C. Benhamed et al., Int. J. of Design & Nature and Ecodynamics. Vol. 11, No. 3 (2016) 376–386

LARGE-SCALE IP NETWORK DATA ANALYSIS FOR ANOMALIES DETECTION THANKS TO SVM

C. BENHAMED[1], S. MEKAOUI[1] & K. GHOUMID[2]
[1]University of Science and Technology Houari Boumediene, Algérie.
[2]Complexe Universitaire, Oujda, Maroc.

ABSTRACT
An SVM (Support Machine Vector) algorithm has been implemented to sense traffic anomalies through a large-scale IP Network. We have applied this algorithm on data provided by the well-known large-scale American IP Network (Abilene Network). The developed SVM algorithm can classify the Network traffic into two categories of classes namely: normal; and abnormal. The implementation of this algorithm has been performed on real collected data thanks to Netflow protocol and has yielded satisfactory results with a classification rate going over 96% and a false alarms rate lower than 10%.
Keywords: anomaly detection, genetic algorithms – SMO, IP network- supervised learning, support vector machines (SVM), true negative ratio, true positive ratio.

1 INTRODUCTION

In an IP Network, the direct traffic measurement or the estimation of the Traffic Matrix (TM) allows to know the traffic volume that goes through the nodes and the links of that Network. Network management operators use the aggregated data within the TMs in different applications and management operations. A typical example of such operations is the detection of anomalies in the Network based on the knowledge of the traffic Volume within the nodes and the links where these anomalies can occur. In fact, the Anomalies Detection Systems (ADS) have been proposed for the first time by Anderson [1], who has exploited the idea that attacks on the systems cause failures and are sources of anomalies in it. So, it appears that anomalies can be detected as significant deviations of the global network behavior. Since the work of Denning [1], many techniques have been developed, Lakhina, [2], has used a specific algorithm named Kernel Based On Line Anomaly Detection (KOAD). This algorithm is based on the recursive least squares kernels, then, Lakhina [3], devised an algorithm for detecting anomalies in a Network using the technique of Principal Component Analysis (PCA). Mekaoui and Benhamed,[4], took the advantage of the Kalman filter with a threshold to detect anomalies in an IP/MPLS Network. Ghosh and Schwartzbard [5, 6], have tried neural networks to attain the same goal. Other authors [7–10], have used statistics or seasonal series modeling to detect anomalies in a Telecommunications Network. Barford and Kline, [11], have defined four main types of anomalies among which they specified the Denial of Service (D.O.S), and the Distributed Denial of Service (D.D.O.S.). Lakhina [12], has also been interested by studying other kinds of anomalies that may affect the correlation between several links of the traffic within a Network. Anomaly detection remains a delicate and a complicated operation. In the literature, we can find many authors, [1, 13–15] that have tried to use the SVM to detect anomalies. Our approach is to tackle the problem of anomalies detection in an IP Network by also using an SVM classifier considering that this detection should be interpreted as a matter of classification of the Network traffic into two classes, normal

WIT CONFERENCES
This paper is part of the Proceedings of the International Conference on Big Data (Big Data 2016)
www.witconferences.com

ISSN: 1755-7437 (paper format), ISSN: 1755-7445 (online), http://www.witpress.com/journals
DOI: 10.2495/DNE-V11-N3-376-386

and abnormal traffic. To attain this goal, we are using an SVM algorithm but in a way that satisfies two major conditions; obtaining a high rate of detected anomalies and getting the minimum false alarms rate at a time. This algorithm needs a supervised Learning phase. Before proceeding to this latter step, our process performs a phase of pretreatment of the analysed data using genetic algorithms that are able to cross and do the mutation (transfer) of the data segments. This is done to maximise the classification of the SVM algorithm. The paper is organised in such a way that in Section 2, we present the used data and the pretreatment on the same data. In Section 3, basics on SVM are exposed. Section 4 details our SVM algorithm, whereas in Section 5 our algorithm approach is given explaining how simultaneously the process had been applied to the collected data. In Section 6, we discuss our results and test the robustness. Finally a conclusion is drawn in Section 7.

2 DATA PRETREATMENT

2.1 Data

To evaluate our process, we have used the data provided by the well-known American Abilene Network (USA observatory Network). The topology of such a Network is illustrated in Figure 1 given hereafter. As we can observe in the latter, the Network comprises twelve (12) main nodes which imply subsequently 144 Origin-Destination (OD) pairs and 54 physical links between the nodes. The data are collected every 5 minutes and correspond to the period from 01-03-2004 to 07-03-2004 (one week). So, this allowed us 288 samples a day and 2016 samples a week. These data are provided in formats of TMs measured by a specific protocol Netflow protocol in the indicated period.

2.2 Pretreatment

This step is necessary and allows us the calculation of the parameters that are required by the classification process (e.g.; means, variances …).

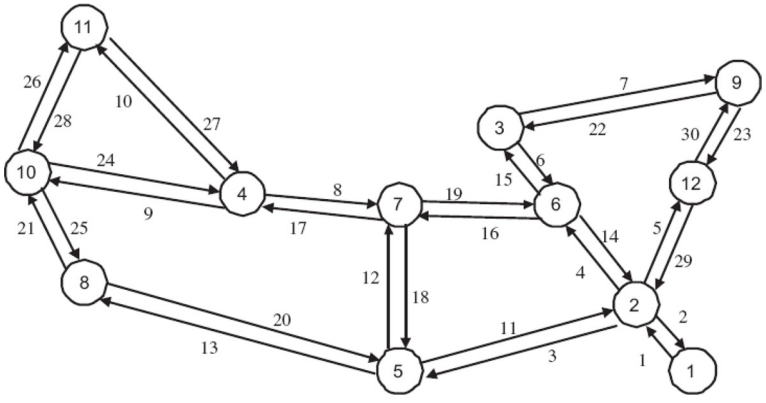

Figure 1: Topology of the network under study (Abilene Network).

Table 1 Datasets under study.

Network	Date	Duration	Resolution	Size
Abilene	March 2004	1 Week	5 mn	144×2,016

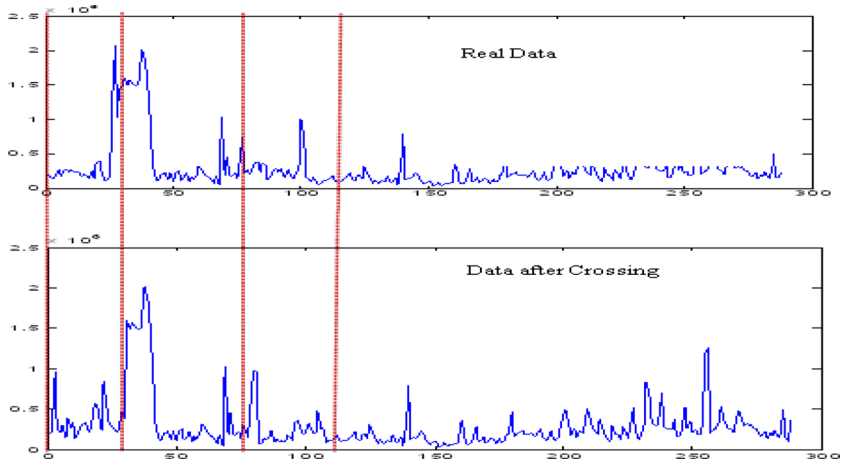

Figure 2: Data crossing.

Our objective still remains the anomaly detection of Denial of Service (DOS) and Distributed Denial of Service (DDOS) types by the mean of a supervised learning. The supervised learning algorithm is based upon a genetic algorithm which is inspired from the biological modeling. As we know well in Biology, the genetic algorithm performs three important operations namely: Selection, Crossing (Crossing over) and mutation. In our case, the genetic algorithm will perform these three important sub-operations on the data segments or vectors. The Selection operation facilitates the choice of individuals (days) represented by the measured traffic vectors in the specified day. The simple or multiple Crossing operation along with the mutation operation (transfer operation) generates other individuals where the anomaly can be easily detected like in genetic, see Figure 2.

3 BASICS ON SVM

3.1 Background

The main idea in Support Machine Vectors (SVM) is a supervised classification that allows to draw a hyper plan, which maximises the margin between two classes (say positive and negative).

In this case, the hyper plan is optimal and occupies the middle of the geometric configuration as depicted in Figure 3. For more details on this, SVM geometric method of data classification and other varieties, we kindly refer the reader to [16–18] whose authors expressly state that: If the data are linearly separable, then, there exists a Hyper plane whose equation is given by; $\langle W,X\rangle + b = 0$; such that we get from the geometric configuration (Figure 3) :

With H_1 and H_2 as hyper planes and H the optimal hyper plane, [17].

$$W.X + b \geq 0 \text{ if } y_i = +1; \tag{1}$$

$$W.X + b \leq 0 \text{ if } y_i = -1 \tag{2}$$

Getting:

$$y_i\,(W.X + b) \geq +1 \tag{3}$$

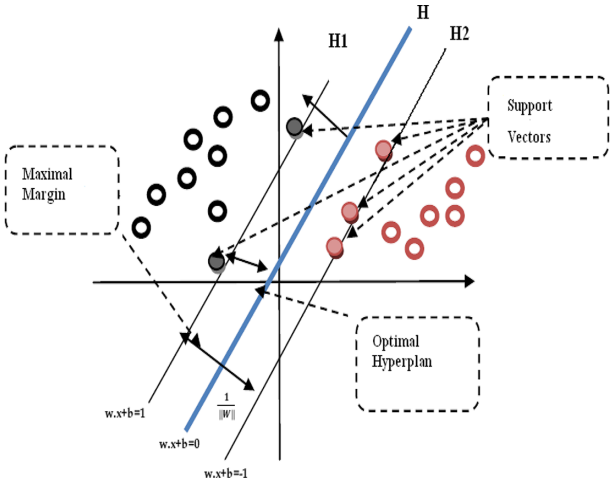

Figure 3: Optimal Hyper plane, support vectors and maximal margin.

According to these conditions (See Figure 4), the perpendicular distance from the Origin to the Hyper plane satisfies the following conditions:

$$H_1: W.X+b=1; \text{ and } x=\frac{|1-b|}{||W||} \tag{4}$$

$$H_2: W.X+b=-1; \text{ and } x=\frac{|1+b|}{||W||} \tag{5}$$

If the distance between one point located on H1 and the Hyper plane H is given by:

$$\frac{|\mathbf{W}.\mathbf{X}+\mathbf{b}|}{||\mathbf{W}||}=\frac{1}{||\mathbf{W}||} \tag{6}$$

Then, the margin between H1 and H2 is : $\frac{2}{||\mathbf{W}||}$

Maximising the previous quantity consists in minimising the quantity: $\frac{||\mathbf{W}||}{2}$, which can be stated as : min $\frac{1}{2}w^Tw$ with always staying in the limits fixed by the initial condition :

$$y_i(W.x+b)\geq+1 \text{ and } \min(\frac{1}{2}w^Tw) \tag{7}$$

This kind of optimisation problem can be solved by associating a Lagrangian multiplication operator $\boldsymbol{\alpha_i}$ ($\boldsymbol{\alpha_i \geq 0}$). It is generally defined by:

$$L(w,b,\alpha)=\frac{1}{2}w^Tw-\sum_{i=1}^{n}\alpha_i\left(y_i\left((x_i.w)+b\right)-1\right) \tag{8}$$

Maximising (8) is equivalent to find out the parameters **αi** and **w** that made equal to zero the partial derivatives of L (w,b,α), then:

$$\frac{\partial L(w,b,\alpha)}{\partial w} = 0, \quad \frac{\partial L(w,b,\alpha)}{\partial b} = 0 \text{ and } \alpha_i \geq 0 \tag{9}$$

Then, we get:

$$w = \sum_{i=1}^{n} \alpha_i y_i x_i \quad \text{et} \quad \sum_{i=1}^{n} \alpha_i y_i x_i = 0 \tag{10}$$

Solving eqns. (9) and (10) will finally yield the expression of the dual Lagrangian, given by:

$$L_{dual}(w,b,\alpha) = \sum_{i=1}^{n} \alpha_i - \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j x_i x_j \tag{11}$$

The resolution of αi, [17], gives the value of the vector (data vector) and can classify a new target following its feature vector x satisfying the function, below:

$$f(x) = sign(w.x + b) = sign\left(\sum_{i=1}^{n} \alpha_i y_i x_i .x + b\right) \tag{12}$$

3.2 Non linearly separable case

In the case of non-linearly separable data, we introduce the concept of deviations variables denoted ε_i (i = 1..N) (values of ε) with $\varepsilon > 0$ and whose constraints become:

$$W.X + b \geq 1 - \varepsilon \text{ if } y_i = +1 \tag{13}$$

$$W.X + b \leq -1 + \varepsilon \text{ if } y_i = -1 \tag{14}$$

Then, the margin between H and Hi becomes: $\frac{||W||}{2} + C\left(\sum \varepsilon_i\right)$. The parameter C ($C \geq 0$) is interpreted as a tolerance of the classification noise. For high values of C, only very small values of ε are authorised. Consequently, only a small number of data segments (points) will be badly classified, this number is usually non-significant. Whereas, if the value of C is too small, values of ε will be high enough. In this case, we tolerate more classification errors. In the case of non- linearly separable data, which mean in fact that the separating surface is non-linear, we transpose the problem in another higher dimension space *F* than the previous one to recover the points linearity and hence make again the two classes (Normal and Abnormal) again separable within the points clouds . This operation requires the use of a specific Transform denoted ϕ, such as:

$$\phi : x - \rightarrow \phi \quad \Phi \in F$$

and whose decision function can be defined by the following inner product:

$$\Phi^T(x_i) * \Phi(x_j), \tag{15}$$

Eqn. (15) can be substituted by a specific function noted K(xi, yi) and called Mercer kernel function, [17]:

$$K(x_i, x_j) = \Phi^T(x_i) * \Phi(x_j)$$

Generally, the function is of Radial Basis Function (RBF) class and of Gaussian type:

$$K(x_i, x_j) = (1/\sqrt{2\sigma^2})\ \exp(-\|x_i - x_j\|^2/2\sigma^2);$$

Where: σ is a regulation parameter

Subsequently, we can derive the calculus of the following maximum:

$$\mathrm{Max}\left(\sum_{i=1}^{n}\alpha_i - \sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j K\left(x_i, x_j\right)\right) \tag{16}$$

It is worth noting that all of the functions that solve (16) are based on Karush-Kuhn-Tucker (KKT), [17], satisfying at a time the following constraints:

$$0 \le \alpha_i \le C \ i = 1,\ldots., n \text{ and } \sum_{i=1}^{n}\alpha_i y_i = 0 \text{ with } i = 1,2,\ldots\ldots n$$

To decide whether a data **x** belongs to the first or the second class; it is enough to find the sign of the decision function:

$$f(x) = sign\left(\sum_{i=1}^{n}\alpha_i y_i K(x_i, x) + b\right); \tag{17}$$

For the learning step, we have implemented the Sequential Minimal Optimization (SMO) algorithm to insure the learning and training step.

4 ALGORITHM

Prior to proceed to our algorithm and for adapting our data to it, we first calculate the necessary parameters that allow the SVM to perform a sharp and accurate classification between Normal and Abnormal traffic. The dynamic flow rate on the Network links is random. To avoid false alarms of anomalies, we calculate the flow rate distances that occur in the Network, and we compare, their means and their variances and we also measure the times and instants of occurrence.

4.1 Pretreatment

We well know that one specification of the DOS (Denial of Service) or DDOS (Distributed Denial of Service) is their continuity as being an overload of the bandwidth. Thus, we consider our evaluation on a three temporal dimensional space (di−1, di, di+1) that will be represented by three time instants. (t−1, t, t+1), respectively. So, we have:

- di−1:(measured and muted flow rate – flow rates mean before mutation) at t−1.
- di: (measured and muted flow rate – flow rates mean before mutation) at instant t.
- di+1: (measured and muted flow rate – flow rates mean before mutation) at instant t+1.

For each measurement, every five minutes, we will examine three samples which mean that an attack will last at least 15 minutes or more. To these three dimensions, we add another one namely the variance that will characterise the stability aspect of the anomaly if it does exist. The variance is then symbolised by the acronym Var. *1.0e+006

Table 2: Sample of data preprocessing result within extraction of parameters.

D*i*−1	D*i*	D*i*+1	Var.
0.3571	0.6137	1.0181	1,110
0.6137	1.0181	0.5450	6,538
0.181	0.5450	0.0183	2,500
0.2406	0.2028	0.5354	3,316
0.5774	0.3385	0.1835	3,940
0.3385	0.1835	0.6047	4,541

4.2 Learning step

This step consists in selecting a model function that belongs to F ($f \in F$ see Section III). So, the principle is based on finding an estimation of the Lagrangian coefficients α_i and the scalar b of the optimal classifying Hyper plane. The selection of these parameters is insured by a learning algorithm called Sequential Minimal Optimization algorithm (SMO) that receives as input the set of the N-data of the training phase. The distances (d*i*-1, d*i*, d*i*+**1**) of one OD pair represent each class (x*i*) along with the target vector y*i* (+1 or -1) which indicates the class for which each traffic measure of one OD« $x_{i\,»}$ is belonging. The target vector yi also indicates a set of learning parameters with a tolerance of about $\left(\frac{1}{||W||}\right)$, σ, the parameter of regulation, the tolerance parameter to the classification noise C and the type of the kernel function that has to be used if the data are linearly separable. The algorithm steps are given below:

```
Initial step
Extraction of Data from traffic Matrix X(i,j);
For i=0 to 144 do;
Abnormal=false;
While Abnormal= false do;
Step 1
Crossing Data;
Mutation (generate random traffic between the mean traffic of the day and the
maximum traffic flux of the OD link).
    Step 2
    Compute the distances (di-1, di, di+1) and the variance.
    If ((di-1>0 and di>0 and di+1>0 and VAR<variances of the day)
    Then
    Proclaim normal traffic
    Call SVM (SMO) training
    Else
    Proclaim abnormal traffic
```

```
Call SVM (SMO) training
End If
Call the classifier f(x)
If f(x)=-1
Abnormal=true
End while
End For
```

5 RESULTS AND DISCUSSION

In order to validate the implemented algorithm, we have used universal metrics. These metrics consist in the anomalies detection ratio and the false alarm sensitivity ratio. These two metrics are symbolised by the following acronyms namely; TPR, True Positive Rate and TNR, True Negative and are given by the following equalities:

$$TPR = \frac{TP}{(TP + FN)}; \text{ and } \qquad TNR = \frac{TN}{(TN + FP)}; \tag{18}$$

Where: TP is the True Positive; FP is the False Positive; TN the True Negative and FN the False Negative.

In our case, we consider that the value of C is high enough, and it varies from 600 to 1,400 so that we get only a limited number of classification errors and authorise only kernel RBF (Radial Basis Function) based SVM whose width $\sigma = 1$ and an acceptable error rate of 0.1%. We have obtained many significant results. The result depicted in Table 3 given hereafter is only one sample among many. This table is clearly showing that our algorithm had yielded good results in the process of classification with a rate of classification that goes over 96%. We can observe from the same table that the false alarm ratio reaches a level lower than 1% and can attain in some cases (see in the table the result for OD(Origin-Destination pair #89) 0.11% with a classification rate of about 97.92%. In fact, the figures in Table 3 reveal the robustness of the SVM algorithm in the detection of Abnormal Traffic. Even for a false alarms rate of about 9.8%, the SVM classification rate remains satisfactory around 96%.

Table 3: Sample of results.

OD number	Error rate	C	σ	Classification Rate (%)	False Alarm rate (%)
49	0.001	900	1	97.0910	5.71
50	0.001	900	1	96.0938	6.17
51	0.001	800	1	97.0938	5.82
52	0.001	600	1	97.2554	0.18
53	0.001	1,000	1	96.4039	5.71
54	0.001	800	1	96.1985	0.11
55	0.001	1,300	1	95.0943	3.52
56	0.001	1,100	1	97.0599	5.13
57	0.001	1,100	1	97.9238	0.11
58	0.001	900	1	96.8494	3.52

OD number	Error rate	C	σ	Classification Rate (%)	False Alarm rate (%)
59	0.001	1,400	1	96.0738	5.31
60	0.001	1,200	1	98.2275	9.11
61	0.001	1,500	1	96.6753	9.82
62	0.001	1,300	1	97.7182	5.81
63	0.001	1,200	1	96.0634	2.90
64	0.001	1,000	1	97.5610	5.71
65	0.001	1,100	1	96.2774	0.72
66	0.001	900	1	97.8088	5.71
67	0.001	900	1	96.6333	6.17
68	0.001	700	1	96.0146	5.82
69	0.001	800	1	97.2610	0.16
70	0.001	600	1	96.9054	5.71
71	0.001	900	1	96.7483	9.11
72	0.001	700	1	96.2201	5.82
73	0.001	800	1	96.0721	2.90
74	0.001	600	1	96.4193	5.03
75	0.001	300	1	97.7321	0.72
76	0.001	100	1	96.4961	5.71
77	0.001	1,000	1	96.5272	6.17
78	0.001	800	1	97.9809	5.82
79	0.001	1,000	1	96.3855	0.16
80	0.001	800	1	96.1156	5.71
81	0.001	700	1	95.7964	0.11
82	0.001	500	1	96.2574	3.52
83	0.001	200	1	96.9054	5.13
84	0.001	100	1	96.7483	9.11
85	0.001	1,300	1	96.2201	9.81
86	0.001	1,100	1	96.0721	0.82
87	0.001	1,900	1	96.0193	3.52
88	0.001	1,700	1	96.4393	5.13
89	0.001	2,100	1	97.7321	0.11
90	0.001	1,900	1	96.4961	3.52

6 CONCLUSION

We have proposed in this paper a novel approach based on the pretreatment of the processed initial data using genetic algorithms for anomalies detection of the traffic throughout an IP Network. In presence of noise, certain false alarms can occur with a rate lower than 10% blurring sometimes the real detection of real anomalies, but the detection of abnormal (real DOS or DDOS anomalies)

traffic remains satisfactory. Nevertheless, this method presents a certain disadvantage when we compare it with those in literature developed by other authors and reported in references [15, 19, 20]. Ours is experiencing a longer learning time in our learning step and hence requires a bigger and wider memory space. This is in fact, the main drawback of our SVM classifier. Otherwise, once the learning is achieved, then classification takes a very short time. Work is in progress to devise and realise a more precise classifier that may use many SVM classifiers in a tree structure.

REFERENCES

[1] Ahmed, T., Coates, M. & Lakhina, A., Multivariate online anomaly detection using kernel recursive least squares. *IEEE Infocom Anchorage*, AK, Boston University, 2007.

[2] Ringberg, H., Soule, A., Rexford, J. & Diote, C., Sensitivity of PCA for traffic anomaly detecion, Department of computer Science princeton University, Thomson Research, New Jersey USA, 2007.

[3] Mekaoui, S., Benhamed, C. & Ghoumid, K., Sensing anomalies with an optimal filter applied to the trafic matrix of an IP telecommunications network. In *International Conference on Multimedia Computing and Systems*, Morocco, 2012.

[4] Benhamed, C. & Mekaoui, S., A PCA Based algorithm for detecting volume traffic anomalies in IP Networks. JLCPTS 2015, *USTHB 14 & 15 Janvier 2015*, Alger, Algérie, 2015.

[5] Dao, V.N.P. & Vemuri, V.R., A performance comparison of different back propagation neural networks methods in computer network intrusion detection. *Differential Equations and Dynamical Systems*, **10**(1–2), pp. 201–214, 2002.

[6] Farraposo, S., Owezarski, P. & Monteiro, E., Détection, classification et identification d'anomalies de trafic, hal-00250220, version 1, 2008.

[7] Lakhina, A., Crovella, M. & Diot, C., Mining anomalies using traffic feature distributions. *ACM SIGCOMM, 2005 M. Young, The Technical Writer's Handbook*, Mill Valley, CA: University Science, Philadelphia, Pennsylvania, USA, 2005.
http://dx.doi.org/10.1145/1080091.1080118

[8] Soule, A., Salamatian, K. & Taft, N., Combining filtering and statistical methods for anomaly detection, USENIX/ACM IMC, Boston, 2005.
http://dx.doi.org/10.1145/1330107.1330147

[9] Brutlag, J., Aberrant behavior detection in time series for network monotoring. In *Proceeding of the USENIX System Administration Conference LISA XIV*, USENIX Association Berkeley, CA, USA December, pp. 139–146, 2000.

[10] Soule, A., Salamatian, K. & Taft, N., Combining filtering and statistical methods for anomaly detection, LIP6-UPMC, Intel research 2005. In *Proceedings of IFIP Networking*, Waterloo, Ontario, Canada, 2005.

[11] Barford, P., Kline, J., Plonka, D. & Ron, A., A signal analysis of network traffic Anomalies. *ACM SIGCOM Internet Measurment Workshop*, pp. 71–82, 2002.
http://dx.doi.org/10.1145/637201.637210

[12] Lakhina, A., Crovella, M. & Diot, C., « Diagnosing network traffic anomalies in trafic flows », SIGCOM, pp. 219–230, 2004.

[13] Lane, T. & Brodley, C.E., An application of machine learning to anomaly detection. *Proceedings of the 20th National Information Systems Security Conference*, Baltimore, MD, pp. 366–377, 1997.

[14] Mukkamala, S., Janoski, G.I. & Sung, A.H., Intrusion detection using support vector machines. *Proceedings of the High Performance Computing Symposium - HPC* , San Diego, pp. 178–183, 2002.

[15] Hu, W., Liao, Y. & Rao Vemuri, V., Robust anomaly detection using support vector machines. In International Conference on Machine Learning, (ICMLA'03) Los Angeles, California, 2003.
[16] Vapnik, V.N., Statistical Learning Theory, John Wiley&Sons, Inc.: New York, Association for Computing Machinery; Knowl Discov2, pp. 121–167, 1998.
[17] Burges, C.J.C., A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, **2**(2), pp. 121–167, 1998.
http://dx.doi.org/10.1023/A:1009715923555
[18] Cortes, C. & Vapnik, V.N., Support vector network. Machine Learning, **20**(3), pp. 273–297, 1995.
http://dx.doi.org/10.1007/BF00994018
[19] Rawat, S., Pujari, A.K., Gulati, V.P. & Rao Vemuri, V., Intrusion detection using text processing techniques with a binary-weighted cosine metric. *Journal of Information Assurance & Security (JIAS)*, **1**(1), pp. 43–50, 2006.
[20] Liao, Y. & Rao, V., Use of K nearest neighbor classifier for intrusion detection. *Journal of Information Assurance and Security,* **21**(5), pp. 439–448, 2002.
http://dx.doi.org/10.1016/s0167-4048(02)00514-x