Neural Analysis of HTTP Traffic for Web Attack Detection

David Atienza, Álvaro Herrero and Emilio Corchado

Abstract Hypertext Transfer Protocol (HTTP) is the cornerstone for information exchanging over the World Wide Web by a huge variety of devices. It means that a massive amount of information travels over such protocol on a daily basis. Thus, it is an appealing target for attackers and the number of web attacks has increased over recent years. To deal with this matter, neural projection architectures are proposed in present work to analyze HTTP traffic and detect attacks over such protocol. By the advanced and intuitive visualization facilities obtained by neural models, the proposed solution allows providing an overview of HTTP traffic as well as identifying anomalous situations, responding to the challenges presented by volume, dynamics and diversity of that traffic. The applied dimensionality reduction based on Neural Networks, enables the most interesting projections of an HTTP traffic dataset to be extracted.

Keywords Intrusion detection · HTTP · Artificial neural networks · Exploratory projection pursuit

D. Atienza (✉) · Á. Herrero
Department of Civil Engineering, University of Burgos Spain C/Francisco de Vitoria s/n, 09006, Burgos, Spain
e-mail: dag0031@alu.ubu.es

Á. Herrero
e-mail: ahcosio@ubu.es

E. Corchado
Departamento de Informática y Automática, Universidad de Salamanca, Plaza de la Merced s/n, 37008 Salamanca, Spain
e-mail: escorchado@usal.es

© Springer International Publishing Switzerland 2015

Á. Herrero et al. (eds.), *International Joint Conference*, Advances in Intelligent Systems and Computing 369, DOI 10.1007/978-3-319-19713-5_18

1 Introduction

An attack or intrusion to a network would end up affecting any of the three computer security principles: availability, integrity and confidentiality, exploiting for example the Denial of Service, Modification and Destruction vulnerabilities [1]. The ever-changing nature of attack technologies and strategies is one of the most harmful issues of attacks and intrusions, increasing the difficulty of protecting computer systems. It means that new ways of attacking information systems and networks are being developed every single day.

Hypertext Transfer Protocol (HTTP) [2] is a stateless application-level protocol for distributed, collaborative, hypertext information systems. It was initially proposed for information exchanging over the World Wide Web. Nowadays, it is not only the usual way of exchanging information associated to web pages because new uses of such protocol are being proposed. HTTP users include household appliances, stereos, scales, firmware update scripts, command-line programs, mobile apps, and communication devices in a multitude of shapes and sizes [2]. On the other hand, common HTTP origin servers include home automation, units, configurable networking components, office machines, autonomous robots, news feeds, traffic cameras, ad selectors, and video-delivery platforms. This means that it is one of the protocols in the application layer of the TCP/IP stack [3] that is more frequently used and will still be in the coming future. Version 1.1 of such protocol was proposed in 1997 [3] and currently remains as the standard version.

The purpose of a web based attack is significantly different than other attacks related to information systems; in most traditional penetration testing exercises a network or host is the target of attack. Web based attacks focus on an application itself and functions on layer 7 of the Open Systems Interconnection [4]. As described for primitive web attacks, all web application attacks are comprised of at least one normal request or a modified request aimed at taking advantage of poor parameter checking or instruction spoofing [4]. Recent studies [5, 6] confirm that web based attacks continue to be on the rise so the automatic detection of such situations still is an open challenge.

Present study proposes a solution characterized by the use of unsupervised connectionist projection techniques providing a novel approach based on the visual analysis of the internal structure of the flow of HTTP data for the detection of web based attacks. Unsupervised learning is quite useful for identifying unknown or not previously faced attacks, known as 0-day attacks, based on the well-known generalization capability of the Artificial Neural Networks (ANNs).

The analysis of HTTP traffic has been approached from several different points of view up to now, but mainly from the machine learning perspective [7, 8]. Additionally, neural projection techniques have been previously applied to many different data, related to computer/network security [9–11]. Differentiating from those previous studies, present work addresses HTTP traffic to check whether the nature of such data allows neural visualization for the detection of attacks.

2 A Neural Approach for Visualization

This work proposes the application of projection models for the visualization of HTTP data. Visualization techniques have been applied to massive security datasets, such as those generated by network traffic [9], SQL code [10] or honeynets [11]. These techniques are considered a viable approach to information seeking, as humans are able to recognize different features and to detect anomalies by means of visual inspection. The underlying operational assumption of the proposed approach is mainly grounded in the ability to render the high-dimensional traffic data in a consistent yet low-dimensional representation. In most cases, security visualization tools have to deal with massive datasets with a high dimensionality, to obtain a low-dimensional space for presentation.

This problem of identifying patterns that exist across dimensional boundaries in high dimensional datasets can be solved by changing the spatial coordinates of data. However, an a priori decision as to which parameters will reveal most patterns requires prior knowledge of unknown patterns.

Projection methods project high-dimensional data points onto a lower dimensional space in order to identify "interesting" directions in terms of any specific index or projection. Having identified the most interesting projections, the data are then projected onto a lower dimensional subspace plotted in two or three dimensions, which makes it possible to examine the structure with the naked eye.

From the information security perspective, visualization techniques were previously proposed as "*visualizations that depict patterns in massive amounts of data, and methods for interacting with those visualizations can help analysts prepare for unforeseen events*" [12].

Due to the aforementioned reasons and based on previous successful applications [9–11], present study approaches the analysis of HTTP data from a visualization standpoint. That is MATLAB [13] implementations of some neural techniques, described in the following subsections, are applied for the analysis of such data. Additionally, Curvilinear Component Analysis (CCA) [14] has been applied to the data but it is not included in present paper for the sake of brevity as it does not visualize the dataset in a way that many different groups can be identified.

2.1 Principal Component Analysis

Principal Component Analysis (PCA) is a well-known statistical model, introduced in [15] and independently in [16], that describes the variation in a set of multivariate data in terms of a set of uncorrelated variables each, of which is a linear combination of the original variables. From a geometrical point of view, this goal mainly consists of a rotation of the axes of the original coordinate system to a new set of orthogonal axes that are ordered in terms of the amount of variance of the original data they account for.

PCA can be performed by means of ANNs or connectionist models such as [17] or [18]. It should be noted that even if we are able to characterize the data with a few variables, it does not follow that an interpretation will ensue.

2.2 Cooperative Maximum Likelihood Hebbian Learning

The Cooperative Maximum Likelihood Hebbian Learning (CMLHL) model [19] extends the Maximum Likelihood Hebbian Learning [20] model, which is based on Exploration Projection Pursuit. The statistical method of EPP was designed for solving the complex problem of identifying structure in high dimensional data by projecting it onto a lower dimensional subspace in which its structure is searched for by eye. To that end, an "index" must be defined to measure the varying degrees of interest associated with each projection. Subsequently, the data is transformed by maximizing the index and the associated interest. From a statistical point of view the most interesting directions are those that are as non-Gaussian as possible.

Considering an N-dimensional input vector (x), and an M-dimensional output vector (y), with W_{ij} being the weight (linking input j to output i), then CMLHL can be expressed as defined in Eqs. 1–4.

1. Feed-forward step:

$$y_i = \sum_{j=1}^{\mathbf{N}} W_{ij} x_j, \forall i \tag{1}$$

2. Lateral activation passing:

$$y_i(t+1) = [y_i(t) + \tau(b - Ay)]^+ \tag{2}$$

3. Feedback step:

$$e_j = x_j - \sum_{i=1}^{M} W_{ij} y_i, \forall j \tag{3}$$

4. Weight change:

$$\Delta W_{ij} = \eta . y_i . sign(e_j) |e_j|^{p-1} \tag{4}$$

where: η is the learning rate, τ is the "strength" of the lateral connections, b the bias parameter, p a parameter related to the energy function [19–21] and A a symmetric matrix used to modify the response to the data [19]. The effect of this matrix is based on the relation between the distances separating the output neurons.

2.3 Self-Organizing Maps

The widely-used Self-Organizing Map (SOM) [22] was developed as a visualization tool for representing high dimensional data on a low dimensional display. It is also based on the use of unsupervised learning. However, it is a topology preserving mapping model rather than a projection architecture.

To mimic the biological brain maps, the SOM is composed of a discrete array of L nodes arranged on an N-dimensional lattice. These nodes are mapped into a D-dimensional data space while preserving their ordering. The dimensionality of the lattice (N) is normally smaller than that of the data, in order to perform the dimensionality reduction. The SOM can be viewed as a non-linear extension of PCA, where the global map manifold is a non-linear representation of the training data [23].

Typically, the array of nodes is one or two-dimensional, with all nodes connected to the N inputs by an N-dimensional weight vector. The self-organization process is commonly implemented as an iterative on-line algorithm, although a batch version also exists. An input vector is presented to the network and a winning node, whose weight vector W_C is the closest (in terms of Euclidean distance) to the input, is chosen, according to Eq. 5.

$$c = \arg\min_i(\|\mathbf{x} - W_i\|) \tag{5}$$

When this algorithm is sufficiently iterated, the map self-organizes to produce a topology-preserving mapping of the lattice of weight vectors to the input space based on the statistics of the training data.

3 Experiments and Results

As previously mentioned, many neural visualization models (see Sect. 2) have been applied to HTTP traffic to analyze its nature. Present section introduces the analyzed dataset as well as the main obtained results.

3.1 HTTP Dataset CSIC 2010

To check the validity of the proposed techniques, they have been confronted to a real-life publicly-available dataset, known as HTTP Dataset CSIC 2010 [24].

This dataset was automatically generated by creating traffic to an e-commerce web application. It contains several HTTP requests, labeled as normal or anomalous. Each HTTP request is defined by the following features: method, url, protocol, userAgent, pragma, cacheControl, accept, acceptEncoding, acceptCharset, acceptLanguage, host, connection, contentLength, contentType, cookie and payload.

The raw data were process, according to the following process:

1. The following features took a single possible value: protocol, userAgent, pragma, cacheControl, accept, acceptEncoding, acceptCharset, acceptLanguage, connection. As those variables do not provide any information to discriminate between normal and anomalous requests, they were deleted.
2. The cookie feature contains a session id for every request. As this kind of information is useless for applied models, it was removed too. Additionally, the URL feature was also removed from the dataset as it contains URLs from the e-commerce application, that cannot be used.
3. Duplicated HTTP requests were removed.
4. Categorical data were converted into numeric values, according to the following details:

   - method, host and contentType features have limited possible values. For example, possible values for method variable are: GET, POST, PUT. In this type of variable, we replaced each possible value with a different number.
   - contentLength is almost a completely numeric variable. Possible values for this variable are a number greater than 0 or null. Null values were replaced by 0.
   - payload variable contains character strings with different lengths and contents, so the possible values are nearly unlimited. Then, the payload value was replaced by its length.

Finally, the analyzed dataset is composed of 1,916 unique HTTP requests and 5 features for each one of them. Table 1 shows an example of an HTTP request on the final dataset.

3.2 Results

PCA Projection

Figure 1 shows the 2-principal component projection, obtained by applying PCA to the previously described data. Blue crosses and red circles represent normal and anomalous requests respectively. As can be seen in this Figure, normal and anomalous requests cannot be clearly differentiated from PCA projection. However, a deeper analysis has been performed, to extract some common characteristics of requests that are depicted in the same group, by knowing its position on this projection. In order to do that, Fig. 1 is divided into 5 areas. Regarding the

Table 1 Sample row from the final dataset

Method	Host	ContentLength	ContentType	Payload
1	0	221	1	12

method used on the HTTP request, there is a clear clustering in the PCA projection:

- Zones 1 and 2 contain requests that only use the GET method.
- Zones 3 and 4 contain requests that only use the POST method.
- Zone 5 contains requests that only use the PUT method.

The second principal component mainly takes into account the value of payload and contentLength. So, yellow and green zones contain the requests with the highest payload and contentLength values.

CMLHL Projection

Figure 2 shows the CMLHL projection of the analyzed data. **As it is shown in Fig. 2a, several groups can be clearly identified in CMLHL projection. An in-depth analysis of those groups reveals the following data:**

- **Zones 1 and 2** contain requests that use the POST and PUT methods.
- **Zone 3** contains requests that only use the GET method.
- **Zones 4 and 5** contain requests that only use the POST method.
- **Zone 6** contains requests that use the POST and GET methods.

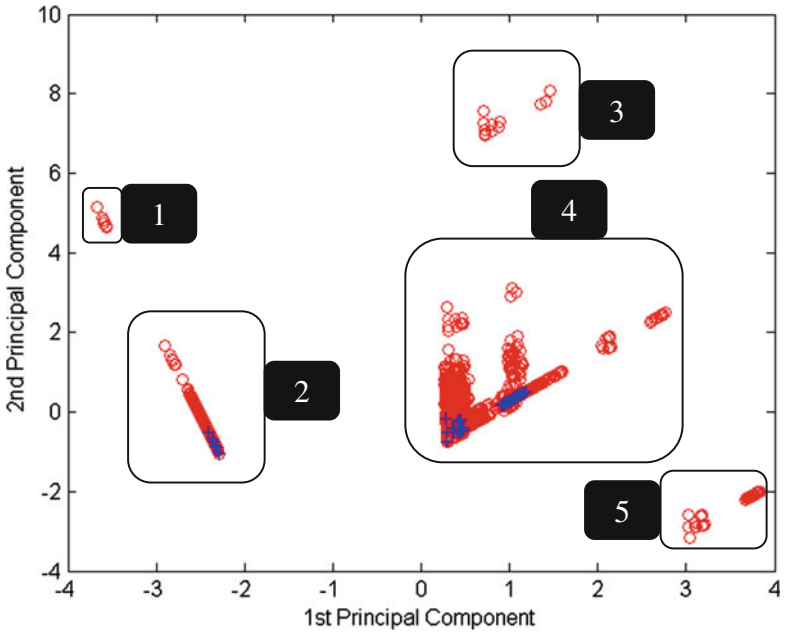
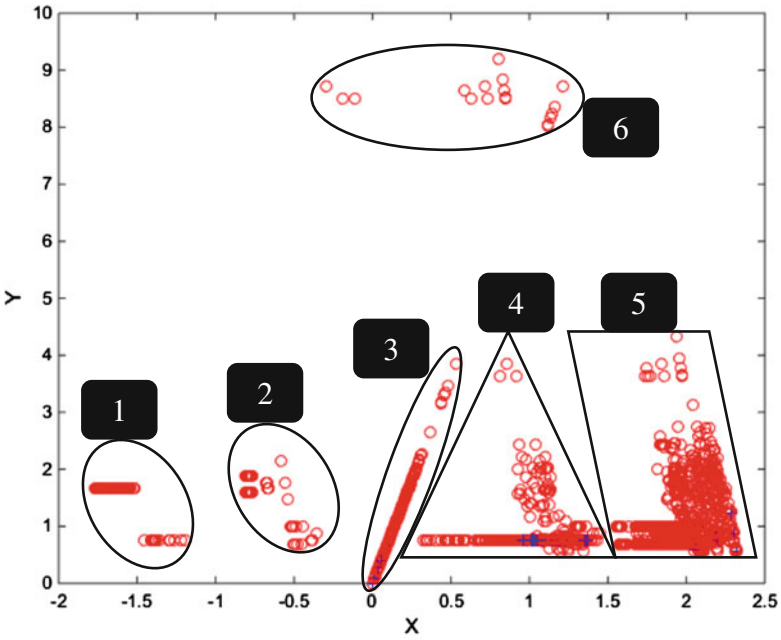
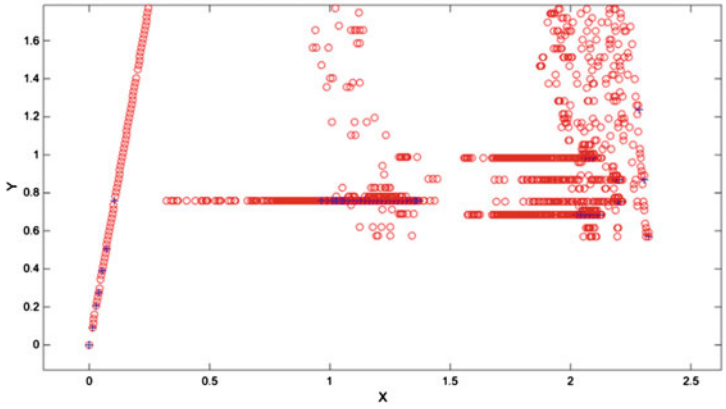


Fig. 1 PCA projection of the HTTP dataset CSIC 2010



2.a. General projection.



2.b. Zoom over zones 3, 4, and 5.

Fig. 2 CMLHL projection of the HTTP Dataset CSIC 2010. **2a** General projection. **2b** Zoom over zones 3, 4, and 5

As it happened in the case of PCA, CMLHL visualize the dataset in different groups. None of them contains only attack requests, but it can be seen (details on Fig. 2b) that all the attack requests are placed at the bottom-left side of groups 3, 4, and 5.

SOM Analysis

To study the SOM results, a 'specialized neuron' ratio has been calculated as performance criteria. It is defined as the number of neurons are specialized (in normal or anomalous data) and then, do not respond to samples of both normal and anomalous requests, divided by the total number of neurons in the given grid.

For the initial test, several experiments were conducted, with the following values for the SOM parameters:

- Size of the network: 4 × 4, 10 × 10, 20 × 20, and 30 × 30.
- Network topology: Grid and hexagonal.
- Distance criteria: 'dist', 'linkdist', 'mandist', and 'boxdist'.

Each combination of the above values was executed 10 times and the mean of the 'specialized neuron' ratio has been calculated, as shown in Table 2.

With the initial test, it was found that the best size for the SOM is 20 × 20 because their results (highest scores in the 'specialized neuron' ratio) are significantly better than those obtained for the other sizes. Additionally, Euclidean and Manhattan distance criteria provide slightly better results.

After obtaining an initial idea on the best parameter values, a second test was designed with the following parameter values:

- Size of the network: 15 × 15, 16 × 16, 17 × 17, 18 × 18, 19 × 19, 21 × 21, 22 × 22, 23 × 23, 24 × 24, and 25 × 25.
- Network topology: Grid and hexagonal.
- Distance criteria: Euclidean and Manhattan.

Table 2 'Specialized neuron' ratio for the different configurations in initial test

		Distance criteria			
Size	Topology	*'dist'*	*'linkdist'*	*'mandist'*	*'boxdist'*
4 × 4	Grid	0.64375	0.65000	0.66875	0.61875
4 × 4	Hex	0.64375	0.57500	0.65625	0.63750
10 × 10	Grid	0.66600	0.65800	0.64300	0.63600
10 × 10	Hex	0.65800	0.65800	0.66600	0.66400
20 × 20	Grid	0.72225	0.71725	0.72775	0.67125
20 × 20	Hex	0.73625	0.69850	0.74675	0.72075
30 × 30	Grid	0.66550	0.66250	0.65940	0.62170
30 × 30	Hex	0.65710	0.63760	0.67980	0.65780

As for the first test, each parameter combination is executed 10 times and the mean value is calculated. The performance of this second test is shown in Table 3.

As a result, it can be concluded that grid size close to 20 × 20 obtains good results. Manhattan distance works slightly better than Euclidean distance and hexagonal topology seems to be the best option. However, none of the experiments obtained a value of 1 in the 'specialized neuron' ratio.

Finally, for the SOM analysis, a representation of the best configuration found (21 × 21 size, hexagonal topology and Manhattan distance) is shown in Fig. 3. Each neuron is colored according to the type of HTTP requests it responds to:

- Gray: the neuron responds to no requests.
- Red: the neuron responds only to anomalous requests.
- Blue: the neuron responds only to normal requests.
- Orange: the neuron responds to both normal and anomalous requests.

Additionally, figures inside each neuron indicates the number of normal / anomalous HTTP requests the neuron responds to. As can be seen in Fig. 3, none of the neurons responds only to normal requests (blue color), and then normal request are mixed with anomalous ones (orange color).

Table 3 'Specialized neuron' ratio for the different configurations in second test

		Distance criteria	
Size	Topology	*'dist'*	*'mandist'*
15 × 15	Grid	0.6951	0.7036
15 × 15	Hex	0.7120	0.7351
16 × 16	Grid	0.6973	0.6945
16 × 16	Hex	0.7105	0.7273
17 × 17	Grid	0.7118	0.7073
17 × 17	Hex	0.7142	0.7239
18 × 18	Grid	0.7188	0.7194
18 × 18	Hex	0.7204	0.7420
19 × 19	Grid	0.7249	0.7202
19 × 19	Hex	0.7288	0.7357
21 × 21	Grid	0.7195	0.7168
21 × 21	Hex	0.7231	0.7485
22 × 22	Grid	0.7207	0.7157
22 × 22	Hex	0.7283	0.7390
23 × 23	Grid	0.7130	0.7089
23 × 23	Hex	0.7161	0.7456
24 × 24	Grid	0.7122	0.7038
24 × 24	Hex	0.7097	0.7420
25 × 25	Grid	0.7034	0.7021
25 × 25	Hex	0.7014	0.7362

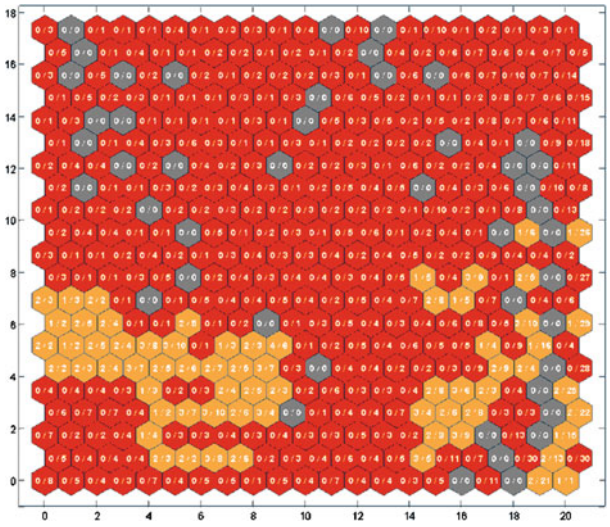


Fig. 3 SOM representation for the best configuration

4 Conclusions and Future Work

In present paper, several neural models have been proposed for the analysis of HTTP traffic, trying to detect web based attacks. As described in previous section, none of the applied models have been able to clearly differentiate normal from anomalous traffic. There are some kinds of data that can be certainly identified as normal or anomalous (see some groups in Sect. 3).

One of the main reasons for the low performance of neural visualization applied to the HTTP Dataset CSIC 2010 is that, the preprocessing process implied removing certain information that may support differentiating the different kinds of traffic. Thus, future work will be based on different ways of collecting and converting HTTP data for improving their neural visualization. More precisely, the original features of the dataset will be processed in a way that more information can be provided to the neural models.

References

1. Myerson, J.M.: Identifying enterprise network vulnerabilities. Int. J. Network Manage **12**(3), 135–144 (2002)
2. Fielding, R., Reschke, J.: Hypertext transfer protocol (HTTP/1.1): message syntax and routing. IETF RFC **7230** (2014)
3. Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Berners-Lee, T.: Hypertext transfer protocol – HTTP/1.1. IETF RFC **2068** (1997)

4. Crist, J.: Web based attacks. SANS institute - infosec reading room (2007)
5. Ponemon Institute - Cost of Cyber Crime Study (2014)
6. Kaspersky Security Bulletin 2014 (2014)
7. Pastrana, S., Torrano-Gimenez, C., Nguyen, H., Orfila, A.: Anomalous web payload detection: evaluating the resilience of 1-grams based classifiers. In: Camacho, D., Braubach, L., Venticinque, S., Badica, C. (eds.) Intelligent Distributed Computing VIII, vol. 570, pp. 195–200. Springer International Publishing (2015)
8. Choraś, M., Kozik, R.: Machine learning techniques applied to detect cyber attacks on web applications. Logic J. IGPL **23**(1), 45–56 (2014)
9. Corchado, E., Herrero, Á.: Neural visualization of network traffic data for intrusion detection. Appl. Soft. Comput. **11**(2), 2042–2056 (2011)
10. Pinzón, C.I., De Paz, J.F., Herrero, Á., Corchado, E., Bajo, J., Corchado, J.M.: idMAS-SQL: intrusion detection based on MAS to detect and block SQL injection through data mining. Inf. Sci. **231**, 15–31 (2013)
11. Herrero, Á., Zurutuza, U., Corchado, E.: A neural-visualization IDS for honeynet data. Int. J. Neural Syst. **22**(2), 1–18 (2012)
12. D'Amico, A.D., Goodall, J.R., Tesone, D.R., Kopylec, J.K.: Visual discovery in computer network defense. IEEE Comput. Graphics Appl. **27**(5), 20–27 (2007)
13. The MathWorks, Inc., Natick, Massachusetts, United States.: MATLAB (2014)
14. Demartines, P., Herault, J.: Curvilinear component analysis: a self-organizing neural network for nonlinear mapping of data sets. IEEE Trans. Neural Networks **8**(1), 148–154 (1997)
15. Pearson, K.: On lines and planes of closest fit to systems of points in space. Phil. Mag. **2**(6), 559–572 (1901)
16. Hotelling, H.: Analysis of a complex of statistical variables into principal components. J. Educ. Psychol. **24**, 417–444 (1933)
17. Oja, E.: Principal components, minor components, and linear neural networks. Neural Networks **5**(6), 927–935 (1992)
18. Fyfe, C.: A neural network for PCA and beyond. Neural Process. Lett. **6**(1–2), 33–41 (1997)
19. Corchado, E., Fyfe, C.: Connectionist techniques for the identification and suppression of interfering underlying factors. Int. J. Pattern Recognit Artif Intell. **17**(8), 1447–1466 (2003)
20. Corchado, E., MacDonald, D., Fyfe, C.: Maximum and minimum likelihood hebbian learning for exploratory projection pursuit. Data Min. Knowl. Disc. **8**(3), 203–225 (2004)
21. Fyfe, C., Corchado, E.: Maximum likelihood hebbian rules. In: 10th European Symposium on Artificial Neural Networks (ESANN 2002), pp. 143–148 (2002)
22. Kohonen, T.: The self-organizing map. Proc. IEEE **78**(9), 1464–1480 (1990)
23. Ritter, H., Martinetz, T., Schulten, K.: Neural Computation and Self-Organizing Maps; An Introduction. Addison-Wesley Longman Publishing Co., Inc., Chicago (1992)
24. HTTP DATASET CSIC 2010: http://www.isi.csic.es/dataset/