Digital Signature to Help Network Management Using Principal Component Analysis and K-Means Clustering

Gilberto Fernandes Jr.*, Alexandro M. Zacaron*, Joel J. P. C. Rodrigues† and Mario Lemes Proença Jr.*
*Computer Science Department, State University of Londrina (UEL), Londrina, Brazil
†Instituto de Telecomunicações, University of Beira Interior, Portugal
Email: {gil.fernandes6, zacaron}@gmail.com, joeljr@ieee.org, proenca@uel.br

***Abstract*—The complexity of a network nowadays and its increasingly amount of traffic data has contributed to the occurrence of problems and anomalies. A traffic characterization, called Digital Signature for Network Segment using Flow Analysis (DSNSF) is important to help Network Management in avoiding these problems. We propose two methods to generate a digital signature capable of describing the traffic behavior. For this purpose, we used the statistical method Principal Component Analysis (PCA) and the clustering algorithm K-Means. The resulting DSNSFs are then submitted to testing with real data to evaluate its precision.**

***Index Terms*—DSNSF; PCA; K-Means; Flows; Traffic Characterization;**

I. Introduction

The popularization of Internet services has contributed to an explosive growth in the number of computers connected to a network, and with new technologies arising, connections are increasingly getting faster. All of that leads to a huge volume of traffic data, and as it happens, problems and failures may occur more often.

Server crashes, link congestion, software failures and Denial of Service (DoS) attacks, are just some examples of several problems that threaten the entire operability of a network, compromising the connection, or even blocking users access to some services.

To overcome these problems, Network Management is an important area that has become crucial to ensure a network's availability and reliability. By monitoring the network's behaviour, it is possible to identify traffic patterns to avoid potential problems. However, in today's large scale and complex network systems, there is no way of performing manual tracking and analysis. Hence, the idea of automating this management process thought tools and techniques has emerged, and it is called Autonomic Management [1].

A fundamental step for Network Autonomic Management is the adoption of an efficient method for network traffic characterization [2]. For this purpose, we use the Digital Signature for Network Segment using Flow Analysis (DSNSF), which does not require a network administrator to be monitoring traffic graphics all the time. The DSNSF will act as a threshold to generate alarms, so the administrators could direct its efforts only where there might be a problem or anomaly.

As presented in Denning [3] and Patcha [4], the anomalies detection can be classified in two ways: based in signature, in which you have a prior knowledge about the type of attack; and based in a profile that presents a history of the network behavior, by using statistical methods, data mining tools and other techniques. That profile is the DSNSF we propose in this paper.

The two methods we propose in this article aims to create a DSNSF by extracting traffic data traces from IP flow records. A Flow is defined as a set of packets passing in a network observation point during a certain period, in which they share a common set of properties, like source and destination IP addresses, ports and other features which constitutes a more accurate network planning and analysis [5].

In this paper, we introduce PCADS (Principal Component Analysis for Digital Signature), a traffic characterization method founded on Principal Component Analysis (PCA). PCA is a statistical procedure used for dimensionality reduction of a multivariate dataset [6]. The PCADS algorithm analyses the data set and identifies the network traffic movement that has the largest eigenvalue. This feature means that the traffic movement identified stands out among the data set, and it can be used to efficiently represent the normal behaviour of a network segment.

Also, we present KMDS (K-means for Digital Signature), which uses K-means clustering [7], a data mining tool used to find and quantify similarities between points of a determined group of data. This process seeks to minimize the variance between elements of a given group, and maximize them in relation to other groups.

The main contributions of this paper consists of generating two signatures through PCADS and KMDS, so that they can describe the behavior of a network segment, and also the learning time required for each algorithm for traffic characterization. To evaluate the proposed methods, we tested them at the gateway of Federal University of Technology - Paraná (UTFPR) - Toledo Campus.

The remainder of this paper is organized as follows: Section II presents the related work; Section III details the implementation of the two characterization methods; Section IV delivers the experimental results; finally, Section V concludes the paper.

978-1-4673-3122-7/13/$31.00 ©2013 IEEE

II. Related Work

Although it is not the focus of this paper, it is important to quote Lakhina et al. [8] anomaly detection method, which uses the Principal Component Analysis. PCA was used to efficiently separate the anomalous subspace, which is more noisy and contains the significant traffic spikes, from normal network-wide traffic, that is dominated by predictable traffic. After that subspace separation, it was possible to accurately diagnose volume anomalies.

Kanda [9] seeks to identify hosts infected by worms. By the use of flow-based communication patterns per host as a metric to identify anomalies using k-means clustering, the authors seek to find patterns that characterize anomalies.

Molnar and Moczar [10] propose a framework for traffic characterization of P2P, gaming, social networks and video playback applications. The usage of clustering make it possible to define groups that can represent each type of traffic using, what the authors call, characterization in three dimensions. This characterization is composed of three traffic features: size, duration and rate. With these three features. it is possible to identify which applications are travelling on the network. The authors identify that the behaviour of a social network packet size range is from 1kB to 350kB. Now, on YouTube, it is from 320kB to 26MB.

Rossi et al. [11] developed an algorithm that exploits behavioural flows for network traffic classification. Their classifier is an extension of the classification algorithm, and behavioural Abacus seeks to identify an application using two fields of flow records, bytes and packets. Results indicate a precision of 90% in the worst case for traffic volume.

III. Digital Signature of Network Segment using Flow analysis

In this section we present two methods to create the DNSNF using bits per second. The first one performs the traffic characterization using the Principal Component Analysis (PCA), a non-parametric method. And the other is based on K-Means clustering to achieve the same objective.

A. Principal Component Analysis for Digital Signature

Introduced by Karl Pearson in 1901 [12], the Principal Component Analysis (PCA) is an statistical method used in multivariate problems. It aims to reduce its dimensionality, by analysing the variance among all the input dimensions [6]. Then, that initial multivariate data set can be represented by a small set of dimensions without much loss of information.

The PCADS (Principal Component Analysis for Digital Signature) method presented in this paper uses a different interpretation of PCA algorithm in order to create a DSNSF for traffic characterization. In classic PCA, the input is a $n \times p$ matrix, composed by p columns representing the p variables (dimensions) of the problem and n lines, as the n samples of each variable. However, in PCADS, the input $n \times p$ matrix is constructed as follows: The p columns will represent each traffic movement day used to train the algorithm to generate the DSNSF, and the n lines will be the n samples of bits per second extracted from the flow records.

```
Algorithm 1 - PCADS used to DSNSF creation.
Input: Set of bits/s collected from historic database in the range of 5 minutes arranged in a $n \times p$ matrix
Output: $\mu$: Vector representing the bits/s sets of a day arranged in 288 intervals of 5 minutes, i.e. the DSNSF.

Step 1 Normalize the input data (mean deviation form).
Step 2 Calculate the covariance matrix.
Step 3 Calculate the eigenvectors and eigenvalues.
Step 4 Choose the eigenvector with the highest associated eigenvalue.
$\mu$ = eigenvector × original_data (1)
```

The PCA method used in the PCADS method is presented in Algorithm 1. First of all, it is required to subtract off the mean from each column in the input matrix. This is called mean deviation form, and it is important because it eases the covariance matrix calculation, and avoids distorted results due to differences in mean link utilization. Then, with the $n \times p$ matrix in mean deviation form, the algorithm calculates de covariance matrix, which is used to compute two important structures, the eigenvectors and eigenvalues. Each dimension has an associated eigenvector that points toward the variance of data, and an eigenvalue, a numerical value that indicates the significance of its associated dimension among the others.

The next step consists of selecting the eigenvector with the highest eigenvalue to reduce the data set to only one dimension. The chosen eigenvector indicates that the associated network traffic is the most relevant between the traffic of all days used in the input of the algorithm. In addition, it represents a higher percentage of the data variation. It is called the principal component. In equation (1), the eigenvector is multiplied with the original data in order to produce the final result. The original data is the input matrix $n \times p$ without subtracting off the mean.

$$new\_data\_set = eigenvector \times original\_data \qquad (1)$$

Finally, the algorithm returns a new dataset of just one dimension that will be used as the DSNSF to characterize the network traffic. The final result is based on the network traffic that is selected as capable of characterizing in a good way the input traffic data after the principal component analysis.

B. K-Means for Digital Signature

K-means (KM) is a process that divides a n-dimensional population in K groups based on a sample. KM partitions points of the data matrix, which can be a vector or a matrix, in k clusters. The matrix rows correspond to the points, and the columns correspond to variables [7].

With the aid of clustering, which is a data mining technique, we seek to quantify similar data on certain groups. This process seeks to minimize the distance between the points of a given group, and increase the distance between groups. The equation that measures the similarity between the data is called the objective function and is described by (2).

$$J(p) = \sum_{k=1}^{K} \sum_{s=1}^{S} \sqrt{|P_s^k - c^k|^2} \tag{2}$$

Where K is the number of clusters, S is the number of points, P_s^k is the value of points belonging to the cluster k, and c^k correspond to the center of the cluster k. The purpose of using clustering is to create a template from which to extract a pattern of information. And also being able to identify data that has a default behavior and move away from this pattern.

To determine how many clusters would be used in the process of clustering, the Silhouette method [13] was used for interpretation and validation of clusters. Tests were performed for 3 to 6 clusters, and after the tests, the best results obtained were using four centers. Therefore, we used k = 4 for KMDS algorithm.

The Algorithm 2 hereafter demonstrates the pseudo code of KMDS.

Algorithm 2 - KMDS used to DSNSF creation.

Input: Set of bits/s collected in the range of 5 minutes, number of clusters.
Output: μ: Vector representing the bits/s sets of a day arranged in 288 intervals of 5 minutes, i.e. the DSNSF.

Step 1 Place k points in space that represent the points to be clustered. These points represent the initial set of centroid.

Step 2 Assign each point to the nearest group of the centroid.

Step 3 When all the points have been allocated, recalculates the position of k centroids.

Step 4 Repeat steps 2 and 3 until the centroid does not move more or the number of iterations is exceeded.

```
for i = 1 : total cluster
  if number of points in the cluster $k(i) < \gamma$
    ignore the cluster $k(i)$
  end-if
end-for
```

μ = weight average between the centers

To avoid and escape the problem of local optimal of KM, KMDS algorithm was used with several repetitions and your centers starting randomly, so that a possible global optimum solution can be found, thereby ensuring that the KMDS obtains a better data clustering.

In order to be representative, a cluster must have at least γ points, preventing possible outliers or anomalies from compromising the traffic characterization. KMDS result is the sum of the weighted average clusters that have the most representative points.

IV. Evaluating the Proposed Methods

To generate a DSNSF with the proposed methods, we used real flow data collected at the Federal University of Technology - Paraná (UTFPR) - Toledo Campus.

We used Softflowd application [14] to export flows to the collector in the version 9 of NetFlow. Softflowd is a network analyzer capable of exporting data within the NetFlow pattern. The collector saves the exported flows in binary files every five minutes to be processed by NFDUMP tools [15] and NfSen [16], so they can be used later by the methods presented in this paper.

The two methods were developed using the $bl7$ methodology, introduced by Proença [2], in which a DSNSF is generated for each workday, based on the history of its previous weeks. To generate the DSNSFs, we analysed flow records collected from workdays of February, March, April and the first week of May of 2012, and then, we used the following week of May to compare and evaluate if the DSNSFs could express the normal network behavior of those days.

Therefore, to validate the results obtained, we used three evaluation techniques for an accurate analysis of the reliability of the two methods against the real traffic movement: Correlation, which indicates how much the DSNSFs are related with the real movement; Normalized Mean Square Error (NMSE), which evaluates the difference between the expected and what was actually verified [17]; and Fractional Standard Deviation (FSD), a performance measure that also compares the observed data with the real data.

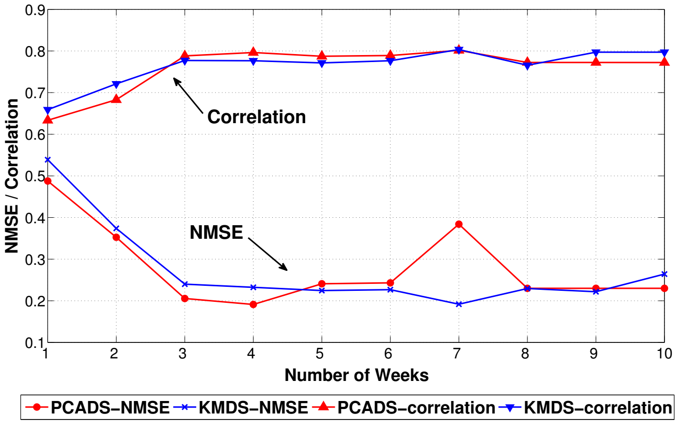

Fig. 2. NMSE and Correlation indices over the DSNSFs generated and the real movement of analyzed days using from 1 to 10 weeks

However, before any tests comparing the two methods, we aimed to find out the number of weeks PCADS and KMDS would achieve better results in generating the DSNSF. Figure 2 shows the Correlation and the NMSE of the DSNSFs created using from one to ten previous weeks of the second week of May, which was used to compare the DSNSFs with its real traffic movement. As we can see in Figure 2, PCADS method started to produce better results through the use of three weeks, but according to NMSE and the Correlation tests together, it achieves a better performance when using 4 weeks.

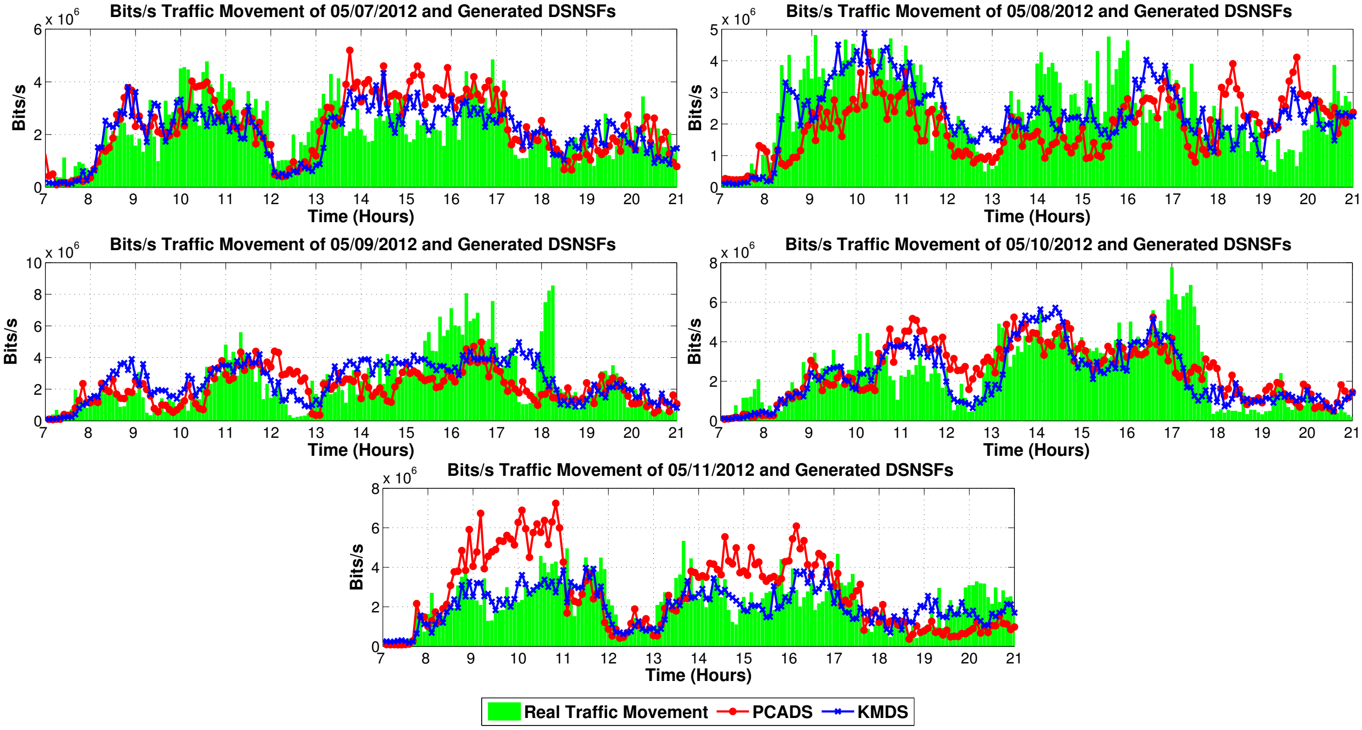

Fig. 1. Comparison between the actual traffic of bits per second and generated DSNSFs

Moreover, KMDS needs seven weeks to obtain good NMSE and Correlation indices.

In order to validate the methods, we generated digital signatures for each workday (from Monday to Friday), using four weeks to train PCADS and seven weeks for KMDS. Then, the two following weeks of May, from day 07 to 11 and from day 14 to 18, were used to compare the DSNSF curves provided by the methods with the real movement of the network segment analyzed in bits/s.

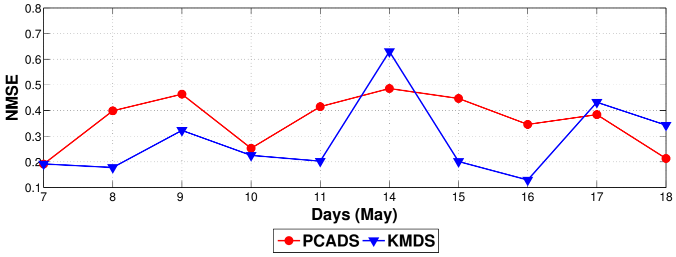

Fig. 3. NMSE tests between the generated DSNSFs and the movement of analyzed days

Figure 1 shows the generated DSNSFs for each workday compared with the traffic observed of an entire week of May 2012. The period is from 7th to 11th May. Each DSNSF is created to describe all the 24 hours of a day, but in the traffic figures of Figure 1 we used only the time interval from 7 to 21 hours to illustrated the results, since it is the period when the network is most used at UTFPR. As we can observe, the digital signature curves of PCADS and KMDS could estimate and characterize very well the real traffic, both accomplishing similar results.

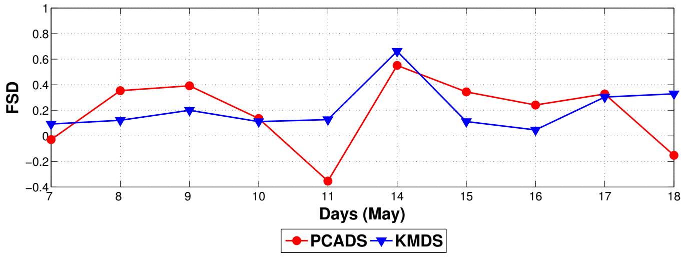

Fig. 4. Fractional Standard Deviation (FSD) tests between the generated DSNSFs and the movement of analyzed days

The two methods were submitted to the Normalized Mean Square Error (NMSE) test. As presented in Figure 3, KMDS showed good results, overlapping PCADS NMSE indices in many points. PCADS had higher error values comparing to KMDS, but they were not higher than 0.5, which is a good mark.

At last, Figure 4 delivers the results when applying the Fractional Standard Deviation. Its indices varies between -2 (underestimate) and +2 (super estimate), with 0 (zero) value as the optimal measure. Again, KMDS achieved an uniform behavior in the range of 0 to 0.65 whereas PCADS had a little variation, but remaining with indices between -0.4 and 0.6.

A. Computational Complexity

For PCADS method, computing all the principal components of a given $n \times p$ matrix X, according to Lakhina et al. [8], is equivalent to solving the symmetric eigenvalue problem for a covariance matrix $X^T X$. To solve this problem, it is necessary to compute the Singular Value Decomposition (SVD), a method used to get the eigenvectors and eigenvalues of a matrix X. So, the computational complexity of a complete SVD of a $n \times p$ matrix is limited by O(np^2).

KMDS has an initial complexity in the order of O(X), with X as the size of the data set to be clustered, and for each point in X, KMDS calculates an association. Since KMDS is based on centers, a cost C is generated every time a center updates, turning the complexity into O(XC). Another parameter used is the number of dimensions D and the number of iterations of the collected data, resulting in a final complexity of O(XCDI).

By comparing the complexity results of each method, and also how much time each one takes to generate the DSNSF under a 3.0 GHz Intel-based processor, PCADS has reached a better performance. For PCADS, the square term of its asymptotic notation is p, which represent the number of days used to generate the DNSNF, and since the method can achieve better results using only four previous days, it takes only less than three seconds to create a DSNSF. Furthermore, KMDS depends on four variables and, according to our tests, seven previous days to create a digital signature. This implies on the usage of a little more computational resources than PCADS does.

V. Conclusion

In this paper, we proposed and evaluated two new approaches to create a digital signature to help network management: PCADS and KMDS. According to the tests we made, both methods achieved good results, pointing out that they can efficiently work as a traffic characterization method.

KMDS had better results than PCADS in both NMSE and FSD tests, with an average of 0.3 for NMSE and 0.2 for FSD. This means good results since they are close to 0 (zero), which is the optimal value of these tests. Besides that, PCADS showed promise due to its smaller learning time for traffic characterization, and the usage of only four previous weeks, obtained through NMSE and Correlation tests. Also, PCADS has a smaller computational complexity and running time.

For future work, the goal is to improve both methods by adding an autonomous alarm system for anomaly detection, since they already can describe the normal behavior of a network segment through the DSNSF. Also, we will analyse more flow features, such packets/s, number of flows, ports and IP addresses.

Acknowledgment

This work was supported by Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), due to the concession of scholarships; SETI/Fundação Araucária and MCT/CNPq for Rigel and Bellatrix Projects' financial support; Instituto de Telecomunicações, Next Generation Networks and Applications Group (NetGNA), Portugal; and by National Funding from the FCT – Fundação para a Ciência e a Tecnologia through the PEst-OE/EEI/LA0008/2011 Project.

References

[1] A. G. Prieto and R. Stadler, "Adaptive real-time monitoring for large-scale networked systems," in *Proceedings of the 11th IFIP/IEEE international conference on Symposium on Integrated Network Management*, ser. IM'09. Piscataway, NJ, USA: IEEE Press, 2009, pp. 790–795. [Online]. Available: http://dl.acm.org/citation.cfm?id=1688933.1689038

[2] M. Proena, C. Coppelmans, M. Bottoli, and L. Souza Mendes, "Baseline to help with network management," in *e-Business and Telecommunication Networks*, J. Ascenso, L. Vasiu, C. Belo, and M. Saramago, Eds. Springer Netherlands, 2006, pp. 158–166.

[3] D. Denning, "An intrusion-detection model," *Software Engineering, IEEE Transactions on*, vol. SE-13, no. 2, pp. 222 – 232, feb. 1987.

[4] A. Patcha and J.-M. Park, "An overview of anomaly detection techniques: Existing solutions and latest technological trends," *Computer Networks*, vol. 51, no. 12, pp. 3448 – 3470, 2007. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S138912860700062X

[5] F. Fatemipour and M. H. Yaghmaee, "Design and implementation of a monitoring system based on ipfix protocol," in *Proceedings of the The Third Advanced International Conference on Telecommunications*, ser. AICT '07. Washington, DC, USA: IEEE Computer Society, 2007, pp. 22–. [Online]. Available: http://dx.doi.org/10.1109/AICT.2007.18

[6] I. Jolliffe, *Principal component analysis*. New York: Springer Verlag, 2002.

[7] J. B. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, L. M. L. Cam and J. Neyman, Eds., vol. 1. University of California Press, 1967, pp. 281–297.

[8] A. Lakhina, M. Crovella, and C. Diot, "Diagnosing network-wide traffic anomalies," in *Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications*, ser. SIGCOMM '04. New York, NY, USA: ACM, 2004, pp. 219–230. [Online]. Available: http://doi.acm.org/10.1145/1015467.1015492

[9] Y. Kanda, K. Fukuda, and T. Sugawara, "A flow analysis for mining traffic anomalies," in *Communications (ICC), 2010 IEEE International Conference on*, may 2010, pp. 1 –5.

[10] S. Molnar and Z. Moczar, "Three-dimensional characterization of internet flows," in *Communications (ICC), 2011 IEEE International Conference on*, june 2011, pp. 1 –6.

[11] D. Rossi and S. Valenti, "Fine-grained traffic classification with netflow data," in *Proceedings of the 6th International Wireless Communications and Mobile Computing Conference*, ser. IWCMC '10. New York, NY, USA: ACM, 2010, pp. 479–483. [Online]. Available: http://doi.acm.org/10.1145/1815396.1815507

[12] K. Pearson, "On lines and planes of closest fit to systems of points in space," *Philosophical Magazine*, vol. 2, no. 6, pp. 559–572, 1901.

[13] P. J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *Journal of Computational and Applied Mathematics*, vol. 20, no. 0, pp. 53 – 65, 1987. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0377042787901257

[14] D. Miller, "Softflowd - traffic flow monitoring," 2010, [Online; accessed 28-May-2011]. [Online]. Available: http://manpages.ubuntu.com/manpages/maverick/man8/softflowd.8.html

[15] P. Haag, "NFDUMP - NetFlow processing tools," Sep. 2004. [Online]. Available: //nfdump.sourceforge.net

[16] ——, "NetFlow visualisation and investigation tool," Mar. 2005. [Online]. Available: //nfsen.sourceforge.net

[17] A. A. Poli and M. C. Cirillo, "On the use of the normalized mean square error in evaluating dispersion model performance," *Atmospheric Environment. Part A. General Topics*, vol. 27, no. 15, pp. 2427 – 2434, 1993. [Online]. Available: http://www.sciencedirect.com/science/article/pii/096016869390410Z