2016 Pattern Recognition Association of South Africa and Robotics and Mechatronics International Conference (PRASA-RobMech)
Stellenbosch, South Africa

Using Exponentially Weighted Moving Average Algorithm to Defend Against DDoS Attacks

Pheeha Machaka
University of South Africa, Decision Sciences, Club-One Pinaster Avenue, Hazelwood
Pretoria, South Africa, 0003
machap@unisa.ac.za

Antoine Bagula
University of Western Cape, Computer Science,
Robert Sobukwe Road
Bellville, South Africa
bbagula@uwc.ac.za

Fulufhelo Nelwamondo
Council for Scientific and Industrial Research, MDS
Meiring Naude Rd, Pretoria, 0184, South Africa
FNelwamondo@csir.co.za

***Abstract*— This paper seeks to investigate the performance of the Exponentially Weighted Moving Average (EWMA) for mining big data and detection of DDoS attacks in Internet of Things (IoT) infrastructure. The paper will investigate the trade-off between the algorithm's detection rate, false alarm and detection delay. The paper seeks to further investigate how the performance of the algorithm is affected by the tuning parameters and how various network attack intensity affect its performance. The performance results are analyzed and discussed and further suggestion is also discussed.**

Keywords*— *Change Detection; Distributed Denial of Service; TCP-SYN Flooding; Exponentially Weighted Moving Average.

I. INTRODUCTION

We are living in a rapidly changing information age, were information is available at our fingertips. The use of Information Communications Technology (ICT) has made access to information-on-demand relatively easy and cheaper. The range of services supported by ICT has constantly been expanding and in recent years has included critical infrastructure service applications such as energy, aerospace, stock exchanges, health and tactical military networks. These applications rely heavily on the internet and network communication technologies for storing, processing and transmission.

The proliferate use of internet and network technologies has led to significant dependence of society on ICT systems. Consequently, any malfunction and disruption to the services provided by these systems directly affects major aspects of society. This interruption may be sharply felt even if it is momentary. For example, an interruption in a business organization or government's ICT infrastructure may have a substantial impact on their day-to-day activities. This may lead to significant financial losses (business and law suits) and increased operational costs from fraudulent activities.

With the recent developments in the Internet of Things (IoT) as an ICT infrastructure, developing countries are becoming adopters of the IoT. Sectors like energy; agriculture; government; business are to a large degree becoming dependent on IoT. A momentary disruption in this infrastructure can be worst for developing countries. The resulting disruptions may be due to a hacker's attempt to disrupt services using Denial of Service (DoS) attacks. A DoS attack is a malicious attempt by an attacker to disrupt the online services of a service provider to make it unavailable to its legitimate users. A large scale variant of DoS is the Distributed Denial of Service (DDoS).

This kind of attack on an organization may have catastrophic results. This may lead to disgruntled service consumers and major financial losses; it may also lead to losses in an organization's intellectual property which in turn affects the long term competitiveness of businesses and governments in industrial and military espionage incidents [1]. It is therefore important that organizations and governments deploy methods and techniques that will help them to accurately and reliably detect the onset and occurrence of the DDoS attacks.

This paper seeks to investigate the performance of the well-known Exponentially Weighted Moving Average (EWMA) as an application for data mining and detection of DDoS attacks. The paper will investigate the trade-off between the algorithm's detection rate, false alarm and detection delay. The paper seeks to further investigate how the performance of the algorithm is affected by the tuning parameters and how various network attack intensity affect its performance.

The remainder of the paper is organised as follows: In section 2 that follows will give a brief background review of the TCP SYN flooding attack, our focus for this paper. A brief overview of anomaly and change point detection techniques is also discussed. Furthermore, a detailed discussion of the EWMA detection technique is provided together with other related detection techniques by other scholars. Section 3 will describe the research and experiment design.

.

II. BACKGROUND AND RELATED WORK

A hacker uses DDoS attacks in order to prevent legitimate users from accessing the service of a provider. The attacker does this through the use of an attack that streams multiple illegitimate requests to the victim, i.e. a High-Rate Flooding (HRF) attack. There have been various classifications of DDoS attacks in the literature [1-6], however the focus of this paper

978-1-5090-3335-5/16/$31.00 ©2016 IEEE

will be on the malicious and widely used TCP SYN flooding attack.TCP SYN Flooding Attack

A TCP SYN flooding attack is an example of a network layer flooding attack, and it is one of the most common and powerful flooding methods. It exploits the vulnerabilities of the TCP three-way handshake. In a normal TCP connection, the client initiates the connection by sending a SYN packet to the server, as a way of requesting a connection. Upon receiving the connection request, the server will open a connection session and respond with a SYN_ACK packet; by doing this the server stores details of the requested TCP connection in the memory stack and allocates resources to this open session. The connection remains in a half-open state, i.e. the SYN_RECVD state. To complete the three-way handshake with the server, the client will need to confirm the connection and respond with an ACK packet. The server will then check the memory stack for an existing connection request, and the TCP connection will be moved from the SYN_RECVD state to ESTABLISHED state. If there is no ACK packet sent within a specific period of time, the connection will timeout and therefore releasing the allocated resources.

In a TCP SYN flooding attack, the attacker streams large volumes of SYN packets towards the victim server. These packets normally contain spoofed IP addresses, i.e. IP addresses that are non-existent or are not utilised. TCP SYN floods can also be launched using compromised machines with legitimate IP addresses, however the machines need to be configured in such a way that it does not respond or acknowledge a SYN_ACK packet from the victim server. In this way the server will not receive any ACK packet from the clients for the 'half-open' connection request. During the high rate flooding attack, and for a period of time, the server will maintain a large volume of incomplete three-way handshake and allocates resource towards the fictitious connection requests. The server will gather more fictitious requests and eventually exhaust its resources. This will prevent new requests, including legitimate client requests, from being further processed by the server.

A. Anomaly and Change Detection Algorithms

In the event of a DDoS attack, abrupt changes in network traffic can be observed. Similarly, an abrupt change in statistical properties of detection parameters can be observed. Thus, the problem of anomaly detection can be constructed as change point detection problem [7, 8]. The aim of change detection techniques are to help detect a change in statistical properties of observed parameters with minimal detection delay and false positive rate [9]. The approach first starts by applying filter to the traffic data by desired parameters and arraigning the data into a time series data. For change detection, if there was a DDoS attack at time λ, the time series will show a significant statistical change around or at a time greater than λ [10].

Detecting changes in statistical properties of observed parameters has been studied extensively and applied in various fields like image processing, network traffic and financial analysis. There are a number of techniques that are used for change detection and amongst them the most common technique used for the detection of DDoS attacks is the Exponentially Weighted Moving Average (EWMA) [11].

B. Exponentially Weighted Moving Average Algorithms

EWMA was first introduced by Roberts [12], it analyses whether the value of the parameter(s) being observed (in this case it is the number of SYN packets), in a given time interval (hourly, daily or weekly etc.), exceeds a particular threshold value. The algorithm adaptively calculates the threshold value (the parameter mean value of recent observations in each sampling interval) in order to take into account the instead of using a predefined threshold value.

$$\text{If } X_n \geq (\alpha + 1)\, \bar{\mu}_{n-1}, \tag{1}$$

then an alarm is signalled at time n, where $0 < \alpha \leq 1$ is a tuning parameter that indicates the percentage above the mean value that we consider to be an indication of anomalous behaviour. This tuning parameter is used for computing the alarm threshold. The mean $\bar{\mu}_n$ can be computed over some past time window or using an exponential weighted moving average of previous measurements.

$$\bar{\mu}_n = \beta\, \bar{\mu}_{n-1} + (1 - \beta)\, X_n , \tag{2}$$

Where the tuning parameter $0 < \beta \leq 1$, is the weighting factor parameter. The parameter β determines the rate at which "older" data enter into the calculation of the EWMA statistic. A value of $\beta = 1$ implies that only the most recent measurement influences the EWMA. Thus, a large value of $\beta = 1$ gives more weight to recent data and less weight to older data; a small value of β gives more weight to older data.

However, if the algorithm is applied in its original format, it will yield a higher rate of false alarms. To improve the performance a modification to the algorithm was to raise an alarm after a minimum number of successive violations of the threshold. Therefore,

$$\text{If } \sum_{i-n-k+1}^{n} 1_{x_i \geq (\alpha+1)\bar{\mu}_{i-1}} \geq k$$

Then an alarm is raised at time *n*, where $k > 1$ is a tuning parameter that indicates the number of successive intervals the threshold must be violated before an alarm can be raised.

The tuning parameters for the EWMA algorithm are the threshold value (amplitude factor) α, the EWMA factor β, and *k* which signifies the number of successive threshold violations before raising an alarm.

C. Related Work

There have been a number of variations of the EWMA algorithm that was used for intrusion detection and flooding attacks. Further related work and a detailed critique of the literature can be found here [13], and the section that follows will highlight some of the important literature.

Siris et al [14] proposed an adaptive threshold algorithm, which is a variation of the EWMA technique. They used real

traffic traces to analyse and compare the performance based on detection delay, false alarm rate and detection accuracy. The algorithm adaptively learns the normal behaviour of the network traffic. The algorithm revealed satisfactory results for high intensity attacks, however the performance declined for low intensity attacks. However, it is of paramount importance to detect the onset of an attack whose intensity increases slowly.

Ye et al. [15, 16] investigated and applied EWMA techniques to help detect anomalous changes in the events intensity for intrusion detections. The techniques were applied on the large DARPA datasets. Their findings revealed that the EWMA techniques can work well for detecting abrupt changes in event intensity, and also small mean shifts through the gradually increased or decreased event intensity.

Mönz et al. [17] investigated and evaluated the network traffic anomaly detection capabilities of the Shewhart, CUSUM and EWMA techniques. In order to cope with seasonal variation and serial variation, a time series of prediction errors was used instead of using direct time-series of the traffic data. The traffic data was collected from an Internet Service Provider (ISP). From their experiments it was found that CUSUM does not perform better than Shewart and EWMA when applied to time-series of prediction errors.

III. The Research Design

The section that follows will describe the methods and techniques used to carry out the research presented in this paper. The performance metrics considered for these experiments were the algorithm's detection rate, false positive rate and detection delay.

The experiments were conducted using actual network traffic data from the MIT Lincoln Laboratory. The data contains trace data taken during a day of network activity. In this experiment we considered trace data where there were significant traffic activities. We therefore considered trace data between the times 08h00-19h00, and thus an 11hour of real network packets was used for experiments. In the investigations SYN packets were considered.

The investigation considered SYN packet measurement of 10 seconds intervals. To allow for investigations of the algorithm's performance across different types of attack characteristics, the attacks were generated synthetically. The synthetically generated attack was designed to last for 300 seconds (5 minutes) over 30 time intervals (using a 10 second time interval). To consider all possible attacks within the 11 hour network packet trace, used for these experiments, every 5 minute window was injected with attack data.

In these experiments we consider and simulate two types of attack characteristics: high intensity and low intensity attacks. The details of these characteristics are expanded in the subsections that follow.

A. Low Intensity Attacks

Low intensity attacks are those attacks whose intensity increases gradually. In these experiments we considered the case of a low intensity attack to be an attack that, within the 5 minute attack interval, has it's mean amplitude to be 50% above the actual attack free traffic's mean rate. This is depicted by figure 1 (a). The attacks were synthetically injected between intevals 20-40.

B. High Intensity Attacks

High intensity attack are those attacks whose intensity increases abruptly and reach a peak amplitude within one time interval. High intensity attacks were considered to be attacks that are 250% higher than the peak rate within the 5 minute attack interval. This can be between intervals 20-40 of the figure 1 (b).

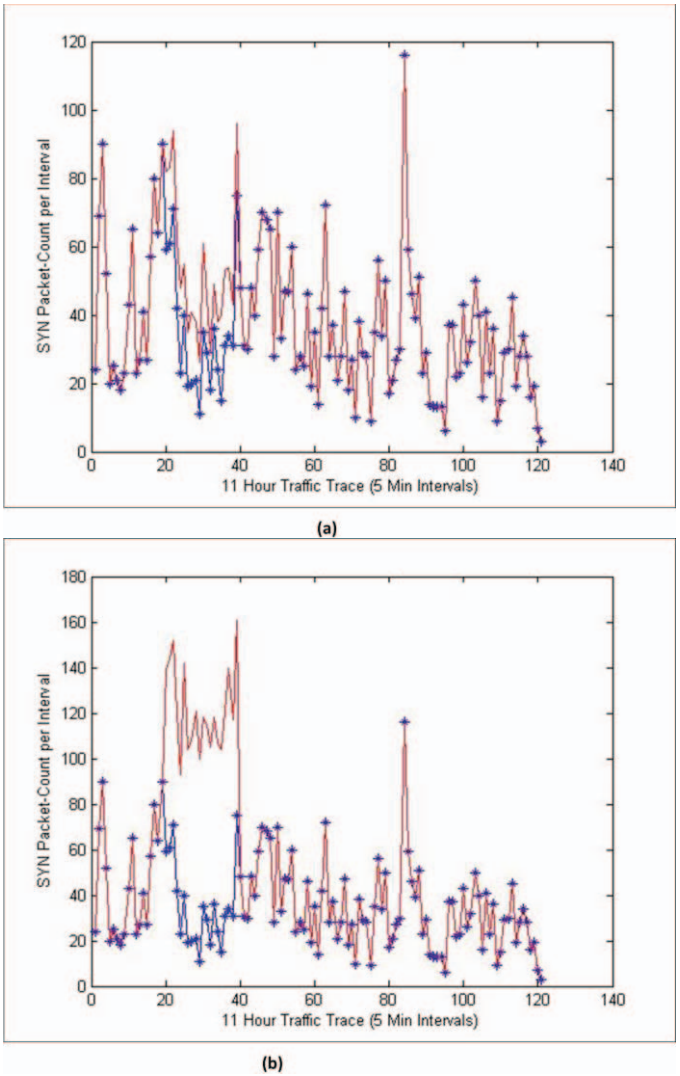

Fig. 1. **(a)** Low intensity attacks; **(b)** High intensity attacks.

IV. Results and Discussions

These experiments were investigating the performance of the EWMA algorithm for both low and high intensity attacks,

testing the following: (1) the effect, on detection-rate, of the alarm threshold α, tuning parameter; (2) the effect, on detection rate, of the EWMA weighting factor β, tuning parameter; (3) the trade-off between detection rate and the false positive rate; (4) the trade-off between the detection rate and detection delay. The result and discussion from the experiments are expanded in the sub-sections that follow.

A. The effect of the alarm threshold (α)

In this section we seek to investigate the effect of the threshold value (α) on the detection rate and the false positive rate. In this part of the experiments, the value of the EWMA factor was held constant $\beta = 0.8$; successive violations $k = 4$; while the threshold value was varied between [0.05; 1.0].

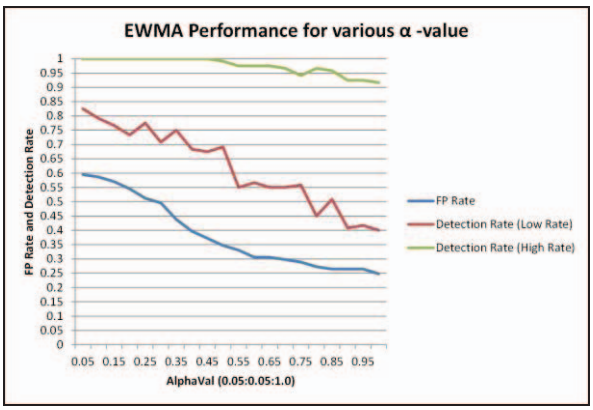

Fig. 2. Detection rate for varied alarm threshold value (α), for low rate attacks and high rate attacks.

The figure 2 depicts the results of the experiments. From the figure 2 it can be observed that for low rate attacks, the detection algorithm yields a detection rate between 60%-85% and a false positive between 30%-60% for values of $0 < \alpha \leq 0.5$. As the value of $0.5 < \alpha \leq 1.0$ the detection rate deteriorate further while the false positive rate improves.

From the figure 2, it can be observed that for high rate attacks and values of $0 < \alpha \leq 0.5$, the detection algorithm yields a 100% detection rate while having a false positive rate between 30%-60%. However, for values of $0.5 < \alpha \leq 1.0$, the algorithm's detection rate deteriorates while the false positive rate improves. It can be seen that there is a trade-off between detection rate and false positive rate.

B. The effect of the EWMA factor (β)

In this section we seek to investigate the effect of the value of the EWMA factor (β) on the detection rate and the false positive rate. In this part of the experiments, the value of the alarm threshold was made constant $\alpha = 0.5$; successive violations $k = 4$; while the value of the EWMA factor was varied between [0.80; 1.0].

From the figure 3 it can be observed that the performance of the detection algorithm performs better at high values of the EWMA factor (β). For low rate attacks, the algorithm reached a 100% detection rate for $\beta \geq 0.98$, while the false positive rate was below 40%. For high rate attacks the detection rate is higher than that of low rate attacks. The detection rate reached 100% for values of $\beta \geq 0.95$ while the false positive rate remained below 40%. In both cases of low and high rate attacks, for the values of $\beta \geq 0.99$, the false positive rate deteriorated substantially.

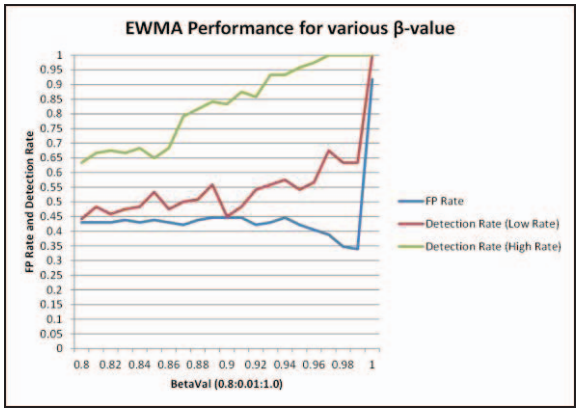

Fig. 3. Detection rate for varied alarm threshold value (α), for low rate attacks and high rate attacks.

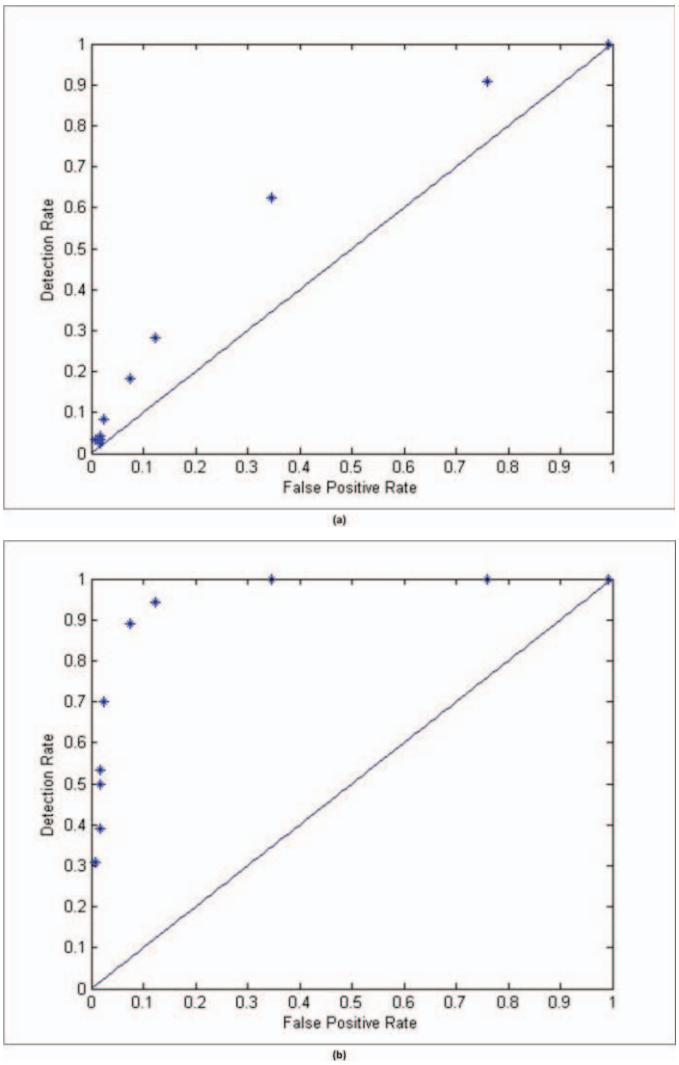

Fig. 4. Receiver Operating Curves for trade-off between FP Rate and Detection Rate for (a) low rate attacks and (b) high rate attacks.

C. *Trade-off between False Positive Rate and Detection Rate*

In this set of experiments we were investigating the trade-off between false positive rate and detection rate. The values for the tuning parameters were as follows: the alarm threshold $\alpha = 0.5$; the EWMA factor $\beta = 0.98$. The value of k was varied from 1-10.

Figure 4 (a) shows results for the experiments simulating low rate attacks. Each point corresponds to a varying value of k. Good operating points on the graph are those points is that are closer to the upper-left corner of the graph. In the case of low rate attacks, an increase in the algorithm's detection accuracy was accompanied by a sharp increase in the false alarm rate. Therefore higher detection accuracy will also result in a higher false alarm rate. This is a performance that is not desired in a detection algorithm. Therefore the EWMA algorithm did not perform very well for cases of low rate attacks.

Figure 4 (b) depicts the performance of the EWMA algorithm in the case of high rate attacks simulation. Most of the operating points are closer to the upper-left corner of the graph. This is also indicative that for a higher detection rate there is a slight increase in the false alarm rate. This is an improved algorithm performance when compared with the low rate attack simulations.

D. *Trade-off between Detection Rate and Detection Delay*

In the next set of experiments we further analyzed the trade-off between detection rate and detection delay. The results are shown in figure 4 below. Detection delay in this case is the average time taken by the algorithm to successfully detect an attack, from the onset of that attack. Each point corresponds to a pair of detection rate and average detection delay. The values for the tuning parameters were as follows: the alarm threshold $\alpha = 0.5$; the EWMA factor $\beta = 0.98$. The value of k was varied from 1-10.

Figure 5 (a) depicts the trade-off between detection rate and average detection delay performance of the EWMA algorithm for low rate attack simulations. From the graph it can be observed that as the detection rate decreases, the accompanying detection delay also increases. The experiment with 100% detection rate had a detection delay that was just below 40s (36.9s) for the low rate attacks simulation.

Figure 5 (b) displays results for the simulations with high rate attacks. The algorithm had an improved performance for high rate attacks. For a 100% detection rate the average detection delays was at 11.75s, 23.83s and 48.67s for various k-values. It can also be observed that for lower detection rate performance the average detection delay also increases.

V. CONCLUSIONS

In this paper we described, analyzed and discussed how the EWMA algorithm can be used for detecting DDoS attacks. In the simulation experiments we investigate how the performance of the algorithm is affected by the tuning parameters (α, β and k). These were efforts to find optimal parameter tuning for best EWMA algorithm performance. We also investigate the trade-off between detection rate and false positive rate; detection rate and average detection delay. Furthermore, the experiments were conducted on real network traffic data with simulations for attack data synthetically generated for various attack intensity, i.e. low rate to high rate attacks.

In these experiments it was found that optimal EWMA parameter tuning for this network traffic was: $\alpha = 0.5$, $\beta = 0.98$ and $k = 3$. Furthermore, it was found that the EWMA algorithm performs well for high rate attacks, however its performance collapses for low rate attacks. This further confirms the findings by the authors in [14].

Ongoing research work will include performance comparison of the EWMA with other anomaly detection algorithm, similar to the work of authors in [18-20]. This will also include efforts to improve current anomaly detection and change detection algorithms by developing algorithms that perform well under various characteristics of attacks.

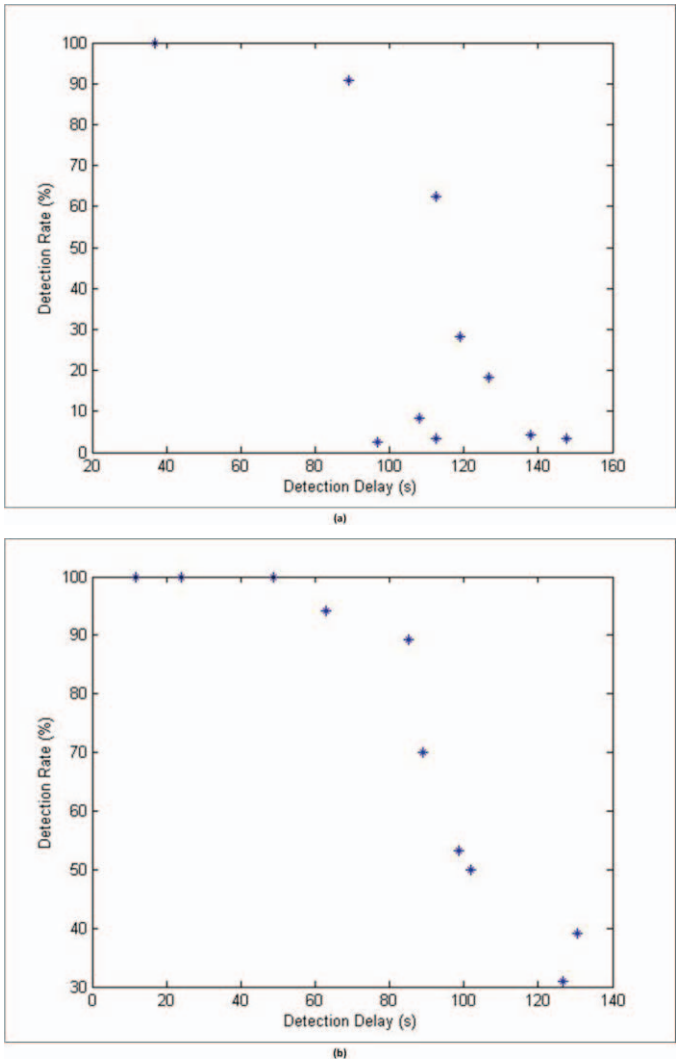

Fig. 5. Graph displaying the trade-off between Detection Rate and average Detection Delay for (a) low rate attacks and (b) high rate attacks.

VI. REFERENCES

[1] A. Lazarevic, V. Kumar and J. Srivastava. "Intrusion detection: A survey," in Managing Cyber ThreatsAnonymous 2005, .

[2] D. K. Bhattacharyya and J. K. Kalita. Network Anomaly Detection: A Machine Learning Perspective 2013.

[3] S. T. Zargar, J. Joshi and D. Tipper. A survey of defense mechanisms against distributed denial of service (DDoS) flooding attacks. Communications Surveys & Tutorials, IEEE 15(4), pp. 2046-2069. 2013.

[4] J. Mirkovic and P. Reiher. A taxonomy of DDoS attack and DDoS defense mechanisms. ACM SIGCOMM Computer Communication Review 34(2), pp. 39-53. 2004.

[5] C. Douligeris and A. Mitrokotsa. DDoS attacks and defense mechanisms: Classification and state-of-the-art. Computer Networks 44(5), pp. 643-666. 2004.

[6] C. V. Zhou, C. Leckie and S. Karunasekera. A survey of coordinated attacks and collaborative intrusion detection. Comput. Secur. 29(1), pp. 124-140. 2010.

[7] A. G. Tartakovsky, B. L. Rozovskii, R. B. Blazek and H. Kim. A novel approach to detection of intrusions in computer networks via adaptive sequential and batch-sequential change-point detection methods. Signal Processing, IEEE Transactions On 54(9), pp. 3372-3382. 2006.

[8] A. G. Tartakovsky, A. S. Polunchenko and G. Sokolov. Efficient computer network anomaly detection by changepoint detection methods. Selected Topics in Signal Processing, IEEE Journal Of 7(1), pp. 4-11. 2013.

[9] M. Basseville and I. V. Nikiforov. Detection of Abrupt Changes: Theory and Application 1993104.

[10] H. V. Poor and O. Hadjiliadis. Quickest Detection 200940.

[11] G. Carl, G. Kesidis, R. R. Brooks and S. Rai. Denial-of-service attack-detection techniques. Internet Computing, IEEE 10(1), pp. 82-89. 2006.

[12] S. Roberts. Control chart tests based on geometric moving averages. Technometrics 1(3), pp. 239-250. 1959.

[13] P. Machaka and F. Nelwamondo. "Data mining techniques for distributed denial of service attacks detection in the internet of things: A research survey," in Data Mining Trends and Applications in Criminal Science and Investigations 2016, .

[14] V. A. Siris and F. Papagalou. Application of anomaly detection algorithms for detecting SYN flooding attacks. Comput. Commun. 29(9), pp. 1433-1442. 2006.

[15] N. Ye, C. Borror and Y. Zhang. EWMA techniques for computer intrusion detection through anomalous changes in event intensity. Qual. Reliab. Eng. Int. 18(6), pp. 443-451. 2002.

[16] N. Ye, S. Vilbert and Q. Chen. Computer intrusion detection through EWMA for autocorrelated and uncorrelated data. Reliability, IEEE Transactions On 52(1), pp. 75-82. 2003.

[17] G. Münz and G. Carle. Application of forecasting techniques and control charts for traffic anomaly detection. Presented at Proc. 19th ITC Specialist Seminar on Network Usage and Traffic, Berlin, Germany. 2008, .

[18] P. Machaka, A. McDonald, F. Nelwamondo and A. Bagula. Using the cumulative sum algorithm against distributed denial of service attacks in internet of things. Presented at International Conference on Context-Aware Systems and Applications. 2015, .

[19] P. Machaka and A. Bagula. "An investigation of scalable anomaly detection techniques for a large network of wi-fi hotspots," in Scalable Information Systems 2014, .

[20] P. Machaka. "Drought monitoring: A performance investigation of three machine learning techniques," in Context-Aware Systems and Applications 2014.