Supporting Cyber Attack Detection via Non-Linear Analytic Prediction of IP Addresses: A Big Data Analytics Technique

Alfredo Cuzzocrea*

iDEA Lab, University of Calabria, Rende, Italy
and LORIA, Nancy, France
alfredo.cuzzocrea@unical.it

Enzo Mumolo

University of Trieste, Trieste, Italy
mumolo@units.it

Edoardo Fadda

Politecnico di Torino and ISIRES, Torino, Italy
edoardo.fadda@polito.it

Marco Tessarotto

University of Trieste, Trieste, Italy
marco.tessarotto@regione.fvg.it

Abstract

Computer network systems are often subject to several types of attacks. For example the distributed Denial of Service (DDoS) attack introduces an excessive traffic load to a web server to make it unusable. A popular method for detecting attacks is to use the sequence of source IP addresses to detect possible anomalies. With the aim of predicting the next IP address, the Probability Density Function of the IP address sequence is estimated. Prediction of source IP address in the future access to the server is meant to detect anomalous requests. In other words, during an access to the server, only predicted IP addresses are permitted and all others are blocked. The approaches used to estimate the Probability Density Function of IP addresses range from the sequence of IP addresses seen previously and stored in a database to address clustering, normally used by combining the K-Means algorithm. Instead, in this paper we consider the sequence of IP addresses as a numerical sequence and develop the nonlinear analysis of the numerical sequence. We used nonlinear analysis based on Volterra's kernels and Hammerstein's models.

1 Introduction

User modeling is an important task for web applications dealing with large traffic flows. They can be used for a variety of applications such as to predict future situations or classify current states. Furthermore, user modeling can improve detection or mitigation of Distributed Denial of Service (DDoS) attack [11, 15, 13],

DOI reference number: 10.18293/DMSVIVA2020-018

* This research has been made in the context of the Excellence Chair in Computer Engineering – Big Data Management and Analytics at LORIA, Nancy, France

improve the quality of service (QoS) [16, 10], individuate click fraud detection and optimize traffic management. In peer-to-peer (P2P) overlay networks, IP models can also be used for optimizing request routing [1]. Those techniques are used by severs for deciding how to manage the actual traffic. In this context, also outlier detection methods are often used if only one class is known. If, for example, an Intrusion Prevention System wants to mitigate DDoS attacks, it usually has only seen the normal traffic class before and it has to detect the outlier class by its different behaviour. In this paper we deal with the management of DDos because nowadays it has become a major threat in the internet. Those attacks are done by using a large scaled networks of infected PCs (bots or zombies) that combine their bandwidth and computational power in order to overload a publicly available service and denial it for legal users. Due to the open structure of the internet, all public servers are vulnerable to DDoS attacks. The bots are usually acquired automatically by hackers who use software tools to scan through the network, detecting vulnerabilities and exploiting the target machine. Furthermore, there is also a strong need to mitigate DDoS attacks near the target, which seems to be the only solution to the problem in the current internet infrastructure. The aim of such a protection system is to limit their destabilizing effect on the server through identifying malicious requests. There are multiple strategies with dealing with DDoS attacks. The most effective ones are the near-target filtering solutions. They estimates normal user behavior based on IP packet header information. Then, during an attack the access of outliers is denied. One parameter that all methods have in common is the source IP address of the users. It is the main discriminant for DDoS traffic classification. However, the methods of storing IP addresses and estimating their density in the huge IP address space, are different. In this paper, we present a novel approach based on system identification techniques and, in particular, on the Hammerstein models. A broader overview of state-of-the-art research on the available methods for DDoS traffic classification is given by [9]. The paper is organized as follows. In Sections 2 and 3 we present our proposed a technique based based on Hammerstein models and we recall some similar model. Although DDoS mitigation is the most important practical application for IP density estimation, we do not restrict the following work on this topic. Our generic view on IP density estimation may be valuable to other applications as well. One might think of preferring regular customers in overload situations (flash crowd events), identifying non-regular users on websites during high click rates on online advertisements (click fraud detection) or optimizing routing in peer-to-peer networks. Finally, in Section 4 we draw conclusions and indicate future work. The extended version of this paper appears in [7].

2 Analytic Prediction

Data driven identification of mathematical models of physical systems (i.e. nonlinear) starts with representing the systems as a black box. In other terms, while we may have access to the inputs and outputs, the internal mechanisms are totally unknown to us. Once a model type is chosen to represent the system, its parameters are estimated through an optimization algorithm so that eventually the model mimics at a certain level of fidelity the inner mechanism of the nonlinear system or process using its inputs and outputs. This approach is, for instance, widely used in the related *big data analytics* area (e.g., [6, 3, 5, 8])

In this work, we consider a particular sub-class of nonlinear predictors: the Linear-in-the-parameters (LIP) predictors. LIP predictors are characterized by a linear dependence of the predictor output on the predictor coefficients. Such predictors are inherently stable, and that they can converge to a globally minimum solution (in contrast to other types of nonlinear filters whose cost function may exhibit many local minima) avoiding the undesired possibility of getting stuck in a local minimum. Let us consider a causal, time-invariant, finite-memory,continuous nonlinear predictor as described in (1).

$$\hat{s}(n) = f[s(n-1), \ldots, s(n-N)] \tag{1}$$

where $f[\cdot]$ is a continuous function, $s(n)$ is the input signal and $\hat{s}(n)$ is the predicted sample. We can expand $f[\cdot]$ with a series of basis functions $f_i(n)$, as shown in (2).

$$\hat{s}(n) = \sum_{i=1}^{\infty} h(i) f_i[s(n-i)] \tag{2}$$

where $h(i)$ a re proper coefficients. To make (2) realizable we truncate the series to the first N terms, thus we obtain

$$\hat{s}(n) = \sum_{i=1}^{N} h(i) f_i[s(n-i)] \tag{3}$$

In the general case, a linear-in-the-parameters nonlinear predictor is described by the input-output relationship reported in (4).

$$\hat{s}(n) = \vec{H}^T \vec{X}(n) \tag{4}$$

where $\vec{H}^T$ is a row vector containing predictor coefficients and $\vec{X}(n)$ is the corresponding column vector whose elements are nonlinear combinations and/or expansions of the input samples.

2.1 Linear Predictor

Linear prediction is a well known technique with a long history [12]. Given a time series $\vec{X}$, linear prediction is the optimum approximation of sample $x(n)$ with a linear combination of the N most recent samples. That means that the linear predictor is described as eq. (5).

$$\hat{s}(n) = \sum_{i=1}^{N} h_1(i)s(n-i) \tag{5}$$

or in matrix form as

$$\hat{s}(n) = \vec{H}^T \vec{X}(n) \tag{6}$$

where the coefficient and input vectors are reported in (7) and (8).

$$\vec{H}^T = \begin{bmatrix} h_1(1) & h_1(2) & \dots & h_1(N) \end{bmatrix} \tag{7}$$

$$\vec{X}^T = \begin{bmatrix} s(n-1) & s(n-2) & \dots & s(n-N) \end{bmatrix} \tag{8}$$

2.2 Non-Linear Predictor based on Volterra Series

As well as Linear Prediction, Non Linear Prediction is the optimum approximation of sample $x(n)$ with a non linear combination of the N most recent samples. Popular nonlinear predictors are based on Volterra series [14]. A Volterra predictor based on a Volterra series truncated to the second term is reported in (9).

$$\hat{x}(n) = \sum_{i=1}^{N_1} h_1(i)x(n-i) + \sum_{i=1}^{N_2}\sum_{j=i}^{N_2} h_2(i,j)x(n-i)x(n-j) \tag{9}$$

where the symmetry of the Volterra kernel (the h coefficients) is considered. In matrix terms, the Volterra predictor is represented in (10).

$$\hat{s}(n) = \vec{H}^T \vec{X}(n) \tag{10}$$

where the coefficient and input vectors are reported in (12) and (12).

$$\vec{H}^T = \begin{bmatrix} h_1(1) & h_1(2)\dots h_1(N) \\ h_2(1,1) & h_2(1,2)\dots h_2(N_2,N_2) \end{bmatrix} \tag{11}$$

$$\vec{X}^T = \begin{bmatrix} s(n-1) & s(n-2)\dots s(n-N_1) \\ s^2(n-1) & s(n-1)s(n-2)\dots s^2(n-N_2) \end{bmatrix} \tag{12}$$

2.3 Non-Linear Predictor based on Functional Link Artificial Neural Networks (FLANN)

FLANN is a single layer neural network without hidden layer. The nonlinear relationships between input and output are captured through function expansion of the input signal exploiting suitable orthogonal polynomials. Many authors used for examples trigonometric, Legendre and Chebyshev polynomials. However, the most frequently used basis function used in FLANN for function expansion are trigonometric polynomials [17]. The FLANN predictor can be represented by eq.(13).

$$\hat{s}(n) = \sum_{i=1}^{N} h_1(i)s(n-i) + \sum_{i=1}^{N}\sum_{j=1}^{N} h_2(i,j)\cos[i\pi s(n-j)] + \sum_{i=1}^{N}\sum_{j=1}^{N} h_2(i,j)\sin[i\pi s(n-j)] \tag{13}$$

Also in this case the Flann predictor can be represented using the matrix form reported in (14).

$$\hat{s}(n) = \vec{H}^T \vec{X}(n) \tag{14}$$

where the coefficient and input vectors of FLANN predictors are reported in (15) and (16).

$$\vec{H}^T = \begin{bmatrix} h_1(1) & h_1(2)\dots h_1(N) \\ h_2(1,1) & h_2(1,2)\dots h_2(N_2,N_2) \\ h_3(1,1) & h_3(1,2)\dots h_3(N_2,N_2) \end{bmatrix} \tag{15}$$

$$\vec{X}^T = \begin{bmatrix} s(n-1) & \dots & s(n-N) & \\ \cos[\pi s(n-1)] & \dots & \dots & \cos[\pi s(n-N_2)] \\ \sin[\pi s(n-1)] & \dots & \dots & \sin[\pi x(s-N_2)] \end{bmatrix} \tag{16}$$

2.4 Non-Linear Predictors based on Hammerstein Models

Previous research [2] shown that many real nonlinear systems, spanning from electromechanical systems to audio systems, can be modeled using a static non-linearity. These terms capture the system nonlinearities, in series with a linear function, which capture the system dynamics as shown in Figure 1.

Indeed, the front-end of the so called Hammerstein Model is formed by a nonlinear function whose input is the system input. Of course the type of non-linearity depends on the actual physical system to be modeled.

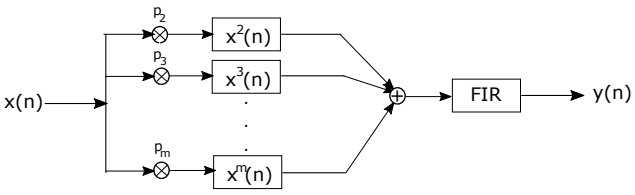

Figure 1. Representation of the Hammerstein Models

The output of the nonlinear function is hidden and is fed as input of the linear function. In the following, we assume that the non-linearity is a finite polynomial expansion, and the linear dynamic is realized with a Finite Impulse Response (FIR) filter. Furthermore, in contrast with [2], we assume a mean error analysis and we postpone the analysis in the robust framework in future work. In other word,

$$z(n) = p(2)x^2(n) + p(3)x^3(n) + \dots p(m)x^m(n) = \\ = \sum_{i=2}^{M} p(i)x^i(n) \quad (17)$$

On the other hand, the output of the FIR filter is:

$$y(n) = h_0(1)z(n-1) + \dots + h_0(N)z(n-N) = \\ = \sum_{j=1}^{N} h_0(j)z(n-j) \quad (18)$$

Substituting (17) in (18) we have:

$$y(n) = \\ \sum_{i=1}^{N} h_0(i)z(n-i) = \sum_{j=1}^{N} h_0(j) \sum_{i=2}^{M} p(i)x^i(n-j) = \\ \sum_{i=2}^{M} \sum_{j=1}^{N} h_0(j)p(i)x^i(n-j) \quad (19)$$

Setting $c(i,j) = h_0(j)p(i)$ we write

$$y(n) = \sum_{i=2}^{M} \sum_{j=1}^{N} c(i,j)x^i(n-j) \quad (20)$$

This equation can be written in matrix form as

$$\hat{s}(n) = \vec{H}^T \vec{X}(n) \quad (21)$$

where

$$\vec{H}^T = \begin{bmatrix} c(2,1) & c(2,2) \dots c(2,N_2) \\ c(3,1) & c(3,2) \dots c(3,N_2) \\ \dots c(M,1) & c(M,2) \dots c(M,N) \end{bmatrix} \quad (22)$$

$$\vec{X}^T = \\ \begin{bmatrix} s^2(n-2) & \dots & s^2(n-N) & \\ s^3(n-2) & \dots & s^3(n-N) & \\ s^M(n-2) & \dots & s^M(n-1) & \dots s^M(n-N) \end{bmatrix} \quad (23)$$

3 Estimation of Predictor Parameters

So far we saw that all the predictors can be expressed, at time instant n, as

$$\hat{s}(n) = \vec{H}^T \vec{X}(n) \quad (24)$$

with different definitions of the input, $\vec{X}(n)$, end parameters vectors $\vec{H}^T$. There are two well known possibilities for estimating the optimal parameter vector.

3.1 Block-based Approach

The Minimum Mean Square estimation is based on the minimization of the mathematical expectation of the squared prediction error $e(n) = s(n) - \hat{s}(n)$

$$E[e^2] = E[(s(n) - \hat{s}(n))^2] = E[(s(n) - \vec{H}^T \vec{X}(n))^2] \quad (25)$$

The minimization of (25) is obtain by setting to zero the Laplacian of the mathematical expectation of the squared prediction error:

$$\nabla_H E[e^2] = E[\nabla_H e^2] = E[2e(n)\nabla_H e] = 0 \quad (26)$$

which leads to the well known unique solution

$$\vec{H}_{opt} = \vec{R}_{xx}^{-1} \vec{R}_{sx} \quad (27)$$

where

$$\vec{R}_{xx}(n) = E[\vec{X}(n)\vec{X}^T(n)] \quad (28)$$

is the statistical auto-correlation matrix of the input vector $\vec{X}(n)$ and

$$\vec{R}_{sx}(n) = E[s(n)\vec{X}(n)] \quad (29)$$

is the statistical cross-correlation vector between the signal $s(n)$ and the input vector $\vec{X}(n)$. The mathematical expectations of the auto and cross correlation are estimated using

$$\vec{R}_{xx}(n) = \frac{\sum_{k=1}^{n} \vec{X}(n)\vec{X}^T(n)}{n} \quad (30)$$

is the statistical auto-correlation matrix of the input vector $\vec{X}(n)$ and

$$\vec{R}_{sx}(n) = \frac{\sum_{k=1}^{n} s(k)(n)\vec{X}(n)}{n} \quad (31)$$

3.2 Adaptive Approach

Let us consider a general second order terms of a Volterra predictor

$$y(n) = \sum_{k=0}^{N-1} \sum_{r=0}^{N-1} h_2(k,r)x(n-k)x(n-r) \tag{32}$$

It can be generalized for higher order term as

$$\sum_{k_1=1}^{N} \cdots \sum_{k_p=1}^{N} c_{k_1} \cdots c_{k_p} H_p \left\{x_{k_1}(n), \cdots x_{k_p}(n)\right\} \tag{33}$$

where

$$\sum_{k=1}^{N} c_k x_k(n). \tag{34}$$

For the sake of simplicity and without loss of generality, we consider a Volterra predictor based on a Volterra series truncated to the second term

$$\hat{r}(n) = \sum_{i=1}^{N_1} h_1(i)r(n-i) + \sum_{i=1}^{N_2} \sum_{j=i}^{N_2} h_2(i,j)r(n-i)r(n-j) \tag{35}$$

By defining

$$H^T(n) = |h_1(1), \cdots, h_1(N_1), h_2(1,1), \cdots, h_2(N_2, N_2)| \tag{36}$$

and

$$\begin{array}{c} X^T(n) = \big|r(n-1), \cdots, r(n-N_1), r^2(n-1) \\ r(n-1)r(n-2), \cdots, r^2(n-N_2)\,| \end{array} \tag{37}$$

Eq (35) can be rewritten as follows

$$\hat{r}(n) = H^T(n)X(n). \tag{38}$$

In order to estimate the best parameters H, we consider the following loss function

$$J_n(H) = \sum_{k=0}^{n} \lambda^{n-k} \left[\hat{r}(k) - H^T(n)X(k)\right]^2 \tag{39}$$

where λ^{n-k} weights the relative importance of each squared error. In order to find the H that minimizes the convex function (39) it is enough to impose its gradient to zero, i.e.,

$$\nabla_H J_n(H) = 0 \tag{40}$$

That is equivalent to

$$R_{XX}(n)H(n) = R_{rX}(n) \tag{41}$$

where

$$\begin{array}{c} R_{XX}(n) = \sum_{k=0}^{n} \lambda^{n-k} X(k)X^T(k) \\ R_{rX}(n) = \sum_{k=0}^{n} \lambda^{n-k} r(k)X(k) \end{array} \tag{42}$$

It follows that the best H can be computed by

$$H(n) = R_{XX}^{-1}(n)R_{rX}(n) \tag{43}$$

Since

$$R_{XX}(n) = \lambda R_{XX}(n-1) + X(n)X^T(n) \tag{44}$$

it follows that

$$R_{XX}^{-1}(n) = \frac{1}{\lambda}\left[R_{XX}^{-1}(n-1) - k(n)X^T(n)R_{XX}^{-1}(n-1)\right] \tag{45}$$

where $k(n)$ is equal to

$$k(n) = \frac{R_{XX}^{-1}(n-1)X(n)}{\lambda + X^T(n)R_{XX}^{-1}(n-1)X(n)} \tag{46}$$

Instead, matrix $R_{rX}(n)$ in (43) can be written as

$$R_{rX}(n) = \lambda R_{rX}(n-1) + r(n)X(n) \tag{47}$$

Thus, inserting Eq (47) and Eq (45) in Eq (43) and rearranging the terms, we obtain

$$H(n) = H(n-1) + R_{XX}^{-1}(n)X(n)\epsilon(n) \tag{48}$$

where

$$\epsilon = \hat{r}(n) - H^T(n-1)X(n) \tag{49}$$

By recalling Eq. (46), we can write Eq. (48) as

$$H(n) = H(n-1) + \epsilon(n)k(n) \tag{50}$$

By introducing, the new notation,

$$F(n) = S^T(n-1)X(n) \tag{51}$$

The previous equations can be resumed by the following system

$$\begin{cases} L(n) = S(n-1)F(n) \\ \beta(n) = \lambda + F^T(n)F(n) \\ \alpha(n) = \frac{1}{\beta(n) + \sqrt{\lambda\beta(n)}} \\ S(n) = \frac{1}{\sqrt{\lambda}}\left[S(n-1) - \alpha(n)L(n)F^T(n)\right] \\ \epsilon(n) = \hat{r}(n-1) - \alpha(n)L(n)F^T(n) \\ \epsilon(n) = H(n-1) + L(n)\frac{\epsilon(n)}{\beta(n)} \end{cases} \tag{52}$$

It should be noted that by using Eq (52) the estimation adapts in each step in order to decrease the error. Thus, the system structure is somehow similar to the Kalman filter.

Finally, we define the estimation error as

$$e(n) = r(n) - H^T(n)X(n) \tag{53}$$

It is worth noting that the computation of the predicted value from Eq. (38) requires $6N_{\text{tot}} + 2N_{\text{tot}}^2$ operations, where $N_{\text{tot}} = N_1 + N_2\left(N_2 + 1\right)/2$.

4 Conclusions

In this paper, we presented a new way to deal with cyber attack by using Hammerstein models. Future work will have two objectives. First, we want to consider the problem in a stochastic optimization settings. Second, we want to test the approach on other case studies, by also exploiting *knowledge management methodologies* (e.g., [4]).

Acknowledgements

This research has been partially supported by the French PIA project "Lorraine Université d'Excellence", reference ANR-15-IDEX-04-LUE.

References

[1] A. Agrawal and H. Casanova. Clustering hosts in p2p and global computing platforms. pages 367– 373, 06 2003.

[2] V. Cerone, E. Fadda, and D. Regruto. A robust optimization approach to kernel-based nonparametric error-in-variables identification in the presence of bounded noise. In *2017 American Control Conference (ACC)*. IEEE, may 2017.

[3] G. Chatzimilioudis, A. Cuzzocrea, D. Gunopulos, and N. Mamoulis. A novel distributed framework for optimizing query routing trees in wireless sensor networks via optimal operator placement. *J. Comput. Syst. Sci.*, 79(3):349–368, 2013.

[4] A. Cuzzocrea. Combining multidimensional user models and knowledge representation and management techniques for making web services knowledge-aware. *Web Intelligence and Agent Systems*, 4(3):289–312, 2006.

[5] A. Cuzzocrea and E. Bertino. Privacy preserving OLAP over distributed XML data: A theoretically-sound secure-multiparty-computation approach. *J. Comput. Syst. Sci.*, 77(6):965–987, 2011.

[6] A. Cuzzocrea, R. Moussa, and G. Xu. Olap*: Effectively and efficiently supporting parallel OLAP over big data. In *Model and Data Engineering - Third International Conference, MEDI 2013, Amantea, Italy, September 25-27, 2013. Proceedings*, pages 38–49, 2013.

[7] A. Cuzzocrea, E. Mumolo, E. Fadda, and M. Tessarotto. A novel big data analytics approach for supporting cyber attack detection via non-linear analytic prediction of ip addresses. In *Computational Science and Its Applications - ICCSA 2020 - 20th International Conference, Cagliari, Italy, July 1-4, 2020, Proceedings*, 2020.

[8] A. Cuzzocrea and V. Russo. Privacy preserving OLAP and OLAP security. In *Encyclopedia of Data Warehousing and Mining, Second Edition (4 Volumes)*, pages 1575–1581. 2009.

[9] S. Dietrich, N. Long, and D. Dittrich. Analyzing distributed denial of service tools: The shaft case. pages 329–339, 12 2000.

[10] E. Fadda, P. Plebani, and M. Vitali. Optimizing monitorability of multi-cloud applications. pages 411–426, 06 2016.

[11] M. Goldstein, C. Lampert, M. Reif, A. Stahl, and T. Breuel. Bayes optimal ddos mitigation by adaptive history-based ip filtering. In *Seventh International Conference on Networking (icn 2008)*, pages 174–179, 2008.

[12] J. Makhoul. Linear prediction: A tutorial review. *Proceedings of the IEEE*, 63(4):561–580, 1975.

[13] G. Pack, J. Yoon, E. Collins, and C. Estan. On filtering of ddos attacks based on source address prefixes. pages 1–12, 08 2006.

[14] Z. Peng and C. Changming. Volterra series theory: A state-of-the-art review. *Chinese Science Bulletin (Chinese Version)*, 60:1874, 01 2015.

[15] H.-X. Tan and W. Seah. Framework for statistical filtering against ddos attacks in manets. pages 8 pp.–, 01 2006.

[16] Y. Yang and C.-H. Lung. The role of traffic forecasting in qos routing - a case study of time-dependent routing. pages 224 – 228 Vol. 1, 06 2005.

[17] H. Zhao and J. Zhang. Adaptively combined fir and functional link artificial neural network equalizer for nonlinear communication channel. *IEEE Transactions on Neural Networks*, 20(4):665–674, 2009.