Spectral Fractal Dimension Trajectory to Measure Cognitive Complexity of Malicious DNS Traffic

Muhammad Salman Khan, Sana Siddiqui, Ken Ferens, and Witold Kinsner
Dept. of Electrical and Computer Engineering, University of Manitoba, Winnipeg, MB, Canada
muhammadsalman.khan@umanitoba.ca, siddiqu5@myumanitoba.ca, ken.ferens@umanitoba.ca, witold.kinsner@umanitoba.ca

Abstract—Internet traffic exhibits long range dependence (persistence), scale invariance and self-similarity or self-affinity which are the known characteristics of fractals. Moreover, these characteristics of fractals can be extracted and quantified from an internet data time series using non-integer dimensions (fractal dimensions). The notion of cognitive complexity is also very well represented by the fractal dimensions, e.g., high value of fractal dimension of an object implies that the complexity of this object is higher than the one with lower fractal dimension. In addition, a multifractal object is more complex than a monofractal object and this can also be characterized to identify the degree of complexity. In this work, we have shown that the complexity introduced by distributed denial of service (DDoS) attack packets in DNS (Domain Name System) traffic is higher than the complexity of DNS traffic with no DDoS attack packets. A power spectrum density of the data series was used to calculate the spectral fractal dimension, and the performance of the proposed algorithm is validated using mathematical fractal Brownian motion process (fBm) and the real data sets. A sequence of spectral fractal dimension measurements of the time series (also known as a trajectory of spectral fractal dimension measurements or spectral fractal dimension trajectory (SFDT)) was generated to show the changing complexity of the series in time domain.

Keywords—Denial of service, Domain Name System (DNS), cyber threats, complexity, multifractal, power spectrum density, time series, spectral fractal dimension trajectory (SFDT), variance fractal dimension trajectory, malicious traffic.

I. Introduction

Certain cyber attackers exploit the vulnerabilities of DNS (Domain Name System) protocol to disrupt DNS services using various methods. Distributed denial of service (DDoS) DNS amplification attack is one of such methods which uses legitimate DNS servers to piggy back and amplify the payload of DNS packets. There is no useful information contained in such packets and they reduce the available bandwidth of the network. The attack is launched by sending a broadcast message to the legitimate computer nodes after manipulating the source and destination IP addresses of the message such that the receiver nodes receive these messages from an authentic node which acts as a piggy back node for the attacker. The victim nodes receive the DNS traffic continuously from the DNS servers without generating any DNS request [1] [2]. Since many authentic servers send these DNS response packets to the victim node, continuously, the resulting persistent high rate of traffic overwhelms the victim node. The victim node becomes unable to process the packets received at the rate they are being sent, and this causes the victim node to loose/drop packets, including packets received from other legitimate sources. Consequently, the victim node is unable to process other legitimate network requests, thus resulting in a denial of DNS service of those legitimate network requests. Furthermore, the attacker cannot be traced because the attack is launched using authentic source nodes and the attacker remains anonymous.

II. Literature Review

In order to detect DNS denial of service attacks with high accuracy, it is required to devise a solution that can differentiate accurately between normal and anomalous packet flows. Signature based methods cannot accurately detect DNS attacks, because there is no known signatures of DNS attack packets that can be used to differentiate between normal and attack packets. In other words, DNS attack packets resemble authentic DNS packets. However, there are various methods in the literature, which attempt to detect DNS DDoS amplification attacks. The authors in [2] describe a method of mapping and monitoring the DNS

ISBN: 1-60132-445-6, CSREA Press ©

mechanism of requests and responses to detect anomaly in the packet flows. This method shows better results in detection, but is limited due to the scaling issues in a large network. Moreover, it is useful for local DNS servers only. In [3], the authors utilized hardware based Bloom filters to analyze DNS packets to detect DNS amplification attacks. Also, as mentioned in [4] [5], there are location based and time based methods to detect DNS DDoS amplification attacks. There are various methods to detect these attacks and include packet based payload analysis and node based collaborative techniques. Fractal based estimation techniques are gaining popularity in anomaly detection algorithms due to their ability in looking at traffic patterns at multiple scales simultaneously. For example, authors in [6] proposed a correlation based fractal dimension for the detection of DDoS attacks using DARPA data set. They used a supervised learning mechanism to detect changes in fractal features. Cognitive security [7] and cognitive computing algorithms [8] [9] [10] [11] have shown their ability to mimic the apparent cognitive process that humans use to classify normal and anomalous traffic. In this work, the authors hypothesized that human's mental model to process information and classify normal and anomalous flows in a packet stream does involve an evaluation of the level of complexity of these flows. In the absence of attack signatures, cyber security experts utilize a cognitive model to differentiate between the complexities of a normal traffic flow from a malicious flow which results in further analysis of the malicious flow for a possible detection of a new threat. Malicious flows will have different level of complexity than that of legitimate traffic flow.

III. FRACTAL ANALYSIS

Typically, time series or data streams are analyzed using single(mono) scale analysis where any analysis i.e. statistical, spectral and/or transformation is performed on the data time series with equal sampling intervals. Multiscale analysis refers to analyzing the data series on multiple level of resolutions such that same data series is analyzed multiple times. Wavelet analysis is an example where independent time and frequency analysis is done that is equivalent to studying the time series at multiple resolution scales. Fractal based multiscale analysis is a revolutionary idea [12] [13] where a relationship is found between multiple resolution levels known as fractal dimension. This is akin to multiscale analysis simultaneously (and not independently as in wavelets) and finding how these multiscale levels are related which is equivalent of finding self-similarity. Multifractal analysis extracts the nature of fractality (fractional, or singularity, or non-integer behavior) of the object i.e. data time series. Mono-scale analysis is appropriate for any time series but cannot describe the relationship among various level of resolutions or scales. If the time series is self-affine then single scale analysis is not sufficient and multifractal analysis is required to extract the features (relationship among scales) [14]. In the science of cognitive detection, this is equivalent of measuring the complexity of the time series [15]. If the series is not very complex, then the value of fractal dimension is low and/or it may show monofractal behavior, while high values of fractal dimensions represent increasing level of complexity and/or the time series will have multifractal behavior. Fractal dimensions are always bounded by an upper value known as embedded dimension. Value of the embedding dimension represents the number of integer dimension of a time series. For example, for a single dimension time series (i.e. time series representing only one parameter such as count series of a variable), fractal dimension is bounded between 1 and 2 [14] [16].

IV. SPECTRAL FRACTAL DIMENSION TRAJECTORY

Spectral fractal dimension analysis, which is an extension of variance fractal dimension analysis [14], is a class of statistical/information based fractal dimension analysis where second order frequency analysis using power spectral density is performed at multiple scales and a relationship among those scales is found simultaneously using log-log relationship of multiple scales [14] [17]. Spectral fractal dimension analysis provides fractal dimension within the embedding topological dimensions of an object. For example, in this work, DNS packet count time series is a single dimension (1D) time series of packet count and therefore, the lower limit of topological dimension is 1 and upper limit is 2. For a 1D time series, spectral fractal dimension of 2 represents that the time series is generated from a statistical pink noise process while spectral fractal dimension of 1 represents a black noise process [18].

If a time series is a self-similar (or self-affine) fractal, the power spectrum density satisfies the following power law [17] [18] [19]:

$$\mathrm{P(f,T)} \sim (\frac{1}{f})^{d} \quad (1)$$

$\mathrm{P(f,T)}$ represents the power spectrum density of the time series as a function of the frequency and the window time T of the time series over which power spectrum density is calculated. Exponent d represents the slope of the least square fit of the line over power spectrum density plot. As shown in Figure 1, a line of slope 1 represents a

negative dimension over a PSD plot. This happens because, one sided PSD plot is considered to estimate the best least square fit which is a negative slope line. Therefore, we are required to reverse the sign in our calculations to ensure that dimensions remain positive. It is equivalent of considering the single sided negative frequency spectrum

As shown in Figure 1, lines having varying slopes over a log-log plot of PSD are shown [13] [14]. If the exponent of the equation (1) is −1, then the resulting PSD would be of blue noise where higher frequency components are amplified. Similarly, if d=0, then the resulting PSD would result due to white noise. If the frequency exponent is 1 then it represents PSD of a pink noise. For d=2, PSD is generated from brown noise or standard Brownian motion process. For d=3, it becomes black noise. Also increasing d from 1 till 3 will result in increasing attenuation of higher frequency components and the correlation will increase. Black noise is also called as broadband noise.

Moreover, this noise phenomenon is also called integer noise [18]. There are fractional noises that are not integer and lie between these integer limits. For example, if the exponent lies between 1 and 3, it is called fractional Brownian motion process [18]. As an example, Figure 2 shows a PSD plot of a Gaussian pulse while Figure 3 shows a one-sided plot of the same PSD plot. In order to find the spectral fractal dimension, Figure 4 shows a linear fit of the log-log plot of single sided PSD. Slope of this line is the magnitude of the exponent of equation 1. As there is only a single slope of the log-log plot of single sided PSD, this process is called a mono-fractal.

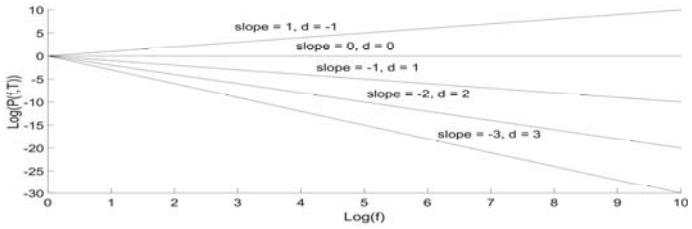

Figure 1: Slopes representing spectral fractal dimensions.

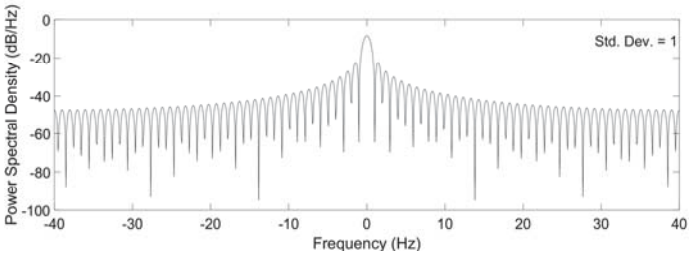

Figure 2: A double sided Power Spectral Density of a Gaussian pulse.

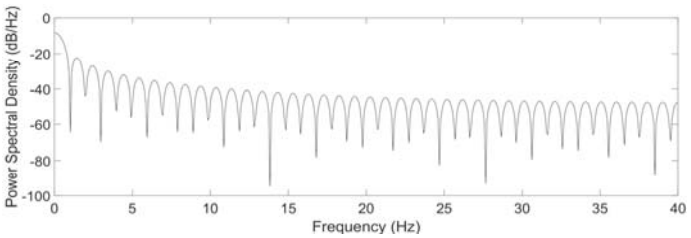

Figure 3: A single sided Power Spectral Density of a Gaussian pulse.

If there are more than one slope then the process is called multifractal and in this case we have to set the data window size T such that increasing the window size shows correct changes in the slope of the log-log plot of the PSD of the window. Therefore, if we estimate the spectral fractal dimension of a time series in a sliding window fashion, it will generate a multifractal trajectory that will represent the pattern of changing fractal dimension within the upper and lower limits of the topological dimensions [18].

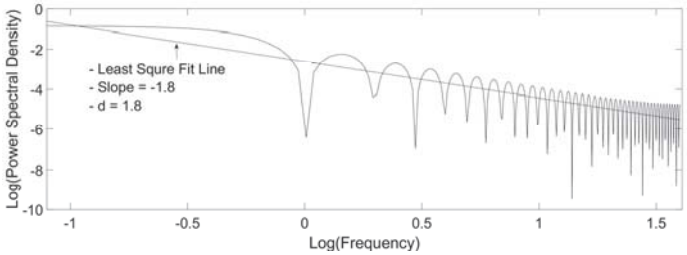

Figure 4: Log-Log plot and least square fit.

In order to find the spectral fractal dimension, following is the relationship between slope and the fractal dimension [13] [17]:

$$D_s = E + \frac{3 - d}{2} \quad (2)$$

where E is the number of dimensions or number of features represented by the time series. In this work, E =1, since there is one feature of the time series i.e. DNS packet count. Therefore, equation (2) is reduced to:

$$D_s = \frac{5 - d}{2} \quad (3)$$

Therefore, in this work, if spectral exponent is in the range $1 < d < 3$, then the spectral fractal dimension accordingly falls in the range $1 < D_s < 2$.

This work is a continuation of our ongoing research to explore and characterize complexity of internet data sets. Earlier [16] [20], authors have illustrated an algorithm using variance fractal dimension trajectory (VFDT) to characterize DNS time series using a moving window of data samples that is varying to ensure weak sense stochastic stationarity. In this work, an algorithm of

spectral fractal dimension trajectory (SFDT) is developed and tested on various DNS data sets to characterize the complexity of normal and attack DNS traffic. Main advantage of using spectral fractal dimension is that it does not require data samples to render weak sense stationarity which is a necessary requirement for variance fractal dimension trajectory. Spectral fractal dimension is an information based fractal dimension which is considered a frequency transform of variance fractal dimension and therefore, their theoretical computation results are bound to be same [21]. However, within a computational accuracy, this equivalence is not apparent. But as the results of this paper reveal, spectral fractal dimension is able to differentiate normal and attack traffic which is similar in performance of the variance fractal dimension. Moreover, in this work, 2 new data sets of normal DNS traffic are also used to test the performance of the proposed algorithm.

V. ESTIMATION OF POWER SPECTRAL DENSITY

There are 2 methods of estimating PSD of a time series [22]; non parametric methods and parametric methods. Non parametric methods estimate the PSD from the data itself using Fast Fourier Transform (FFT) and overlapping the adjacent windows. These methods introduce redundancy in the statistical information. Parametric methods are superior in performance because these methods seek to estimate the parameters of a linear or nonlinear model that is generating the time series. Typically these models mimic the model using statistical white process. Parametric methods are sometimes referred as Auto-Regressive (AR) processes whose order defines the type of non-linearity expected in the time series. In this work, authors have implemented a second order AR processes known as Yule-Walker method [23]. A time series can be represented as follows:

$$y[n] = a_0 x[n] + a_1 x[n-1] + \cdots a_N x[n-N] \qquad (4)$$

where $y[n]$ represents the time series of the window having an order of N. Parameters $[a_0, a_1, a_2 \ldots a_N]$ are required to be calculated to estimate the model generating the samples.

According to Yule-Walker model of estimating PSD of a time series, following is the calculation method [22]:

1. Set the model order N a-priori.
2. Find the autocorrelation function (ACF) of the N ordered AR process.
3. Find the FFT of the ACF which will provide an N'th order PSD estimate of the time series.

VI. DATA SET AND PROGRAMMING PLATFORM

Authors have used 3 data sets of DNS packets to analyze the performance of the proposed algorithm. Following is the summary of information about these data sets:

1. Data set from PREDICT USA [24] which contains traces of a DNS distributed denial of service attack (DDOS). This data set is composed of various packet capture (ERF file format) files taken from a real DNS attack scenario and is anonymized to ensure data confidentiality. It contains total 59,928,920 packets out of which there are total 358,019 DNS packets. DNS denial of service amplification attack was recorded for 10 minutes while the total capture time was 32 minutes and 47 seconds. One target IP and 6 DNS server IPs were already known and total 340,865 DNS denial of service packets were recorded. Total ERF file size is 5.3 GB. According to the USC-Lander [24], this data set was composed of one DNS Denial of Service Amplification attack staged between USC/ISI, Marina-del -Rey, Los Angeles, California to Colorado State University, Fort Collins, Colorado.
2. Data set from CAIDA USA [25] which contains internet traces from optical fiber internet connectivity from 2002 and 2003. Traces of April 24 2003 are used which captures 75,74,005 packets from 7:00:00 GMT till 7:04:59 GMT and contains 32,358 DNS packets within this duration. This does not contain any malicious traffic.
3. Data set from our experiment in which a PCAP file is captured from a lab computer which is being used for browsing and software development for the cyber security project. This computer is connected to internet and has MS Windows 7 installed. Firefox browsers are used for browsing and multiple windows and tabs are opened where many websites, cloud applications and services are connected to the internet. Approximate memory usage of Firefox based internet connectivity is 1.2 GB. 10,39,460 packets were captured on Nov 27 2015 from 00:08:06 GMT till 02:07:31 GMT. Out of these, total 11,721 DNS queries were made from 00:08:07 GMT till 02:07:21 GMT. This computer is heavily guarded against any malicious threats by the network administrators.

The proposed algorithm is developed using Matlab programming platform. DNS data time series is created using Matlab based parsing program such that the time series represents DNS time series at individual end points

of a network. DNS data time series is generated by sampling DNS packets at 100 milliseconds in order to include the network latency to complete a packet round trip is covered sufficiently [26].

VII. ALGORITHM

Spectral Fractal Dimension Trajectory (SFDT)

1. Set the following parameters:
   a. Data pointer: d_p.
   b. Window size: lag. (use 1024 samples for DNS packets which is less than 2 minutes of window for 100 ms sampling and is aligned with DNS traffic patterns i.e. normal DNS traffic is not very frequent compare to HTTP or FTP traffic)
   c. Window= d_p+ lag.
   d. Auto-Regression process degree: d=2.
   e. Feature Dimension: E=1.
2. Initialize $\boldsymbol{d_p}$ at first sample of the data series.
3. Run a main loop till the end of data series.
4. Pass the window $N = d_p + lag$.
5. Set M-point FFT:
$$MFFT = 2^{\text{ceil}(\log_2(N))}$$
6. Call **PSD_AR(N, *MFFT*,d)** function and get one-sided estimate of power spectral density (PSD).
7. Take logarithm of both PSD and frequency.
8. Estimate the least square slope of the log-log plot of PSD.
9. If slope is greater than -1, then remove high frequency components of PSD till slope $<$ -1.
10. If slope is less than -3, then remove low frequency components of PSD till slope $>$ -3.
11. If slope is not defined due to computation limitations or zeros in the data set (for which log is undefined), use previous value of slope because having a zero slope or log of zero means that the power spectral density is either a flat line which means it is a white process or there is a zero value whose fractal dimension is zero.
12. Change the sign of slope and calculate spectral fractal dimension according to equation (3).

Power Spectral Density Estimate Function- PSD_AR( )

1. Let N samples are considered in a window of data X: *main window* X_N.
2. Remove DC component as follows:
$$X_C = X_N - \frac{sum(X_N)}{N}$$
3. Calculate sampling frequency:
$$F_s = \frac{N-1}{\max(X_N)}$$
4. Append 0's at the end of X_N corresponding to the difference: $N - MFFT$.
5. Estimate parameters a_0, a_1, a_2 of the AR(2) process using Yule-Walker model. Find the autocorrelation vector ACF_x of the $MFFT$ samples.
6. Calculate the Fast Fourier Transform of the ACF_x that will output a vector of the least square estimate of the power spectral density (PSD) vector of the time seriesX_N.
7. If ACF_x has even elements, return following number of elements of estimated PSD vector:
$$\frac{MFFT}{2} + 1$$
Else return following number of elements of estimated PSD vector:
$$\frac{MFFT + 1}{2}$$

VIII. EXPERIMENT AND RESULTS

In order to validate the performance of spectral fractal dimension trajectory, authors have tested the algorithm over fractal Brownian motion (fBm) process. We have generated different samples of fBm process using varying values of Hurst parameter between 0.1 and 0.9. As spectral fractal dimension trajectory follows equation 3 for a single feature, therefore, spectral fractal dimension trajectory is obtained that varies between 1.9 and 1.1 respectively for the above values of Hurst parameters. As shown in Figure 5, 30780 samples of a concatenated fBm process are shown which are generated with 3 values of Hurst parameter; 10260 samples for H=0.1, next 10260 samples for H=0.5 and the rest with H=0.9. As can be seen in Figure 6, SFDT shows marked variations in the trajectory when Ds changes. Spectral fractal dimension trajectory of this concatenated fBm process clearly shows that SFDT for the 3 different fBm processes is distinctly apparent and the first and last samples of each fractal process are following closely with the process itself.

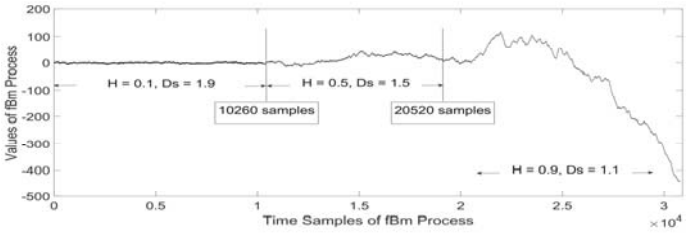

Figure 5: Concatenated fBm process with varying Ds.

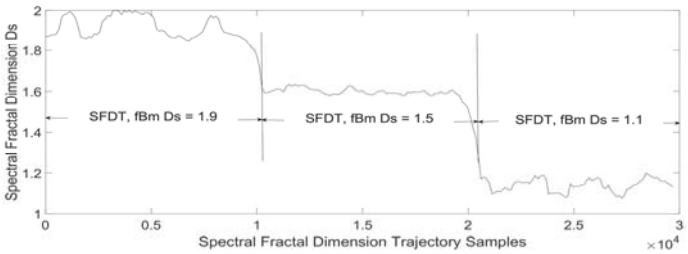

Figure 6: SFDT of Concatenated fBm process.

In this work, we have tested the spectral fractal dimension trajectory algorithm to characterize the time series of DNS packet counts using data set from CAIDA [25], PREDICT [24] and our own experimental data set.

As Figure 7 shows, a sampled time series of DNS packet counts from CAIDA data set is generated. We can see that it shows many samples with no DNS packets. In addition, there is only one sample that has maximum count of 30 while most non-zero samples lies close to the count of 10. As evident from Figure 8, the spectral fractal dimension trajectory of this time series is showing a constant fractal dimension within a precision order of 13 decimal digits which can be rounded to a fractal dimension of 1.99. This is a monofractal and as already discussed in Figure 1, this behavior can be approximated to a random fractional Brownian motion process which in turn is equivalent to a Brown noise process. Therefore, the time series of DNS packet count would have strong correlation which is a property of Brown noise process. In order to ensure and further validate that normal DNS packet counts are monfractal, we have tested the algorithm over an experimental data set of normal DNS packets whose DNS count time series sampled at 100 millisecond is shown in Figure 9. This series conforms the pattern of CAIDA data set but the max DNS packet count is 12 which is lower than CAIDA data set. It is expected since CAIDA data set is generated by monitoring traffic at the network gateway of an optical fiber link while this experiment is performed over a single computer in our lab. However, most samples are found below a count of 6 and there are many samples with zero DNS packet counts. As depicted in Figure 10, spectral fractal dimension trajectory is showing monfractality within the range of 1.99. Again, it confirms our hypothesis that normal DNS packets are fractal Brownian process and does not show higher degree of complexity as multifractals do.

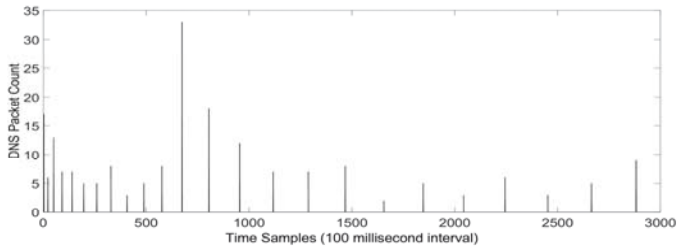

Figure 7: Time series of DNS packet counts sampled at 100 ms – CAIDA data set.

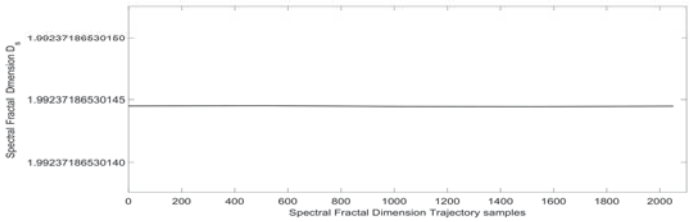

Figure 8: SFDT of DNS packet counts – CAIDA data set.

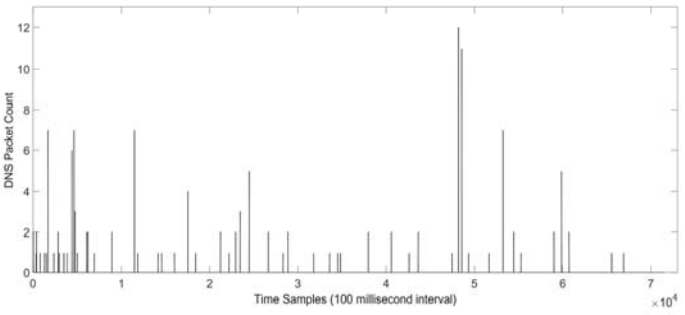

Figure 9: Time series of DNS packet counts sampled at 100 ms – Experiment data set.

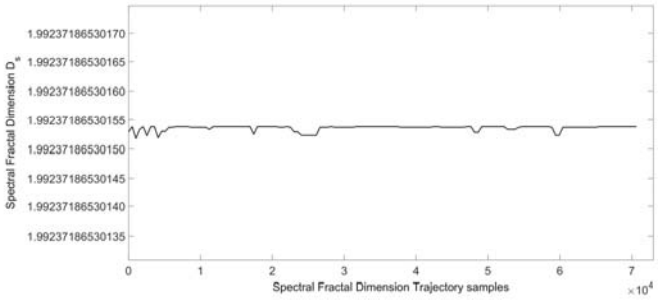

Figure 10: SFDT of DNS packet counts – Experiment data set.

As shown in Figure 11, DNS packet count time series sampled at 100 millisecond is shown for PREDICT data set. It contains both normal and attack traffic. SFDT algorithm is applied and Figure 12 shows the trajectory of spectral fractal dimension of DNS time series. This time series of DNS packet counts over the target IP is generated with equal sampling interval of 100 millisecond. Attack started at sample number 805 and ended at sample number 9146. There are total 10260 samples. Moreover, at sample number 72, there is a large burst of DNS packet count that

happens when the node starts sending and receiving DNS broadcast to resolve queries and build local DNS cache etc. As can be seen, attack starts when the spectral fractal dimension is above 1.8 and ends when it goes below 1.8. During the start and end time, spectral fractal dimension trjectory shows higher dimension close to 1.9. These values of start and end of the attack are dependent on data set. However, we can also state that the presence of attack has introduced higher level of complexity as depicted by the change (increase) in spectral fractal dimension i.e. multifractal. Moreover, there is a significant distinction between normal and attack DNS traffic. Correspondingly, the attack traffic is not showing a monofractal behavior and has varying fractal dimension between the range of 1.8 and 1.9.

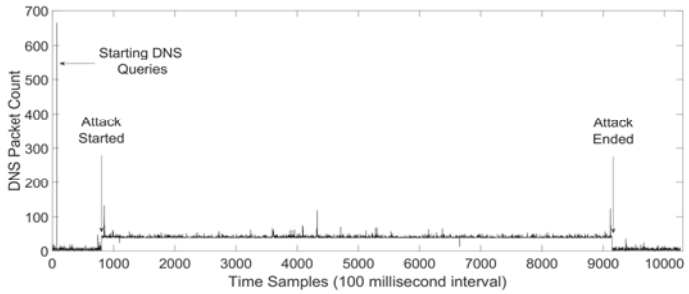

Figure 11: Time series of DNS packet counts sampled at 100ms – PREDICT data set.

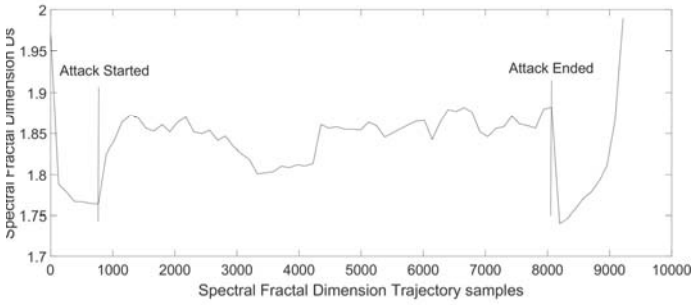

Figure 12: SFDT of DNS packet counts containing attack – PREDICT data set

IX. DISCUSSION

The spectral fractal dimension trajectory (SFDT) is a proposed method that calculates the cognitive complexity of a time series in a sliding window fashion by estimating the self-similarity or self-affinity of the sliding window of the time series. As shown, a mono-fractal such as a single dimension fractal Brownian motion process shows a mean value of Ds which is a least-square estimate of the spectral fractal dimension. Moreover, multifractal which shows varying spectral dimension over different time intervals (time windows), exhibits significant variations in fractal dimension trajectory (SFDT) as the self-similarity or fractal dimension changes with the course of time. Equivalently, multifractals show a varying degree of persistence (contrast to monofractal which have single degree of persistence) from high degree of persistence at one extreme to high degree of anti-persistence at other extreme. It can be stated that multifractals are more complex because they have varying fractal dimensions which is equivalent to varying degree of persistence. It is important to note that spectral fractal dimension trajectory is sensitive to errors in calculation due to outliers and saturation points. As spectral fractal dimension is calculated by estimating the slope of a least-square fit of line over a log-log plot of the single-sided power spectral density of time series, it is critical to remove the saturation points and outliers in the data. If the dimension is going below the lower limit of the topological dimension i.e. E =1, then remove low frequency components which introduce saturation. Moreover, if the dimension is going above the higher limit of the topological dimension i.e. E=2, then remove the higher frequency components which introduce outliers and increases the slope of the PSD. Moreover, if a window of time series contains zero values, then it is an indication of no fractality. Also, if there is a zero slope in the power spectral density, then it should be treated as random process.

In addition, it is considerably important to pre-process the packets carefully. In our work, we have provided an initial proof of concept to detect an attack by estimating the spectral fractal dimension in a sliding window fashion. We call this spectral fractal dimension trajectory (SFDT). However, SFDT algorithm requires that the time series should follow Nyquist sampling criterion to generate statistically valid samples. This requires that the standard characteristics of the observation data are known a-priori i.e. round trip time of the DNS packet, so that the time series represents meaningful features i.e. DNS packet count in our work.

As stated in the experiment and result section, our preliminary analysis shows that normal DNS packet count time series shows high degree of mono-fractality which is a sign of low complexity as compared with multifractal time series where normal and attack traffic shows different fractal behavior (or fractal dimensions). It is also shown that if there is a consistent DDoS DNS attack, then the multifractal dimensions will show lot of complexity/fractal variations especially during the time of start and end of an attack. It can be validated through visual plots and could provide an automated way to alert the network administrators.

It can be argued that a DNS DDoS attack can be detected visually then why there is a need to automate the detection process. It is imperative to note here that with the state of the art security technologies that includes firewall, Security Information and Event Management Systems (SIEM) and Intrusion Detection Systems (IDS), human analysis is required to either analyze and/or configure the statistical results of the internet flows. Moreover, main purpose of DDoS attack is to deny availability of a network service to a network or a node in a network. As most of the security systems are based on perimeter security (network security), it is hard to detect a DDoS attack if it is aimed at a particular node because for the network security administrators, detecting an anomalous increase in packet counts for a node would be equivalent of finding a needle in the haystack of humongous network data to detect. In addition, machine learning can be used to detect DDoS attacks autonomously, but as shown in Figure 7 and Figure 9, normal network data does not show consistent pattern that is necessary to benchmark machine learning algorithms based detection of DDoS attacks. As authors have already shown in [27] , probability of false alarms is relatively high and lot of fine tuning and reconfiguration of machine learning algorithm is necessary. Therefore, in this work, it is presented that the detection of DDoS attacks can be offloaded to a cognitive algorithm which is based on spectral fractal dimension trajectory.

X. CONCLUSION

In this work, authors have presented a new fractal based cognitive algorithm called spectral fractal dimension trajectory (SFDT) to detect variations in the complexity of the DNS packet time series using a sliding window. SFDT generates statistically valid spectral fractal dimensions over a sliding window of time series and the power spectrum density of the sliding window is estimated using second order auto-regression process. Also, authors have validated the performance of the algorithm using mathematical fractal Brownian motion process. The proposed algorithm is prone to the high variability of the time series and can capture variations in the complexity of the time series due to the presence of an attack. Also, it is shown that normal DNS traffic either at a network gateway or at a node in a network shows a monofractal behavior with persistent fractal dimension while in the case of DNS DDoS attack, spectral fractal dimension trajectory shows multifractal behavior which is an indication of an increase in degree of complexity from monofractal to multifractal.

XI. ACKNOWLEDGMENT

This work is supported in part through a research fellowship from Mitacs-Accelerate Canada. Authors are also thankful to PREDICT USA and CAIDA USA for providing state-of-the-art data sets.

XII. REFERENCES

[1] Muhammad Salman Khan, Ken Ferens and Witold Kinsner, "A cognitive multifractal approach to characterize complexity of non-stationary and malicious DNS data traffic using adaptive sliding window," in *Proceedings of IEEE 14th International Conference on Cognitive Informatics & Cognitive Computing (ICCI*CC), 2015*, Beijing, China, 2015.

[2] Georgios Kambourakis, Tassos Moschos, Dimitris Geneiatakis and Stefanos Gritzalis, "Detecting DNS amplification attacks," *Lecture Notes in Computer Science,* vol. 5141, pp. 185-196, 2008.

[3] Changhua Sun, Bin Liu and Lei Shi, "Efficient and low-cost hardware defense against DNS amplification attacks," in *IEEE GLOBECOM*, 2008.

[4] Saman Taghavi Zargar, James Joshi and David Tipper,, "A survey of defense mechanisms against distributed denial of service (DDoS) flooding attacks," in *IEEE Communications Surveys & Tutorials*, 2013.

[5] Tao Peng, Christopher Leckie and Kotagiri Ramamohanarao, "Survey of network-based defense mechanisms countering the DoS and DDoS problems," *Journal ACM Computing Surveys (CSUR),* vol. 39, no. 1, 2007.

[6] Zhengmin Xia, Songnian Lu and Jianhua Li, "DDoS Flood Attack Detection Based on Fractal Parameters," in *Proceedings of 8th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM)*, Shanghai, China , 2012.

[7] Witold Kinsner , "Towards cognitive security systems," in *Proc. 11th IEEE Intern. Conf. on Cognitive Informatics and Cognitive Computing*, Kyoto, Japan; August 22-24, 2012, 2012.

[8] Yingxu Wang, "On cognitive informatics," in *Proc. 1st IEEE Intern. Conf. Cognitive Informatics*, Calgary, 2002.

[9] Simon Haykin , Cognitive dynamic systems: Perception-Action cycle, Cambridge, UK: Cambridge University Press, 2012, p. 322.

[10] Penti O.A. Haikonen , The cognitive approach to consious machines, New York, NY: Academic, 2003.

[11] Yingxu Wang, Du Zhang and Witold Kinsner, Advances in cognitive informatics and cognitive computing, vol. SCI 323, Berlin: Springer Verlag, 2010, pp. 265-295.

[12] Witold Kinsner , "It's time for multiscale analysis and synthesis in cognitive systems," in *Proc. IEEE 10th Intl. Conf. Cognitive Informatics & Cognitive Computing (ICCI*CC11)*, Banff, AB, 2011.

[13] Michael Potter and Witold Kinsner, "Multifractal characterization of synthetic ECG in the presence of coloured noise," in *Proc. IEEE Canadian Conference on Electrical and Computer Engineering*, 2004.

[14] Witold Kinsner, "A unified approach to fractal dimensions," *Int'l Journal of Cognitive Informatics and Natural Intelligence,* vol. 1, no. 4, pp. 26-46, 2007.

[15] Witold Kinsner, "Towards cognitive machines: Multiscale measures and analysis," *Intern. J. Cognitive Informatics and,* vol. 1, no. 1, p. 28–38, 2007.

[16] Muhammad Salman Khan, Ken Ferens and Witold Kinsner, "A cognitive multifractal approach to characterize complexity of non-stationary and malicious DNS data traffic using adaptive sliding window," in *Proc. IEEE 13th Intl. Conf. Cognitive Informatics & Cognitive Computing (ICCI*CC14)*, 2015.

[17] Joao B. Florindo and Odemir M. Bruno , "Fourier fractal descriptors for colored texture analysis," in *Lecture Notes in Computer Science, Advanced Concepts for Intelligent Vision Systems*, 2011.

[18] Witold Kinsner, *Graduate lectures on Fractal and Chaos Engineering,* Winnipeg, MB, Canada, 2015.

[19] Joao B. Florindo and Odemir M. Bruno, "Closed contour fractal dimension estimation by the fourier transform," *Chaos, Solitons and Fractals,* vol. 44, no. 10, pp. 851--861, 2011.

[20] Muhammad Salman Khan, Ken Ferens and Witold Kinsner, "A polyscale autonomous sliding window for cognitive machine classification of malicious Internet traffic," in *The 14th International Conference on Security and Management (SAM'15), WORLDCOMP'15*, Las Vegas, 2015.

[21] Witold Kinsner, "Towards Cognitive Machines: Multiscale Measures and Analysis," in *Novel Approaches in Cognitive Informatics and Natural Intelligence*, 2009, pp. 188-199.

[22] Inan Guler, M.Kemal Kiymik, Mehmet Akin and Ahmet Alkan, "AR spectral analysis of EEG signals by using maximum likelihood estimation," *Computers in Biology and Medicine,* vol. 31, no. 6, p. 441–450, 2001.

[23] Larry Marple, "A new autoregressive spectrum analysis algorithm," *IEEE Transactions on Acoustics, Specch and Singal Processing,* Vols. ASSP-28, pp. 441 - 454, 1980.

[24] PREDICT, "USC/Lander - Scrambled Internet Measurement, PREDICT ID USC-Lander".

[25] CAIDA UCSD , 2003. [Online]. Available: https://data.caida.org/datasets/oc48/oc48-original/.

[26] AT&T Inc. USA, "Network latency," 2015. [Online]. Available: https://ipnetwork.bgtmo.ip.att.net/pws/network_delay.html. [Accessed 2015].

[27] Muhammad Salman Khan, Ken Ferens and Witold Kinsner, "A chaotic complexity measure for cognitive machine classification of cyber-attacks on computer networks," *International Journal of Cognitive Informatics and Natural Intelligence (IJCINI),* vol. 8(3), 2014.

[28] Delio Brignoli, "A Masters thesis on "DDoS detection based on traffic self-similarity"," University of Canterbury Research Repository, 2008.