Schedae Informaticae Vol. 24 (2015): 63–71
doi: 10.4467/20838476SI.15.006.3028

Incoherent Dictionary Learning for Sparse Representation in Network Anomaly Detection

Tomasz Andrysiak, Łukasz Saganowski
UTP University of Science and Technology
Institute of Telecommunications
ul. Kaliskiego 7, 85-789 Bydgoszcz, Poland
e-mail: *tomasz.andrysiak@utp.edu.pl, lukasz.saganowski@utp.edu.pl*

Abstract. In this article we present the use of sparse representation of a signal and incoherent dictionary learning method for the purpose of network traffic analysis. In learning process we use 1D INK-SVD algorithm to detect proper dictionary structure. Anomaly detection is realized by parameter estimation of the analyzed signal and its comparative analysis to network traffic profiles. Efficiency of our method is examined with the use of extended set of test traces from real network traffic. Received experimental results confirm effectiveness of the presented method.

Keywords: dictionary learning, sparse representation, anomaly detection.

1. Introduction

Dynamic and continuous development of local and global information systems requires proper protection against cybercriminal attacks. In their attempts, they apply more and more excellent techniques of penetration and hacking infrastructures of network systems. Most often, it is realised by dynamically spreading network malware. Growing threat and universality of using information resources stimulate ongoing development of safety and protection network systems [1]. Essential mechanisms of safety supervision over computer networks are Intrusion Detection/Prevention Systems (IDS/IPS). Their action consists in monitoring and detecting attacks directed onto resources of computer systems. The main aim of IDS systems is detecting successful attacks. However, they are also applied in monitoring and registering of attempts to break securities of an attacked computer system [2, 3]. IDS intrusion detection systems can be classified as belonging to one of two groups using diverse techniques of threat identification. One is based on detection of known attacks with the use of defined and specific features, named signatures. The other uses the idea of monitoring normal work of a given system in order to detect anomalies that may be indicative of an intruder. It allows for detecting attempts of intrusion consisting of numerous network connections [4]. Primary advantage of methods based on anomaly detection is their ability to indicate unknown attacks. The above mentioned approach uses knowledge of not how a particular attack looks like, but what does not correspond to specified network traffic norms. Therefore, IDS/IPS systems based on the use of anomalies are more effective in comparison with systems applying signatures while detecting unknown, new types of attacks. In this article we present the use of incoherent dictionary learning method and sparse representation of a signal for specific time series describing the analysed network traffic. Anomaly detection is realized by estimated parameters of the sparse representation of a signal and comparative analysis to network traffic profiles [5]. This paper is organized as follows: after the introduction, in Section 2, the sparse representation of a signal for date traffic prediction is described in details. Then, in Section 3 the incoherent dictionary learning method based on 1D INK-SVD algorithm estimation is shown. Implementation details and experimental results are described in Section 4. Conclusions are given thereafter.

2. Sparse representation of a signal

The difficulty of locating the sparse representation of a signal in a specific overcomplete dictionary might be described in the following way: sparse representation is searching for the sparse explanation of decomposition coefficients C, standing for the signal S, over the redundant dictionary, where the balance is smaller than a given constant T, and it can be presented as:

$$\min \|c\|_0 \ \textit{subject to} \ \left\| S - \sum\nolimits_{k=0}^{K-1} c_k d_k \right\| < \varepsilon, \tag{1}$$

where $\|\cdot\|_0$ is the l^0 norm figuring up the nonzero items of a vector, $c_k \in C$ representing a set of projection coefficients and d_k are the elements of the redundant dictionary D. It is difficult to obtain the outcome of the equation (1) because of its specification of combinational optimization. We can find the best possible solutions to the issues by iterative methods, e.g. the matching pursuit algorithm, as well as the orthogonal pursuit algorithm. The matching pursuit (MP) algorithm is proposed in [6]. The goal of the above mentioned is to approximate to the input signal S by sequential choice of vectors from the dictionary D. The algorithm proceeds greedy tactics in which the basis vector best aligned with the residual vector is selected at each iteration. Signal S can be noted down as the weighted sum of these items

$$S = \sum\nolimits_{i=0}^{n-1} c_i d_i + r^n s, \tag{2}$$

where $r^n s$ is residual in an n-term total. The orthogonal matching pursuit (OMP) algorithm is an upgrade of MP algorithm and is presented in [7, 6]. Resemblance between two algorithms has a greedy structure, however, what concerns discrepancies is that OMP algorithm requires selected atoms to be orthogonal in each decomposition stage. The algorithm chooses φ_p in the p^{th} iteration by spotting the vector best aligned with the residual gained by projecting $r^p s$ onto the dictionary elements, that is

$$\varphi_p = \arg\max\nolimits_{i \in \Phi_p} \left| \langle r^p s, d_i \rangle \right|, \ \varphi_p \notin \Phi_{p-1} \tag{3}$$

and

$$c_p = \left\langle r^{p-1} s, d_{\varphi_p} \right\rangle, \tag{4}$$

where $\Phi_p = \{\varphi_1, \varphi_2, ..., \varphi_{p-1}, \varphi_p\}$ is the index vector and $D_p = \{d_{\varphi_1}, d_{\varphi_2}, ..., d_{\varphi_p}\}$ is the columns of the dictionary D_p. The initial set of value is $\Phi_0 = \emptyset$ and $D_0 = \emptyset$. The complication with the re-selection is to be omitted with the stored dictionary. If $\varphi_p \notin \varphi_{p-1}$ then the index set is updated as $\Phi_p = \Phi_{p-1} \cup \varphi_p$ and $D_p = D_{p-1} \cup d_{\varphi_p}$. In other case, $\Phi_p = \Phi_{p-1}$ and $D_p = D_{p-1}$. The residual $r^p s$ is calculated as

$$r^p s = r^{p-1} s - D_p \left(D_p^T D_p\right)^{-1} D_p^T r^{p-1} s, \tag{5}$$

where $D_p^T D_p$ is the Gram matrix. The algorithm concludes when residual of a signal is smaller than adequate limit

$$\|r^p s\| < th, \tag{6}$$

where th is the approximation error.

The final complication of OMP can be diminished by using the Cholesky Factorization, the QR Factorization, or the Matrix Inversion Lemma [8, 9, 10]. In our practical realization we use a progressive Cholesky upgrade process to diminish the effort connected with the matrix inversion.

3. Dictionary learning

A dictionary learning problem for sparse approximation is finding the best dictionary D to represent the signal S as sparse composition, by solving

$$\min\nolimits_{D,C} \left\{ \|S - DC\|_F^2 \right\} \ \textit{subject to} \ \forall_i \, \|c_i\|_0 \leq T, \tag{7}$$

where $\|\cdot\|_F^2$ is the Frobenius norm and T is a fixed and predetermined number of nonzero entries. The commonly applied strategy to solve this problem is to start with an initial dictionary and alternate between the fallowing steps: sparse approximation and dictionary update [11].

3.1. The 1D K-SVD algorithm for search dictionary of signal

Many dictionary learning algorithms follow an iterative solution that alternates between update of D and C to minimize the cost function 7. In our work we use the 1D K-SVD algorithm. This solution is efficient for adapting dictionaries in order to achieve sparse signal representations [12].

The algorithm iterates in two steps:

The Sparse Approximation Step: Provided D is fixed. We use the orthogonal matching pursuit algorithm (mentioned in section 2) to estimate M sparse coefficients c_i for each sample of signal S, by approximation of the solution of

$$\min_C \left\{ \|s_i - Dc_i\|_F^2 \right\} \textit{ subject to } \|c_i\|_0 \le T, \;\; i = 1, 2, ...M, \tag{8}$$

where s_i is a sample of signal S.

The Dictionary Update Step: Provided both C and D are fixed. We focus on an atom d_k of the dictionary and its corresponding sparse vector c_T^k (i.e. row k of c_T^k), the corresponding objective function in 7 can be written as:

$$\|S - DC\|_F^2 = \left\| S - \sum\nolimits_{j=1}^{K} d_j c_T^j \right\|_F^2 =$$

$$= \left\| \left(S - \sum\nolimits_{j \ne k} d_j c_T^j \right) - d_k c_T^k \right\|_F^2 = \left\| E_k - d_k c_T^k \right\|_F^2, \tag{9}$$

where E_k indicates the representation error of the sample of signal after removing the k^{th} atom and its fixed. The next steps of the dictionary update stages are:

- Define the set of indexes that use the atom d_k , which means the case c_T^k (i) is non-zero as

$$\omega_k = \left\{ i | 1 \le i \le M, c_T^k(i) \ne 0 \right\}. \tag{10}$$

- Define the matrix Ω_k as a matrix with ones on the $(\omega_k(i), i) - th$ entries and zeros elsewhere.

- Then compute

$$E_k = S - \sum\nolimits_{j \ne k} d_j c_T^j \tag{11}$$

  and restrict E_k by choosing the columns corresponding to ω_k so that we obtain E_k^R.

- Finally, apply SVD to decompose $E_k^R = U \Delta V^T$ and update d_k to be first column of U and coefficient vector c_T^k to be first column of V multiplied by $\Delta(1, 1)$.

All dictionary atoms are updated is this way. Iterating through the two steps will produce dictionary that approximates given signal S sparsely and accurately. A detailed description of the presented algorithm can be found in the work of [11, 12].

3.2. An incoherent dictionary learning problem

The coherence $\mu(D)$ of a dictionary D measures the maximal correlation of different atoms

$$\mu\left(D\right)=\max\nolimits_{d_i,d_j\in D\ i\neq j}\left|\left\langle\frac{d_i}{\left|d_i\right|_2},\frac{d_j}{\left|d_j\right|_2}\right\rangle\right|, \tag{12}$$

where μ is the function, and is valued between 0 and 1. The minimum is reached for an orthogonal dictionary and the maximum μ_0 for a dictionary containing at least two collinear atoms. The general problem of the incoherent dictionary learning is to find the closest dictionary $\widehat{D}$ to a given dictionary $\bar{D}$, with a coherence lower than a given μ_0. The dictionary $\bar{D}$ is described as

$$\widehat{D}=\arg\min\nolimits_{D\in\Gamma}\left\|D-\bar{D}\right\|_F^2 \tag{13}$$

and

$$\Gamma=\left\{D|\mu\left(D\right)\leq\mu_0\wedge\left\|d_i\right\|_2=1,i\in\{1,...,M\}\right\}. \tag{14}$$

A good strategy for this problem can be pursued by including a decorrelation step to the iterative scheme in section 3.1. At each iteration of the dictionary learning algorithm consisting of sparse approximation followed by dictionary update, we add the following optimization problem

$$\widehat{D}=\arg\min\nolimits_{D\in\Gamma}\mu\left(D\right) \tag{15}$$

and

$$\Gamma'=\left\{D|\left\|D-\bar{D}\right\|_F^2\leq\theta\wedge\left\|d_i\right\|_2=1,i\in\{1,...,M\}\right\}, \tag{16}$$

where θ is the unknown minimum value reached by the criterion (13). In our implementation we solve the problem by inserting a decorrelation step in the K-SVD loop after the dictionary update. The modified algorithm is called the Incoherent – KSVD INK-KSVD. For more information about this algorithm, please see [13].

4. Experimental results

In this section we compare results achieved for INK-KSVD and KSVD based anomaly detection to SNORT [14] based preprocessor which we proposed in [3]. Preprocessor uses DWT – Discrete Wavelet Transform (Mallat implementation [6, 15]) for anomaly detection.

Efficiency of INK -KSVD based anomaly detection algorithm was evaluated by simulating different real world attacks on test LAN network. We used Kali Linux [16] distribution in order to simulate different attacks such as: Application specific DDos, various port scanning, DoS, DDoS, Syn Flooding, pocket fragmentation, spoofing and

Table 1. Network traffic features used for experiments.

Feature	Traffic feature description
f_1	number of TCP packets
f_2	in TCP packets
f_3	out TCP packets
f_4	number of TCP packets in LAN
f_5	number of UDP datagrams
f_6	in UDP datagrams
f_7	out UDP datagrams
f_8	number of UDP datagrams in LAN
f_9	number of ICMP packets
f_{10}	out ICMP packets
f_{11}	in ICMP packets
f_{12}	number of ICMP packets in LAN
f_{13}	number of TCP packets with SYN and ACK flags
f_{14}	out TCP packets (port 80)
f_{15}	in TCP packets (port 80)
f_{16}	out UDP datagrams (port 53)
f_{17}	in UDP datagrams (port 53)
f_{18}	out IP traffic [kB/s]
f_{19}	in IP traffic [kB/s]
f_{20}	out TCP traffic (port 80) [kB/s]
f_{21}	in TCP traffic (port 80) [kB/s]
f_{22}	out UDP traffic [kB/s]
f_{23}	in UDP traffic [kB/s]
f_{24}	out UDP traffic (port 53) [kB/s]
f_{25}	in UDP traffic (port 53) [kB/s]

others. We used the same set of attacks as in [3] in order to compare INK-KSVD based solution to algorithms based on KSVD and DWT [3]. In order to classify anomalies we create profiles of normal traffic behavior based on network traffic features with assumption that there is no attack in this traffic. For algorithms evaluation 25 traffic features was extracted from network traffic (see Table 1). Traffic features are represented as one dimensional – 1D vector of values. In Tables 2 and 3 there are results of DR detection rates and FP false positive, respectively. We can see that for a given test, INK-KSVD give us better results in comparison to KSVD and DWT based anomaly detection methods. We can notice that detection rate and false positive strongly depend on given traffic feature. Attack has got direct impact only on selected traffic features from Table 1. f_9 and f_{10} features give us the best results. DR [%] for f_9 and f_{10} changes in boundaries 90.73 – 98.43 in turn FP [%] changes in boundaries 0.32 – 5.12.

Additionally, we tested our method with basic traffic base [17] for evaluating algorithm performance. In Table 4 there are results of detection rate for two testing days.

Table 2. Detection Rate DR [%] for a given network traffic features.

Feature	KSVD	INK-KSVD	Mallat
f_1	5.26	8.26	5.26
f_2	5.26	12.52	10.52
f_3	0.00	12.52	10.52
f_4	15.78	10.52	10.52
f_5	10.52	14.52	10.52
f_6	0.00	0.00	0.00
f_7	0.00	0.00	0.00
f_8	25.22	35.24	31.58
f_9	90.73	98.43	94.73
f_{10}	83.68	96.43	94.73
f_{11}	7.24	10.26	5.26
f_{12}	80.42	85.95	78.95
f_{13}	10.52	14.22	10.52
f_{14}	0.00	8.26	5.26
f_{15}	0.00	14.22	10.52
f_{16}	0.00	0.00	0.00
f_{17}	5.26	8.26	5.26
f_{18}	10.52	14.52	10.52
f_{19}	5.26	8.22	5.26
f_{20}	10.52	15.24	5.26
f_{21}	12.26	14.24	10.52
f_{22}	0.00	0.00	0.00
f_{23}	0.00	0.00	0.00
f_{24}	0.00	0.00	0.00
f_{25}	5.26	8.24	0.00

Table 3. False Positive FP [%] for a given network traffic features.

Feature	KSVD	INK-KSVD	Mallat
f_1	5.46	4.23	7.43
f_2	5.17	4.84	7.99
f_3	5.45	4.22	7.96
f_4	5.44	4.02	6.06
f_5	5.64	4.23	5.62
f_6	3.96	3.02	4.14
f_7	5.18	3.50	5.33
f_8	5.24	4.24	8.28
f_9	7.68	6.12	9.13
f_{10}	1.22	0.32	0.48
f_{11}	5.12	4.12	12.06
f_{12}	6.34	4.20	4.34
f_{13}	5.23	4.56	7.07
f_{14}	4.58	3.26	7.48
f_{15}	4.86	3.52	7.17
f_{16}	0.02	0.02	0.02
f_{17}	0.40	0.25	0.39
f_{18}	4.80	3.72	8.74
f_{19}	5.24	4.46	8.36
f_{20}	4.52	3.18	8.50
f_{21}	4.23	3.12	7.09
f_{22}	3.46	2.52	3.08
f_{23}	4.82	2.82	3.07
f_{24}	0.02	0.00	0.00
f_{25}	0.37	0.03	0.02

Table 4. Detection Rate for W5D5 (Week5, Day5) and W5D1 DARPA [17] trace.

Network Traffic	DR[%]	DR[%]	DR[%]	DR[%]
Feature	KSVD	INK-KSVD	KSVD	INK-KSVD
	W5D5	W5D5	W5D1	W5D1
icmp flows/min.	64.7	95.60	94.52	95.42
icmp in bytes/min.	79.14	92.22	93.15	98.42
icmp in frames/min.	85.29	94.28	93.15	94.82
icmp out bytes/min.	79.41	85.25	89.04	98.12
icmp out frames/min.	88.23	94.85	75.34	94.22
tcp flows/min.	48.52	85.28	63.01	98.63
tcp in bytes/min.	55.88	92.64	90.41	94.25
tcp in frames/min.	60.29	90.72	97.26	98.54
tcp out bytes/min.	36.76	94.25	84.93	96.54
tcp out frames/min.	38.23	85.24	89.04	96.54
udp flows/min.	85.29	98.25	90.41	96.64
udp in bytes/min.	76.47	100.00	87.67	98.82
udp in frames/min.	85.29	98.75	68.49	100.00
udp out bytes/min.	89.7	96.22	98.63	100.00
udp out frames/min.	91.17	100.00	98.63	100.00

5. Conclusions

In this article we describe the complete procedure of building sparse representation of a signal and propose to identify anomalies based on network traffic prediction. In learning processes we apply modified 1D INK-SVD algorithm to detect incoherent dictionary on the basis of network traffic which does not contain anomalies. The classification is performed with the use of normal network traffic profiles and sparse representation parameters of the analyzed signal.The computed results clearly showed that abnormal activities included in the traffic signal can be detected by the proposed methods.

6. References

[1] Choraś M., Saganowski Ł., Renk R., Hołubowicz W., *Statistical and signal-based network traffic recognition for anomaly detection.* Expert Systems, 2012, 29(3), pp. 232–245.

[2] Garcia-Teodoro P., Diaz-Verdejo J., Maciá-Fernández G., Vázquez E., *Anomaly-based network intrusion detection: Techniques, systems and challenges.* Comput-

ers & security, 2009, 28(1), pp. 18–28.

[3] Saganowski Ł., Goncerzewicz M., Andrysiak T., Anomaly detection preprocessor for snort ids system. In: *Image Processing and Communications Challenges 4.* Springer 2013, pp. 225–232.

[4] FP7 INTERSECTION Project, *Deliverable d.2.1: Solutions for securing heterogeneous networks: A state of the art analysis.*

[5] Hwang K., Cai M., Chen Y., Qin M., *Hybrid intrusion detection with weighted signature generation over anomalous internet episodes.* Dependable and Secure Computing, IEEE Transactions on, 2007, 4(1), pp. 41–55.

[6] Mallat S.G., Zhang Z., *Matching pursuits with time-frequency dictionaries.* Signal Processing, IEEE Transactions on, 1993, 41(12), pp. 3397–3415.

[7] Pati Y.C., Rezaiifar R., Krishnaprasad P., Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. In: *Signals, Systems and Computers, 1993. 1993 Conference Record of the Twenty-Seventh Asilomar Conference on*, IEEE, 1993, pp. 40–44.

[8] Davis G., Mallat S., Avellaneda M., *Adaptive greedy approximations.* Constructive approximation, 1997, 13(1), pp. 57–98.

[9] Tropp J.A., *Greed is good: Algorithmic results for sparse approximation.* Information Theory, IEEE Transactions on, 2004, 50(10), pp. 2231–2242.

[10] Gribonval R., *Fast matching pursuit with a multiscale dictionary of Gaussian chirps.* Signal Processing, IEEE Transactions on, 2001, 49(5), pp. 994–1001.

[11] Elad M., From Exact to Approximate Solutions. In: *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing.* Springer, New York, 2010 pp. 79–109.

[12] Aharon M., Elad M., Bruckstein A., *K-svd: An algorithm for designing overcomplete dictionaries for sparse representation.* Signal Processing, IEEE Transactions on, 2006, 54(11), pp. 4311–4322.

[13] Barchiesi D., Plumbley M.D., *Learning incoherent dictionaries for sparse approximation using iterative projections and rotations.* Signal Processing, IEEE Transactions on, 2013, 61(8), pp. 2055–2065.

[14] *Snort – intrusion detection system.* https://www.snort.org/, Accessed: 2014-12-30.

[15] Dainotti A., Pescapé A., Ventre G., Wavelet-based detection of dos attacks. In: *Global Telecommunications Conference, 2006. GLOBECOM'06. IEEE*, IEEE, 2006, pp. 1–6.

[16] *Kali linux.* https://www.kali.org/, Accessed: 2014-12-30.

[17] *Defense advanced research projects agency darpa intrusion detection evaluation data set.* http://www.ll.mit.edu/mission/communications/ist/corpora/ideval /data/index.html, Accessed: 2014-12-30.