Statistical analysis of DDoS attacks

A B. Tech Project Resport Submitted
in Partial Fulfillment of the Requirements
for the Degree of

Bachelor of Technology

by

Vatsal Gupta, Pranshu Kandoi
(200101105,200101086)

under the guidance of

Pinaki Mitra

to the

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY GUWAHATI
GUWAHATI - 781039, ASSAM

CERTIFICATE

*This is to certify that the work contained in this thesis entitled "**Statistical analysis of DDoS attacks**" is a bonafide work of **Vatsal Gupta, Pranshu Kandoi** (**Roll No. 200101105,200101086**), carried out in the Department of Computer Science and Engineering, Indian Institute of Technology Guwahati under my supervision and that it has not been submitted elsewhere for a degree.*

Supervisor: **Pinaki Mitra**

Associate Professor,

May, 2024

Department of Computer Science & Engineering,Guwahati.

Indian Institute of Technology Guwahati, Assam.

Acknowledgements

We want to extend our appreciation to *Prof. Dr. Pinaki Mitra* our supervisor, for his outstanding guidance during the semester when we worked on this project. Our gratitude also goes to *Prof. Sukumar Nandi*, Professor in the Department of Computer Science and Engineering, for his valuable suggestions for the project. Last but not least, we express our thanks to our friends and peers who may have offered guidance or provided innovative suggestions during the spirited discussions we engaged in.

Contents

List of Figures

List of Tables

Abstract

Distributed Denial-of-Service (DDoS) attacks pose a significant threat to the stability and security of Internet-of-Things (IoT) systems. This paper proposes a novel approach for DDoS attack detection by leveraging statistical techniques commonly used in financial analysis. The methodology involves resampling a captured network dataset based on a chosen time window. We then apply these techniques to the windowed data to identify anomalies that deviate from normal traffic patterns. This includes exploring the optimal window size for maximizing attack detection accuracy while maintaining computational efficiency. Our findings suggest that statistical analysis of packet arrival rates, informed by insights from financial analysis, offers a promising and efficient technique for DDoS attack detection in IoT systems.

Chapter 1

Introduction

The rapidly growing Internet-of-Things (IoT) has revolutionised how we interact with the outside world, but it also brings with it security risks, most notably Distributed Denial-of-Service (DDoS) attacks, which seriously threat the stability and security of IoT systems.

Conventional DDoS detection methods often rely on signature-based approaches, which struggle to identify emerging attack variants. In response, this study looks , through a new lens, into IoT systems via the Quant systems and advocates for a novel detection strategy tailored for IoT systems, drawing inspiration from statistical techniques commonly utilized in financial and quantitative analysis.

Our methodology involves resampling network datasets within defined time windows, treating network traffic patterns akin to financial time series- hence constructing an analogue for stock prices by considering total number of packets forwaded in a given time window. By applying statistical methods typical in financial analysis to this data, we aim to discern deviations from normal traffic patterns, potentially signaling a DDoS attack.

Emphasis is placed on exploring optimal window sizes to maximize detection accuracy while upholding computational efficiency. Additionally, the adaptability and effectiveness of this approach are showcased through its application beyond DDoS detection, including anomaly detection in various domains.

In light of the need for rapid response in online systems, the study underscores the agility

of statistical techniques compared to traditional machine learning approaches by comparing and contrasting the same. Furthermore, it delves into the exploration of incremental calculations by adapting existing algorithms, ensuring real-time responsiveness in detecting DDoS attacks.

Moreover, we present several proposed detection functions designed to swiftly recognize the onset of a DDoS attack, leveraging statistical parameters' responses. These methods are readily implementable for monitoring live IP traffic, offering early recognition and mitigation capabilities.

1.1 Required definitions

1.1.1 Common Abbreviations

Term	**Definition**
ML/DL	Machine Learning/Deep Learning
IDS	Intrustion Detection Systems
DDoS	Distributed Denial of Service (Attacks)
IoT	Internet of Things
indic.	Technical indicators and Statistical Metrics
IP FLow	Internet Protocol/ Packet Flow
OHLC	Open-High-Low-Close

Table 1.1: Common Abbreviations

Note: Other abbreviations for statistical metrics as well as indicators are mentioned at appropriate places where they are first defined and explained.

1.1.2 Moving averages

Moving averages (MAs) are a widely used method in series analysis (time and finance both) for exploring patterns. These are used in quant finance analysis and can hence be extended to DDoS attacks for IoT systems to predict feature (price) direction and to ascertain if the price (of a security) a certain feature is rising or falling. There are three main types of moving averages:

- **Simple Moving Average (SMA)**

SMA is computed by taking the average of the datapoints within a window. The formula is:

$$SMA(t) = \frac{X(t-n+1) + X(t-n+2) + \ldots + X(t)}{n}$$

where $X(t)$ is the value at time t, n is the number of periods in the average (window size), and t represents the current time point.

- **Weighted Moving Average (WMA)**

  The WMA assigns different weights to data points within the window, with more recent values typically receiving higher weights. This allows the WMA to react more quickly to changes in the data. The formula is:

$$\text{WMA}(t) = \frac{\sum_i (W_i \times X(t-i+1))}{\sum_i W_i}$$

  where W_i is the weight assigned to data point i.

- **Exponential Moving Average (EMA)**

  The EMA gives exponentially decreasing weights to past data points, with the most recent value having the strongest influence. The formula for calculating EMA is:

$$EMA(t) = \alpha \times X(t) + (1-\alpha) \times EMA(t-1)$$

  where α is a smoothing factor ($0 < \alpha \leq 1$) i.e. weight given to recent datapoint ($X(t)$), and $EMA(t-1)$ is the EMA from the previous time step.

1.2 CICDDoS-2019 Dataset

CICDDoS2019 contains benign and the most up-to-date common DDoS attacks, which resembles the true real-world data (PCAPs). It contains multiple features, out of which we sample on the most frequent Destination IP Address and focus on "Total Fwd Packets". The Testbed architecture and types of attacks are as in Fig. 1.3 and Fig. 1.2 resp.

In this study, our focus lies on **UDP attacks, specifically those categorized as exploitation-based attacks**. These attacks involve masking the attacker's identity by utilizing 3rd party components. The method entails directing packets to reflector servers, with the attacker's source IP address appearing as if it were the target victim's IP address. The objective is to overwhelm the victim with response packets.

These attacks can exploit protocols such as TCP and UDP. The dataset includes TCP based SYN-Flood attacks and UDP based UDP/UDP-Lag attacks. In a UDP flood attack, the remote host is inundated with a significant volume of UDP packets. These packets are sent to random ports on the target machine at an extremely high rate, resulting in depletion of network bandwidth, system crashes, and causes performance degradation.

The attack times are given. Day 1 is taken as training files and Day 2 as testing. This is as per convention for other similar work done on this dataset [9],[2],[1].

Days	Attacks	Attack Time
First Day	PortMap	9:43 - 9:51
	NetBIOS	10:00 - 10:09
	LDAP	10:21 - 10:30
	MSSQL	10:33 - 10:42
	UDP	10:53 - 11:03
	UDP-Lag	11:14 - 11:24
	SYN	11:28 - 17:35
Second Day	NTP	10:35 - 10:45
	DNS	10:52 - 11:05
	LDAP	11:22 - 11:32
	MSSQL	11:36 - 11:45
	NetBIOS	11:50 - 12:00
	SNMP	12:12 - 12:23
	SSDP	12:27 - 12:37
	UDP	12:45 - 13:09
	UDP-Lag	13:11 - 13:15
	WebDDoS	13:18 - 13:29
	SYN	13:29 - 13:34
	TFTP	13:35 - 17:15

Fig. 1.1: Attacks upon CICDDoS-2019 dataset.

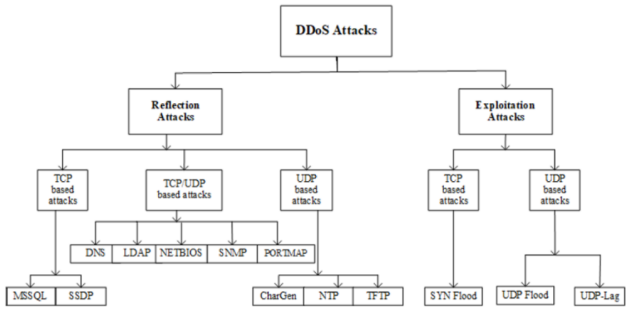


Fig. 1.2: Types of Attacks in CICDDoS2019 dataset.

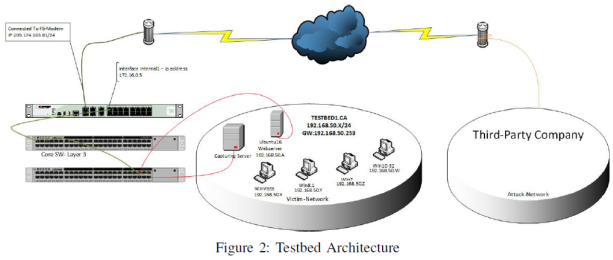


Figure 2: Testbed Architecture

Fig. 1.3: Testbed for CICDDoS-2019 dataset

Chapter 2

Review of Prior Works

DDoS attacks pose a significant threat to the stability and security of networks, which is a growing concern for IoT systems. Traditional signature-based detection methods struggle to identify new attack variants.We analyse existing research on DDoS detection techniques, focusing on approaches that leverage statistical analysis without relying on financial indicators.

Statistical Techniques for DDoS Detection Several studies demonstrate the potential of statistical analysis for DDoS detection. [5] propose using the Hurst exponent, autoregression coefficients, and the coefficient of variation for attack recognition. [3] explore methods based on entropy and frequency-sorted distributions of packet attributes. Similarly, [14] investigate the effectiveness of statistical parameters like coefficients of variation, kurtosis, skewness, and the Hurst exponent. [4] highlight the potential of time-series based analysis using statistical measures like periodicity, kurtosis, skewness, and self-similarity to distinguish DDoS attacks from normal traffic. These studies showcase the ability of statistical analysis to capture inherent characteristics of network traffic and identify deviations indicative of DDoS attacks. Additionally, [7] explore covariance analysis for DDoS detection, particularly for identifying SYN flooding attacks. This approach leverages the relationships between different network traffic attributes to differentiate between normal and attack traffic.

Advantages and Limitations While there are several well established statistical parameters to detect attacks and improve IDS we present the improvements by drawing **Inspiration from Financial Analysis** as our work draws techniques from statistical methods commonly used in financial time series analysis. This novel approach leverages the analogy between network traffic patterns and financial data to identify anomalies potentially indicative of DDoS attacks and by **focusing on IoT Systems** This study specifically tailors the proposed statistical analysis approach for DDoS detection in IoT systems, considering the unique characteristics of IoT network traffic patterns and analysing the CICDDoS-2019 dataset.

Other Detection Techniques Beyond statistical analysis, a range of techniques exist for DDoS detection.[10] propose a method using cluster analysis to exploit attacker behavior in selecting compromised devices for launching attacks. By analyzing network traffic patterns and identifying clusters of compromised devices exhibiting abnormal behavior, this technique can potentially detect DDoS attacks in their early stages.ML algorithms have emerged as a promising avenue for DDoS detection. [10] propose a neural network-based approach, while [18] utilize decision trees. Other studies by [12] and [17] explore various machine learning algorithms, demonstrating their effectiveness in identifying different types of DDoS attacks. Recent advancements in DL have also been applied.[11] and [8] propose methods that utilize Long Short-Term Memory (LSTM) networks, capable of learning long-term dependencies within data sequences. By applying LSTMs to network traffic data, these approaches can potentially identify complex patterns associated with DDoS attacks, even if they deviate from previously known attack signatures.

Chapter 3

Proposed Approach

Fig. 3.1: Methodology Flowchart.

3.1 Methodology

We utilize the CIC-DDoS2019 dataset for our study, aiming to develop mathematical techniques for real-time detection of potential DDoS attacks without relying on pre-learned information from test datasets. This datasets consists of UDP based attack (Fig. 1.2). The attack time for train set and test set are 10:52:00 and 12:45:00 respectively (Table 1.1). We evaluate indicators based on **proximity** (signal detection as close as possible to attack time) and **frequency** (number of signals received after attack time). These strategy and indicators can be readily adapated to other attacks/datasets and their utility is studied in this report.

3.1.1 Notation and pre-processing

Drawing from the work of [5], we construct a time series data for IP flow to model network behavior. We describe network flow using process $A(r, s)$ (arrival process), representing the number of packets in the interval $[r, s]$. $A(t)$ denotes the number of packets until time t, computed by dividing the time series into windows a_i representing the number of packets in the i-th time slot (1 unit slot).

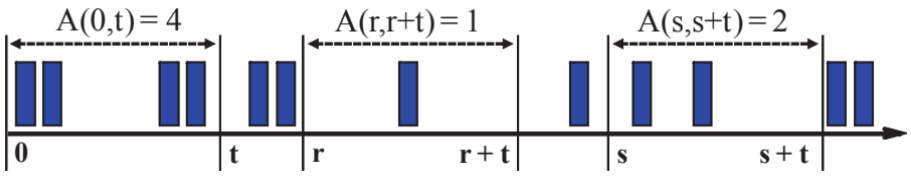


Fig. 3.2: Stationary flow process example.

$$A(t) = \sum_{i=1}^{t} a(i), \quad a(t) = A(t) - A(t-1)$$

No packet occurs at time zero, therefore $A(0) = a(0) = 0$

The procedure as shown in in Fig 3.1 is as follows:

- We pre-process the CIC-DDoS2019 dataset by resampling to the smallest granularity (1 ms) to obtain the total number of packets flowing in each millisecond for the UDP attack. This is used henceforth.

- Next, we compute larger time quantum windows where any window of size $k$ms consists of all packets flown within that window. We then select the optimal window size by contrasting the time taken to run algorithms of $\mathcal{O}(n^k)$ time complexity, where k is the number of windows. We identify the knee point in the time vs. window size graph using techniques similar to [13] and choose 1 second as the granularity for further computations.

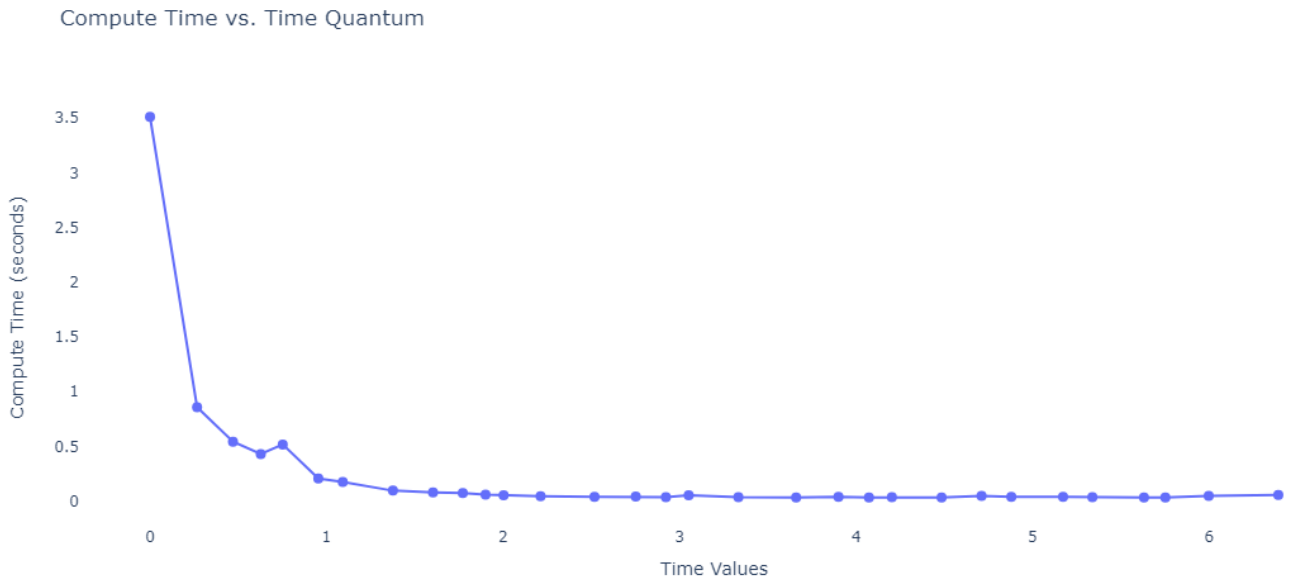


Fig. 3.3: Compute time(logarithmic scale) v/s Time Quantum.

Note: Bollinger bands were computed (upper and lower) on the entire time series obtained after dividing into slots of size q where q is the time quantum (window width).

- The value a_t is envisioned/adapted to stock prices in the finance world. We hence get two types of datasets - **Regular Closing Prices Data** one containing a single feature for a_i which is the total number of packets flown in i^{th} time window, and another containing **OHLC data**[1] where 4 features are available for any i^{th} time series:

  1. **Open**: Which is the opening price at start of window i.

  2. **High**: Which is the highest price within window i.

[1]Note: To do this sampling, we first sample to 500ms with the Regular Closing prices and then consider a'_{2i} and $a'_{(2i-1)}$ to construct a_i where a are windows of OHLC data with 1s granularity and a' are windows of Regular Closing prices with 500ms granularity.

3. **Low**: Which is the lowest price within window i.

4. **Close**: Which is the closing price at end of window i.

a_t vs t graphs are as shown in Fig 3.7.

- We apply financial and non-financial indicators and perform hyper-parameter optimization to select the best values by tuning within a search window on the training set. The values and thresholds are determined based on the training set.

- Signals isolated from the above algorithms are evaluated for efficiency using a metric that combines signal density and distance from known attack times. These algorithms are then tested on the same attacks in the test sets.

3.1.2 Types of indicators/statistical measures

We employ three classes of indicators as explained in tables 3.1, 3.2, 3.3.

1. **Trend & Momentum** Trend indicators provide an objective measure of trend direction by smoothing price data and representing the trend with a single line, such as moving averages. Momentum indicators gauge the strength or weakness of a stock's price by measuring the rate of price rise or fall. *Note: These indicators use Regular Closing Prices Data.*

2. **Multi-OHLC:** These indicators are utilized after creating candlestick bar charts from the resampled data. OHLC charts show open, high, low, and closing prices for each period, aiding in employing other indicators. *These indicators use OHLC Data.*

3. **Non-Financial:** These indicators are non-financial and use classical methods for computing pattern changes. *Note: These indicators use Regular Closing Prices Data.*

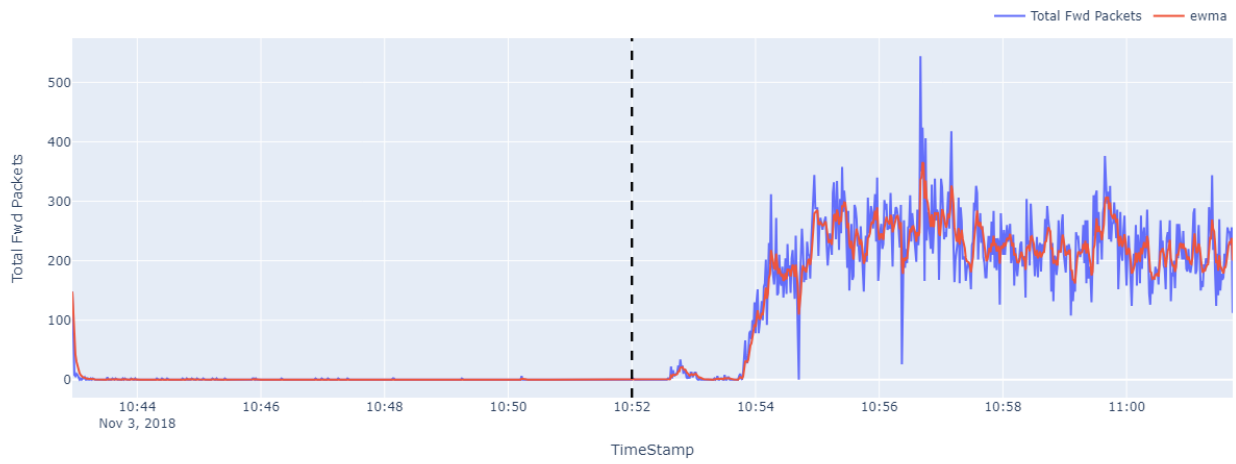


Fig. 3.4: a_t v/s t for Regular Data along with EMA(1.1.2) with α=0.3

Fig. 3.5: a_t v/s t for OHLC data, shown as candlestick charts

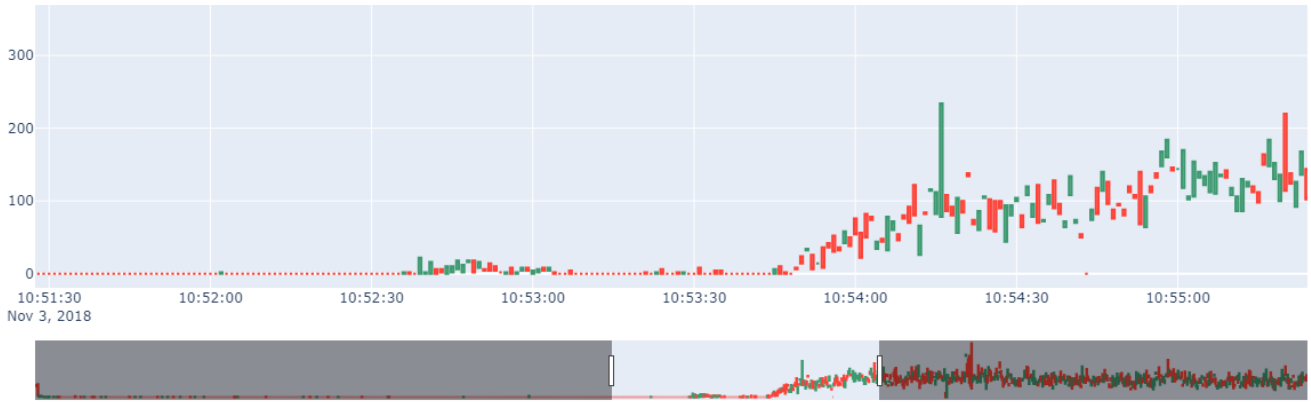


Fig. 3.6: Zoomed-in OHLC candlestick charts at attack time (10:53)

Fig. 3.7: We adapt IoT systems to financial world and show stock-price like charts , a_t is considered to be stock price for applying financial indicators.

3.2 Technical indicators

Tables 3.2,3.1,3.3 highlight the indicators and statistical measures (indic.) used in this study. The indicators are mainly inspired from financial analysis [6, 16, 15]. Technical analysis is the study of market action, primarily through the use of charts(in our case based solely on price), for the purpose of forecasting future price trends.

C, H, L= Current Close, High, Low Price
C_{avg}, C_P = Average and previous close price.
n_i, k = Arbitrary constants.
w_f = Fast window, w_s = Slow window ($w_f > w_s$)
ROC_i, EMA_i = Rate of Change and EMA in period i
σ= Std. Dev.
P, Q= Arbitrary probability distribution

Strategy	**Formula**	**Brief Description**
Average True Range (ATR)	$\text{ATR}_n = \frac{\text{ATR}_{n-1} + \text{TR}_n}{n}$ $\text{TR} = \max(\text{H - L}, \lvert\text{H - C}_P\rvert, \lvert\text{L - C}_P\rvert)$	The ATR considers the range of price movements within a given period, providing insight into how much an asset typically moves over that timeframe.
Keltner Channel High Band Indicator (KCHB)	$KC = EMA_{w_f}(C) + k \times ATR$	The Keltner channel uses ATR, this makes the upper band and deviation from it by$k$ gives a signal.
Awesome Oscillator (AO)	$AO = \text{SMA}\left(\frac{H+L}{2}, w_f\right)$ $- \text{SMA}\left(\frac{H+L}{2}, w_s\right)$	Measures momentum using the difference between two short-term moving averages of price midpoints.

Strategy	Formula	Brief Description
Stochastic Oscillator (%K) (Stoch)	$\%K = \frac{\text{C} - \text{Min}L_{w_s}}{\text{Max}H_{w_s} - \text{Min}L_{w_s}} \times 100$ $\%D = \text{SMA}(\%K, w_f)$	Compares the closing price to the price range over a period, indicating overbought or oversold levels.
Aroon	$\text{AroonUp} = \left(\frac{N - \text{MaxH}_N}{N}\right) \times 100$ $\text{AroonDown} = \left(\frac{N - \text{MinL}_N}{N}\right) \times 100$ MaxH_N= Periods since N period High MinL_N= Periods since N period Low	The indicator measures the time between highs and the time between lows over a time period.
Ulcer Index (UI)	$\text{UI} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(PD_i)^2}$ $PD = \frac{C - MaxC_{14}}{MaxC_{14}}$	The Ulcer Index focuses solely on the magnitude and duration of drawdowns, providing a more accurate assessment of risk.

Table 3.1: Multi OHLC strategies

Strategy	Formula	Brief Description
Bollinger Bands (BB)	$BB = SMA + k \times \sigma$	Volatility bands based on a moving average. Bands widen with higher volatility and contract with lower volatility.
Moving Average Convergence Divergence (MACD)	$MACD = EMA_{w_f} - EMA_{w_s}$	The MACD indicator calculates the difference between two moving averages, to signal potential trend changes and momentum shifts in a feature.

Strategy	**Formula**	**Brief Description**
Relative Strength Index (RSI)	$RSI = 100 - \frac{100}{1 + \frac{\text{AG}}{\text{AL}}}$ $AG = \frac{\sum_{i=1}^{n} \max(\text{C}_i - \text{C}_{i-1}, 0)}{n}$ $AL = \frac{\sum_{i=1}^{n} \max(\text{C}_{i-1} - \text{C}_i, 0)}{n}$	Measures momentum by comparing recent gains and losses. Indicates potential overbought or oversold conditions.
Triple Exponential Moving Average (TRIX)	$\text{TRIX} = EMA^3_{w_f}$	TRIX applies triple smoothing to the price data, making it more responsive to changes in trend compared to traditional moving averages.
Know Sure Thing (KST)	$\text{KST} = \sum_{i=1}^{4} \text{SMA}(\text{ROC}_i, n_i) \times i$	The KST incorporates multiple moving averages of different periods to provide a smoother and more reliable signal.
True Strength Index (TSI)	$TSI = \frac{\text{PCDS}}{\text{APCDS}} \times 100$ $\text{PC} = \text{C} - \text{C}_P \quad , \quad \text{APC} = \text{C}_{avg} - \text{C}_P$ $\text{PCDS} = \text{EMA}_{w_f}(\text{EMA}_{w_s}(\text{PC}))$ $\text{APCDS} = \text{EMA}_{w_f}(\text{EMA}_{w_s}(APC))$	The TSI combines multiple EMAs of price changes to smooth out price fluctuations and highlight underlying trends.
StochRSI	$\text{StochRSI} = \frac{\text{RSI} - \text{Min RSI}_K}{\text{Max RSI}_K - \text{Min RSI}_K}$	It aims to provide more sensitive signals by applying the stochastic oscillator formula to the RSI values rather than to the price itself.

Strategy	Formula	Brief Description
RAPD	It is an algorithm which can best be described as a modification on BB the pseudocde for signal generation on any series is presented in Algorithm:1.	It is based on the principle of dispersion: if a new datapoint is a given x number of standard deviations away from a moving mean, the algorithm gives a signal. The algorithm is very robust because it constructs a separate moving mean and deviation, such that previous signals do not corrupt the signalling threshold for future signals. The sensitivity of the algorithm is therefore robust to previous signals.

Table 3.2: Momentum and Trend Based Strategies

Strategy	Formula	Brief Description
Coefficient of Variation (C/V)	$$CV = \sigma/\mu$$ $$\mu = \frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)$$ $$\sigma = \left[\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2\right]^{\frac{1}{2}}$$	This statistic measures the relative dispersion of data points around the mean.
Skewness Coefficient (SKEW)	$$Skew = \frac{\mu_3}{\sigma^3}$$ $$\mu_3 = \frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^3$$	It measures the asymmetry of a distribution relative to a symmetrical (normal) distribution.
Kurtosis Coefficient (KURT)	$$Kurt = \frac{\mu_4}{\sigma^4}$$ $$\mu_4 = \frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^4$$	It measures the tailedness of a distribution compared to a normal distribution.

Strategy	Formula	Brief Description
Hurst Exponent using Rescaled Range (R/S)	$H = \log\left(\frac{Y_{max} - Y_{min}}{\sigma}\right) / \log(n)$ $Y_{max} = \max_{1 \le i \le n} (\sum_{j=1}^{i} (X_j - \mu))$ $Y_{min} = \min_{1 \le i \le n} (\sum_{j=1}^{i} (X_j - \mu))$	The R/S analysis estimates the Hurst Exponent by comparing the range (difference between the maximum and minimum) of the data over different time windows with its standard deviation.
Kullback-Leibler divergence (D_{KL})	$D_{KL}(P\|\|Q) = \sum_i P(i) \log\left(\frac{P(i)}{Q(i)}\right)$ P(i): probability of i under P. Q(i): probability of i under Q.	KL divergence is a measure of how one probability distribution diverges from a second, expected probability distribution.
Autoregressive coefficient (c)	$c = \frac{\sum_{i=1}^{N-1} X_i^2}{\sum_{i=1}^{N-1} X_i X_{i+1}}$	Autoregressive models are a class of models used in time series analysis where the current value of a variable is modeled as a linear combination of its past values.

Table 3.3: Other strategies

Algorithm 1 RAPD Algorithm (Non-Incremental)

for $i = lag + 1$ to t **do**
  if $|y(i) - \text{avgFilter}(i-1)| > \text{threshold} \times \text{stdFilter}(i-1)$ **then**
    if $y(i) > \text{avgFilter}(i-1)$ **then**
      set signals(i) to +1; ▷ Positive signal
    else
      set signals(i) to -1; ▷ Negative signal
    end if
    set filteredY(i) to influence $\times\ y(i) + (1 - \text{influence}) \times \text{filteredY}(i-1)$;
  else
    set signals(i) to 0; ▷ No signal
    set filteredY(i) to $y(i)$;
  end if
  set avgFilter(i) to mean(filteredY(i-lag+1),...,filteredY(i));
  set stdFilter(i) to std(filteredY(i-lag+1),...,filteredY(i));
end for

Algorithm 2 Efficient RAPD Algorithm (Incremental)

for $i = \text{lag} + 1$ to t **do**
    Compute $avgFilter(i)$ and $stdFilter(i)$ incrementally;
    if $|y(i) - \text{avgFilter}(i-1)| > \text{threshold} \times \text{stdFilter}(i-1)$ **then**
        $signals(i) = \text{sign}(y(i) - \text{avgFilter}(i-1))$;
        $filteredY(i) = \text{influence} \times y(i) + (1 - \text{influence}) \times \text{filteredY}(i-1)$;
    else
        $signals(i) = 0$;
        $filteredY(i) = y(i)$;
    end if
end for

3.3 Incremental Algorithms

An incremental algorithm operates on time-series data. As the data stream progresses, the algorithm computes new values without re-performing fundamental calculations. This makes them particularly useful for online tasks where fast computation is crucial. In our study, we have predominantly employed financial indicators designed to function incrementally. However, the non-financial indicators have been approached in a non-incremental manner, as illustrated in [14, 5]. This choice facilitates a comparative analysis between incremental and non-incremental methodologies, shedding light on their respective strengths and limitations.

Chapter 4

Results and Analysis

Online Tasks It's important to recognize that financial indicators are essentially created for operating in online tasks over a stream of data and can hence be correctly adapted in IoT systems as well. An online IDS focuses primarily on compute time and efficiency.

Small Scale IoT devices This implies that by appropriately adjusting parameters, these indicators can seamlessly integrate with IoT devices. Even compact IoT devices can leverage statistical metrics, providing them with a competitive advantage over more computationally intensive machine learning approaches which may not be implementable in such devices.

Quantifying performance Thus, we propose defining a metric to evaluate and compare these indicators.

- Compute time
- Metric(sec. 4.1) which captures efficiency through early detection as well as frequency (number of signals).
- False Positives (before attack instant)

The indicators when above(or below) a certain threshold return either a 1 or a 0 indicating whether or not an attack has started (True/False) and hence return a signal array after being applied on the a_t time series. The returned signal array is hence binary in nature.

4.1 Metric Established

Usual metrics lack a temporal aspect to them. For instance, simply detecting whether or not an attack has occurred and is labelled correctly misses out on how early the attack was detected and classified correctly. This calls for a better metric with adjustable parameters to present a trade-off between **frequency** (number of signals after the attack instant) and **proximity** (closeness of signal to the attack time). We establish a metric μ with an exponential decay function as follows:

$$\boxed{\mu_\alpha(i) = e^{-\frac{x_i - \lambda}{\alpha}}} \tag{4.1}$$

The total metric is given as the sum of this over all values in the signal array which are 1 and obtained after the known attack time.

$$\boxed{\mu_\alpha = \sum_i \mu_\alpha(i), i \text{ ranges over all 1 in the signal array where } x > \lambda.} \tag{4.2}$$

1. λ is the attack instant.
2. x_i is the time the signal is obtained ($x_i > \lambda$).
3. $\alpha > 0$ is an arbitrary value.

The α value presents a trade-off between frequency and temporal accuracy. More specifically, the number of signals and the closeness of signals to the attack instant. As $\alpha \to \infty$, $\mu_\alpha \to$ total number of signals. As $\alpha \to 0$, μ_α is dominated by the 1^{st} "True" signal with a value $e^{-\frac{x-\lambda}{\alpha}}$ where x is the instant signal is received.

4.2 Indicators Analysis

Regular Close prices along with EMA and OHLC data are as shown in Fig. 3.7. Note the vertical dashed black line is the time from when the attack begins (10:52:00).

Indic.	FP	TP	$\mu_{0.01}$	$\mu_{0.1}$	$\mu_{0.5}$	μ_1	μ_2	μ_{10}
BB	1	29	3.24×10^{-28}	6.51×10^{-3}	2.40	6.90	13.41	24.65
RAPD	2	41	3.85×10^{-28}	$\mathbf{1.08 \times 10^{-2}}$	3.95	10.65	19.80	35.13
KST	0	31	4.10×10^{-91}	5.80×10^{-9}	0.30	3.04	9.68	24.55
MACD	0	57	1.10×10^{-123}	3.23×10^{-12}	0.09	2.18	11.05	40.99
RSI	0	33	3.12×10^{-28}	2.73×10^{-3}	1.16	5.05	12.41	27.00
TSI	0	142	1.15×10^{-89}	8.14×10^{-9}	0.50	7.06	29.99	**103.10**
TRIX	0	21	2.59×10^{-30}	6.81×10^{-3}	3.91	9.02	13.75	19.29
StochRSI	1	7	$\mathbf{5.54 \times 10^{-26}}$	7.18×10^{-3}	0.95	2.13	3.61	6.06
AWSM	0	3	1.11×10^{-29}	1.27×10^{-3}	0.29	0.67	1.13	2.22
Stoch	1	32	6.11×10^{-29}	3.31×10^{-3}	1.03	3.44	9.16	24.50
KCHB	0	127	7.27×10^{-29}	9.30×10^{-3}	**4.48**	**16.15**	**40.62**	99.52

Table 4.1: Evaluations for financial indicators.
FP: False positives (signals before attack time). **TP**: True positives i.e. **Frequency** (signals after attack time). $\boldsymbol{\mu_i}$: Metric μ evaluated with $\alpha = i$.

Indic.	Compute Time (μs)
BB	1515.2935
RAPD	39124.363
Efficient RAPD [a]	14973.2
KST	5983.1953
MACD	870.7396
RSI	3401.3828
TSI	2561.6479
TRIX	1391.1973
StochRSI	5138.1931
AWSM	945.697
Stoch	1557.7529
KCHB	12482.8988
UI	8516.8644
ATR	10069.961
AROON	5886.4548

Fig. 4.1: Compute time in μs for financial indicators (avg. over 10000 iterations).

[a] As in Algo 2

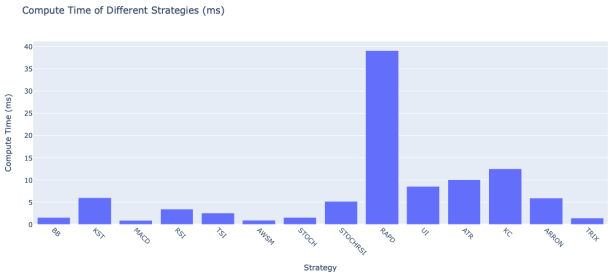


Fig. 4.2: Bar chart comparing compute times for financial indicators *Note: Value is high for RAPD when using Algo. 1.*

4.2.1 Financial Indicators

Bollinger Bands (BB)

Bollinger Bands are created on regular data with SMA over 300 points and $k = 5$. Under these settings, the bollinger band indicator performs quite well and gives reliable as well as early signals as is evident from a low $\mu_{0.01}$ score from Table 4.1.

Know Sure Thing (KST)

KST is taken to give a signal when the $KST - \text{EMA}(KST) > 3000$ (EMA smoothed over a 150-window period). $n_1 = n_2 = n_3$=300, n_4=250. KST gives a quite clear and evident peak as is

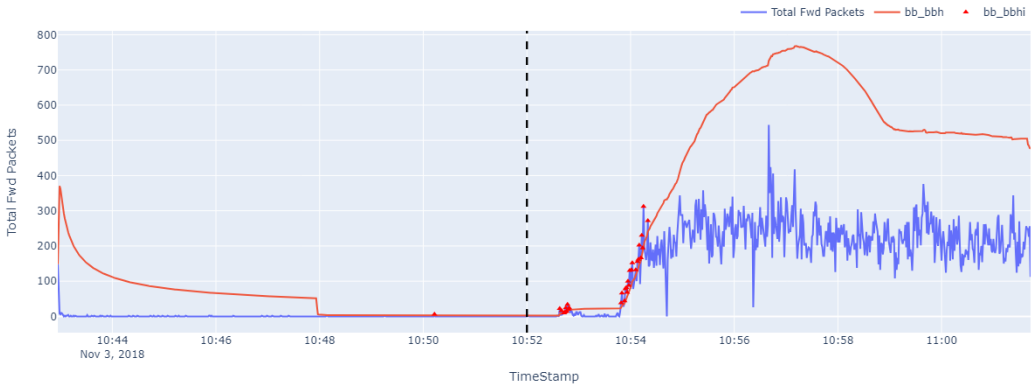


Fig. 4.3: BB on a_t

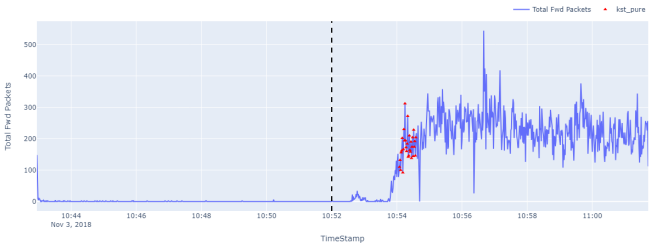


Fig. 4.4: KST indicator on a_t

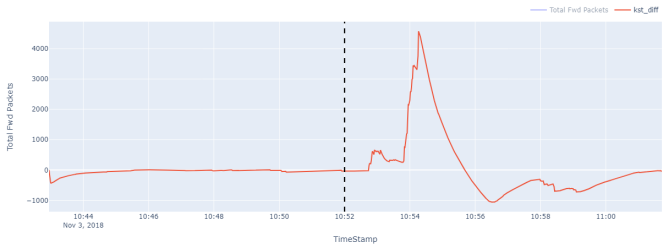


Fig. 4.5: KST-EMA(KST)

Fig. 4.6: KST indicator and signal line

evident in Fig. 4.8. Even with a lower threshold value on KST ine, we can prevent fals positives easily and can get more frequent signals. However this is avoided in our study and a high threshold is set to prevent overfitting.

Moving Average Convergence Divergence (MACD)

MACD is taken to give a signal when the $MACD - \text{EMA}(MACD) > 5$ (EMA smoothed over a 100-window period). w_f=250, w_s=300. MACD gives frequent true positives evident from the count of true positives and a high μ_{10} value from Table 4.1 and no false positives. It performs

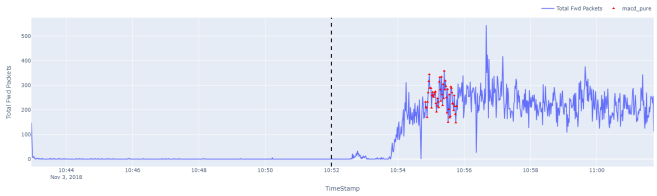


Fig. 4.7: MACD indicator on a_t

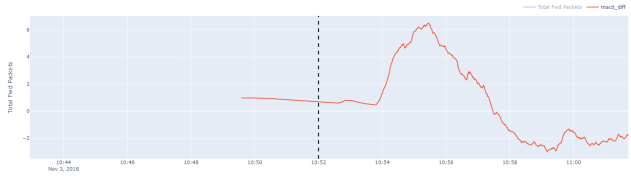


Fig. 4.8: MACD-EMA(MACD)

Fig. 4.9: MACD indicator and signal line

quite similar to KST.[1]

Relative Strength Index (RSI)

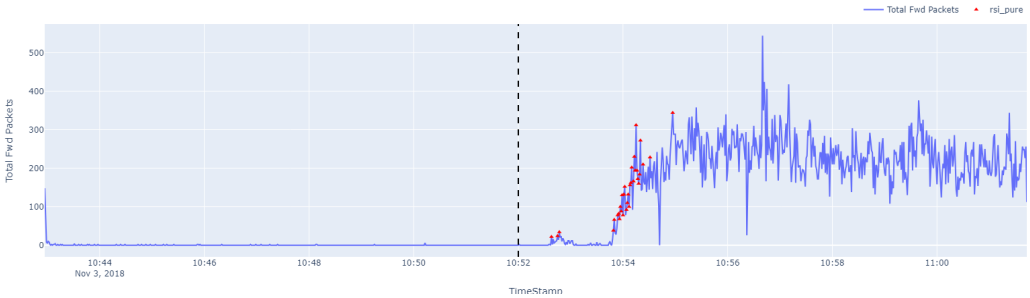


Fig. 4.10: RSI on a_t

RSI is set with with n=200. We take the threshold to be 55 and take a signal when $RSI > 55$. RSI gives quite early signals wihch is very evident from the higher $\mu_{0.01}$ and $\mu_{0.1}$ values of RSI inspite of less frequency (TP). Infact for KST which has higher TP value than RSI, all μ_k values are higher for RSI (refer Table 4.1). Hence RSI becomes a crucial asset for IDS where early detection is of high value.

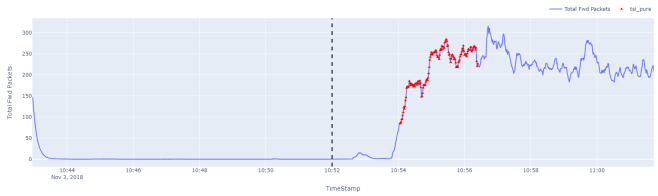


Fig. 4.11: TSI indicator on EMA(a_t)

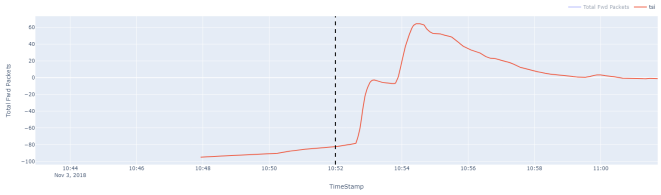


Fig. 4.12: TSI value

Fig. 4.13: TSI indicator and value

True Strength Index (TSI)

TSI is modified and the formula is applied on the smoothed EMA(a_t)(14-window period) instead of a_t [2]. A signal is taken when $TSI > 28$. This threshold gives a high frequency of signals (TP and μ_{10} values justify high frequency from Table 4.1).

Awesome Oscillator (AO)

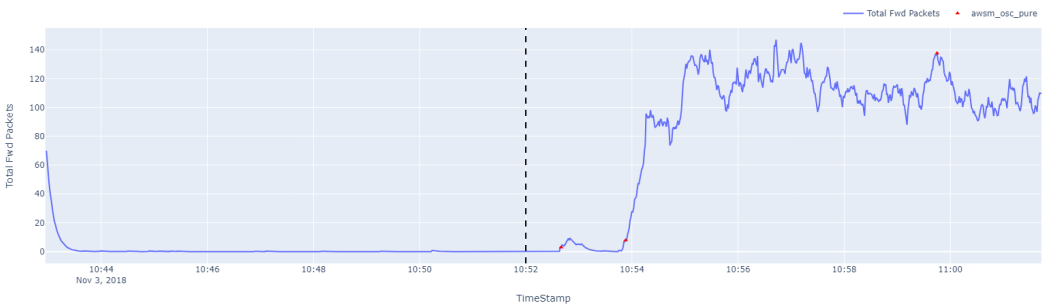


Fig. 4.14: AO on EMA(a_t)

Awesome Oscillator was applied on EMA of a_t over a 14 window period and refined to give a signal when the current value is positive and previous 3 values are negative (detecting crossover).

[1]Note both KST and MACD are taken after subtracting their smoothened EMA values from themselves. Such indicators are usually laggy in nature but aim to give a surely correct signal and prevent false positives.

[2]Note unlike the detection for other indicators, this is signal detection is done unconventionally for TSI as it was seen to perform better experimentally

Although the signals are less frequent, they are quite early and in this sense the AO indicator is capable of detecting an attack an early on and if a positive signal is given, one can be quite sure that it is indeed an attack as the indicator remains 0 throughout in noisy or non-attack scenarios.

Stochastic Oscillator (Stoch)

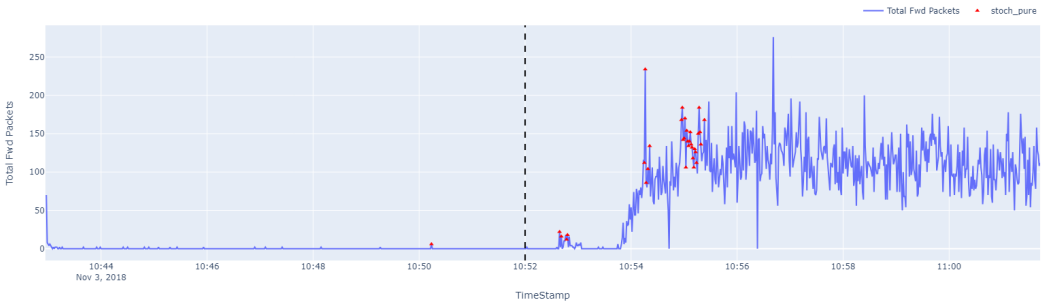


Fig. 4.15: Stoch on a_t

A true signal is taken when the $Stoch > 90$. w_f=50, w_s=150 is taken. The stochastic oscillator gives quite early signals and detects peaks correctly just as the upward trend gets over. It infact shows comparable performance to or beats KST and MACD on most μ values, showing its better **proximity** nature and its usefulness in IDS where time and early detection is of importance.

Stochastic Oscillator RSI (StochRSI)

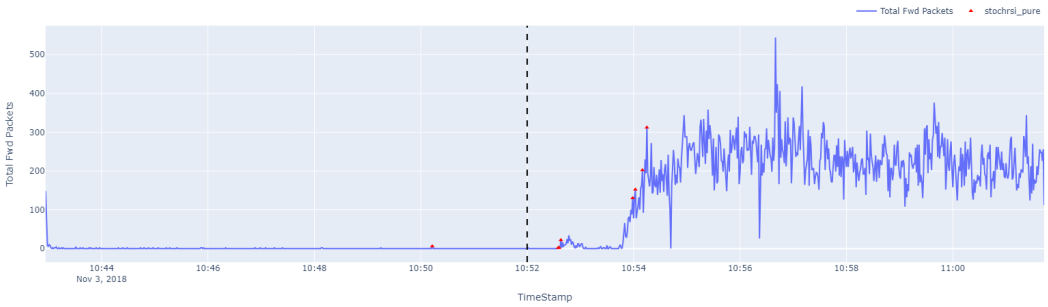


Fig. 4.16: StochRSI on a_t

A true signal is taken when the $StochRSI > 95$. k=100 is taken and RSI is taken over 200 window period. StochRSI gives the earliest signal and even though the frequency is comparatively lesser than other signals, **proximity** to attack instant is high. this is justified by the highest $\mu_{0.01}$ value seen in Table 4.1.

Robust Peak Detection Algorithm (RAPD)

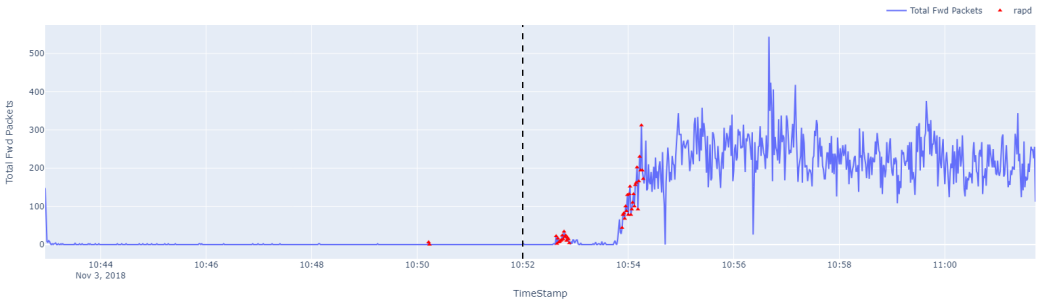


Fig. 4.17: RAPD on a_t

RAPD is a niche algorithm proposed for time series peak detection [3].We have used follwing values:

$$lag = 200, threshold = 5, influence = 0.5$$

Comparison between Algo 1 and Algo 2 show that incremental versions can give upto 3X better performance as in Table 4.2. It provides a decent balance between frequency as well as proximity shown by a high enough $\mu_{0.01}$ and $\mu_{0.5}$ value and the highest $\mu_{0.1}$ value as in Table 4.1. It can be used in IDS and is often seen as a modification on BB. The algorithm is very robust because it constructs a separate moving mean and deviation, such that previous signals do not corrupt the signalling threshold for future signals. The sensitivity of the algorithm is therefore robust to previous signals and so it can be used in IDS where robustness to nature of traffic is important.

Keltner Channel High Band (KCHB)

KCHB is applied on the EMA smoothed values of OHLC data with 14 window period. $k = 2$,$w_f = 200$. KCHB gives frequent signals after the attack time as the Keltner band lags as is visible in 4.18. This is also justified by highest $\mu_{0.5}, \mu_1, \mu_2$ values. However it is important to note that the signals are given quite early as well because when compared with TSI, even though TSI has more TP, KCHB gives better performance on $\mu_{0.5}, \mu_1, \mu_2$. So we can say that overall the indicator performs better than TSI and presents a good number of signals without comprising too much on early detection and can hence contribute to a robust IDS.

[3]refer: https://stackoverflow.com/questions/22583391/peak-signal-detection-in-realtime-timeseries-data

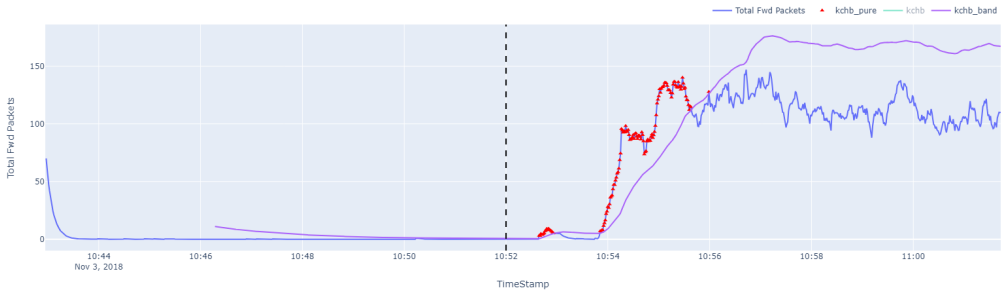


Fig. 4.18: KCHB on OHLC data with EMA

Triple Exponential Moving Average (TRIX)

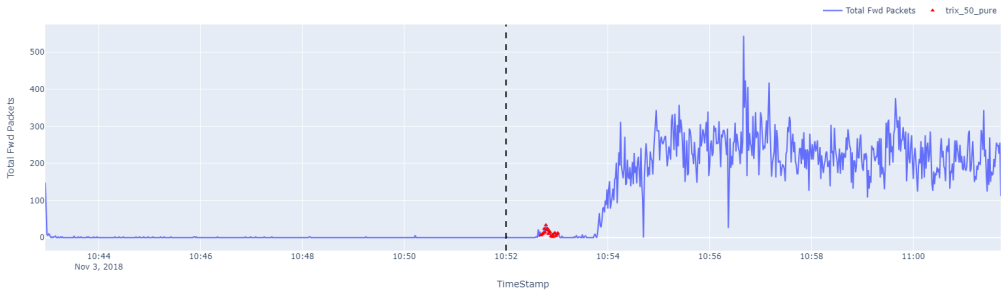


Fig. 4.19: TRIX on a_t

TRIX is applied with $w_f = 50$ and taken to give a signal when $TRIX > 10$. A high $\mu_{0.5}$ value justifies the balance between proximity an frequency for the TRIX indicator and the fact that there are no False Positives can make it useful in IDS systems where nature of traffic is unknown for the most part and robust signals are needed much like RAPD.

Other Indicators

Ulcer Index (UI) and Aroon indicators are seen to not perform so well and it is tough to select a parameter value which can create clear signals but from rough estimates upon plotting multiple graphs, we propose the following window values for these indicators: 50 for UI (blue line giving a sharp and clear signal) and 80-100 for Aroon (Red and green line givnig sharp peaks). Similar tuning was done for other indicators and hence the metrics and thresholds were calculated.

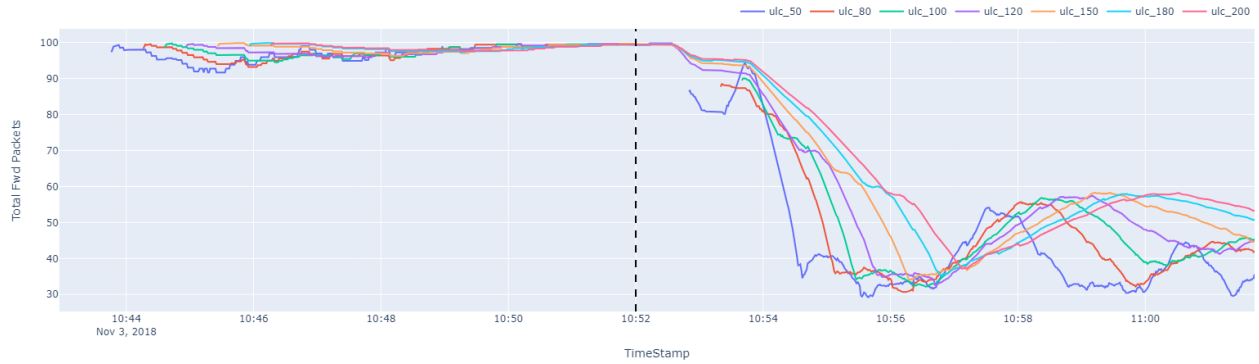


Fig. 4.20: Ulcer indicator on a_t

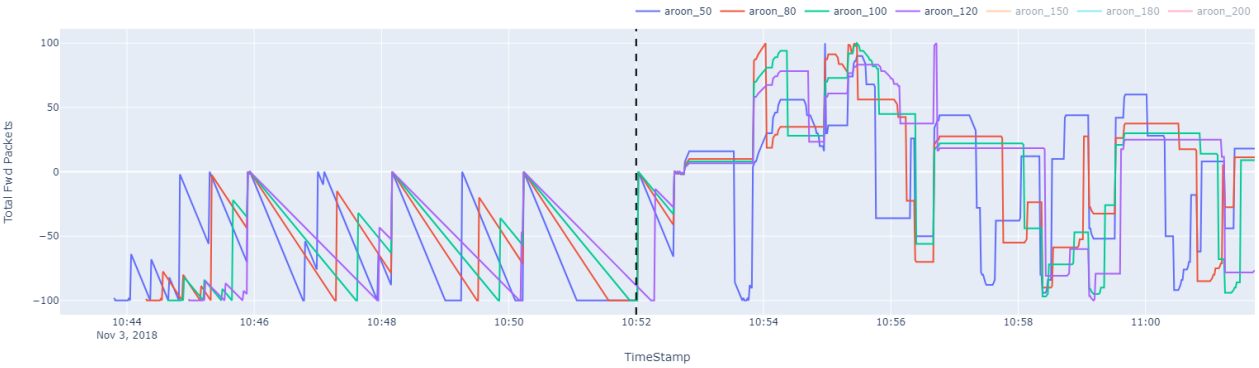


Fig. 4.21: Aroon indicator on a_t

Fig. 4.22: Other Financial Indicators

4.2.2 Non-Financial Indicators

Indic.	**FP**	**TP**	$\boldsymbol{\mu_{0.01}}$	$\boldsymbol{\mu_{0.1}}$	$\boldsymbol{\mu_{0.5}}$	$\boldsymbol{\mu_{1}}$	$\boldsymbol{\mu_{2}}$	$\boldsymbol{\mu_{10}}$
SKEW	1	2	3.71×10^{-28}	3.12×10^{-3}	0.55	1.05	1.45	1.87
KURT	1	2	3.61×10^{-28}	2.80×10^{-3}	0.55	1.06	1.84	2.11
R/S	1	46	3.12×10^{-28}	1.80×10^{-3}	0.88	5.60	15.74	37.03

Table 4.2: Evaluations for Non-Financial indicators.
FP: False positives (signals before attack time). **TP**: True positives i.e. **Frequency** (signals after attack time). $\boldsymbol{\mu_i}$: Metric μ evaluated with $\alpha = i$.

Indic.	Compute Time (μs)
cv	161955.3
skew	704448.2
kurt	619225.6
entropy	61555.8
hurst(r/s)	3103680.1
autoreg. coef.	2183049.4
kl div.	354295.1

Table 4.3: Compute time in μs for non-financial indicators (avg. over 100 iterations).

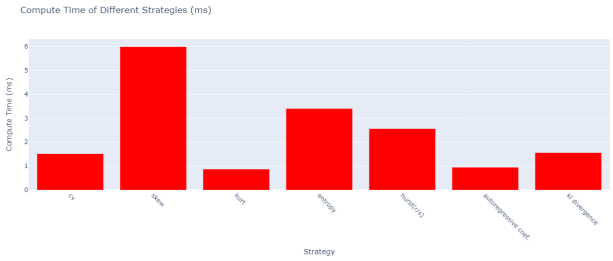


Fig. 4.23: Bar chart comparing compute times for non-financial indicators

Skewness Coefficient (SKEW) and Kurtosis Coefficient (KURT)

KURT and SKEW both act very similar to each other this is because the only differnce between them is in the order of moment $(3^{rd}, 4^{th})$ and are capable of giving correct signals as well as early detection as their $\mu_{0.01}$ is of similar order as some of the financial indicators. However TP is quite low and FP aren't negligible in comparison either as evident from Table 4.2.

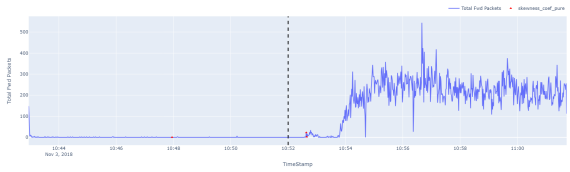


Fig. 4.24: SKEW on a_t

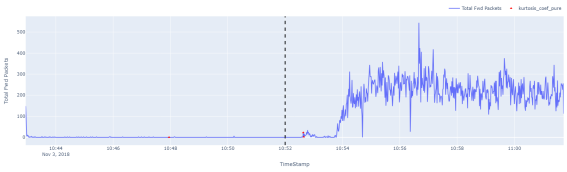


Fig. 4.25: KURT on a_t

Fig. 4.26: KURT and SKEW indicator on regular data

Hurst Exponent using Rescaled Range (R/S)

R/S is constructed with a look-back window of size 300 and is said to give a signal when the value is above 0.5. It seems to be the best performing non-financial indicator giving quite high μ_{10}(refer Table 4.2) value even on comparison to financial indicators table 4.1.

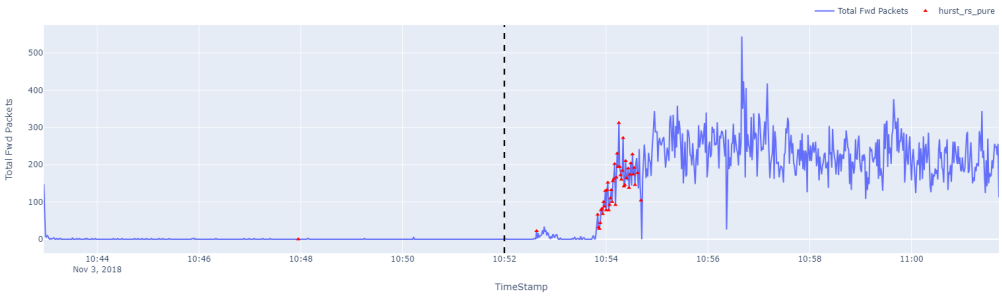


Fig. 4.27: R/S on a_t

Other Indicators

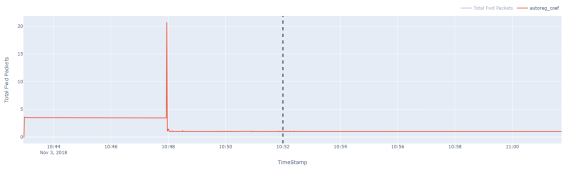


Fig. 4.28: Autoregressive coefficient(c) on a_t

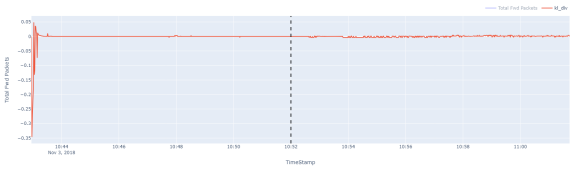


Fig. 4.29: Kullback-Leibler divergence (D_{KL}) on a_t

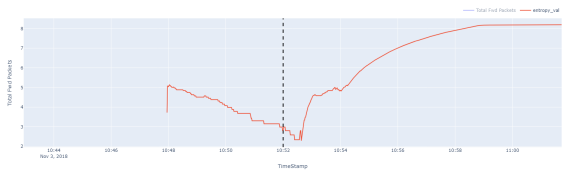


Fig. 4.30: Entropy on a_t

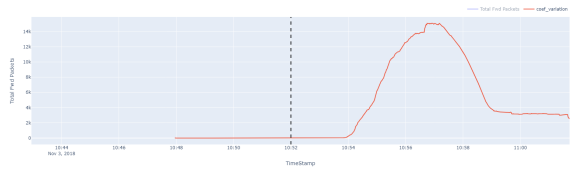


Fig. 4.31: Coefficient of Variation (CV) on a_t

Fig. 4.32: Other Non Financial Indicators

Other Indicators perform quite poorly giving either unclear signals or noisy random signals as is evident from Fig 4.32. Amongst these CV (Fig.4.31) has some potential to be a decent indicator but the peak is too broad and the signal is too spread out to be of any use in early detection and good capturing for short and bursty traffic unlike other indicators which catch a small peak above a certain threshold as well. However, this may work well in conjunction with other indicators. Entropy performs poorly and this finding is in agreement with [5, 14].

Chapter 5

Conclusion

Enhanced Performance with Financial Indicators Financial indicators outperform typical non-financial indicators, enhancing the diversity of Intrusion Detection Systems (IDS). Their utility in an online and compact IoT environment is hence made apparent by analysing their robustness, efficiency, and reliability for IDS implementations.

Evaluation Metric The μ metric proves effective in assessing indicator performance for **proximity** as well as **frequency** (as described in Sec 4.1). It can therefore be of use in the evaluation and testing of IDS systems across other studies to contrast and evaluate different techniques/strategies for DDoS attack detection. We have also effectively evaluated compute times and compared indicators on this scale as well as shown in Tables 4.3, 4.1 and Figs. 4.23, 4.2.

Optimal Indicator Selection We provide computational insights into the incremental nature of financial indicators, recommending the utilization of RAPD and TRIX for a well-balanced IDS. Additionally, indicators such as RSI and Stoch are advocated for their proficiency in early threat detection. The frequent signaling capability of indicators like KCHB contributes to creating a resilient and diverse IDS architecture.

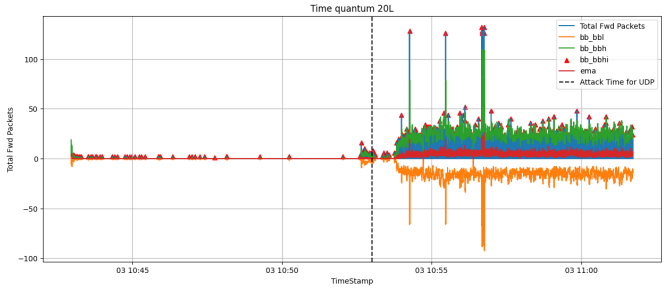
Non-Financial Indicator Performance Non-financial indicators like R/S demonstrate effectiveness in detecting UDP attacks. Indicators such as C/V, with generalized spread-out peaks, serve as valuable components in conjunction with other signals to mitigate false positives.

Robustness through Indicator Combination A strategic combination of indicators enhances system robustness - considering an attack only when the ensemble of multiple indicators are in agreement, thereby reducing false alarms and improving overall IDS reliability and resilience.

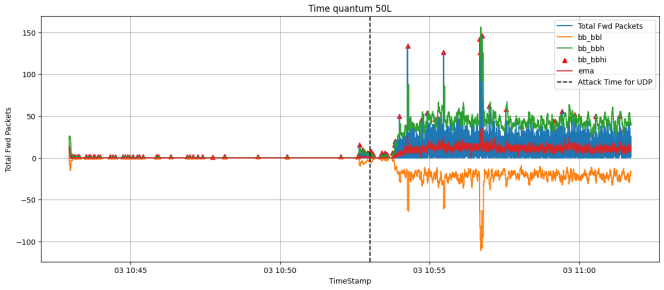
Chapter 6

Appendix

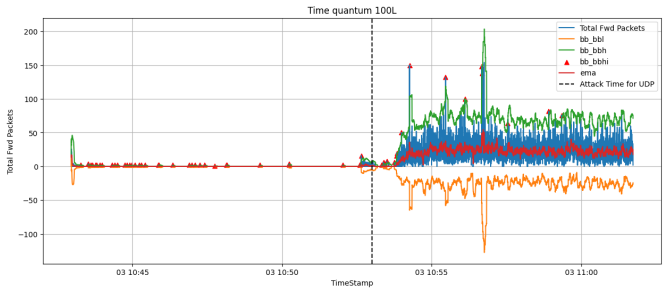
We have run BB on various time windows to obtain the correct compute window size and handle the tradeoff between compute time and window size. Smaller window sizes can aid in early detection which is crucial and give more information on the changing nature of traffic, however they take more time to compute as is explained in Sec. 3.1.1. The graphs are as below



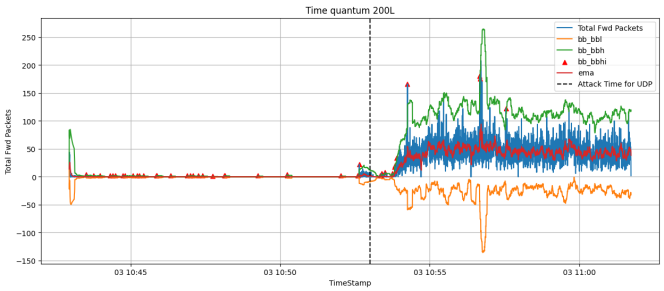
BB on 20ms window



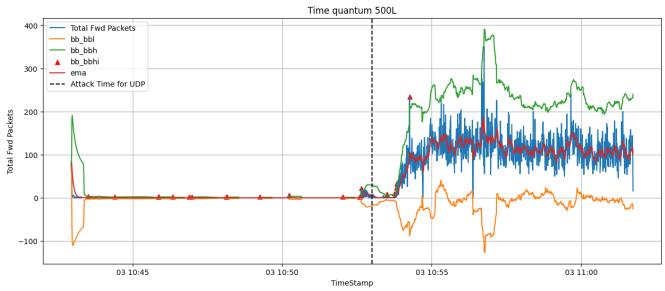
BB on 50ms window



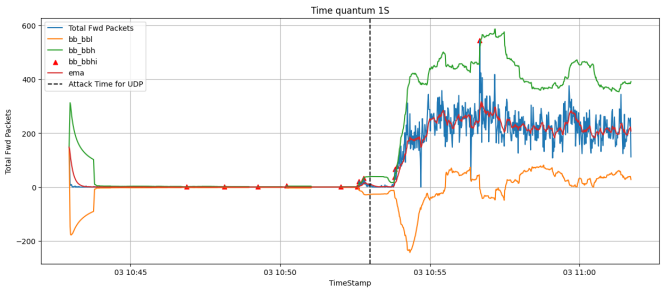
BB on 100ms window



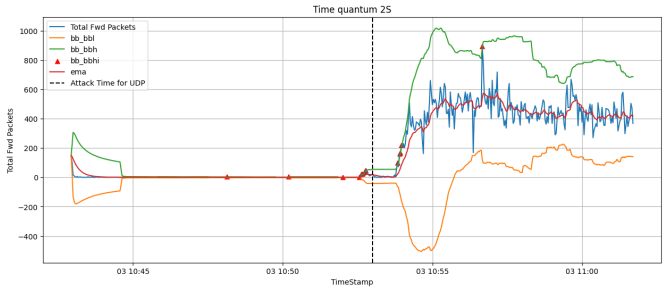
BB on 200ms window



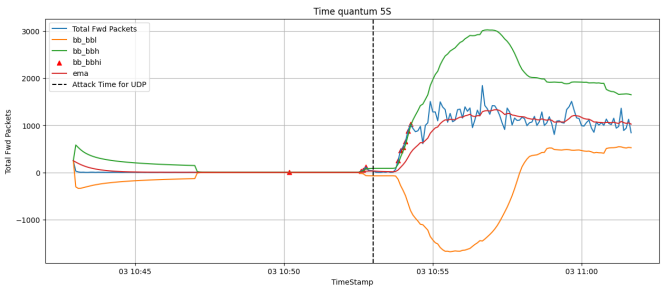
BB on 500ms window



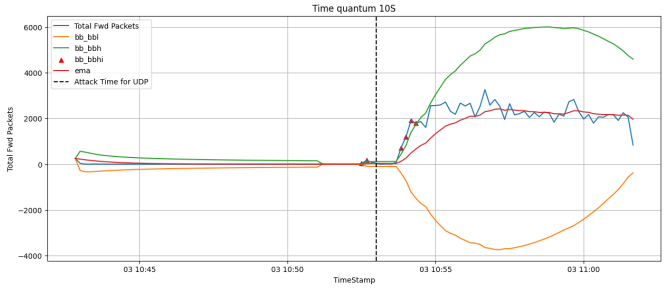
BB on 1S window



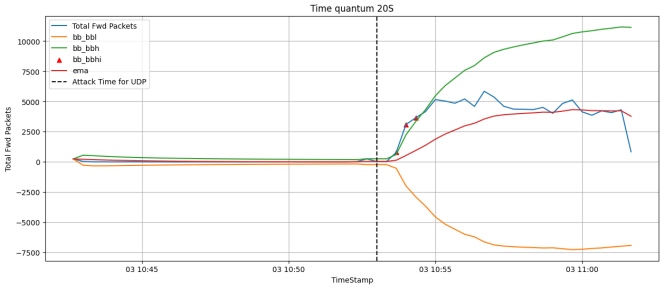
BB on 2S window



BB on 5S window



BB on 10S window



BB on 20S window

References

[1] Daniyal Alghazzawi, Omaimah Bamasag, Hayat Ullah, and Muhammad Zubair Asghar. Efficient detection of ddos attacks using a hybrid deep learning model with improved feature selection. *Applied Sciences*, 11(24):11634, 2021.

[2] Abdullah Emir Cil, Kazim Yildiz, and Ali Buldu. Detection of ddos attacks with feed forward based deep neural network model. *Expert Systems with Applications*, 169:114520, 2021.

[3] Laura Feinstein, Dan Schnackenberg, Ravindra Balupari, and Darrell Kindred. Statistical approaches to ddos attack detection and response. In *Proceedings DARPA information survivability conference and exposition*, volume 1, pages 303–314. IEEE, 2003.

[4] Ramin Fadaei Fouladi, Cemil Eren Kayatas, and Emin Anarim. Statistical measures: Promising features for time series based ddos attack detection. In *Proceedings*, volume 2, page 96. MDPI, 2018.

[5] Roman Hajtmanek, Martin Kontšek, Juraj Smieško, and Jana Uramová. One-parameter statistical methods to recognize ddos attacks. *Symmetry*, 14(11):2388, 2022.

[6] Yufeng Han, Yang Liu, Guofu Zhou, and Yingzi Zhu. Technical analysis in the stock market: A review. *SSRN Electronic Journal*, 01 2021.

[7] Shuyuan Jin and Daniel S Yeung. A covariance analysis model for ddos attack detection. In *2004 IEEE international conference on communications (IEEE Cat. No. 04CH37577)*, volume 4, pages 1882–1886. IEEE, 2004.

[8] Thapanarath Khempetch and Pongpisit Wuttidittachotti. Ddos attack detection using deep learning. *IAES International Journal of Artificial Intelligence*, 10(2):382, 2021.

[9] Deepak Kumar, RK Pateriya, Rajeev Kumar Gupta, Vasudev Dehalwar, and Ashutosh Sharma. Ddos detection using deep learning. *Procedia Computer Science*, 218:2420–2429, 2023.

[10] Jin Li, Yong Liu, and Lin Gu. Ddos attack detection based on neural network. In *2010 2nd international symposium on aware computing*, pages 196–199. IEEE, 2010.

[11] Yan Li and Yifei Lu. Lstm-ba: Ddos detection approach combining lstm and bayes. In *2019 seventh international conference on advanced cloud and big data (CBD)*, pages 180–185. IEEE, 2019.

[12] Francisco Sales de Lima Filho, Frederico AF Silveira, Agostinho de Medeiros Brito Junior, Genoveva Vargas-Solar, and Luiz F Silveira. Smart detection: an online approach for dos/ddos attack detection using machine learning. *Security and Communication Networks*, 2019:1–15, 2019.

[13] Congming Shi, Bingtao Wei, Shoulin Wei, Wen Wang, Hai Liu, and Jialei Liu. A quantitative discriminant method of elbow point for the optimal number of clusters in clustering algorithm. *Eurasip Journal on Wireless Communications and Networking*, 2021:1–16, 2021.

[14] Juraj Smiesko, Pavel Segec, and Martin Kontsek. Machine recognition of ddos attacks using statistical parameters. *Mathematics*, 12(1):142, 2023.

[15] Jelena Stanković, Ivana Marković, and Miloš Stojanović. Investment strategy optimization using technical analysis and predictive modeling in emerging markets. *Procedia Economics and Finance*, 19:51–62, 2015. The Economies of Balkan and Eastern Europe Countries in the Changed World (EBEEC 2014).

[16] Varshini Venu, Bhavya Vikas, and Charithra CM. Equity research using technical analysis. Volume III:2454–6186, 07 2019.

[17] Shreekh Wankhede and Deepak Kshirsagar. Dos attack detection using machine learning and neural network. In *2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)*, pages 1–5. IEEE, 2018.

[18] Marwane Zekri, Said El Kafhali, Noureddine Aboutabit, and Youssef Saadi. Ddos attack detection using machine learning techniques in cloud computing environments. In *2017 3rd international conference of cloud computing technologies and applications (CloudTech)*, pages 1–7. IEEE, 2017.