Systematic Downloading: Analysis and Detection

Akshay Bhandari, Shreya Khare and Hema A. Murthy
Department of Computer Science and Engineering
Indian Institute of Technology Madras,Chennai-600036
Email :{akshayb, shreya, hema}@cse.iitm.ac.in

***Abstract*—The Internet has become a vital source of information. This comes with the attendant problems, namely misuse. Systematic downloading of academic and digitized media have become commonplace. Academic institutions in particular get blacklisted owing to free availability of Internet across campus.**

The objective of this paper is to pro-actively detect systematic downloading. Time series of number of requests are analyzed with pattern analysis techniques. A characterization of the model in the Z-domain shows that the roots of the transfer function form separate clusters during normal and abnormal behavior of traffic. Stability of the system has been used as a cue to detect systematic downloading. Analyzing the trajectory and location of roots to detect systematic downloading involves complex decisions which may not be robust to evolving traffic. This issue is addressed by using Support Vector Data Description that learns a hypersphere enclosing normal traffic. Our empirical evaluations with features like Auto-Regressive model roots and Line Spectral Pairs (LSP) obtained from the request time series along with data description show a lot of promise for detecting systematic downloading with F-measure and accuracy as high as 0.90 and 99.5%, respectively. This hybrid approach ensures low false alarms, misses and guarantees robustness of the system.

Keywords - Traffic modeling, Systematic downloading, Data description, Anomaly detection, Auto-Regressive (AR) model.

I. Introduction

Internet today provides high speed access to various kinds of information. Although there are many benefits to this, the downside is the misuse of electronic resources and copyright violations of licensed content. In the past, monitoring of electronic sources by vendors like IEEE [1], ACM [2], JSTOR [3] has revealed incidents wherein large volumes of material were downloaded in a short period of time. Such acts, termed as systematic downloading or excessive downloading, violate terms of usage between the publisher and subscriber followed by cessation of access to the content. It degrades publisher's server performance along with blocking or increasing the response time of the request [4].

In 2010, 4.8 million articles were downloaded from JSTOR using the MIT network leading to indictment of Aaron Swartz [5]. Other incidents of similar nature are reported in [6]. In the event of policy violations, the vendor automatically blacklists the offending IP address or range of IP addresses and penalizes with a new pricing model. A mid-sized campus has about 10,000 users. A single malicious user can affect an entire campus. Vendors use automated tools to identify excessive or systematic downloading [6] before it adversely affects their systems. They are reluctant to reveal specifics about detection of misuse as hackers may use this information to bypass their controls. Hence, it is necessary to detect systematic downloading at the level of campus access network.

Administrative measures like performing user authorization, scanning of proxy servers at regular intervals, awareness of policies, etc. are common techniques to prevent systematic downloading [6]. Generally, the threshold on the number of daily downloads set by the publisher is not revealed. Moreover, techniques for adaptation of the threshold as the ground rule may increase the false alarms. Hence, there is a need to devise automated techniques which can capture the intent of the user and pro-actively detect systematic downloading.

Tracking significant changes in the traffic pattern may provide an insight for detecting anomalies. [7], [8] use time series to detect anomalies like Denial of Service (DoS) attack. Further, [9] exploits time series models to capture prominent network traffic characteristics. [10], [11] deploy time series data obtained from management information base variables (MIB) to raise alarms for network anomalies. In [10], abrupt changes in data are detected by modeling network data as an Auto Regressive (AR) process and two adjacent non-overlapping windows of time series are compared to detect anomalies. Time series have also been used in spam detection, system

978-1-4799-5255-7/14/$31.00 ©2014 IEEE

failure detection and bio-surveillance.

Rule based, pattern matching and statistical analysis approaches are the popular methods of detecting anomalies in time series [12]. Rule based methods require an exhaustive database containing the rules of behavior of the faulty system [13]. Rule based anomaly detection systems are too slow for real time applications and do not scale for the evolving network environment. Techniques that learn from changing characteristics are required for effective detection.

Statistical pattern matching techniques attempt to address this variability by building traffic profiles from the environment using online learning [12]. In pattern matching based techniques, anomalies are defined as a deviation from normal behavior that can be associated with parametric and non-parametric changes evolving gradually in the system [14]. The efficiency of the system depends on the accuracy of the traffic profile generated.

In the statistical approach, anomalies manifest themselves as abrupt changes in the system. Detailed information that covers the variability in network traffic, helps in improving detection of anomalies. [7], [8], [10], [11] and [15] implement various statistical approaches to detect anomalies and intrusions in different systems. Real time statistical changes are characterized in [11] using duration filter heuristics in conjunction with Management Information Base (MIB) variables that precede the occurrence of a fault. These methods reduce the number of false alarms as compared to rule based techniques. However, handling multiple feature variables requires more sophisticated techniques in addition to the statistical approaches for identifying anomalies in systems.

The goal of this paper is to perform anomaly detection using time series models in tandem with pattern analysis techniques. This hybrid approach can be used to design tools that can be used at a campus proxy to detect systematic downloading on the fly. Modeling the request traffic as a time series, provides meaningful statistics and helps in analyzing the dependence between adjacent observations. Further, these statistics are employed to build a one class classification model.

In this paper, the emphasis is on the use of signal processing techniques with pattern recognition for efficient anomaly detection for evolving environments. While signal processing provides a good set of features, pattern analysis makes use of these features for modeling and detection. The number of requests over a given time interval is used as a time series. This feature is then translated to the roots of the Auto Regressive(AR) model [16] and Line Spectral Pairs [17]. The stability of the system and trajectory of the roots have been used to differentiate between normal and systematic download situations. This technique though useful, holds some limitations to detect systematic downloading. To overcome these limitations, support vector data description [18] based techniques have been used to model the normal traffic and identify outliers. Data description techniques make testing for new data simple and reduces number of false alarms and misses.

As a convention in this paper, normal period refers to the period when the download traffic does not contain requests characterizing systematic downloading.

The paper is organized as follows. Section II describes the dataset used for modeling time series. Section III details the proposed work comprising of feature selection followed by methodology adopted for modeling. Experiments and results are described in Section IV followed by conclusion in Section V.

II. Dataset

In order to identify outliers in the data and detect systematic downloading, a normal scenario of user interaction needs to be modeled. At an institutional level, the proxy server acts as an intermediary for requests from clients to remote servers across the Internet. It logs the details of all the requests from the client [19]. Proxy logs contain an entry for each request which contains details of time stamp, user ID, client IP address, size of the requested data, URI etc. User access patterns obtained can be used to train models for pattern analysis. Pattern analysis techniques are mostly supervised and require the ground truthed normal traffic to distinguish between normal and anomalous traffic.

A. Normal traffic request

For experimental analysis, data from Squid proxy logs of a university campus were used. For better characterization of normal traffic request, the data set consisted of traffic requests over the entire institute to the subscribed publisher over a period of four months. Since the license agreement and the normal daily downloads for each publisher are different, traffic to each publisher was modeled separately.

B. Systematic download traffic request

Due to the unavailability of systematic downloading traffic from the campus proxy logs, an emulator was designed. For design, a proxy server replicating the characteristics of the campus proxy server was installed on a machine to capture requests at the time of systematic downloading and generate logs. Apache server from WAMP [20] package was used as a Web server to host the digital media in the form of PDF (Portable Document Format) documents. Figure 1 shows the topology of the emulation.

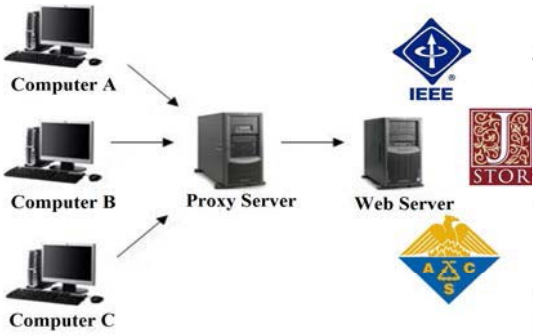


Fig. 1: Systematic downloading emulation topology

Configuration of emulation parameters such as download speed, number of documents and others were fixed with extreme care, to make emulated behavior a true replica of the real time behavior. To imitate a systematic downloading environment, a web server was hosted with a large number of documents. The Web-server hosted about 2200 files with variability in file size. Proxy log information was used to determine the file sizes for simulation. The download speed was varied from 30Kbps to 300Kbps. Download speed during the emulation was controlled by limiting the speed of the Web-server, considering the normal campus environment. Across emulations, the order in which the files were downloaded was randomized to remove correlation between any two requests.

III. Proposed Work

Request logs reflect traffic behavior accurately. However, they could also contain information irrelevant to anomaly detection. For a pattern analysis algorithm to perform well, appropriate features must be identified which can characterize the difference between normal and abnormal traffic. Once features are identified, detection mechanisms can be developed which can differentiate abnormal from normal traffic.

A. Feature Selection

The fundamental criterion for choosing a feature for detection of any type of anomaly is that the feature should exhibit distinct properties during normal and abnormal (anomalous) state. The feature value should remain unaltered by any other external factors. On extraction of data from proxy logs, features identified as potential candidates for modeling the request traffic and detect systematic downloading are

- Number of requests
- Size of the data downloaded

An attempt of systematic downloading involves request for digital content within short time intervals leading to high number of requests when compared to the requests during normal traffic. A thresholding based mechanism can be adopted on the number of requests to detect anomalous requests. However, publishers are averse to sharing the limits on downloads that trigger an event of systematic downloading.

Modeling traffic to obtain a suitable threshold may not be able to adequately represent the burst in the traffic leading to detection of limited anomalies, which may increase the probability of false alarms in detection. Moreover, imposing threshold on per a session basis may restrict users in the campus from downloading licensed content even if their request is normal, causing interrupted client-server interaction. Hence, better modeling of normal traffic requests is required to distinguish anomalous requests from normal traffic.

Besides the number of requests, size of the data downloaded will also increase at the time of an attempted of systematic download. However, size is highly dependent on the number of requests and therefore it is sufficient to model one of them. Moreover, if size is modeled, a single large download may be misconstrued as a systematic download which will lead to an increase in the number of false alarms. Therefore, the time series of number of requests has been used for modeling the download traffic by deriving features.

AR modeling captures the relationship between consecutive requests. This is important for systematic downloading, as a sequence of requests are correlated. The objective is to build a model using the time series

of requests and derive features that can be used for classification of traffic.

1) LPC/AR modeling: Linear prediction (LP) has been used for analysis of discrete signals. LP has been widely used for modeling time series and anomaly detection in network traffic. In LP analysis, the signal is modeled as a linear combination of its current and past inputs and its past outputs [16]. In the Z -domain, the LP coefficients model the roots. An all-pole model is a special case of this model with signal s_n modeled as a linear combination of past values and some input u_n. It is given as:

$$s_n = -\sum_{k=1}^{p} a_k s_{n-k} + G u_n \tag{1}$$

where a_k, $1 \leq k \leq p$, G is the gain parameter of the system and p is the model order. Taking the Z- transform of Equation 1, the transfer function of the system can be written as

$$H(z) = \frac{1}{A(z)} = \frac{G}{1 + \sum_{k=1}^{p} a_k z^{-k}} \tag{2}$$

Least squares method discussed in [16] is used to determine the LP coefficients. The model order is chosen using the partial auto-correlation plots [21]. Roots of $A(z)$ (inverse filter) have been used for detection of systematic downloading by

- Testing the stability of the system based on the location of the roots vis-a-vis the unit circle.
- Roots as a feature in a one-class classifier.

2) Line Spectrum Pairs: Linear prediction coefficients can not be used directly as features in a classifier. A small perturbation in the coefficient can lead to significant changes in the root locations in the Z-domain. LSPs have better sensitivity to noise and show better interpolation properties when compared to LPC coefficients [17]. As described in Equation 2, an all-pole filter $H(z) = \frac{1}{A(z)}$, $A(z)$ is given by

$$A(z) = 1 + a_1 z^{-1} + ... + a_p z^{-p} \tag{3}$$

where p is the order of the LP analysis and a_i represents the LP coefficients. The inverse filter polynomial can be decomposed into two polynomials

$$P(z) = A(z) + z^{-(p+1)} A(z^{-1}) \tag{4}$$

and

$$Q(z) = A(z) - z^{-(p+1)} A(z^{-1}) \tag{5}$$

Roots of the polynomials $P(z)$ and $Q(z)$ correspond to the LSPs. As contrary to the roots of $A(z)$ which lie inside the unit circle (set of complex numbers of the form $z = e^{i\theta}$), roots of $P(z)$ and $Q(z)$ lie on the unit circle. Thus, LSP representation of $A(z)$ consists only of the angle in the Z-domain, the magnitude is unity as compared to LPC roots which have both magnitude and phase. For every AR root, there are two LSPs ($P(z)$ and $Q(z)$). As LSPs occur in conjugate pairs, only half of the number of roots (between 0 to π) are required. Figure 2 shows roots of P(Z) and Q(Z), interleaved and located on the unit circle. For every AR root, there are two LSPs. LSPs have been used as a feature for the one class classifier.

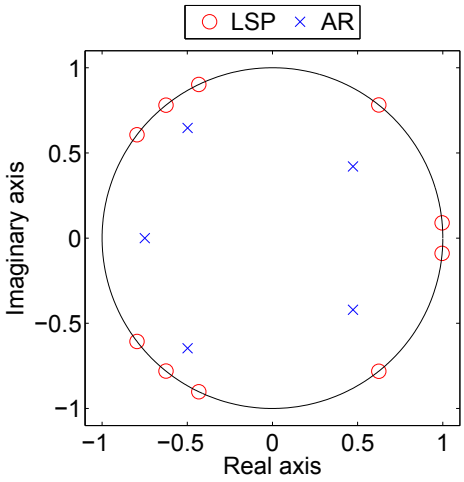


Fig. 2: Roots of P(z), Q(z) and A(z)

B. Anomaly (Systematic Downloading) Detection

Abnormalities in the request traffic can be identified on the fly using AR roots and LSPs. Two criteria have been used for this detection namely, stability of the transfer function in the case of all-pole filter model and using one class classification as mentioned below to detect outliers in the traffic.

1) Stability of system: The poles of an all-pole filter are the roots of the polynomial $A(z)$. If all poles lie inside the unit circle, the filter is stable. If the roots of the inverse filter tend to move towards the boundary of

the unit circle, the system tends to become unstable as the poles lie near the circle boundary. This stability has been used as the criterion for the detection of systematic downloading. Figure 3 shows the movement of the roots towards the boundary of the unit circle and tendency of the system to become unstable. Similar observations were made in [7] in the context of DOS attack detection with TCP half-open counts as a feature.

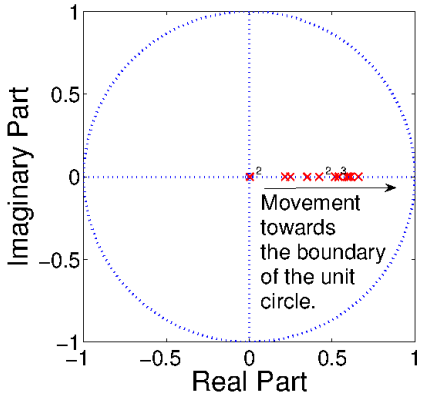


Fig. 3: Movement of roots

2) Data Description: As normal traffic is the norm and systematic downloading is an exception, large number of examples of normal traffic are available in the proxy logs. A classification methodology is required, where only normal traffic is modeled. A measure of dissimilarity is defined to distinguish between normal and abnormal traffic. As the details about specific distribution to be followed by normal data can not be discerned, it is necessary to model the boundary of the normal class instead of modeling the complete density distribution [18]. This method is different from the standard classification methods. This methodology is termed as one class classification, or data description [22]. This is especially useful for identifying new or unknown distributions.

[18] presents Support Vector Data Description (SVDD), a method which is robust against anomalies and provides spherically shaped boundaries around a dataset without estimating data density. SVDD translates this one class scenario to find a minimum enclosing hypersphere (i.e center a and radius R) that contains normal input data. For a function,

$$f(x) = \| \phi(x) - a \|^2 - R^2 \tag{6}$$

the boundary of the hypersphere is described by the set $x\colon f(x) = 0 \wedge x \in \chi$. This set is termed as the set of support vectors. The parameters of $f(x)$ are to be chosen such that $f(x) \leq 0$ for normal data and $f(x) > 0$ for anomalous points. The parameter (i.e center a and radius R) can be calculated by solving the optimization as shown in Equation 7.

$$\begin{gathered} \min_{R,c,\xi} \quad R^2 + C\sum_{i=1}^{n}\xi_i \\ s.t. \forall_{i=1}^{n} f(x) = \| \phi(x) - a \|^2 - R^2 \\ \forall_{i=1}^{n}\colon \xi_i \geq 0 \end{gathered} \tag{7}$$

In [18], the details of solving the above mentioned optimization problem is discussed. The parameter C controls the trade-off between the volume and the errors. Omission of some data points might lead to concise description of data. Slack induced by discarded data points is observed by variables ξ_i. To test an object z, the distance from the center of the sphere is calculated. A test object is classified as normal if the distance of the point lies inside the hypersphere which is given as

$$\| z - a \|^2 \leq R^2 \tag{8}$$

IV. Experiments and Results

Time series of the requests for the licensed content to different publishers was extracted from the proxy server logs over four months. The data generated, as discussed in Section II was used for analysis purposes. Time series of request traffic was extracted and roots of the AR model and LSP were calculated. Detection methods include checking the stability of the AR model. The trajectory of the roots is important for being proactive, since any root that moves towards the unit circle is a clear indication of a possible systematic download. Since checking stability of a system involves complicated analysis like tracking trajectory of the roots, there is a high probability of large number of false alarms and misses. To overcome this, one class classification or data description was used. SVDD can detect novel or outlier trace in the data. An extensively large number of experiments were performed with varying parameters like different polling interval of time series and C, kernel functions and its parameters in case of SVDD, to develop a robust method to detect systematic downloading.

A. Detection using stability of the system

Logs obtained after emulation of systematic downloading were mixed with the normal behavior logs and were processed to perform time series analysis. An all-pole filter was used to model the combined time series. In an all-pole filter model, poles measure the stability of the system. By sequentially tracking one or more roots of the AR polynomial at regular intervals, systematic download can be distinguished from normal downloads. At the time of systematic downloading, roots tend to move towards the boundary of the unit circle as explained in section III-B1.

During systematic downloading, there will be a change in expected mean and variance, which can be captured by estimating the roots over each window of samples. Table I shows the sum of distances of roots from the origin in the unit circle for five emulations.

In Table I PI, N and $S.D$. denote polling interval, normal and systematic downloading period, respectively. It indicates that the average distance of roots across all the emulations is high at the time of systematic downloading when compared to normal period.

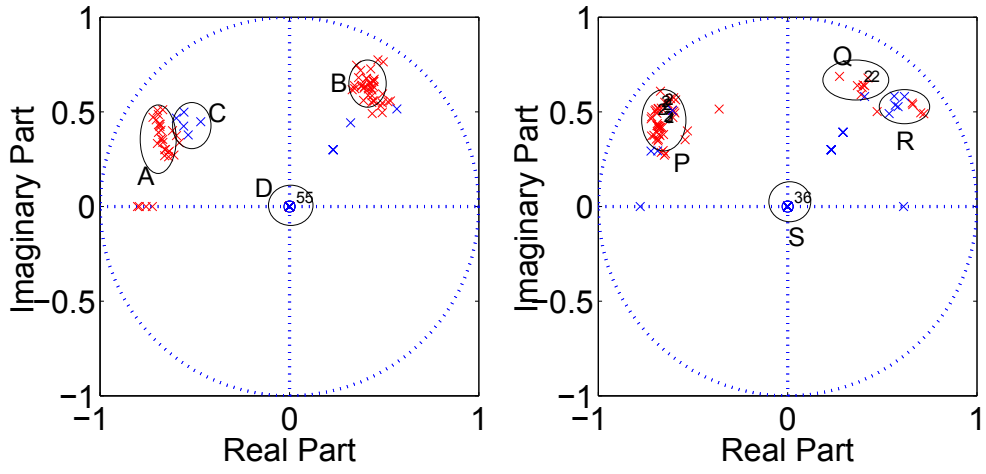
Similar behavior is observed over different polling intervals. Across emulations, variation is observed in the sum of distances of the roots. This variation may be ascribed to randomization in download order of files and the difference in the number of requests at the time

PI	**5 sec**		**10 sec**	
	N	***S.D***	***N***	***S.D***
S1	8.380	149.7	8.850	68.59
S2	23.18	147.0	2.600	66.74
S3	2.590	146.3	2.700	65.00
S4	21.15	145.5	2.380	65.45
S5	2.180	145.8	1.240	66.42

TABLE I: Distance of roots

of systematic downloading and during a normal period. There may be situations when the magnitude of the root is close to unity, but it will not be classified as systematic download, since the behavior of the roots to move towards the unstable region will not persist for long. While, at the time of systematic downloading, the magnitude of the roots will be consistently high for successive frames. Figure 5 shows the trajectory of roots at the time of systematic downloading. It is evident that consecutive roots are closer to the unit circle. Figure 5 also shows the clustering of roots, which suggests that roots pertaining to same download behavior get clustered together.

Clustering is performed on the roots as indicated in Figure 4. The points indicated by red color correspond to that of systematic downloading, while the points



(a) Polling Interval− 5 seconds

(b) Polling Interval− 10 seconds

Fig. 4: Clustering of AR roots. Blue and Red points depict normal and systematic download AR roots, respectively.

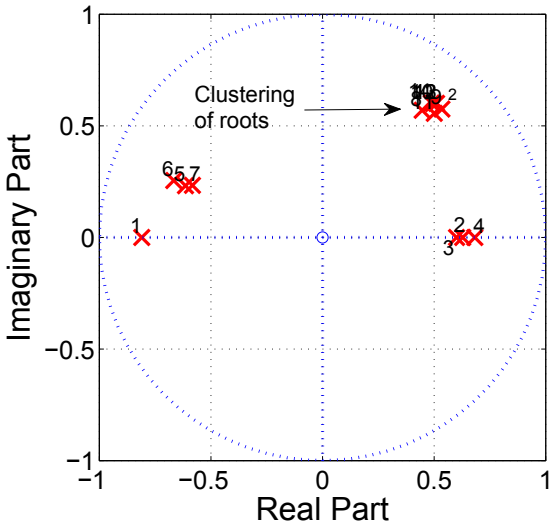


Fig. 5: Trajectory of AR roots for polling interval of 5 seconds

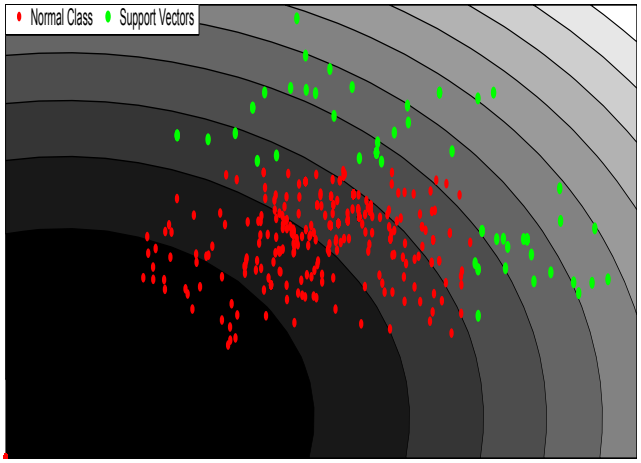


Fig. 6: Description of normal data along with support vectors using Gaussian kernel

indicated by blue color correspond to normal behavior. In Figure 4, it can be observed that the clusters formed are more or less identical across different polling intervals. This character of the time series can be exploited by an agent, to pro-actively quell systematic downloading.

B. Detection using SVDD

As described above, stability of the system can be used to detect systematic downloading. But tracking the trajectory of the roots requires certain rules and may not be robust and efficient. In this section, performance of SVDD has been investigated using the derived features namely, roots of the AR model and LSPs. The following set of features were obtained:

1) AR model roots
2) Magnitude and Phase of AR roots (MPAR)
3) LSP roots
4) Distance between LSP roots (DLSP)

These features were then modeled using SVDD. Figure 6 exemplifies data description, using the distance between the LSP roots of normal data for LP order 4 as a feature. It indicates that normal data is enclosed in a hypersphere bounded with support vectors. represented as green solid points. For better modeling of the features, parameters of SVDD were fine-tuned using the two pass method as mentioned in [23]. The method is described only for C but the same procedure was used for all the parameters. A line search was performed to select the parameters for building the model. In this process, first C was varied from 2^{-12} to 2^{-2}. $BestC$ was determined by choosing maximum accuracy, F-measure and minimum number of support vectors. In the second pass, the value of C was varied in the range $0.2 \times bestC ... 0.9 \times bestC$. These experiments were performed with linear, polynomial and Gaussian kernel. Grid search [23] was used to estimate the parameters of the Gaussian kernel ($variance$ of the Gaussian distribution) and the polynomial kernel ($degree$).

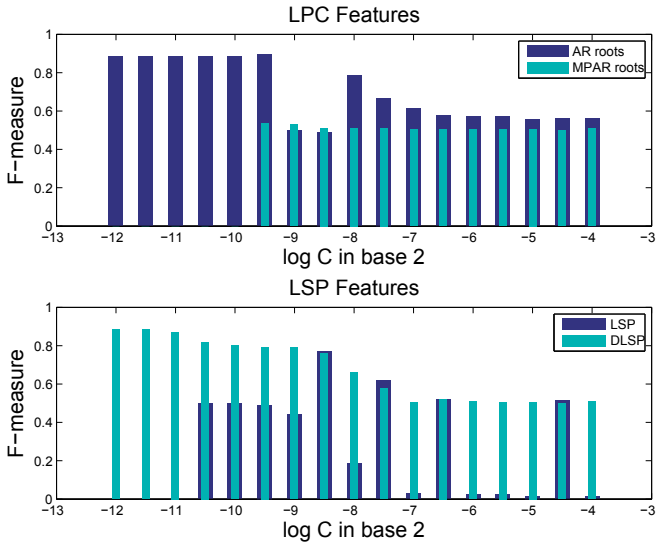


Fig. 7: F-measure for linear kernel at different values of C for all the four features

		Linear		Polynomial		Gaussian	
		5 sec	10 sec	5 sec	10 sec	5 sec	10 sec
AR roots	*F-Measure*	0.88	0.85	0.88	0.86	0.89	0.85
	Accuracy	99.5	99.2	99.5	99.3	99.5	99.39
MPAR	*F-Measure*	0.59	0.69	0.55	0.70	0.54	0.69
	Accuracy	98.4	99.0	99.4	99.4	98.1	99.3
LSP roots	*F-Measure*	0.89	0.87	**0.90**	0.87	**0.90**	0.87
	Accuracy	99.4	99.4	**99.5**	99.3	**99.5**	99.3
DLSP	*F-Measure*	0.88	0.85	0.88	0.80	0.89	0.86
	Accuracy	99.4	99.2	99.4	99.1	99.4	99.3

TABLE II: Accuracy and F-measure using the derived features for linear, polynomial and Gaussian kernel at 5 and 10 seconds polling interval.

Figure 7 shows the F-measure for AR roots and LSP using linear kernel for different values of C with polling interval of the time series being 5 seconds. It can be inferred that, with finely tuned SVDD parameters, distance between LSP roots outperform other features. This fact is further verified in Table II.

Table II shows the best F-measure and accuracy obtained for linear, polynomial and Gaussian kernel at 5 and 10 seconds polling interval. It is observed that LSP roots show better F-measure and accuracy than other features.

V. Conclusion

In this paper, a novel technique has been developed for detection of systematic downloading at the campus proxy using a time series model on download traffic requests. AR model has been employed and roots of the system have been used to distinguish between normal and abnormal behavior of the system. At the time of systematic downloading, roots of the AR polynomial tend to move towards the boundary of the unit circle and form separate clusters in Z-domain. As the magnitude of the roots increases during a systematic download, we conjecture that the system moves to an unstable state during systematic downloading. It was concluded that out of the two set of features obtained, each from AR (roots and magnitude and phase of each root) and LSP (LSP coefficients and distance between the roots of $P(z)$ and $Q(z)$) , LSP roots performed extremely well in detecting anomalies using SVDD with a reduction in number of miss.

Acknowledgment

This work was carried out under the IU-ATC project funded by the Department of Science and Technology (DST), Government of India and the UK EPSRC Digital Economy Programme.

References

[1] "IEEE xplore digital library," http://www.ieee.org/.

[2] "The ACM digital library," http://dl.acm.org/.

[3] Journal storage. http://www.jstor.org/.

[4] A. Waller, "When the licensor calls: Some thoughts on systematic downloading," *Informed Librarian Online*, August 2007. [Online]. Available: http://hdl.handle.net/10760/10206

[5] Journal storage. http://docs.jstor.org/.

[6] G. Baker and C. Tenopir, "Managing the unmanageable: Systematic downloading of electronic resources by library users," *Journal of Library Administration*, vol. 44, pp. 11–24, 2006.

[7] C. James and H. A. Murthy, "Time series models and its relevance to modeling tcp syn based dos attacks." in *NGI*. IEEE, 2011, pp. 1–8.

[8] D. Divakaran, H. Murthy, and T. Gonsalves, "Detection of syn flooding attacks using linear prediction analysis," in *Networks, 2006. ICON '06. 14th IEEE International Conference on*, vol. 1, 2006, pp. 1–6.

[9] B. Zhou, D. He, Z. Sun, and W. H. Ng, "Network traffic modeling and prediction with arima/garch," in *HET-NETsâĂŹ 06 Conference*. Citeseer, 2005, pp. 1–10.

[10] Q. Wu and Z. Shao, "Network anomaly detection using time series analysis," in *Autonomic and Autonomous Systems and International Conference on Networking and Services, 2005. ICAS-ICNS 2005. Joint International Conference on*, 2005, pp. 42–42.

[11] M. Thottan and C. Ji, "Proactive anomaly detection using distributed intelligent agents," *Network, IEEE*, vol. 12, no. 5, pp. 21–27, 1998.

[12] ——, "Anomaly detection in ip networks," *Signal Processing, IEEE Transactions on*, vol. 51, no. 8, pp. 2191–2204, Aug 2003.

[13] T. Ndousse and T. Okuda, "Computational intelligence for distributed fault management in networks using fuzzy cognitive maps," in *Communications, 1996. ICC '96, Conference Record, Converging Technologies for Tomorrow's Applications. 1996 IEEE International Conference on*, vol. 3, Jun 1996, pp. 1558–1562 vol.3.

[14] S. C. Chin, A. Ray, and V. Rajagopalan, "Symbolic time series analysis for anomaly detection: A comparative evaluation," *Signal Processing*, vol. 85, no. 9, pp. 1859 – 1868, 2005. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0165168405001039

[15] H. Liu and M. S. Kim, "Real-time detection of stealthy ddos attacks using time-series decomposition," in *Communications (ICC), 2010 IEEE International Conference on*, May 2010, pp. 1–6.

[16] J. Makhoul, "Linear prediction: A tutorial review," *Proceedings of the IEEE*, vol. 63, no. 4, pp. 561–580, 1975.

[17] K. Paliwal, "On the use of line spectral frequency parameters for speech recognition," *Digital signal processing*, vol. 2, no. 2, pp. 80–87, 1992.

[18] D. M. J. Tax and R. P. W. Duin, "Support vector data description," *Mach. Learn.*, vol. 54, no. 1, pp. 45–66, Jan. 2004. [Online]. Available: http://dx.doi.org/10.1023/B:MACH.0000008084.60811.49

[19] S. Sait, M. Kumar, and H. Murthy, "User traffic classification for proxy-server based internet access control," in *Signal Processing and Communication Systems (ICSPCS), 2012 6th International Conference on*, 2012, pp. 1–9.

[20] Wampserver. http://www.wampserver.com/en/.

[21] G. E. Box, G. M. Jenkins, and G. C. Reinsel, *Time series analysis: forecasting and control*. Wiley. com, 2013.

[22] M. Moya, M. Koch, and L. Hostetler, "One-class classifier networks for target recognition applications," *NASA STI/Recon Technical Report N*, vol. 93, p. 24043, 1993.

[23] C.-W. Hsu, C.-C. Chang, C.-J. Lin *et al.*, "A practical guide to support vector classification," 2003.