**International Journal of Computing and Applications** | **ISSN: 0973-5704**
Volume 13, Number 1, (January-June 2018), pp 83-90


# DETECTING DDOS

## *A Novel Method based on Network Traffic History and Chaos Theory*

**Rohit Wanchoo[1], Bhanu Pratap Chib[2], Nikita Raina[3] and Sahil Koul[4]**

*[1-4]Department of CSE, Model Institute of Engineering and Technology, Jammu, India.*
*Email: [1]rohit26013.cse@mietjammu.in, [2]bhanu27213.cse@mietjammu.in, [3]nikita.cse@mietjammu.in, [4]sahil24313.cse@mietjammu.in*

***Abstract:*** A crippling attack on the traditional architecture of internet today is Denial of service or simply DoS attack. But the most notorious form of DoS attack is the Distributed Denial of Service attack also known as DDoS attack which adds the "many to one" dimension that makes these attacks more difficult to prevent. It compromises the availability of the target network or system by the distributed attacks which rely on recruiting a fleet of compromised hosts that unwittingly join forces to flood the victim server. This paper focusses on proposing a novel method of detection of DDoS attacks by analyzing the network traffic history and combining the forces of this method with the AR Time Series Model and Chaos Theory to form a faster and more stringent Network Anomaly Detection Algorithm.

***Keywords:*** Distributed Denial of Service (DDoS), network history analysis, Chaos Theory, Time Series Model, Autoregressive (AR) Model, Lyapunov exponent.

## 1. INTRODUCTION

A denial of service attack is characterized by an attempt by a hacker to prevent authorized users from using the resources. An attacker may attempt to flood a network and thus reduce a legitimate user's capacity to access it, preventing usage of a service, or disrupt service to a specific system or a user [1].

A Distributed Denial of Service (DDoS) differs in the aspect, that the attacker does not directly attack a victim. Instead he searches and hacks a various number of insecure computers, which are known as Zombies. These zombies then collectively form a botnet to perform a DoS attack on the victim.

There are many different types of DDoS Attacks, as classified in [2, 16], that may be attempted on any victim. But the packet-flooding attack is most commonly type used. In this type of attack, an attacker sends a large number of Transmission Control Protocol (TCP), User Datagram Protocol (UDP) or Internet Control Message Protocol (ICMP) to the victim. As per Peng et. al., [17], escaping from this attack is very difficult for the victim because of following two reasons. Firstly, the collection of zombies in the botnet is very large. This produces a huge rush of traffic which will eventually flood the victim. Second, the zombies also spoof their address under attackers influence. This makes it very difficult to trace back the attack traffic. In this paper, we will be discussing a method to prevent this type of DDoS attack traffic.

Distinguishing a DDoS attack traffic (hereafter, called just as attack traffic), from a normal bursty legitimate traffic is a very difficult task. To achieve this, we are going to study the traffic history pattern incoming to the system and use this traffic history pattern to determine whether the traffic is a attack or legitimate bursty traffic. Also we are going to use AR Time Series Model and Chaos theory to achieve the same.

A time series is a data taken at discrete values of time. The data points are then indexed or listed or graphed in time order. There are various different models by which a time series can represent different stochastic

processes. We will be using the Autoregressive model to process and predict the network traffic data [18].

Also we will also be using Chaos theory to determine the state of the system. The Chaos theory is a branch of mathematics which studies the dynamics of a system, which is very susceptible to initial conditions (and hence appear to be random systems). By studying the Lyapunov exponent of the system (that is the divergence of the predicted data from the actual data) we can tell whether the system is chaotic or not. If the system is chaotic, the lyapunov exponent remains positive, which states that the data is chaotic/random. If lyapunov exponent is negative or zero (both the cases will be discussed later briefly), the system is not in a chaotic state.

The method proposed in this paper can be summarized as follows: In the first phase, we analyze the incoming packet traffic to see whether packet traffic is giving a suspicion of a DDoS attack. If the first phase gives a suspicion about a DDoS attack, the second phase processes the traffic using the Network Anomaly Detection Algorithm (NADA), based upon AR Time series model and Chaos Theory.

The benefit of using this two phase system as proposed in this paper is that, first the NADA algorithm cannot work all the time as it will waste computational resources as it requires to solve some mathematical equations, which may not be suitable for all devices (especially some low end systems or a IOT device). Secondly, the network traffic might increase tremendously during a DDoS attack, that we might even not require a NADA algorithm to determine a DDoS attack. Just by observing the change in traffic we can declare a DDoS attack and take the needed countermeasures. On the other hand, sometimes DDoS attack might be difficult to judge by just traffic history analysis. Thus, our proposed method covers the best of both worlds.

The rest of the paper is organized as follows: In section II, a detailed literature survey regarding NADA algorithm using AR Time Series Modelling and chaos theory, is discussed. Section III discusses the proposed approach and its relevant components. Final results using proposed approach are given in section IV and the paper is concluded and future directions are given in section V and VI.

## 2. RELATED WORK

Due to their distributed nature, a DDoS attack is very difficult to detect, as the origins of the traffic might be miles apart, from various network and geographical locations. For this reason, the DDoS is still a very powerful attack to bring down a network.

The authors in [20] have categorized a DDoS attack into various types of attacks. These are categorized as: Network Device Level, OS Level, Application Level Attacks, Data Flood attacks and Protocol Level attacks.

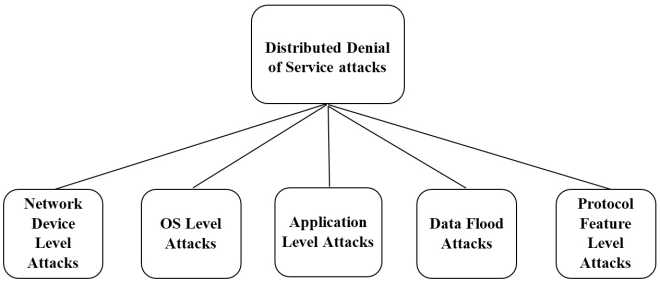


**Figure 1: Types of DDoS attacks**

There are many measures that can be taken to prevent or stop DDoS attacks at various levels of the network. In [21] the authors have mentioned many such detection and prevention techniques to tackle a DDoS attack. However, as these attacks are becoming more and more sophisticated and due to the release of many attacking tools which allow some ordinary person to perform a DDoS attack, these methods are fast becoming ineffective and outdated.

To overcome these problems, the authors in [22] have developed a new method to DDoS attack. This method uses the principles of time series modeling such as AR, ARMA, ARIMA, and FARIMA etc. for analyzing and forecasting the network traffic. Then by find the prediction error and finding the lyapunov exponent (from chaos theory), we can detect whether the incoming traffic is a legitimate traffic or an attack traffic.

## 3. PROPOSED WORK

The architecture (Figure 1) of the proposed work is explained as follows:

1. The network traffic analyzer will constantly monitor the network traffic and will keep the track of the incoming network traffic.
2. The analyzer will analyze the current network traffic and compare it with to that of the network traffic history to see if there is any large increase in the network traffic.
3. Based on the comparative difference between incoming network traffic to that of the traffic history (stored in the traffic history database), the traffic analyzer will decide that whether the traffic is a legitimate traffic or a DDoS attack traffic.
4. Until the traffic analyzer is able to distinguish whether the traffic is legitimate or an attack traffic, the NADA algorithm is not used, hence saving the time and computing resources.
5. If the traffic analyzer will not be able to take a concrete decision to whether as the traffic is attack traffic or normal one, the traffic analyzer will pass control to the NADA (shown in red in Figure 1).
6. NADA first collects network packet and flow information and pre-processes it by cumulatively averaging the sequence, to suppress the network traffic.
7. Finally we do a network traffic analyzation to see if the traffic has any anomaly (i.e. if the traffic is a legitimate burst traffic or a DDoS attack traffic) based on AR time series model and Chaos theory, as shown in equations (ii) through (v).
8. If a DDoS attack is detected, the suspected packets are dropped until the traffic becomes normal again.
9. After this the packets will reach the server.

Let us suppose that in the starting our DDoS detection system has just been installed on the network/server. The system will be fresh in the starting and will not have any history of the traffic on that network. We will then start to record the network traffic history at a regular interval of 15 minutes and store the traffic data along with the time of day and day of the week in the database. The traffic data will be stored as the number of packets that arrive to server in these 15 minutes. The database table will be like this:

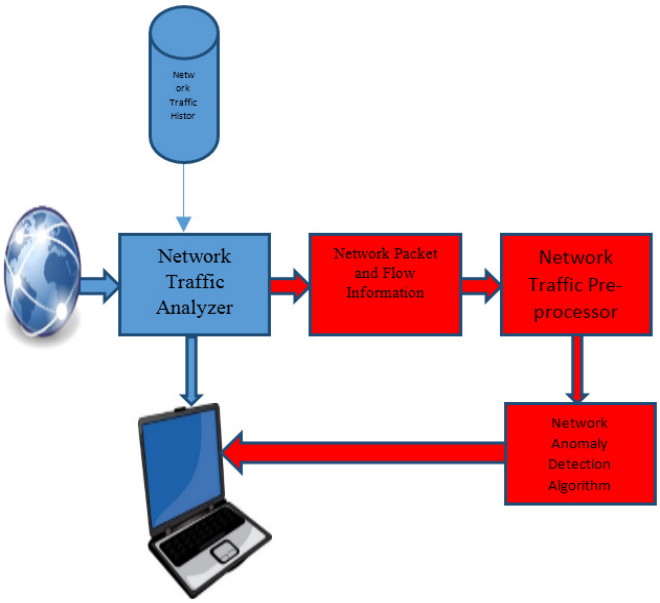


**Figure 2: The architecture of the proposed method.**

**Table 1**
**Database Structure**

| *Time* | *Mon* | *Tue* | *Wed* | *Thu* | *Fri* | *Sat* | *Sun* |
|---|---|---|---|---|---|---|---|
| 00:00 | 345 | 520 | 586 | 578 | 800 | 790 | 850 |
| 00:15 | 350 | 500 | 600 | 620 | 750 | 800 | 1000 |
| 00:30 | 335 | 486 | 602 | 634 | 729 | 654 | 1050 |
| 00:45 | 351 | 492 | 590 | 640 | 744 | 635 | 1053 |
| 01:00 | 374 | 488 | 583 | 642 | 750 | 662 | 1044 |
| 01:15 | 342 | 477 | 569 | 626 | 743 | 521 | 1037 |
| 01:30 | 328 | 470 | 562 | 611 | 724 | 699 | 1021 |
| 01:45 | 326 | 440 | 500 | 601 | 700 | 708 | 800 |
| 02:00 | 300 | 380 | 458 | 582 | 683 | 717 | 749 |
| 02:15 | 314 | 356 | 402 | 570 | 664 | 688 | 812 |
| 02:30 | 286 | 340 | 490 | 509 | 670 | 644 | 899 |
| 02:45 | 294 | 300 | 468 | 515 | 687 | 631 | 967 |
| 03:00 | 273 | 280 | 400 | 518 | 644 | 622 | 932 |
| 03:15 | 286 | 290 | 389 | 494 | 605 | 634 | 901 |
| 03:30 | 249 | 280 | 387 | 422 | 555 | 642 | 876 |
| 03:45 | 264 | 274 | 350 | 387 | 514 | 604 | 867 |
| ⋮ | | | | | | | |

The average field in the end will contain the average packets that a server receives on that particular day. After one week of installation, the system will have a complete history of network traffic for the previous week and then we can start to analyze the network traffic.

To analyze the traffic we will count the number of packets reaching the server for 15 minutes. Then we will compare the traffic volume to the day of the week and time of the day which is same as the current day and time. Means, if the current network traffic has been monitored from 00:00 to 00:15 hours on Monday we will compare it with the 00:15 column and row Monday in the database. If the current traffic volume is either lesser or not too large than the one we are comparing it to in the database, then we can safely say that the traffic coming to the server is legitimate traffic and also if this value is larger than the previously recorded value we will update the value in the database.

If the value is too large then such a traffic might be a DDoS attack. In this case we will analyze the change in the traffic values across the week passed. To do this we will find the change in traffic between first two days and then store this value. Similarly, we will find the change in traffic values across all the consecutive days and store them (See Table 2). Of these values, we will drop the negative values. Then we will find the average change in traffic across the week. This we will compare with the change in traffic of the current day to the previous day.

For this system one of the, three scenarios may arise:

1. If the incoming traffic is lesser than or comparable to the history of the traffic, the traffic can be safely declared as a legitimate traffic. The server can be safely take this load as we know it will have already taken it.
2. If the incoming traffic is greater than the history traffic, we will analyze the incremental change in the traffic over the past week. Let us understand this using an example:

   Consider the traffic values as mentioned in row 2 of Table 1. As we can see the traffic value stored in the database is 350 at 00:15 hrs. But over the course of the week, traffic at this time period is increasing, as is relevant from Table 1. Thus it will be wrong to declare that a incoming traffic value of say 2500 packets will be a DDoS attack. Thus, we will find the incremental change in the network traffic for this period, as depicted in Table 2.

   After doing this step, we will find the average of the incremental change, as shown below.

   Average difference of traffic

   = (150 + 250 + 250 + 500 + 200 + 300)/6

   = 297 which is comparable to 310

   As we can see from the above example, the average change in the traffic over a period of week is comparable to the change in traffic volume on Monday. Hence, this will not be a DDoS attack.

**Table 2**
**Example showing traffic values of each day in a week**

| *Day* | *Traffic* | *Incremental Change in Traffic* |
|---|---|---|
| Monday | 276 | |
| Tuesday | 500 | 224 |
| Wednesday | 750 | 250 |
| Thursday | 1030 | 280 |
| Friday | 1500 | 470 |
| Saturday | 1790 | 290 |
| Sunday | 2060 | 270 |
| Monday | 2370 | 310 |

If the traffic would not be increasing throughout the week, and will increase abruptly in a single day, it will most probably be a DDoS attack. In that case, we will detect whether this bursty traffic will be a legitimate traffic or a DDoS attack traffic by detecting anomalies in the network traffic by Network Anomaly Detection Algorithm (NADA)

Now if we are not able to resolve whether the traffic is a DDoS attack or not (i.e. 3$^{rd}$ case occurs), we will use the method of traffic prediction. The **basic rule of** this method is that we predict the network traffic and then compare it with the original traffic

value. This difference will be called as the prediction error. If we consider this prediction error as chaotic, then we can apply the chaos theory to find nature of traffic.

We use AR Time Series model to predict the network traffic. However, in order to bring stability to the models, we sample the network traffic after collecting the network packets and flow information.

Let $r_n$ be the different states of network traffic. Hence we get a sequence as follows:

$$s_1, s_2, \ldots, s_i, \ldots, s_n \quad \text{(i)}$$

where, $s_i$ is the state of traffic to be predicted.

We can use the average of (i) over time period $t_i$ to make the network traffic stable for accurate prediction, that is:

$$Z_i = (s_1 + s_2 + s_3 + \ldots + s_i)/t_i \quad \text{(ii)}$$

Using autoregressive (AR) model, we can predict $z_i$

$$V_j = \sum_{i=1}^{m} a_I Z_{j-I} \quad \text{(iii)}$$

Hence the sequence $z_i$ can be generated from (ii) and (iii) as follows:

$$z_i = t_i Z_i - t_{i-1} Z_{i-1}$$

where, $z_i$ is the prediction of $s_i$

Hence the prediction error can be found out as:

$$\Delta z_i = s_i - z_i$$

$$s_i = z_i + \Delta z_i$$

We now consider that the sequence $\{z_i\}$ represents normal traffic, whereas $\{\Delta z_i\}$ represents changed traffic due to additional, bursty legitimate or attack traffic.

Also, we will assume that $\{\Delta z_i\}$ behaves 'chaotically' when new traffic enters the system. By making this assumption, we analyze the mean exponential rate of $\{\Delta z_i\}$ which is the divergence between normal traffic $\{z_i\}$ and the real traffic $s_n$. We can then use the Lyapunov exponent to observe change in traffic to see whether it is attack traffic

$$L_i \approx \{\ln[\Delta z_i / \Delta z_0]\}/t_i$$

Now for the value of Lyapunov exponent, one of the three cases arises:

**Case 1:** If $L_i > 0$, that is, the Lyapunov exponent is positive, the change in traffic is chaotic $\{\Delta z_i\}$. This means that the change in traffic $\{\Delta z_i\}$ is caused by new legitimate traffic entering the system.

**Case 2:** If $L_i = 0$, there is no divergence in the network traffic and the predicted network traffic. This means $\{\Delta z_i\}$ is in a steady state that is, network traffic is constant.

**Case 3:** If $L_i < 0$, the $\{\Delta z_i\}$ is steady and not random or chaotic (which it should be in case of a legitimate bursty traffic). This means that the change is caused by a DDoS attack traffic that may be introduced by an attacker affecting the system.

**Table 3**
**DDoS Detection Algorithm**

| | |
|---|---|
| **Step 1** | Get the volume of incoming traffic for a pre-specified period. |
| **Step 2** | Compare the current traffic volume with that of the traffic history. |
| **Step 3** | If the incoming traffic volume is determined by the traffic analyzer to be within limits, no further steps need to be taken |
| **Step 4** | If the traffic analyzer is suspicious we have to analyze the traffic by using NADA algorithm. |
| **Step 5** | Fetch the packet flow information. |
| **Step 6** | Stabilize (pre-process) the network traffic with (ii). |
| **Step 7** | Predict the traffic with (iii) and (iv). |
| **Step 8** | Find the prediction error $\Delta z_i$ and then detect any abnormal network traffic as explained. |
| **Step 9** | Use this data to train neural networks to detect DDoS |

## 4. IMPLEMENTATION AND OBSERVATION

We used Riverbed Modeler to simulate three kinds of network traffic.

The first was the normal non bursty traffic increasing over a week and then becoming somewhat bursty in nature. This kind of traffic did not required to use the NADA phase as it was simply resolvable by comparing it with the history. (See Figure 2)

The second simulation was of a traffic of high bursty nature. This traffic varies rapidly over the period

of time and cannot be compared using traffic history. The Figure 3 depicts this traffic.

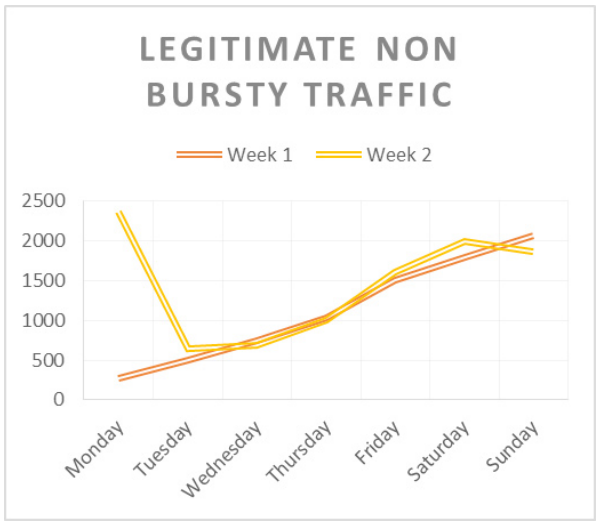


**Figure 3: Graph showing a non bursty legitimate traffic.**

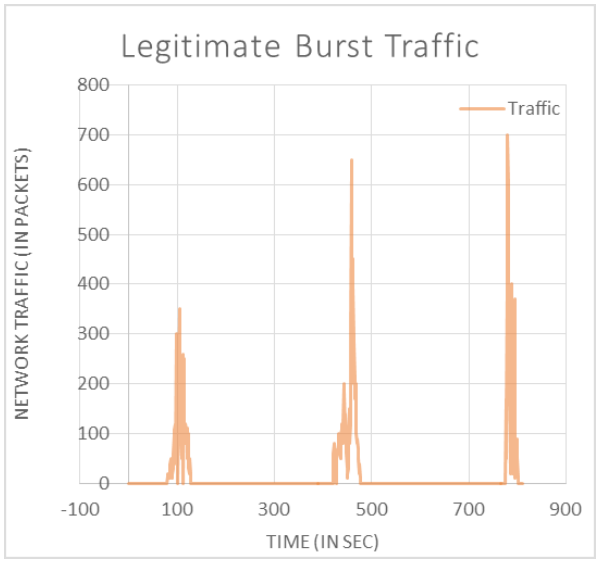


**Figure 4: Graph showing a legitimate bursty traffic.**

The third kind of traffic simulation was of a DDoS attack traffic. Figure 4 shows a DDoS attack traffic. This traffic goes on increasing over the time period, until it is either stopped or the server crashes.

Graph in Figure 5 shows the traffic chaos pattern for a normal legitimate burst traffic (Figure 3). The Lyapunov exponent Li is plotted against the time elapsed on the network. It can be observed that the Lyapunov exponent is positive for most of the time. This indicates that there is divergence between the current traffic and predicted traffic value. So we can say that the system is unpredictable or chaotic. Hence

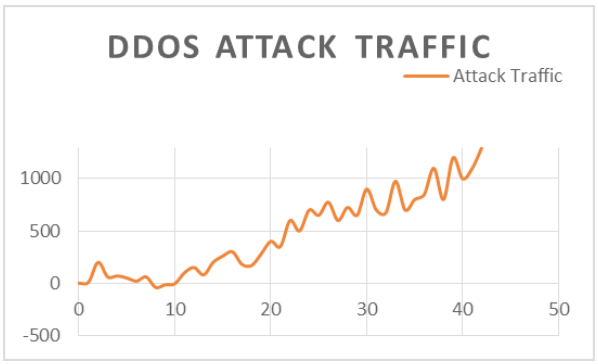


**Figure 5: Graph showing a DDoS attack traffic.**

we can safely say that the traffic is a normal legitimate traffic. Even if the lyapunov exponent had become zero, then it would mean that there is no change in the divergence of the system, in other words there is no traffic entering the system.

The graph in Figure 6 shows the traffic chaos pattern for a DDoS attack traffic. The Lyapunov exponent Li is plotted against the time elapsed on the network. It can be observed that the Lyapunov constant is negative for most of the time. This means there is no

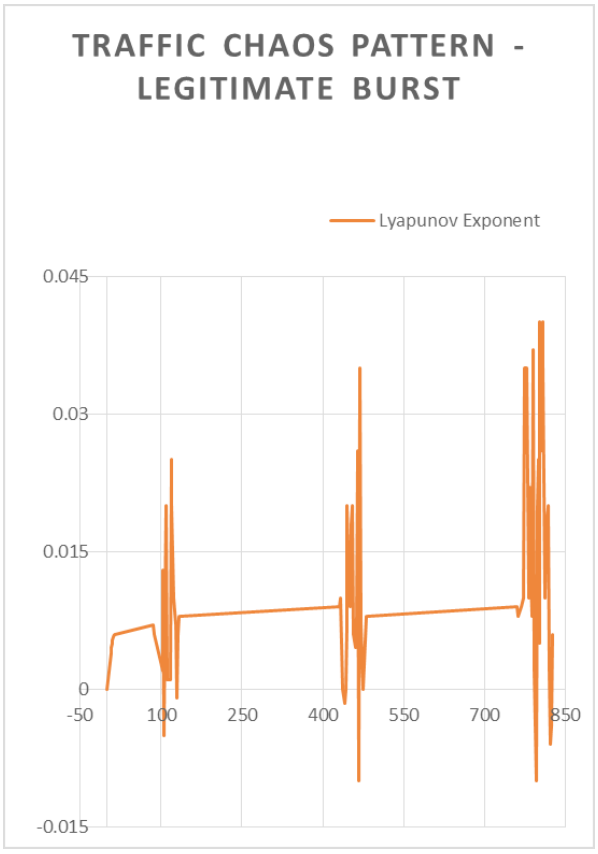


**Figure 6: Graph showing traffic chaos pattern for a bursty traffic.**

divergence in the current and predicted traffic values of the system, which means that the traffic values in the system is predictable and not random or chaotic. Thus the system has gone from a chaotic to a predictable state. This indicates that the incoming traffic is some value changing with some constant rate which it is in case of a DDoS attack traffic (Figure 4).

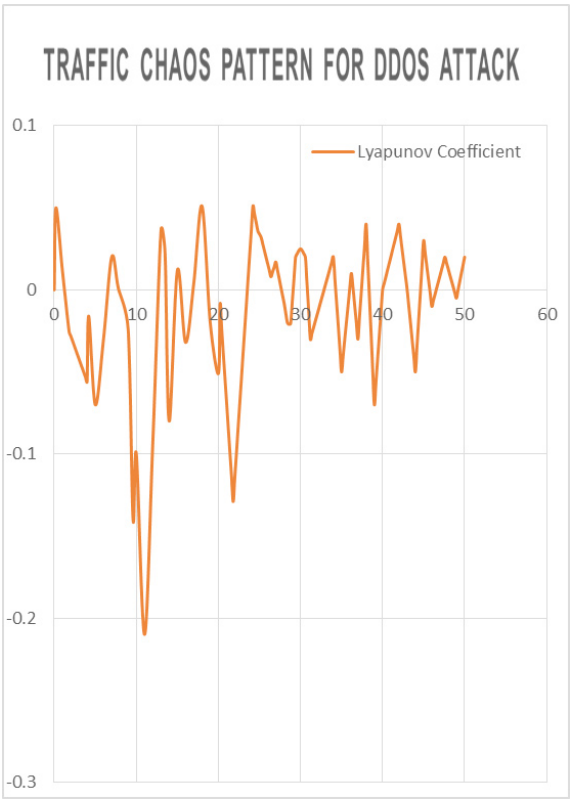


**Figure 7: Graph showing traffic chaos pattern for a DDoS attack.**

## 5. CONCLUSION

The method proposed in this paper is successful in detecting DDoS attacks. It is able to differentiate a DDoS attack based on the history of the incoming traffic and also by using prediction models and chaos theory. The method is much faster and light weight as compared to the simple NADA algorithm but can also detect bursty traffic nature and differentiate it from a DDoS attack.

## 6. FURTHER SCOPE

We can use back-propagation trained neural networks to detect the DDoS attack by using the sample of abnormal traffic. Thus we can increase the detection efficiency up to 96.5%. We can also use a probability distribution function to tell the probability of a particular traffic volume on day instead of just comparing the traffic volumes. This can further increase the detection efficiency of our model.

## *References*

[1] Felix Lau, Stuart H. Rubin, Michael H. Smith, jiljana Trajkovic, "Distributed Denial of Service Attacks",in: 2000 IEEE INTERNATIONAL CONFERENCE ON SYSTEMS, MAN & CYBERNETICS 2000, Nashville, Tennessee, USA.

[2] Douligeris C. and Mitrokotsa A., "DDoS Attacks and Defense Mechanisms: Classification and State of the Art," Computer Journal of Networks, vol. 44, no. 5, pp. 643-666, 2004.

[3] Galli P., "DoS Attack Brings down SUN Grid Demo," http: //www.eweek.com/article2/0,1895,1941574, 00.asp, 2007.

[4] Garber L., "Denial of Service Attacks Rip the Internet," Computer Journal of IEEE, vol. 33, no. 4, pp. 12-17, 2000.

[5] Gibson S., "The Strange Tale of the Denial of Service Attacks Against GRC.COM," http://grc.com/dos/grcdos.htm, 2007.

[6] Gonsalves C., Akamai DDoS Attack Whacks Web Traffic, http: //www. eweek.com/article2/0,1895, 1612739,00. asp, 2007.

[7] Gordon A., Loeb P., Lucysgyn W., and Richardson R., CSI/FBI Computer Crime and Security Survey, CSI Publications, 2006.

[8] Handley M., "Internet Architecture WG: DoS Resistant Internet Subgroup Report," onlineathttp:// www.communications.net/object/download/1543/ doc/mjh-dos-summary.pdf. 2007.

[9] Haris B. and Hunt R., "TCP/IP Security Threats and Attack Methods," Computer Journal of Communications Review, vol. 22, no. 10, pp. 885-897, 1999.

[10] Haymarket Media, "Al-Jazeera Hacked in DoS Attack," http://www. itnews. com.au /newsstory. aspx?CIaNID=17603, 2007.

[11] Howard J., "An Analysis of Security Incidents on the Internet," PhD Dissertation, Carnegie Mellon University, 1997.

[12] ITworld. com, "CERT Hit by DDoS Attack for a Third Day," http://security.itworld.com/4339/IDG010524CERT2/pfindex.html, 2007.

[13] Kumar K., Joshi R., and Singh K., "An Integrated Approach for Defending Against Distributed Denial of Service Attacks," http://www.cs. Iitm.ernet.in/~iriss06/paper.html, 2002.

[14] McAfee, "Personal Firewall," http://www.mcafee.com, 2003.

[15] McCue A., "Bookie Reveals," http://software.silicon.com/security/0, 39024655, 39121278, 00.htm, 2007.

[16] Mirkovic J. and Reiher P., "A Taxonomy of DDoS Attack and DDoS Defense Mechanisms, "Computer Journal of ACM SIGCOMM, vol. 34, no. 2, pp. 39-53, 2004.

[17] Peng T., Leckie C., and Ramamohanarao K., "Survey of Network Based Defense Mechanisms Countering the DoS and DDoS Problems," Computer Journal of ACM Computing Surveys, Vol. 39, No. 1, pp. 123-128, 2007.

[18] T. Vafeiadis, A. Papanikolaou, C. Ilioudis, and S. Charchalakis, "Realtime network data analysis using time series models," Simulation Modelling Practice and Theory, pp. 173–180, 2012.

[19] A. Chonka, J. Singh, and W. Zhou, "Chaos theory based detection against network mimicking DDoS attacks," IEEE Commun. Lett., Vol. 13, No. 9, pp. 717–719, 2009.

[20] Muhammad Aamir and Mustafa Ali Zaidi, "DDoS Attack and Defense: Review of Some Traditional and Current Techniques", SZABIST, Karachi, Pakistan.

[21] Stephen M. Specht and Ruby B. Lee," Distributed Denial of Service: Taxonomies of Attacks, Tools and Countermeasures", 17th International Conference on Parallel and Distributed Computing Systems, 2004 International Workshop on Security in Parallel and Distributed Systems, pp. 543-550, September 2004.

[22] Yonghong Chen, Xinlei Ma and Xinya Wu," DDoS Detection Algorithm Based on Preprocessing Network Traffic Predicted Method and Chaos Theory", IEEE Communications Letters, Vol. 17, No. 5, May.