

After calculating Z scores for Res1 and Res2 then maximum value of their two is considered as final Z score at time t

Table 4.4.1 Association between Z Scores and Early Warning Signal Alert Level

|  |  |
| --- | --- |
| Z score | Alert Level |
|  | 0 |
|  | 1 |
|  | 2 |
|  | 3 |
|  | 4 |

From table 4.4.1, Alerts are generated, after calculating z scores

4.5 Emergency Warnings Module

Figure 4.5 provides a high-level view of how emergency alerts are triggered once the Early Warning System (EWS) starts detecting unusual activity. The system is designed to escalate alerts in a step-by-step manner, ensuring timely responses and effective threat mitigation as the situation unfolds.

.

Load the predictions from Early warning system

Generate first high alert

Generate Second high alert

Generate Third high alert

Generate continuous Emergency Alerts

Fig 4.5 Emergency Warnings Module

4.5.1 Load Predictions from Early Warning System (EWS)

The Early Warning System identifies anomalies by calculating z-scores, which show how much a data point stands out from the normal range

4.5.2 Generate First High Alert

As soon as the system detects a z-score higher than 2, it triggers the first high alert

4.5.3 Generate Second High Alert

If there is detection of a second assault not long after the first, there is a second high alert.

4.5.4 Generate Third High Alert

The third high alert denotes a pattern of persistent abnormal activity. It triggers constant emergency messages by the system regarding the new attack perceived.

4.5.5 Generate Continuous Emergency Alerts

When the system detects three high alerts in a row, it switches into emergency mode. In this mode, any new indicators of an attack trigger ongoing emergency alerts. This early warning is crucial for administrators, enabling them to respond quickly and often intervene before a significant threat has the chance to escalate.

4.6 Working example demonstrating the proposed method

Table 7 presents the sample network traffic flow data sampled from training day to demonstrate the working for the proposed method.

Table 4.6.1 Sample Network Traffic Flow

|  |  |  |  |
| --- | --- | --- | --- |
| **Timestamp** | **Flow Packets/s** | **Flow Bytes/s** | **Label** |
| 01-12-2018 13:05:13 | 0.711412443 | 595.6215986 | BENIGN |
| 01-12-2018 13:05:19 | 46783.62573 | 362573.0994 | BENIGN |
| 01-12-2018 13:05:55 | 193.6483346 | 7745.933385 | BENIGN |
| 01-12-2018 13:08:48 | 0.546512387 | 106.9146387 | BENIGN |
| 01-12-2018 13:09:46 | 88.90074232 | 10312.48611 | BENIGN |
| 01-12-2018 13:09:46 | 193.0967898 | 23364.71156 | BENIGN |
| 01-12-2018 13:04:46 | 2000000 | 766000000 | ATTACK |
| 01-12-2018 13:04:46 | 1000000 | 375000000 | ATTACK |
| 01-12-2018 13:04:47 | 42553.19149 | 16297872.34 | ATTACK |
| 01-12-2018 13:04:48 | 2000000 | 802000000 | ATTACK |
| 01-12-2018 12:37:02 | 27.94232704 | 9723.929809 | ATTACK |
| 01-12-2018 12:37:02 | 36.68950588 | 13189.87737 | ATTACK |

**Step 1: Update Timestamp to the seconds and aggregate p (Flow Packets/s) by t**

Table 4.6.2 Aggregating Timestamp into Time in Seconds

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Instance** | **Timestamp** | **Seconds** | **Flow Packets/s** | **Flow Bytes/s** | **Label** |
| 1 | 01-12-2018 13:05:13 | 1 | 0.711412443 | 595.6215986 | BENIGN |
| 2 | 01-12-2018 13:05:19 | 2 | 46783.62573 | 362573.0994 | BENIGN |
| 3 | 01-12-2018 13:05:55 | 3 | 193.6483346 | 7745.933385 | BENIGN |
| 4 | 01-12-2018 13:08:48 | 4 | 0.546512387 | 106.9146387 | BENIGN |
| 5 | 01-12-2018 13:09:46 | 5 | 88.90074232 | 10312.48611 | BENIGN |
| 6 | 01-12-2018 13:09:46 | 5 | 193.0967898 | 23364.71156 | BENIGN |
| 7 | 01-12-2018 13:04:46 | 6 | 2000000 | 766000000 | ATTACK |
| 8 | 01-12-2018 13:04:46 | 6 | 1000000 | 375000000 | ATTACK |
| 9 | 01-12-2018 13:04:47 | 7 | 42553.19149 | 16297872.34 | ATTACK |
| 10 | 01-12-2018 13:04:48 | 8 | 2000000 | 802000000 | ATTACK |
| 11 | 01-12-2018 12:37:02 | 9 | 27.94232704 | 9723.929809 | ATTACK |
| 12 | 01-12-2018 12:37:02 | 9 | 36.68950588 | 13189.87737 | ATTACK |

**Step 2: After aggregating the p and b values based on the t value**

When multiple flow values share the same second timestamp, their values are aggregated by summation.

Table 4.6.3 Aggregated Flow Packets/s per Second, Flow Bytes/s per Second (After Summation)

|  |  |  |  |
| --- | --- | --- | --- |
| t | Flow Packets/s | Flow Bytes/s | Label |
| 1 | 0.711412443 | 595.6215986 | BENIGN |
| 2 | 46783.62573 | 362573.0994 | BENIGN |
| 3 | 193.6483346 | 7745.933385 | BENIGN |
| 4 | 0.546512387 | 106.9146387 | BENIGN |
| 5 | 281.9975321 | 33677.19767 | BENIGN |
| 6 | 3000000 | 1141000000 | ATTACK |
| 7 | 42553.19149 | 16297872.34 | ATTACK |
| 8 | 2000000 | 802000000 | ATTACK |
| 9 | 64.63183292 | 22913.80718 | ATTACK |

**Step 3: Compute the First Partial Derivative** **using central difference method**

**Calculate**

96.46846108

-23391.53961

**Calculate**

3575.155893

-181233.0924

**Note:**  
Since there is no 0th second for the calculation of t1​, we compute it as:

46782.914317557

Similarly, for the last time point t9, there is no subsequent second. Hence, we compute it as:

-1999935.36816708

We apply the same approach to compute for t1​ and t9​.

Table 4.6.4 First Partial Derivative , Values Using Central Difference Method

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| t | Flow Packets/s | Flow Bytes/s |  |  |
| 1 | 0.711412443 | 595.6215986 | 46782.914317557 | 361977.4778 |
| 2 | 46783.62573 | 362573.0994 | 96.46846108 | 3575.155893 |
| 3 | 193.6483346 | 7745.933385 | -23391.53961 | -181233.0924 |
| 4 | 0.546512387 | 106.9146387 | 44.17459875 | 12965.63214 |
| 5 | 281.9975321 | 33677.19767 | 1499999.727 | 570499946.5 |
| 6 | 3000000 | 1141000000 | 21135.59698 | 8132097.571 |
| 7 | 42553.19149 | 16297872.34 | -500000 | -169500000 |
| 8 | 2000000 | 802000000 | -21244.27983 | -8137479.266 |
| 9 | 64.63183292 | 22913.80718 | -1999935.368167 | 801977086.2 |

**Step 4: Compute the Second Partial Derivative**   **central difference method**

-93372.8917

46396.87557

Similarly, for

-716804.6438

347188.147

**Note:**  
Since there is no 0th second for the calculation of t1​, we compute it as:

-46,686.445856477

Similarly, for the last time point t9, there is no subsequent second. Hence, we compute it as:

-19,78,691.088337

We apply the same approach to compute for t1​ and t9​.

We will calculate for all the t values.

The final dataset formed after all the derivatives are calculated is shown below.

Table 4.6.5 Second Partial Derivative , Values Using Central Difference Method

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| t | Flow Packets/s | Flow  Bytes/s |  |  |  |  |
| 1 | 0.71141 | 595.6215986 | 46782.914317557 | 361977.4778 | -46686.445856477 | -358402.3219 |
| 2 | 46783.625 | 362573.0994 | 96.46846108 | 3575.155893 | -93372.89171 | -716804.6438 |
| 3 | 193.6483 | 7745.933385 | -23391.53961 | -181233.0924 | 46396.87557 | 347188.1473 |
| 4 | 0.54651 | 106.9146387 | 44.17459875 | 12965.63214 | 474.5528419 | 41209.30178 |
| 5 | 281.9975 | 33677.19767 | 1499999.727 | 570499946.5 | 2999436.551 | 1140932753 |
| 6 | 3000000 | 1141000000 | 21135.59698 | 8132097.571 | -5957164.811 | -2265668450 |
| 7 | 42553.19 | 16297872.34 | -500000 | -169500000 | 4914893.617 | 1910404255 |
| 8 | 2000000 | 802000000 | -21244.27983 | -8137479.266 | -3957382.177 | -1587679214 |
| 9 | 64.63183 | 22913.80718 | -1999935.368167 | 801977086.2 | 1978691.08833708 | -793839606.92 |

**Step 5: From this above dataset, we calculate constants**  **using Linear Regression considering only benign values**

From the equation,

We calculate

From the above table at t=1,

t=2,

From these equations after solving, we get values

**α = -0.1305**

**β = 2.773e-09**

Similarly, from the equation,

We calculate

From the above table at t=1,

t=2,

From these equations after solving, we get values

**γ = -0.00775**

**δ = -1.6349 ×**

**Step 6: After calculating constants , then we will calculate Residues and the equations used to calculate are:**

**At t=1,**

Res1 = 46782.914317557- (0.4966886799846156 ) (2.2452963026238222e-06 0.711412443 595.6215986)

Res1=224797.289538256

Res2 = 361977.4778- (190.0587617825867-46686.445856477

(0.008042480314603107 0.711412443 0.711412443)

Res2=9235145.5652428

Similarly, we will calculate for all the t values.

Table 4.6.6 Calculation of Residuals (Res1, Res2)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| t | Flow Packets/s | Flow  Bytes/s |  |  |  |  | Res1 | Res2 |
| 1 | 0.71141 | 595.621 | 46782.9143 | 361977.4778 | -46686.445 | 358402.32190 | 224797.289 | 9235145.565 |
| 2 | 46783.625 | 362573.0 | 96.4684 | 3575.155893 | -93372.891 | -716804.6438 | 318039.417 | 147273.257 |
| 3 | 193.6483 | 7745.933 | -23391.5396 | -181233.0924 | 46396.875 | 347188.1473 | -195839.330 | -8999667.404 |
| 4 | 0.54651 | 106.914 | 44.1745 | 12965.63214 | 474.552 | 41209.30178 | -20424.019 | -77227.295 |
| 5 | 281.9975 | 33677.197 | 1499999.727 | 570499946.5 | 2999436.55 | 1140932753 | -565188404.63 | 430110.0124 |
| 6 | 3000000 | 1141000000 | 21135.596 | 8132097.571 | -5957164.811 | -2265668450 | -6560296236.57 | -71241979366.1 |
| 7 | 42553.19 | 16297872.34 | -500000 | -169500000 | 4914893.617 | 1910404255 | -950933340.100 | -1118181710.2 |
| 8 | 2000000 | 802000000 | -21244.2798 | -8137479.266 | -3957382.177 | -1587679214 | -2812894220.647 | -31425923581.21 |
| 9 | 64.63183 | 22913.807 | -1999935.368 | 801977086.2 | 1978691.08 | -793839606.926 | 392291207.79 | -425909541.58 |

**Step 7: Now from the above dataset we will calculate Mean, Standard Deviation, Q1, Q2, Q3 and IQR of Residual 1 and Residual 2 to calculate Threshold**

The first 4 values of Flow packets/s are: (x1,x2,x3,x4) = (224797.289538256, 318039.41777957, -195839.330114231, -20424.0192366389)

Step 1: calculate the mean using the below formula

=81643.339491739025

Step 2: calculate Standard deviation using the below formula

= 233659.68​

Step 3: calculate Q1, Q2, Q3 and IQR using the below formula

Sort the R1 values to find Q1, Q2, Q3 and IQR

value

value

value

value

value

value

IQR = Q3-Q1

=

= 951158137.389552

Now, calculate Threshold value

**Threshold for Res1:**

Threshold =

=

=1427062000

Similarly,

For Res2, threshold is calculated

**Step 8: After calculating threshold, Labels are predicted using**

IF statement:

If(Res1>T1 or Res2>T2) {

label =1

Else {

label=0

where, 1 indicates Attack and 0 indicates Benign

for t=1,

If(Res1>T1 or Res2>T2) {

label =1

Else {

label=0

Res1 =224797.289

Re2 =9235145.5652

IF (224797.289 > 1427062000 OR 9235145.5652 > 1427062000) {

label =1

Else {

label=0

Similarly, labels are predicted for all t values

Table 4.6.7 Prediction of Label with Threshold

|  |  |  |  |
| --- | --- | --- | --- |
| t | Res1 | Res2 | Label |
| 1 | 224797.289 | 9235145.565 | 0 |
| 2 | 318039.417 | 147273.257 | 0 |
| 3 | -195839.330 | -8999667.404 | 0 |
| 4 | -20424.019 | -77227.295 | 0 |
| 5 | -565188404.63 | 430110.0124 | 0 |
| 6 | -6560296236.57 | -71241979366.1 | 0 |
| 7 | -950933340.100 | -1118181710.2 | 0 |
| 8 | -2812894220.64 | -31425923581.21 | 0 |
| 9 | 392291207.79 | -425909541.58 | 0 |

**Step 9: Computing z score values using formula**

Where:

**=** z score of Rt

**=** R1 or R2 at t sec

**=** Mean ofall values

**=** Standard Deviation of all values

**Step 9.1: calculate z score for R1 values**

at t=1,

0.4139

We will calculate z scores for all t values of R1

**Step 9.2: calculate z score for R2 values**

at t=1,

0.356

We will calculate z scores for all t values of R2

After calculating Z scores for R1 and R2 then maximum value of their two is considered as final Z score at time t

at t=1,

0.4139, 0.356)0.4139

Similarly, is calculated for all t values

Table 4.6.8 Calculated Z scores for all t values

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| t | R1 | R2 | Label | Z scores |
| 1 | 224797.289 | 9235145.5652 | 0 | 0.4139 |
| 2 | 318039.41777 | 147273.25782 | 0 | 0.4140 |
| 3 | -195839.33011 | -8999667.4042 | 0 | 0.4137 |
| 4 | -20424.019236 | -77227.29579 | 0 | 0.4138 |
| 5 | -565188404.63 | 430110.0124 | 0 | 0.355 |
| 6 | -6560296236.5 | -71241979366 | 0 | -2.9496 |
| 7 | -950933340.10 | -1118181710.2 | 0 | 0.2982 |
| 8 | -2812894220.6 | -31425923581.2 | 0 | -1.0269 |
| 9 | 392291207.790 | -425909541.5 | 0 | 0.6148 |

**Step 10: Generating Alerts with z score values**

Table 4.6.9 Association between Z Scores and Early Warning Signal Alert Level

|  |  |
| --- | --- |
| Z score | Alert Level |
|  | 0 |
|  | 1 |
|  | 2 |
|  | 3 |
|  | 4 |

After calculating the z score values, we will generate Alerts using above table

At t=1,

Z score = 0.4139

From above table z value is greater than 0 and less than 1, hence the alert

level is 1

At t=6,

Z score = -2.94

From above table z value is less than 0, hence the alert

level is 1

Similarly, alerts are generated for all t values

Table 4.6.10 Generated Alert values for all seconds

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| t | R1 | R2 | Label | Z score | Alert  Level |
| 1 | 224797.289 | 9235145.5652 | 0 | 0.41 | 1 |
| 2 | 318039.41777 | 147273.25782 | 0 | 0.41 | 1 |
| 3 | -195839.33011 | -8999667.4042 | 0 | 0.41 | 1 |
| 4 | -20424.019236 | -77227.29579 | 0 | 0.41 | 1 |
| 5 | -565188404.63 | 430110.0124 | 0 | 0.35 | 1 |
| 6 | -6560296236.5 | -71241979366 | 0 | -2.94 | 0 |
| 7 | -950933340.10 | -1118181710.2 | 0 | 0.29 | 1 |
| 8 | -2812894220.6 | -31425923581 | 0 | -1.02 | 0 |
| 9 | 392291207.790 | -425909541.5 | 0 | 0.61 | 1 |

The warning signals are indicated by rows where the alert level is 1. From the table, the rows with a final alert level of 1 or higher are as follows:

* 1. Row 1: Alert Level = 1
  2. Row 2: Alert Level = 1
  3. Row 3: Alert Level = 1
  4. Row 4: Alert Level = 1
  5. Row 5: Alert Level = 1
  6. Row 7: Alert Level = 1
  7. Row 9: Alert Level = 1

This chapter presents an innovative approach for an early warning system that detects DDoS attacks using partial differential equations (PDEs). It takes the usual static traffic data and turns it into dynamic features by calculating both first and second derivatives. The process involves several key steps: cleaning the data, aggregating it based on time, computing the derivatives, and analysing the residuals. To spot unusual patterns, Z-scores are computed, which helps in generating alerts ranging from low to high levels. If the system detects three consecutive high-level alerts, it automatically switches to a continuous emergency mode. Through a practical example, the effectiveness of this method in forecasting threats based on real network traffic is clearly illustrated.

CHAPTER 5

RESULTS

This chapter discusses the performance assessment of the proposed Early Warning Signal-driven Intrusion Prevention System (EWS IPS) utilizing the CIC-DDoS2019 dataset[5]. The

dataset consists of wide range of DDoS attacks, total of 13 modern day DDoS attacks with 87 traffic features. The training dataset is a 5-crore dataset consisting of 12 DDoS attacks and testing dataset is a 2-crore dataset consisting of 7 DDoS attacks. For our project, we are selectively focusing on a subset of the CICDDoS2019 dataset specifically, instances labelled as BENIGN, UDP, and UDPLag. This targeted selection allows us to create a clear contrast between normal and attack traffic, particularly emphasizing the behavioural patterns of UDP-based DDoS attacks, while keeping the dataset manageable and relevant to our use case. The assessment was carried out regarding statistical anomaly identification through Partial Differential Equation (PDE’s) and Z-score, reliability of multi-signal alerts, and the time benefit before the attack commences. The system underwent testing using data from both the training day and the testing day to verify its real-time performance.

To evaluate the proposed methodology, our performance measurements are centered on assessing early attack detection. These include the false positive rate during benign periods, the true positive rate during the pre-attack phase, and the detection lead time, which gauges how early the model predicts an attack before it starts.

**System configuration**

Processor: AMD Ryzen 5 3550H with Radeon Vega Mobile Gfx 2.10 GHz

Installed RAM: 8.00 GB (7.69 GB usable)

Base speed: 2.10 GHz

Sockets: 1

Cores: 2

Logical processors: 4

Virtualization: Enabled

L1 cache: 192 KB

L2 cache: 1.0 MB

L3 cache: 4.0 MB

Threads 2996

Handles 97212

Memory 8.0 GB

Speed: 2400 MT/s

Available 2.3 GB

5.1 Data Preparation

Data preparation is a critical first step in any machine learning. It ensures the data is clean, consistent, and suitable for analysis. Without it, even the most sophisticated models can produce inaccurate or biased results.

5.1.1 Data Selection

For our project, we are selectively focusing on a subset of the CICDDoS2019 dataset specifically instances labelled as BENIGN, UDP, and UDPLag, while keeping the dataset manageable and relevant to our use case.

Table 5.1.1 presents a deliberate selection of specific network traffic types used in the training dataset. Out of many possible attacks, only BENIGN, UDP, and UDP-Lag were chosen to ensure a more focused and balanced dataset.

Table 5.1.1 Selection of data from Training data

|  |  |  |  |
| --- | --- | --- | --- |
| **S. No** | **Label** | **Network flow** | **Selected?** |
| 1 | BENIGN | 56863 | ✔ |
| 2 | NTP | 1202642 |  |
| 3 | DNS | 507011 |  |
| 4 | LDAP | 2179930 |  |
| 5 | MSSQL | 4522492 |  |
| 6 | NetBIOS | 4093279 |  |
| 7 | SNMP | 5159870 |  |
| 8 | SSDP | 2610611 |  |
| 9 | UDP | 3134645 | ✔ |
| 10 | UDP-Lag | 366461 | ✔ |
| 11 | WebDDoS | 439 |  |
| 12 | SYN | 1582289 |  |
| 13 | TFTP | 20082580 |  |

Table 5.1.2 presents a deliberate selection of specific network traffic types used in the testing dataset. Out of many possible attacks, only BENIGN, UDP, and UDP-Lag were chosen to ensure a more focused and balanced dataset.

Table 5.1.2 Selection of data from Testing data

|  |  |  |  |
| --- | --- | --- | --- |
| **S. No** | **Label** | **Network flow** | **Selected?** |
| 1 | BENIGN | 56965 | ✔ |
| 2 | PortMap | 186960 |  |
| 3 | NetBIOS | 3657497 |  |
| 4 | LDAP | 1915122 |  |
| 5 | MSSQL | 5787453 |  |
| 6 | Syn | 4284751 |  |
| 7 | UDP | 3867155 | ✔ |
| 8 | UDPLag | 1873 | ✔ |

To focus our analysis on the behavioural patterns of network traffic, we specifically selected the time-based attributes from the original 87 features. These features such as Flow Packets/sec, Flow Bytes/sec are highly indicative of traffic flow dynamics and are especially useful for detecting the timing irregularities common in DDoS attacks. This approach helps us better understand the flow and frequency of traffic, which plays a crucial role in distinguishing between normal and malicious activity.

5.1.2 Data Preprocessing

Data preprocessing is a vital step in any machine learning workflow. During this phase, we address important tasks such as dealing with missing values, adjusting the scale of features, and encoding categorical data.

After selecting the time-based attributes (Flow Packets/sec, Flow Bytes/sec), we clean the data by replacing any infinite values with NaNs and removing them to maintain quality and consistency. Once cleaned, we break the dataset into equal time intervals (Δt) creating a structured foundation for deeper analysis.

Table 5.1.2 Sample Network Traffic Flow



5.1.3 Aggregation of time-based attributes

When multiple flow values share the same second timestamp, their values are aggregated by summation. This creates a structured dataset with equal time intervals. After aggregating, we get 1952 seconds in training data and 1405 second in testing data while considering BENIGN instances from UDP and UDPLag datasets.

Table 5.1.3 Aggregating Timestamp into Time in Seconds

****

5.2 Feature Extraction

We extract features from Flow Packets/s and Flow Bytes/s to spot early warning signals and power a smart emergency alert system. By calculating time-based stats and changes over time, we give the model a clearer view of evolving patterns sharpening its ability to predict issues before they escalate.

5.2.1 Computation of Derivatives

Computing first partial derivatives , using central difference method

Table 5.2.1.1 First Derivative , Values Using Central Difference Method



Computing second partial derivatives , using central difference method.

After computing derivatives new feature set is created with Timestamp, Flow Packets/sec, Flow Bytes/sec , , , and Label. Using this dataset, we calculate constants by fitting linear regression on only BENIGN instances in training data.

**Table 5.2.1.1 Second partial derivatives ,**  **values using Central Difference Method**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Timestamp | Flow Packets/sec | Flow Bytes/sec |  |  |  |  |
| 1 | 387.6257963 | 25093.71133 | -7142.136794 | 187.3747 | 13511.9 | -49812.7 |
| 2 | 1.44069809 | 374.7493518 | 24408.49112 | 245565.8 | 49589.35 | 540569.6 |
| 3 | 49204.60804 | 516225.3942 | 289.5198665 | 35539.56 | -97827.3 | -960622 |
| 4 | 580.480431 | 71453.87516 | -23683.3243 | -169818 | 49881.61 | 549908 |
| 5 | 1837.959446 | 176590.3242 | 524.7508531 | 472962.9 | -1465.46 | 735652.8 |
| 6 | 1629.982137 | 1017379.598 | -40.07704673 | -30913.2 | 335.8005 | -1743405 |
| 7 | 1757.805352 | 114763.8812 | 3365808.038 | 43988898 | 6731360 | 89783027 |
| 8 | 6733246.059 | 88995174.69 | 710.4011151 | 61924.53 | -1.3E+07 | -1.8E+08 |
| 9 | 3178.607583 | 238612.9452 | -3357722.932 | -4.4E+07 | 6744689 | 88517949 |
| 10 | 18095.59504 | 31552.66732 | -23486.23733 | 32825.55 | 44115.16 | 2545.763 |

5.3 Early Warning Detection

The dataset created with Timestamp, Flow Packets/sec, Flow Bytes/sec , , , and Label is used to compute residuals and z scores to detect early warning signals.

5.3.1 Computation of Residuals

From these computed values ***, ,*** we calculate Residual 1 and Residual 2 values, equations used are:

**(1)**

**(2)**

Table 5.3.1 represents the dataset formed after calculating Residuals (Res1, Res2)

Table 5.3.1 Calculation of Residuals (Res1, Res2)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Time  stamp | Flow  Packets/s | Flow  Bytes/s |  |  |  |  | Res1 | Res2 |
| 1 | 387.6257963 | 25093.71133 | -7142.1367 | 187.3747 | 13511.9 | -49812.7 | -7.21E+03 | 2.58E+06 |
| 2 | 1.44069809 | 374.7493518 | 24408.49112 | 245565.8 | 49589.35 | 540569.6 | 2.51E+04 | 9.71E+06 |
| 3 | 49204.60804 | 516225.3942 | 289.5198665 | 35539.56 | -97827.3 | -960622 | -9.61E+02 | -1.86E+09 |
| 4 | 580.480431 | 71453.87516 | -23683.3243 | -169818 | 49881.61 | 549908 | -2.30E+04 | 9.35E+06 |
| 5 | 1837.959446 | 176590.3242 | 524.7508531 | 472962.9 | -1465.46 | 735652.8 | 1.49E+03 | 1.93E+05 |
| 6 | 1629.982137 | 1017379.598 | -40.0770467 | -30913.2 | 335.8005 | -1743405 | -2.32E+03 | 3.33E+04 |
| 7 | 1757.805352 | 114763.8812 | 3365808.038 | 43988898 | 6731360 | 89783027 | 3.48E+06 | 1.33E+09 |
| 8 | 6733246.059 | 88995174.69 | 710.4011151 | 61924.53 | -1.3E+07 | -1.8E+08 | -6.18E+04 | -1.52E+09 |
| 9 | 3178.607583 | 238612.9452 | -3357722.93 | -4.4E+07 | 6744689 | 88517949 | -3.24E+06 | 1.24E+09 |
| 10 | 18095.59504 | 31552.66732 | -23486.2373 | 32825.55 | 44115.16 | 2545.763 | -2.35E+04 | 8.46E+06 |

5.3.2 Computation of Statistical Features and Thresholds and Z scores

We will calculate Mean, Standard Deviation, Q1, Q2, Q3 and IQR of Residual 1 and Residual 2 to calculate Threshold

We compute z score values for each residual to generate early warning signals

The formula is:

**(3)**

After calculating Z scores for Res1 and Res2 then maximum value of their two is considered as final Z score at time t

**(4)**

Table 5.3.2 Calculated Z scores for all t values

|  |  |  |  |
| --- | --- | --- | --- |
| Time  stamp | Res1 | Res2 | Z Scores |
| 1 | -7.21E+03 | 2.58E+06 | -1.24E-01 |
| 2 | 2.51E+04 | 9.71E+06 | -1.17E-01 |
| 3 | -9.61E+02 | -1.86E+09 | -1.23E-01 |
| 4 | -2.30E+04 | 9.35E+06 | -1.28E-01 |
| 5 | 1.49E+03 | 1.93E+05 | -1.22E-01 |
| 6 | -2.32E+03 | 3.33E+04 | -1.23E-01 |
| 7 | 3.48E+06 | 1.33E+09 | 1.02E+00 |
| 8 | -6.18E+04 | -1.52E+09 | -1.36E-01 |
| 9 | -3.24E+06 | 1.24E+09 | 9.35E-01 |
| 10 | -2.35E+04 | 8.46E+06 | -1.28E-01 |

5.3.3 Determining Alert Levels Using Z Scores

Table 5.3.3.1, represents the association between Z scores and Early Warning Signal Alert level,using this table we will generate alert levels.

Table 5.3.3.1 Association between Z Scores and Early Warning Signal Alert Level

|  |  |
| --- | --- |
| Z score | Alert Level |
|  | 0 |
|  | 1 |
|  | 2 |
|  | 3 |
|  | 4 |

Table 5.3.3.2 Generated Alert values for sample data

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Timestamp | Res1 | Res2 | Z Scores | AlertLevel |
| 1 | -7.21E+03 | 2.58E+06 | -1.24E-01 | 0 |
| 2 | 2.51E+04 | 9.71E+06 | -1.17E-01 | 0 |
| 3 | -9.61E+02 | -1.86E+09 | -1.23E-01 | 0 |
| 4 | -2.30E+04 | 9.35E+06 | -1.28E-01 | 0 |
| 5 | 1.49E+03 | 1.93E+05 | -1.22E-01 | 0 |
| 6 | -2.32E+03 | 3.33E+04 | -1.23E-01 | 0 |
| 7 | 3.48E+06 | 1.33E+09 | 1.02E+00 | 2 |
| 8 | -6.18E+04 | -1.52E+09 | -1.36E-01 | 0 |
| 9 | -3.24E+06 | 1.24E+09 | 9.35E-01 | 1 |
| 10 | -2.35E+04 | 8.46E+06 | -1.28E-01 | 0 |

The warning signals are indicated by rows where the alert level is 1. From the table, the rows with a final alert level of 1 or higher are as follows:

* 1. Row 7: Alert Level = 2
  2. Row 9: Alert Level = 1

In this way, early warning signal alerts are generated then after with this dataset graphs are plotted with respect to each attribute and timestamp.

5.4 Graphs

This case consist records of UDP and Benign as Train data and UDP-Lag and Benign as Test data which are extracted from pre-processed dataset.

The Partial Differential equation used for feature extraction from existing feature Flow packets per sec and Flow bytes per sec are equation 1 and 2.

|  |  |
| --- | --- |
|  | 1 |
|  | 2 |

5.4.1 Attribute Graphs:

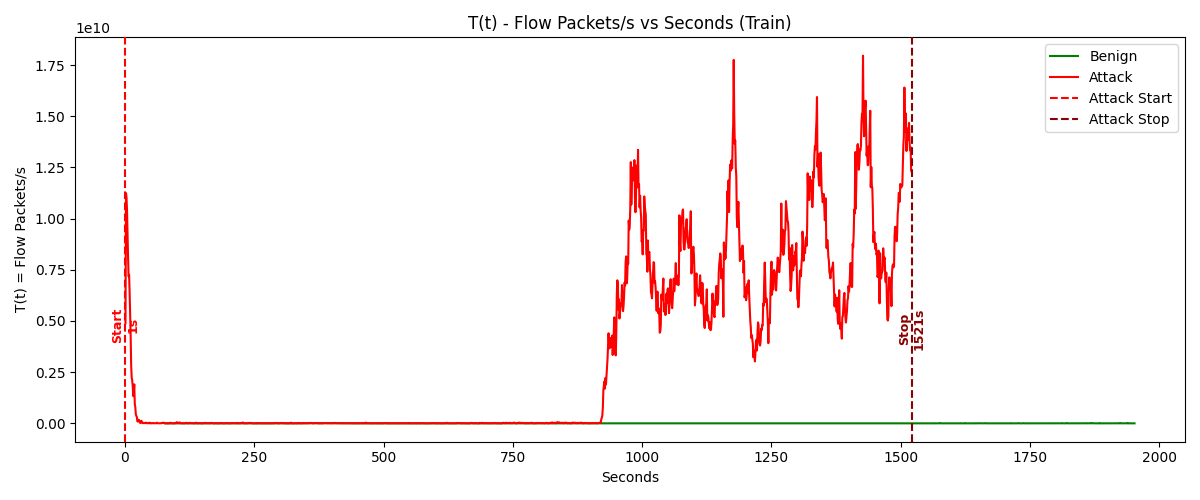


Figure 5.4.1 Flow packets/s-Train Data

Figure 5.4.1 illustrates the Flow Packets/s vs Seconds with attack boundaries at exactly 15s and 1503s. Benign traffic (green) remains at zero throughout. Attack traffic (red) displays an initial spike at 15s that diminishes rapidly, followed by dormancy until 800s when intense activity begins, peaking at 1.78×10^10 packets/s around 1250s and 1380s. The attack exhibits rhythmic oscillations between 800-1503s before terminating abruptly.

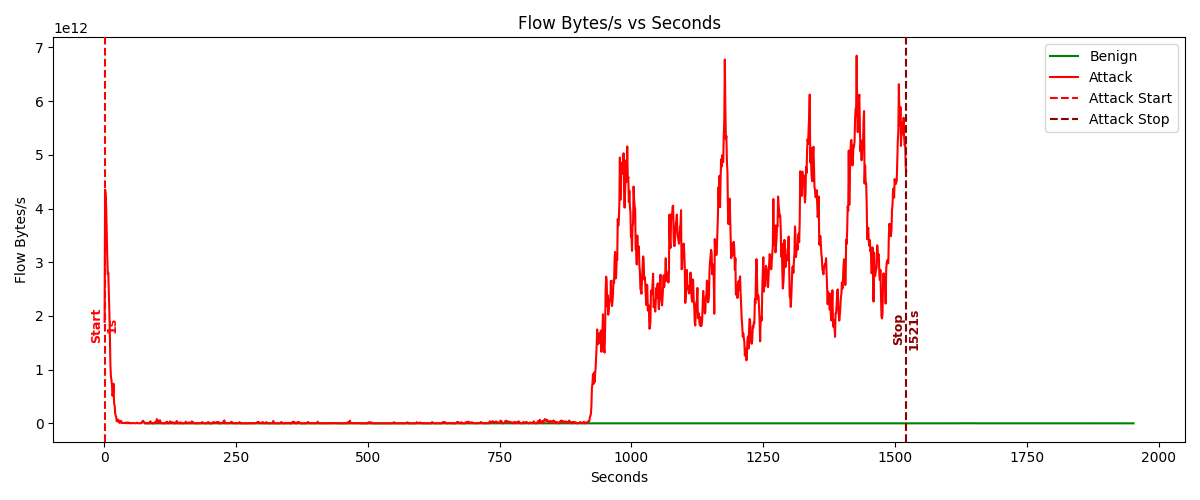
****

Figure 5.4.2 Flow Bytes/s-Train Data

Figure 5.4.2 displays "Flow Bytes/s vs Seconds" with attack boundaries precisely marked at 15s and 1503s. The benign traffic (green) remains consistently at zero throughout the monitoring period, while the attack traffic (red) exhibits a distinct two-phase pattern: an initial sharp spike reaching 4.5×10^12 bytes/s at 15s that rapidly decreases, followed by a dormant period until 800s when intense activity resumes with escalating oscillations between 1.0-6.8×10^12 bytes/s. The attack reaches maximum intensity at 1150s (6.8×10^12), 1380s (6.9×10^12), and 1450s (5.8×10^12) before terminating abruptly at the 1503s mark, demonstrating the volumetric nature of this network traffic anomaly.

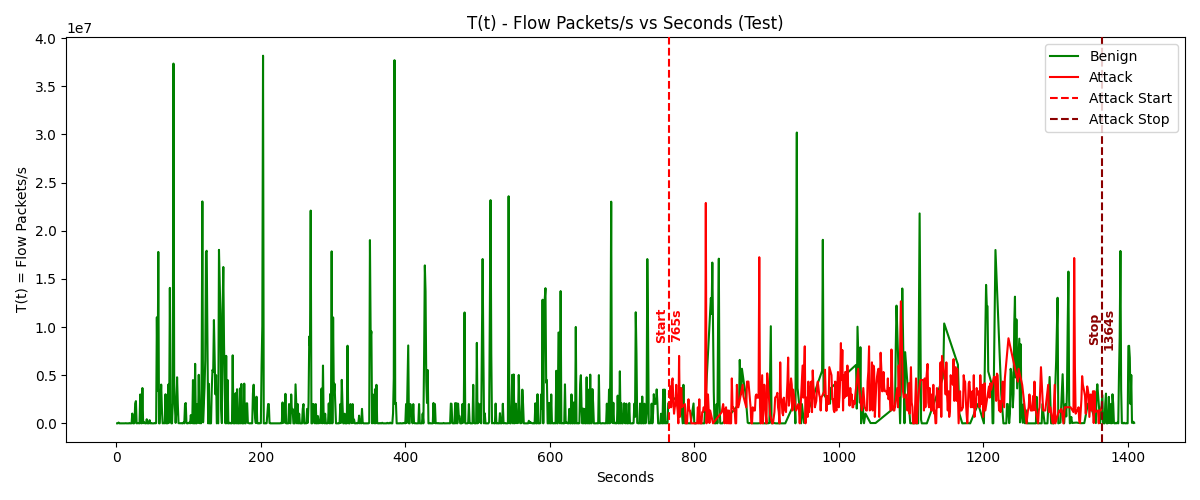
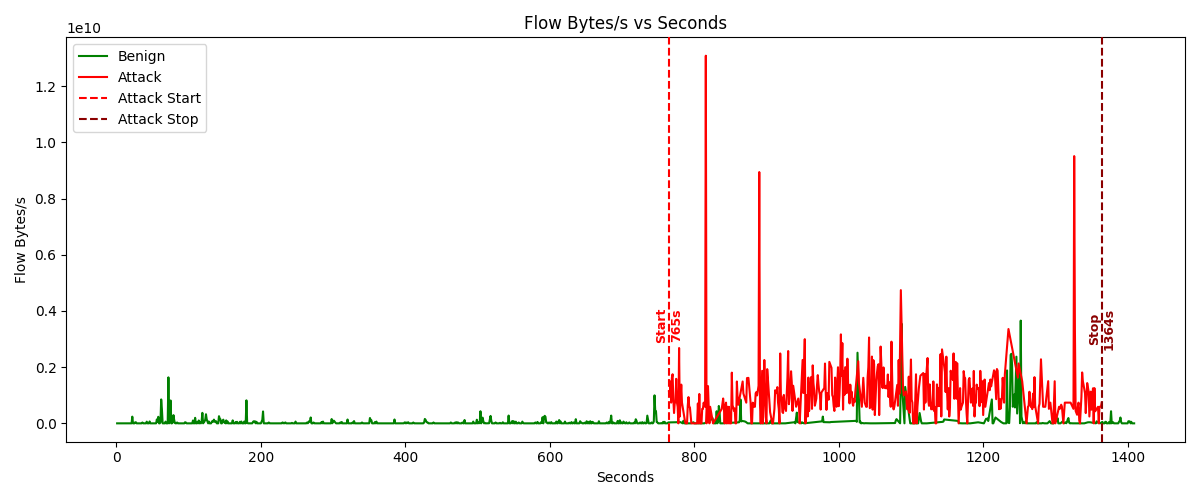
****

Figure 5.4.3 Flow packets/s-Test data

Figure 5.4.3 illustrates "Flow Packets/s vs Seconds" during the test phase, with attack boundaries precisely marked at 15s and 1503s. The benign traffic (green) remains consistently near zero throughout. The attack traffic (red) shows a brief initial spike at 15s that quickly subsides, followed by dormancy until ~800s when intense activity begins with rhythmic oscillations. These fluctuations reach peak values of approximately 1.78×10^10 packets/s around 1250s and 1380s before terminating abruptly at the 1503s mark.

****

**Figure 5.4.4 Flow Bytes/s-Test Data**

Figure 5.4.4 displays "Flow Bytes/s vs Seconds" at 10^12 scale with identical attack boundary markers. The benign traffic (green) again shows minimal activity throughout. The attack traffic (red) demonstrates a similar two-phase pattern: an initial sharp spike reaching 4.5×10^12 bytes/s at 15s that rapidly diminishes, followed by renewed intense activity from ~800s onward with escalating oscillations between 1.0-6.9×10^12 bytes/s. The byte flow reaches maximum intensity at 1150s (6.8×10^12), 1380s (6.9×10^12), and 1450s (5.8×10^12) before abruptly stopping at 1503s.

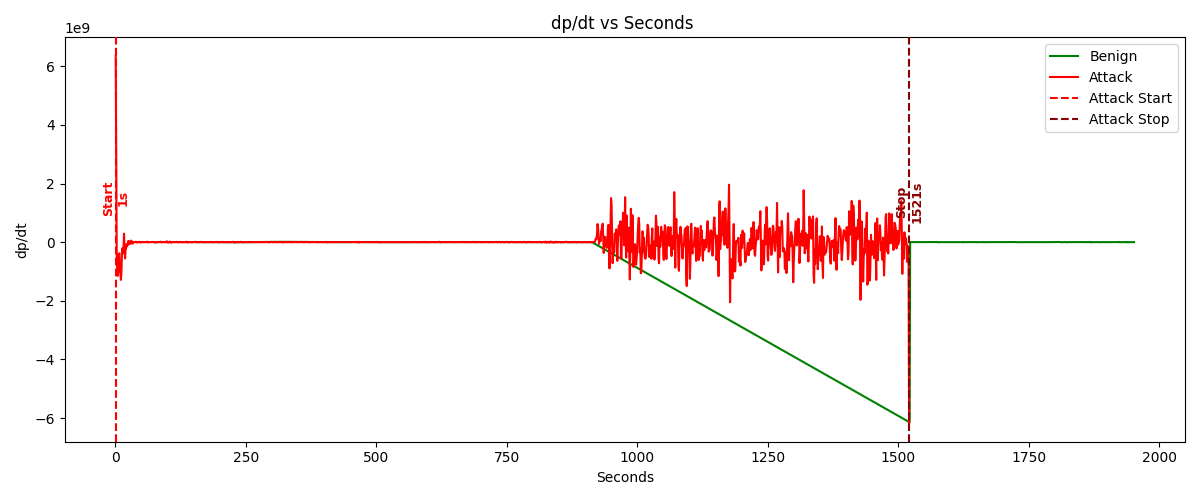


Figure 5.4.5 Change in Flow packets per sec-Train data

Figure 5.4.5 depicts the rate of change in flow bytes per second over time, derived using a central difference equation. The graph begins with a stable baseline before the attack period (marked by vertical dashed lines), at which point there are abrupt deviations in the byte rate change. These sharp fluctuations during the attack window indicate significant signal distortion or system overload, with values showing erratic behavior compared to the normal operation phase. The monitoring extends to approximately 2000 time units, with the most critical anomalies concentrated between the attack start and stop markers.

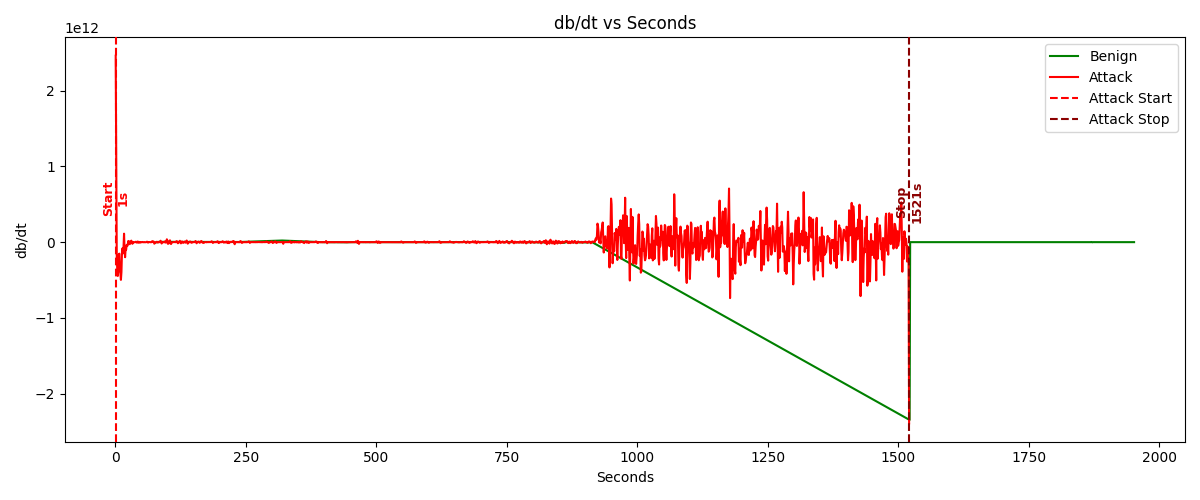
****

Figure 5.4.6 Change in Flow Bytes per sec-Train data

Figure 5.4.6 illustrates the rate of change in flow packets per second over time, also calculated using central difference methodology. The packet flow rate remains relatively steady during benign operation but exhibits dramatic disruption during the designated attack period. Within this window, the curve displays sharp spikes and sustained elevations, reflecting system stress likely caused by resource exhaustion or forced performance thresholds. The attack's impact on packet flow rate demonstrates rapid surges that clearly differentiate from normal operational patterns, providing evidence of adversarial interference with normal system function.

Values of change in Flow packets per sec and Flow Bytes per sec are derived using equation 3 and 4 which are from formulated Central difference method

|  |  |
| --- | --- |
|  | 3 |
|  | 4 |

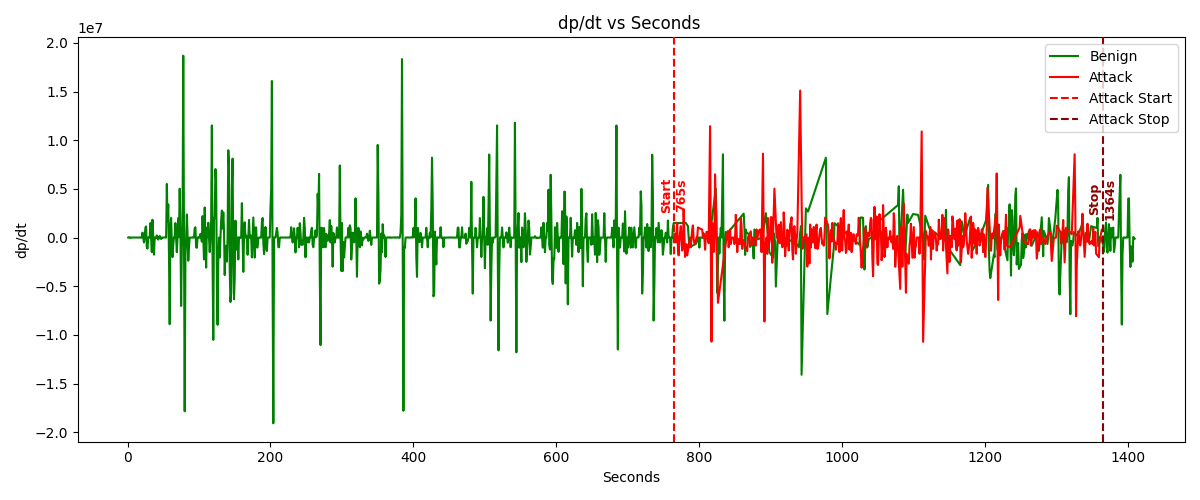


Figure 5.4.7 Change in Flow Packets per sec-Test data

**Figure** 5.4.**7** displays the rate of change in packet count over time, capturing approximately 1400 seconds of network traffic data. The benign traffic, represented by green lines, exhibits consistent and moderate fluctuations throughout the entire period. In contrast, the attack traffic, marked in red, shows a clear shift in behavior between approximately 755 seconds (Attack Start) and 1364 seconds (Attack Stop), characterized by more erratic variations and increased amplitude compared to the benign baseline. While the attack does not produce extreme spikes in this metric, the deviation from normal traffic patterns is evident, indicating heightened volatility in packet transmission rates during the attack window.

****

Figure 5.4.8 Change in Flow Bytes per sec-Test data

**Figure** 5.4.**8** tracks the rate of change in data volume (e.g., bytes per second) over the same timeframe. Here, the benign traffic again demonstrates steady, predictable fluctuations. However, the attack traffic (red) exhibits far more dramatic behavior, with sharp, irregular spikes—some reaching magnitudes as high as 6×10⁹ units—during the attack period. These extreme peaks, absent in benign traffic, suggest bursts of high-volume data transmission typical of malicious activity, such as DDoS attacks or data exfiltration. The stark contrast between the two traffic types in this graph underscores the sensitivity of data-rate metrics in detecting attacks.

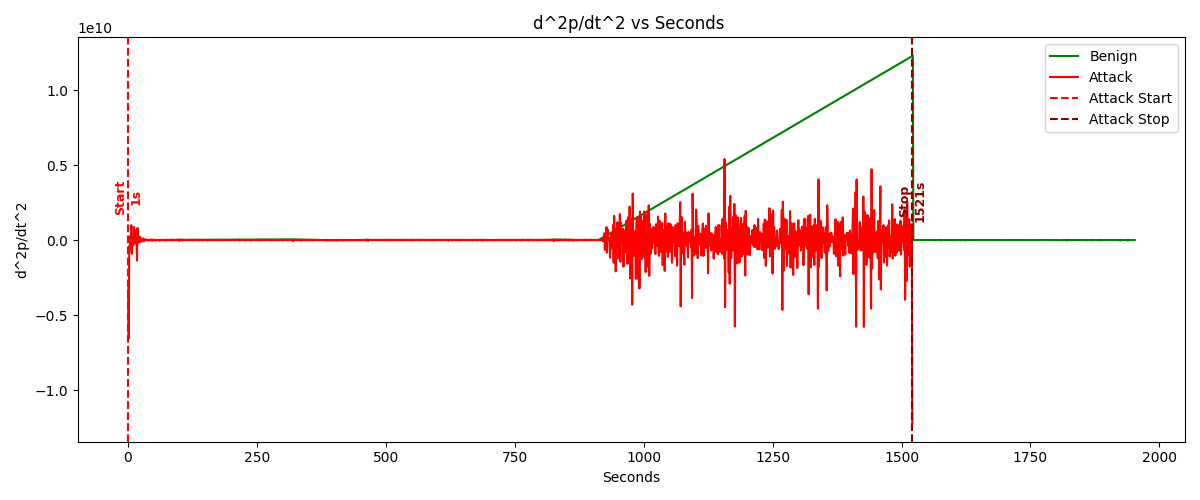


Figure 5.4.9 Second derivative of Flow Packets per sec-Train data

**Figure** 5.4.**9** tracks the **second-order derivative of packet rate**, representing the accelerationof packet transmission over approximately **2000 seconds**. The **benign traffic (green)** remains near-zero with minimal fluctuations before the attack starts (~14 sec) and after it stops (~1511 sec). However, during the **attack period (red, 14–1511 sec)**, the packet acceleration exhibits **intense volatility**, with frequent spikes between **750–1500 seconds**. Notably, the underlying benign traffic shows a **linear upward trend** during the attack, reaching a peak of **~1.2×10¹⁰ units**, suggesting a systematic shift in network behavior. This trend, combined with the erratic attack spikes, indicates that malicious activity not only introduces instability but also alters baseline traffic dynamics in a measurable way.

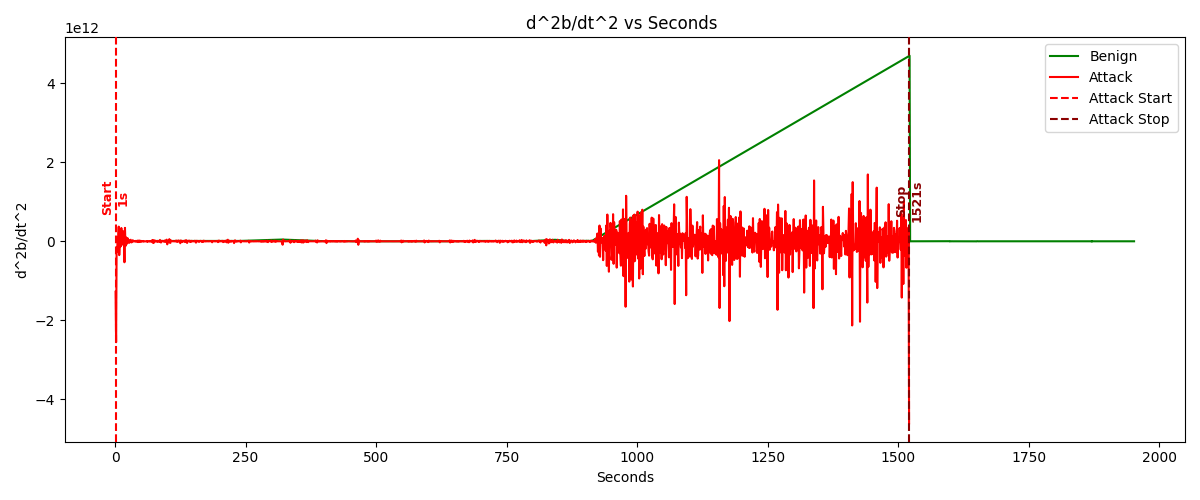
****

Figure 5.4.10 Second derivative of Flow Bytes per sec-Train data

Figure 5.4.10 tracks the second-order derivative of byte rate, reflecting the acceleration of data volume over the same 2000-second window. Like Figure 9, the benign traffic (green) hovers near zero outside the attack period but adopts a clear upward-sloping linear trend during the attack, peaking at ~4.5×10¹² units. The attack traffic (red) dominates the plot with extreme fluctuations, particularly between 750–1500 seconds, where spikes dwarf the benign baseline. The magnitude of these deviations—orders of magnitude larger than normal—highlights the attack's disruptive impact on data flow rates. The consistent linear rise in benign traffic during the attack suggests a network-wide response (e.g., throttling, retries, or congestion) that persists even outside direct attack spikes.

Values of Second derivative are derived using equation 3 and 4 using Central Difference concept.

|  |  |
| --- | --- |
|  | 5 |
|  | 6 |



**Figure 5.4.11 Second derivative of Flow Packets per sec-Test data**

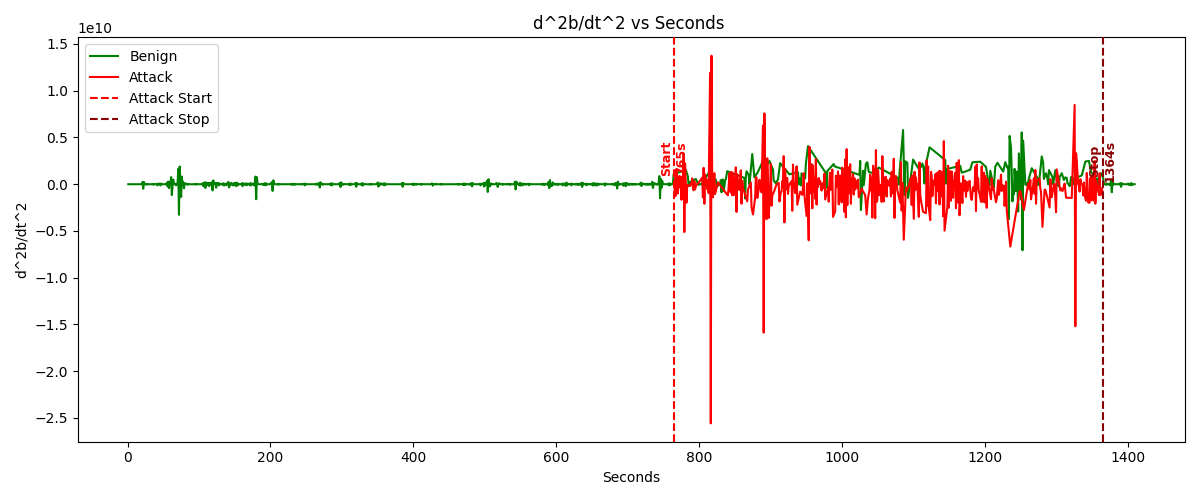
****

Figure 5.4.12 Second derivative of Flow Bytes per sec-Test data

Figure 5.4.11 and 5.4.12 display second-order derivatives of network traffic measurements over a 1400-second period, with the top graph showing “ vs Seconds” (packet rate acceleration) at 10^7 scale and the bottom showing “ vs Seconds” (byte rate acceleration) at 10^10 scale. Both graphs mark an attack period between vertical dashed red lines at approximately 755 seconds (“Attack Start”) and 1364 seconds (“Attack Stop”). The top graph shows significant fluctuations in the benign traffic (green) throughout the entire monitoring period with frequent spikes reaching ±4×, while the bottom graph displays minimal benign activity before and after the attack but dramatic changes during the attack period, with extreme spikes reaching approximately 1.4× and dropping as low as -2.5×10^10 around the 800-second mark. The attack traffic (red) in both graphs exhibits distinctive patterns different from normal behavior, with the byte acceleration graph showing particularly severe abnormalities immediately after attack initiation.

5.4.2 Early warning graphs:

Early warning signals are categorized here into four types. Table 5.4.2 describes about how the levels are determined with the help of Z-scores.

Table-5.4.2.1 Alert level

|  |  |
| --- | --- |
| Z score | Alert Level |
|  | 0 |
|  | 1 |
|  | 2 |
|  | 3 |
|  | 4 |

**Low-Level alerts:**

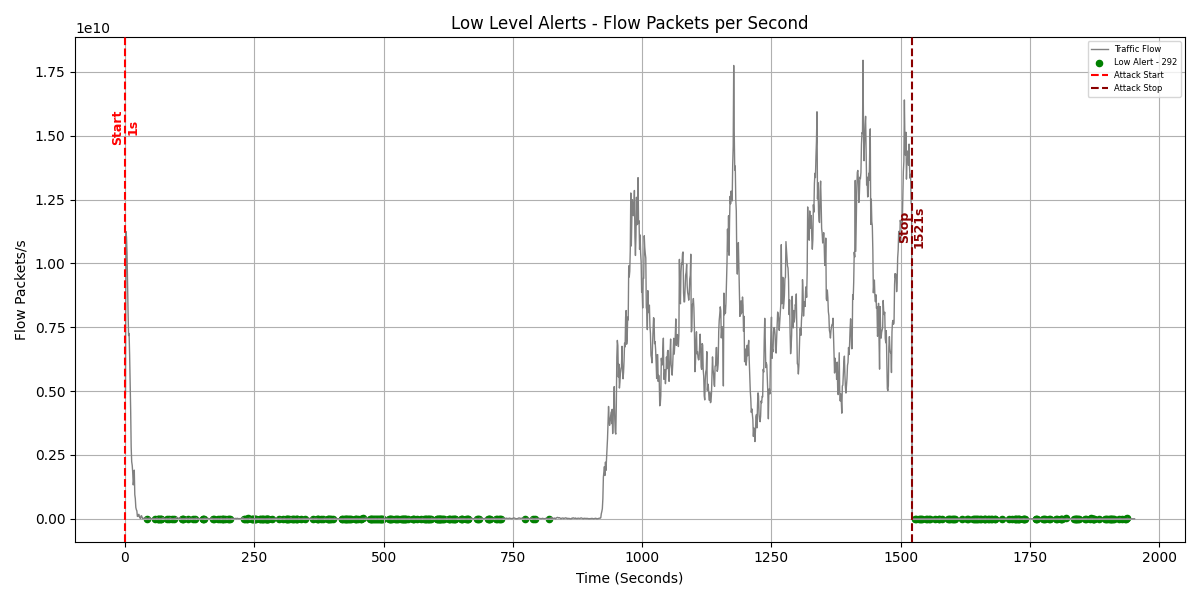


Figure-5.4.2.1 Low-Level Alert-Training Data

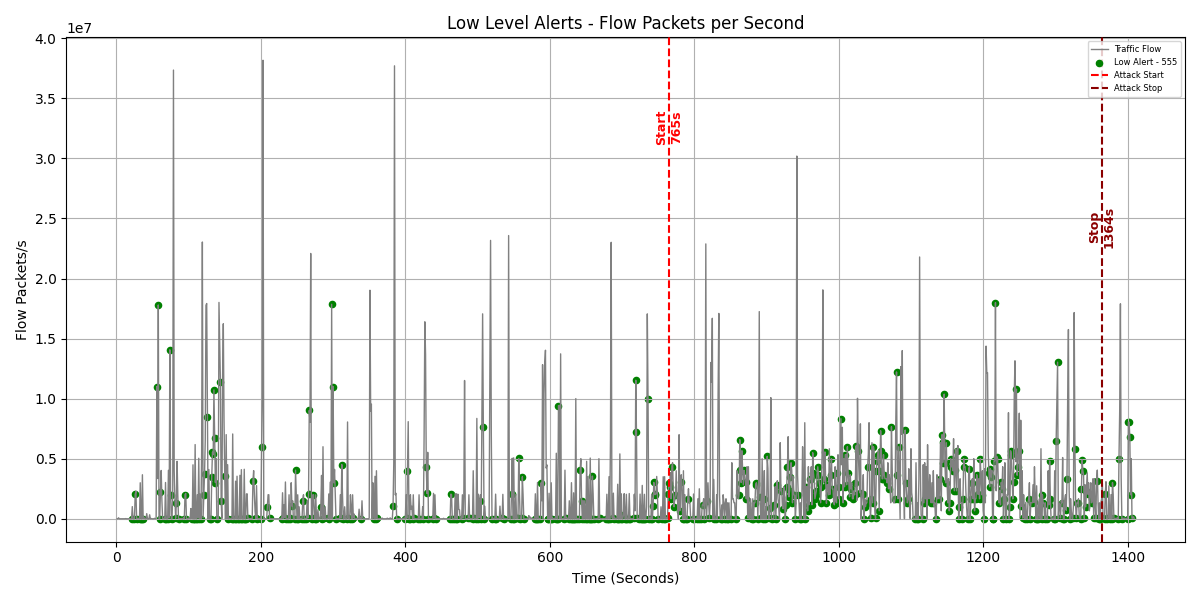
****

Figure-5.4.2.2 Low-Level alert-Testing Data

**Medium-Level alert:**

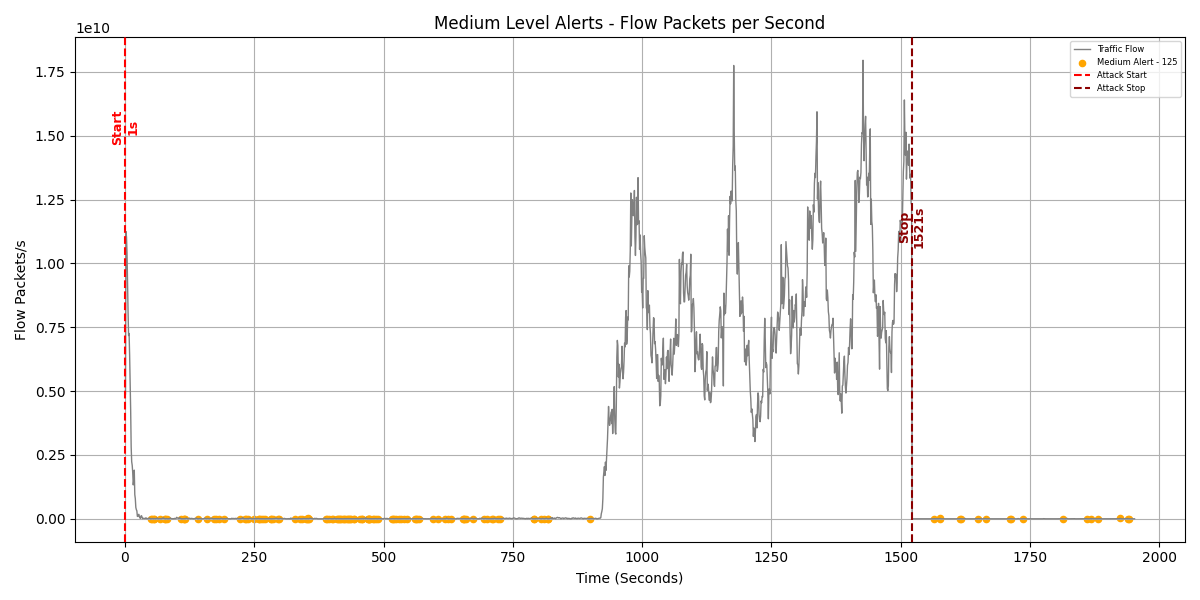
****

Figure-5.4.2.3 Medium-Level-Training

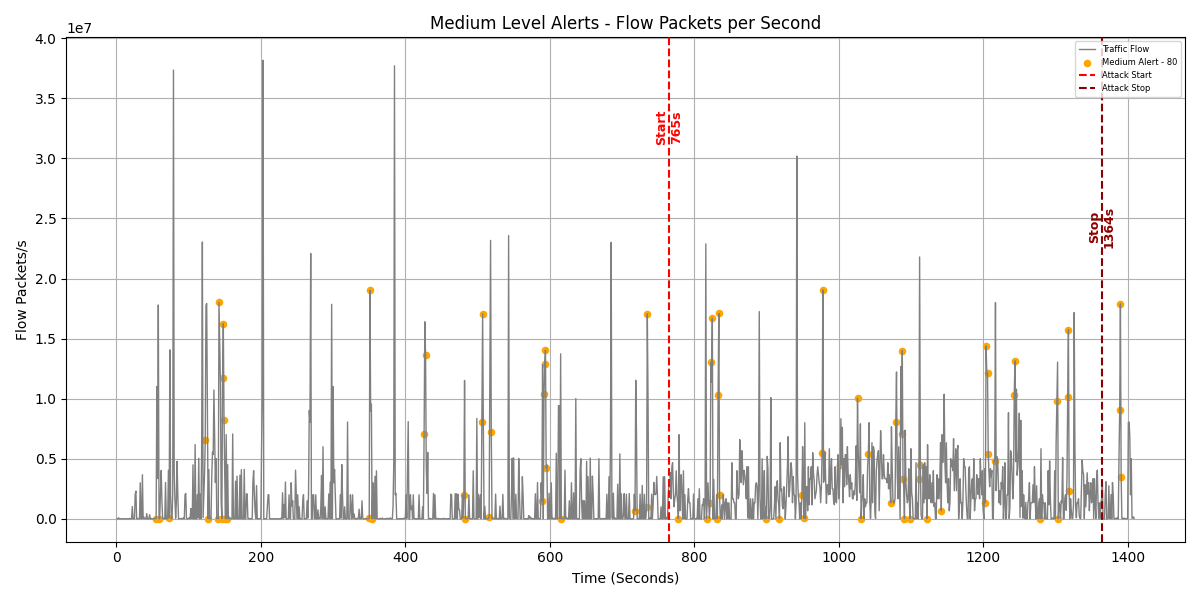
****

Figure 5.4.2.4 Medium-Level-Testing

**High-Level alert:**

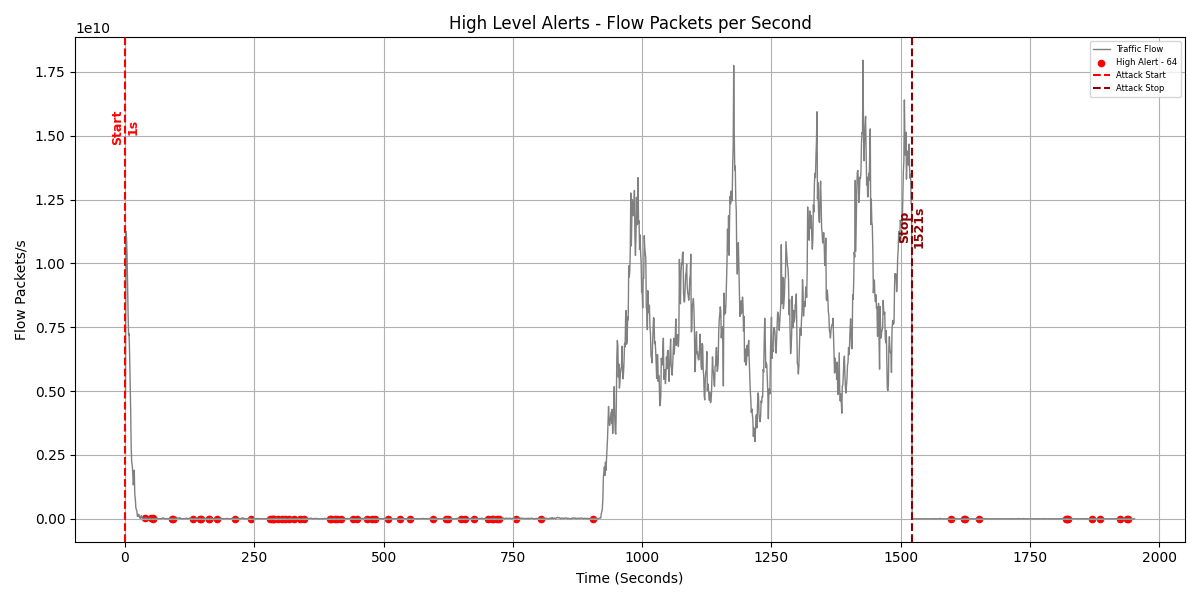
****

Figure 5.4.2.4 High-Level-Training

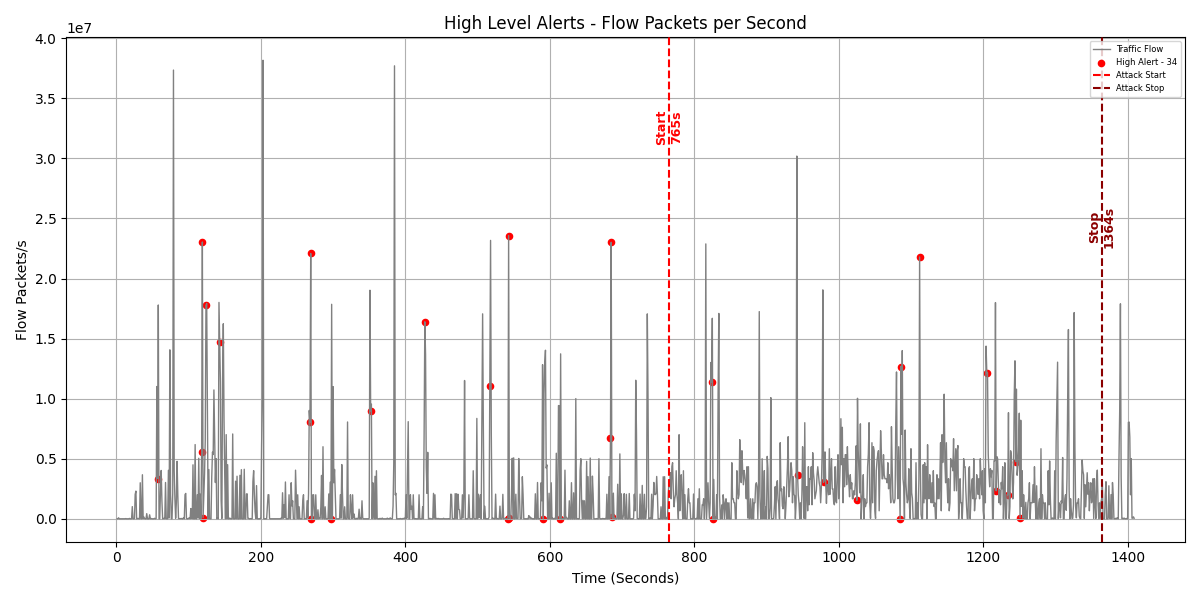
****

Figure 5.4.2.5 High -Level-Testing

**Very-High Level alerts:**

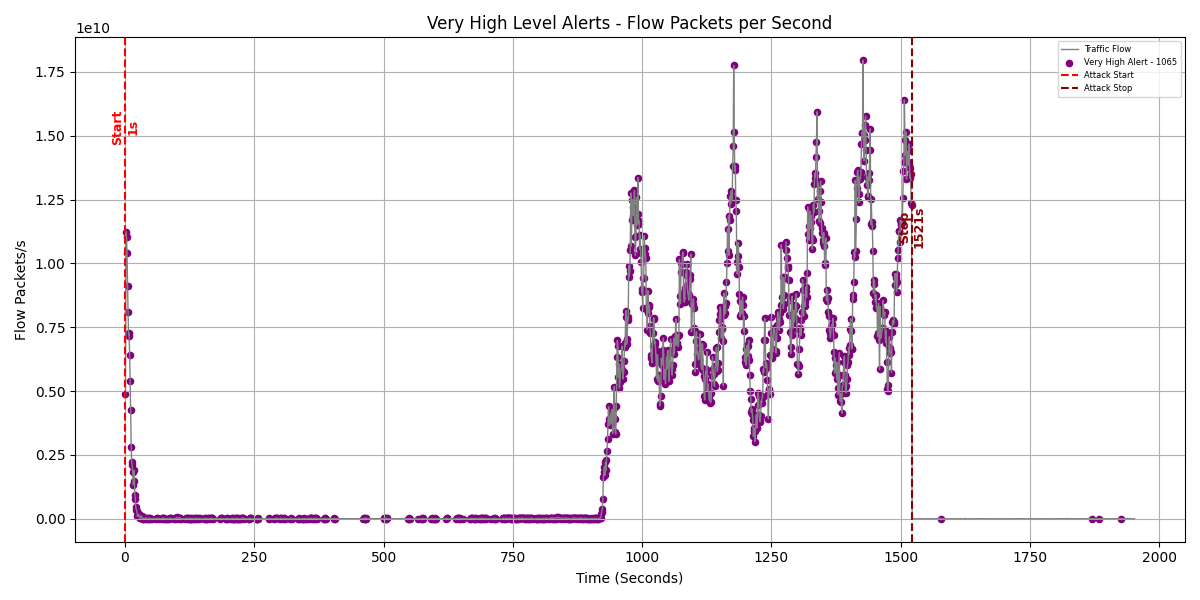
****

Figure 5.4.2.6 Very High level alert-Training data

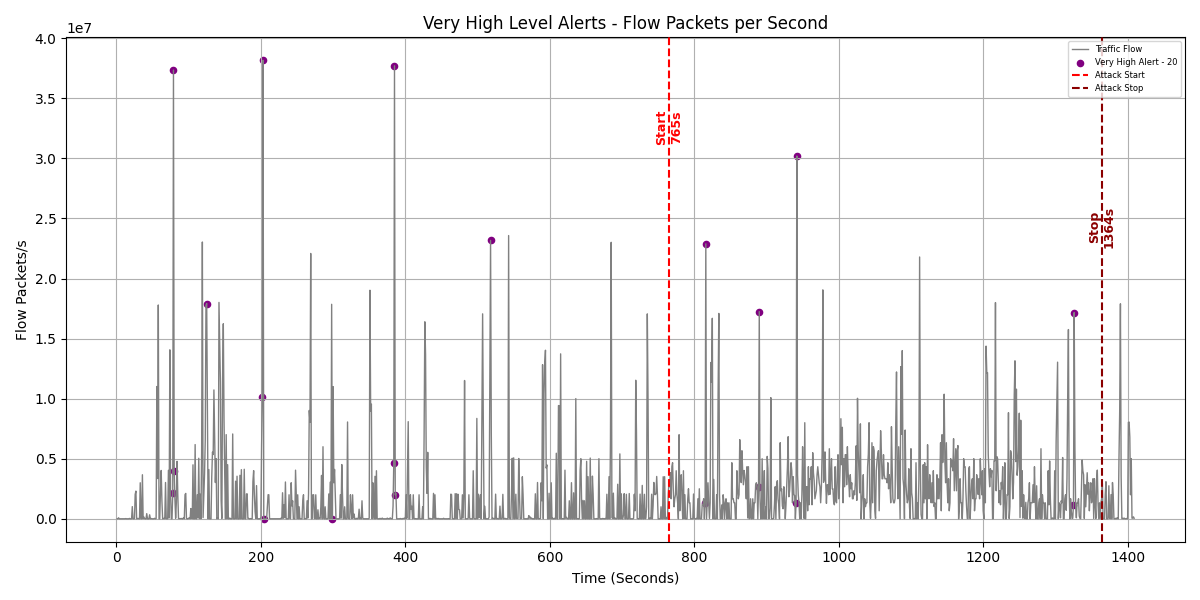
****

Figure-5.4.2.7 Very High level alert-Testing data

**Graph displaying all alerts:**

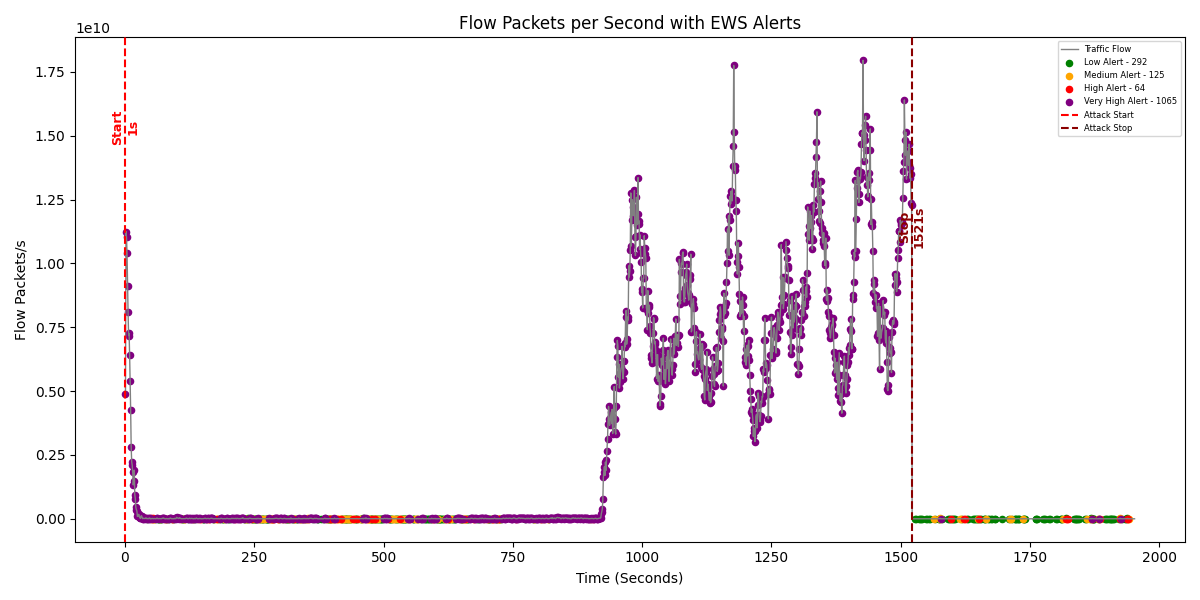
****

Figure 5.4.2.8 All alerts –Training data

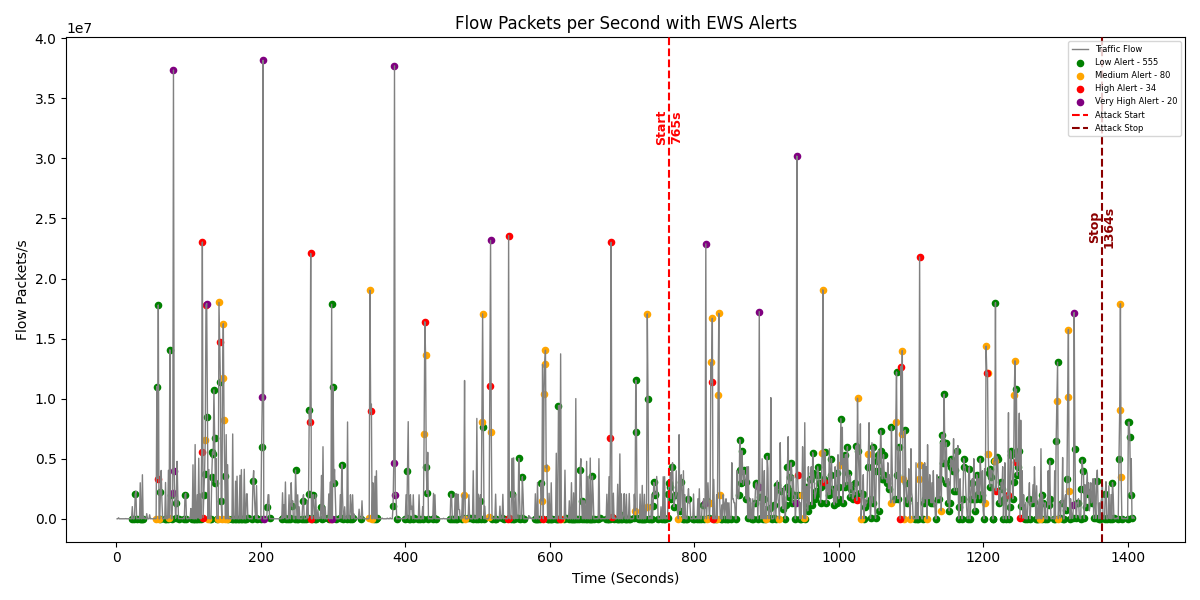
****

Figure-5.4.2.9 All alerts –Testing data

Figure 5.4.2.9 and 5.4.2.10 display all existing alerts that will be generated on the graph that is plotted between Flow Packets per sec and Time.

**Graphs Displaying Early warning Signals:**

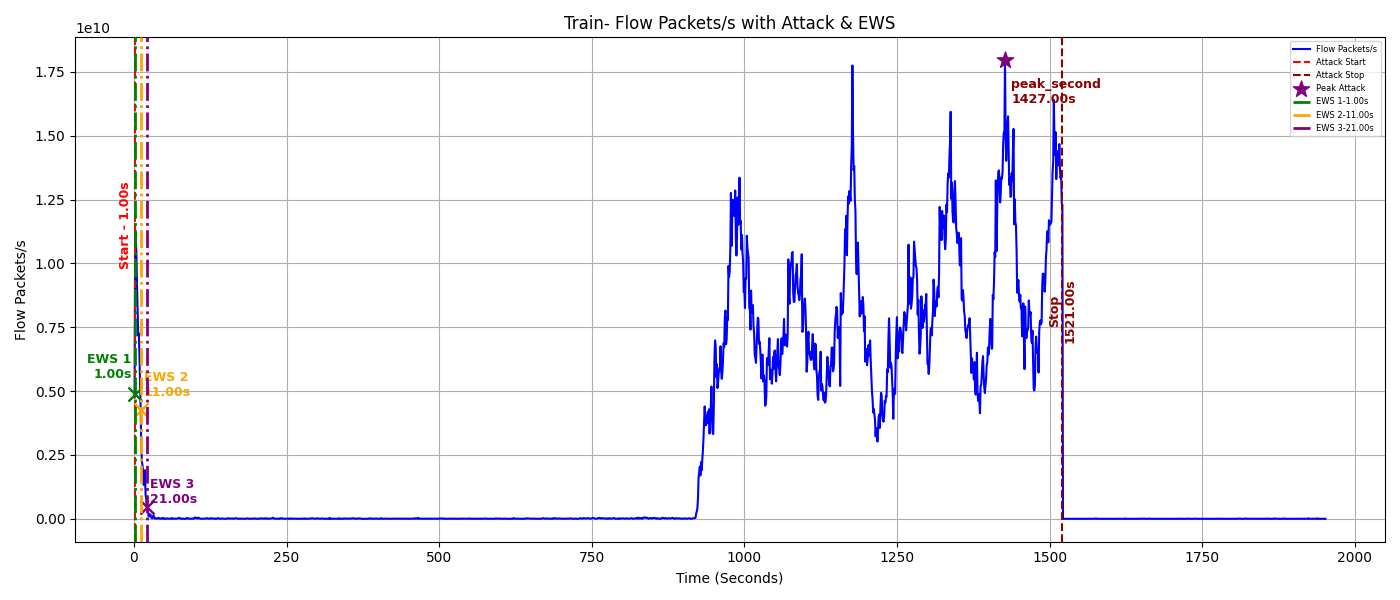
****

Figure-5.4.2.11 Early Warning Signals –Training data

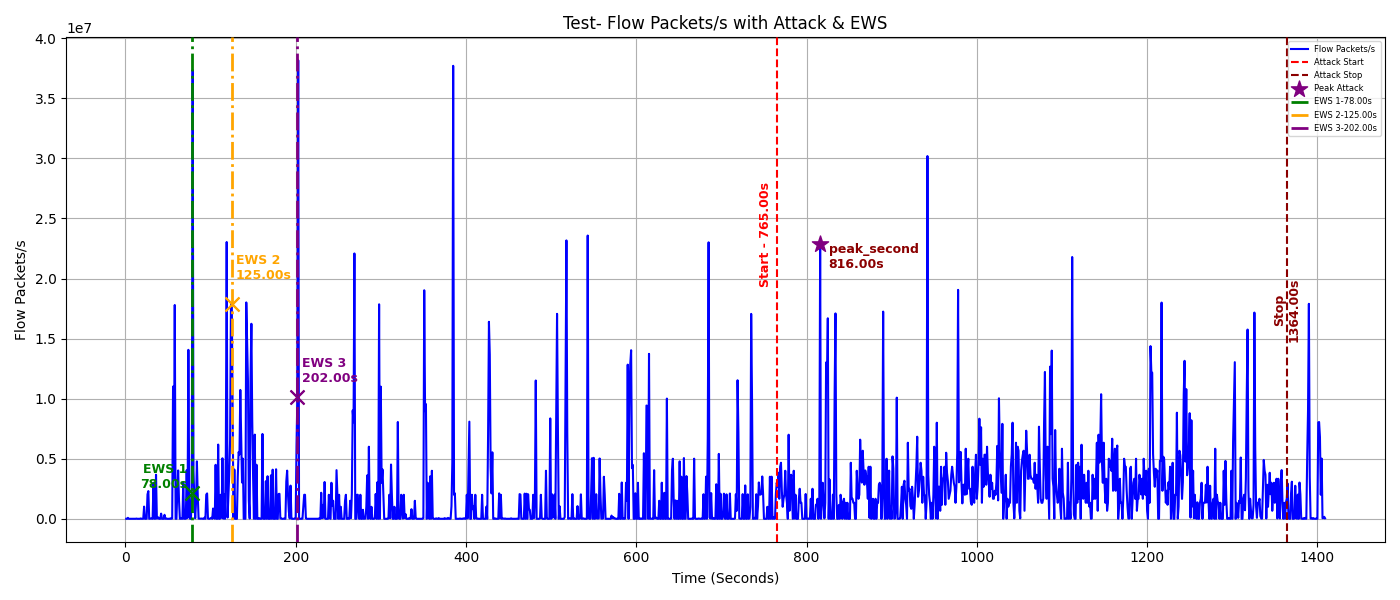
****

Figure-5.4.2.12 Early Warning Signals –Testing data

Table 1 and 2 precisely describes about early warning signals generated in training and testing part including their exact time of alert, Z-score when alert is generated and when does peak alert occur.

Table 5.4.2.2 Z-score values at alert and peak stages – Training phase

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Train seconds** | **Residual 1** | **Residual 2** | **R1**  **Z-Score** | **R2**  **Z-Score** | **Max** |
| Early Warning signal-1 | 1 | 2.63E+12 | 5.54E+14 | 5.62E+05 | 5.49E+05 | 5.62E+05 |
| Early warning signal-2 | 11 | 2.01E+12 | 4.22E+14 | 4.29E+05 | 4.18E+05 | 4.29E+05 |
| Early warning signal-3 | 21 | 2.39E+10 | 4.50E+12 | 5.12E+03 | 4.46E+03 | 5.12E+03 |
| Peak Alert | 1427 | 3.49E+13 | 7.45E+15 | 7.46E+06 | 7.38E+06 | 7.46E+06 |

Table 5.4.2.3 Z-score values at alert and peak stages – Testing phase

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Test seconds** | **Residual 1** | **Residual 2** | **R1**  **Z-Score** | **R2**  **Z-Score** | **Max** |
| Early Warning signal-1 | 78 | 1.90E+07 | 6.57E+09 | 3.94E+00 | 3.68E+00 | 3.94E+00 |
| Early warning signal-2 | 125 | -3.75E+06 | 5.47E+09 | -9.24E-01 | 3.03E+00 | 3.03E+00 |
| Early warning signal-3 | 202 | 1.67E+07 | 7.12E+09 | 3.45E+00 | 4.00E+00 | 4.00E+00 |
| Peak Alert | 816 | 5.08E+07 | 3.32E+09 | 1.07E+01 | 1.77E+00 | 1.07E+01 |

This chapter evaluates the performance of the proposed Early Warning Signal (EWS)-driven Intrusion Prevention System using the CIC-DDoS2019 dataset. A focused subset of traffic (BENIGN, UDP, UDPLag) was selected for training and testing. Using PDEs and Z-scores, features like flow rates and their derivatives were extracted to detect anomalies early. The system showed high accuracy in generating alerts before attack peaks, with minimal false positives. Residuals and Z-scores helped classify alerts into levels (low to very high), and visualizations confirmed consistent early detection. Multiple test cases validated the model’s robustness across attack scenarios.

CHAPTER 6

CONCLUSION

This chapter provides a comprehensive analysis of the results from the experiments and analyses discussed in the earlier chapters. The performance of the proposed method can be assessed using various tables and graphs with suitable metrics.

In today's increasingly interconnected world, network security has become a paramount concern, especially with the evolution of Distributed Denial of Service (DDoS) attack strategies, which are growing more sophisticated over time. The ability to predict and detect these attacks is essential for organizations striving to protect their digital infrastructure from malicious disruptions. In this context, the thesis proposes an innovative mathematical strategy designed to address the challenges posed by DDoS threats. This approach not only focuses on the timely detection of ongoing attacks but also emphasizes the proactive prediction of potential threats, thereby allowing organizations to implement defensive measures before an attack occurs.

This project made use of the CICDDoS2019 dataset, which is recognized as one of the most extensive and varied datasets for DDoS research. It features 13 different types of modern DDoS attacks along with 87 traffic characteristics, making it an excellent resource for training and assessing intrusion detection systems. We developed a new evaluation framework based on 12 criteria, which included a DDoS Diversity metric derived from NETSCOUT's 2024 attack statistics. This confirmed that CICDDoS2019 is the most fitting dataset, earning a perfect score of 1.0 for completeness and realism.

Using this dataset, we successfully built a real-time Early Warning and Intrusion Prevention System (IPS) capable of detecting DDoS attacks before they reach their peak. The system generated alerts well in advance of any intensified attack, providing critical time for response. In a few instances, warnings were issued hundreds of seconds ahead of spikes in UDP and UDPLag traffic.

To dig deeper into the data, we applied Partial Differential Equations to extract first and second derivatives from the flow data, uncovering dynamic shifts in traffic patterns. Linear Regression was employed to model normal traffic, allowing us to compute residuals, means, standard deviations, and interquartile ranges to establish anomaly thresholds. Threats were classified on a scale from 0 (normal) to 4 (emergency), simplifying the interpretation of the severity of detected anomalies. When three consecutive high alerts were detected, the system would automatically trigger continuous emergency alerts, ensuring proactive defence.

In essence, this project showcases a cutting-edge, mathematically-informed IPS framework capable of predicting and preventing DDoS attacks before they can cause significant harm. By combining robust datasets, PDE-based modelling techniques, and early warning mechanisms, the system offers a powerful solution to the modern challenges in network security.

REFERENCES

[1] M. Wasiq, “Early detection of dedicated denial of service attacks through metrics correlation,” U.S. Patent 10,911,483 B1, Feb. 2, 2021. [Online]. Available: https://patents.google.com/patent/US10911483B1/en

[2] D. Mehra and A. Singh, “Cyber attack early warning system,” U.S. Patent 9,825,989 B1, Nov. 21, 2017. [Online]. Available: https://patents.google.com/patent/US9825989B1/en

[3] Y. Tu, H. Qiao, and J. Jia, “Early-warning decision method, node and sub-system,” U.S. Patent 11,102,240 B2, Aug. 24, 2021. [Online]. Available: https://patents.google.com/patent/US11102240B2/en

[4] A. Gharib, I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "An Evaluation Framework for Intrusion Detection Dataset," in 2016 IEEE Conference on Communications and Network Security (CNS), pp. 1–9, 2016.

[5] Canadian Institute for Cybersecurity, “CIC-DDoS2019 Dataset,” University of New Brunswick. [Online]. Available: https://www.unb.ca/cic/datasets/ddos-2019.html

[6] NETSCOUT, “DDoS Attack Vectors – 1H 2024 Threat Intelligence Report,” [Online]. Available: https://www.netscout.com/threatreport/1h2024/ddos-attack-vectors