Università degli Studi di Salerno
Dipartimento di Informatica

Dottorato di Ricerca in Informatica - X Ciclo

Tesi di Dottorato

# Network Anomaly Detection Based On The Observation Of Multi-Scale Traffic Dynamics

Francesco Palmieri

Anno Accademico 2010-2011

| Candidato: | Coordinatore: | Tutor: |
|---|---|---|
| Francesco Palmieri | Prof. Giuseppe Persiano | Prof. Alfredo De Santis |

*To my father Alfonso*

*for being my unwitting accomplice in this adventure*

# Abstract

With the rapid growth and the ever increasing complexity of the modern network infrastructures, the task of identifying and preventing network abuses is getting more and more strategic to ensure an adequate degree of protection from both external and internal menaces. In this scenario many techniques are emerging for inspecting the network traffic and modeling anomalous and normal behaviors to detect undesired or suspicious activities. First of all, the definition of normal or abnormal network behavior depends on several factors related to the day-to-day operations and resource usage. Normal behavior can only be determined by acquiring information about past events, but traffic trends usually take time to be understood and analyzed. This paradox can only be coped with by modeling the future behavior, based on a statistical idealization of the past events and an observation of the present ones and by specifically analyzing and observing some particularly discriminating statistical features and evolutive phenomena that occur on the network traffic. Since anomalous events are now conceived to be a structural part of the overall network traffic, it is more and more important to automatically detect, classify and identify them in order to react promptly and adequately. Accordingly the main focus of this dissertation is on developing a novel approach to network anomaly detection based on the analysis of complex non-stationary properties and "hidden" recurrence patterns occurring in the aggregated IP traffic flows. In the observation of the above transition patterns for detecting anomalous behaviors, we adopted several techniques that are known to be effective in exploring the hidden dynamics and time correlations of

statistical time series, such as wavelet and recurrence quantification analysis. The resulting model, using supervised machine learning techniques to adaptively classify the traffic time series from the aforementioned observations, demonstrated to be effective for providing a deterministic interpretation of nonlinear patterns originated by the complex traffic dynamics observable during the occurrence of "noisy" network anomaly phenomena, characterized by measurable variations in the statistical properties of the traffic time series, and hence for developing qualitative and quantitative observations that can be reliably used in detecting such events.

# Acknowledgements

This Ph.D. thesis takes its origins from some of my research efforts of the last three years, but it would not have been possible without the contributions and support of many people to whom I will always be grateful.

To start with, I would like to deeply thank my supervisor Alfredo De Santis and my course chairman Pino Persiano, for their friendly support and guidance during my studies, as well as for giving me the opportunity of taking this PhD.

I am also infinitely grateful to my dear friends and colleagues Ugo Fiore, who collaborated with me to most of the experiences on which this thesis work is based, and Nello Castiglione, who during this cycle of studies has been an invaluable source of friendship, as well as opportunities, good advices and collaboration.

Last but not least, and this is one of those cases where the "not least" part is the topmost one, thanks to my wife Adele and to my sons Alfonso and Francesca who put up with my endless hours on the computer and dealt with all of my absences with a smile.

Finally, I thank my parents, for their endless support and encouragement and in particular my father, for being the unwitting cause of this adventure. I dedicate this dissertation to him.

# Contents

# List of Figures

# Chapter 1

# Introduction

Network anomalies, circumstances in which the network behavior deviates from its normal operational baseline, can be due to various factors such as network overload conditions, malicious/hostile activity, denial of service (DoS) attacks and network intrusions that disrupt the correct delivery of network connectivity services. Detecting, identifying and classifying these operational and security hazards is very important but at the same time hard. First of all, the definition of normal behavior depends on several elements and considerations associated to the daily human activities accomplished by using the network. These include the traffic volumes generated, the applications running on the network, and the data they process. The main goal of anomaly detection is to devise techniques that are able to model what a normal operating network should look like and immediately report any deviations from such normal behavior. These techniques are typically based on machine learning, data mining or statistical analysis.

A fundamental challenge related to the detection of anomalous events on the network traffic is that these anomalies are a continuously evolving target. It is difficult to precisely and permanently define all the possible types of anomalies that can be experienced, especially in the case of malicious network traffic. New previously unknown anomalies can continuously emerge over time. Consequently, anomaly detection systems should avoid being limited by the knowledge of any

predefined set of anomalies and should be able to flexibly recognize/classify any unknown event affecting the network operations.

Consequently, the ultimate aim of anomaly detection systems is to achieve an adaptive behavior that responds in "real-time", so that all the critical network events can be handled as soon as possible. However, the normal network behavior can only be determined by using the experiential knowledge acquired from the past events history, whereas, on the other hand, reliable historical trends take time to be learnt and analyzed. To cope with this paradox we can only build an effective model of future behavior, based on a statistical idealization of the past events combined with an observation of the present ones (like in weather forecasting) and by specifically analyzing some particularly discriminating statistical features and evolutive phenomena observable on the network traffic. A timely response requires rapid processing of the observations originating from network monitoring devices, that are often capable to collect data at very high rates. Consequently, designing an effective anomaly detection system involves extracting and processing only the really relevant information from a large amount of noisy, multi-dimensional data.

Early integrated approach to the wider theme of *intrusion detection* were based on the assumption that most anomalous events can be revealed from the occurrence of a set of signatures flagging the known hostile activities and their specific communication patterns occurring in the stream of network packets. Unfortunately, while very efficient in real-time response, such systems are clueless when exposed to novel attacks, or even slight modifications of already known ones where the attack pattern does not closely match stored signatures or known communication behavior. In presence of new unknown types of attacks the necessary up-to-date signatures are generated mainly through human intervention as soon as the community become aware of the menace and the first detailed information about the attack dynamics and behavior become available. This may require a not negligible time that is clearly unacceptable when real-time or timely response to anomalies is strictly necessary.

On the other side, the anomaly detectors that we see on today networks are very simplistic: they observe several transport layer statistics such as the ratio of

the bytes sent in each direction, the average size and mean inter-arrival time of the packets etc., and the observation results are compared against constant pre-assigned threshold values that are often independent from the current network utilization and from the number of users. All the instances associated to crossings of the threshold value are flagged as anomalous. While more effective and flexible of signature based ones, the trouble with these anomaly detectors, based on the use of pure statistical methods, is that they take a rather narrow view of what an anomaly means, that is, they rely only on network traffic measurements taken over a short time scale, to "motivate" observed anomalies. Such a limited "view" strongly affects their detection reliability.

The second problem with these anomaly detection systems is due to the fact that, by aggregating the traffic statistics before processing them, they seriously risk to lose the information about the specific flows causing the anomaly. More specifically, the basic observation that makes traditional statistic time-series analysis unreliable for anomaly detection is that there is an inhomogeneous pattern associated to human/computer resource usage, and this pattern is clearly reflected in network resource usage. Furthermore, statistical detection techniques need to be based on very accurate statistical distributions describing the observed traffic phenomena, but its widely recognized that not all the network-related behaviors can be modeled by using only pure statistical methods. Accordingly, most of the known statistical anomaly detection techniques are based on the assumption of quasi-stationariety, which does not hold for most of the traffic data processed by network anomaly detection systems. Of course, all the models that do not need to make this assumption are significantly more complex and heavy in terms of required processing time.

However, statistical-based anomaly detection approaches have the fundamental advantage of not requiring prior knowledge of security flaws and/or the attacks themselves. Being based only on the statistical properties characterizing the traffic flows (in terms of packet exchange activity) under scrutiny such approaches does not require any content inspection activity.

By considering the above strengths, we propose a novel anomaly detection strategy, particularly suitable for IP networks, based on the statistical analysis of nonlinear traffic properties, more precisely, on the evaluation of the non-stationary hidden transition patterns in end-to-end traffic flow time series, that become evident only on multiple time scales. Such strategy takes its origins from several research efforts and experiences developed during the last three years of study ([1][2][3]), that have been collected and re-organized in a promising framework of new ideas and techniques that can be useful for the development of next-generation anomaly detection systems.

## 1.1 Our Contribution

Network traffic features and hence the probability distributions characterizing their IP-layer packets may change dynamically in the time domain [13][14] according to specific network conditions. These features, reflecting different dimensions of observation such as the communication intensity/burstiness (i. e. number of transactions/flows, average packet length, packet inter-arrival times etc.), and connection dispersion (i. e. number of corresponding counterparts), may be depending on other more intrinsic variables determined by the hidden "laws" governing the traffic itself (i.e. power laws and other fractal-like behaviors). These intrinsic variables are clearly closer to the fundamental system dynamics, whose common ground resides in their shared recurrence properties. More precisely, within the dynamical signals expressed by their associated time series we can find several stretches, short or long, of repeating patterns that are likely to be related to some hidden non-linear system properties and are subject to phase transition phenomena. These properties can be used to flag anomalous behavior more sharply than the immediately observable features. That is, each intrinsic components contains a great deal of information about the real traffic properties, and hence, being strictly necessary to describe it, can be more sensitive to specific volume based network anomalies and their related disorders than other directly available statistic features. Such more sensitive components

can be considered as the really useful signals to be inspected in the classification of anomalous or normal phenomena that are less influenced by noise or events observable on a short time scale and hence can be more effective for investigation.

Accordingly, we looked into the traffic time series by examining the fundamental dynamics of aggregated and per flow traffic transmitted over time (stationariety, determinism, linearity, distributions etc.) together with the associated fractal properties (self similarity and long range dependence). In particular we gathered the feature information needed to classify the network traffic as normal or anomalous by using several techniques based on nonlinear analysis and, more precisely, on the evaluation of the hidden recurrences and emerging non-stationary transition patterns in end-to-end traffic.

A preliminary study has been performed with the objective of characterizing the time series in terms of power spectral distribution, stationariety, determinism, linearity, distributions etc. For example, to determine if the original time series were characterized by a stationary Gaussian linear process the Surrogate time series method has been used.

Then, state space reconstruction techniques based on embedding and time delay dimensions were used to carry out the analysis in the reconstructed state space. Recurrence Quantification Analysis (RQA) has been applied to observe and study the aforementioned hidden properties of the traffic transition patterns and to determine a first set of recurrence-based features. We also used several Wavelet-based techniques to perform multi-resolution analysis at different scales to obtain a better understanding of the data generating process as well as of the most characterizing and discriminating traffic dynamics, in form of local feature information, hidden behind long time series. All the previous features were repeatedly estimated at fixed time intervals, or epochs, in order to obtain all the points in the feature space describing the dynamically variable traffic properties to be used in distinguishing anomalous events from normal behavior.

For this sake, we developed an adaptive non-parametric anomaly detection strategy based on a traditional Minimum Volume Set or binary classification machine

learning scheme, driven by support vector machines (SVM) to classify anomalous events occurring within the sequence of feature vectors estimated on each epoch. Such strategy has been conceived to be particularly sensitive to the "manifestation" of each suspicious network activity rather than to the explicit mechanism behind it, which is clearly unknown for new, totally unknown menaces. Accordingly, it can be more effective against specific phenomena explicitly involving measurable variations in the statistical properties of the traffic time series. For the same reason, it is not a viable choice for detecting anomalous behaviors affecting only packet/transactions contents (e. g. buffer overflow or other malicious code exploit attempts) or designed to be undistinguishable from regular user-originated network activities.

Starting from its basic theoretical perspectives, the proposed approach strongly differentiates from all the traditional volume-based detection schemes that make specific assumptions on the mathematical structure of traffic data or on the distribution and content of traffic flows/transactions. In addition it does not rely on assumptions of stationariety and does not need to consider the studied traffic data as the output of a linear dynamical system.

All the architectural choices behind the detection schema demonstrated the possibility of a pure operational use of concepts and techniques derived by complex traffic and systems dynamics for developing deterministic qualitative and quantitative observations that can be reliably end effectively used in flagging anomalous events characterized by measurable variations in the statistical properties of the traffic time series. We provided an extensive evaluation of the proposed framework to assess its detection effectiveness under various test case scenarios, demonstrating a quite satisfactory identification accuracy on real world traffic data.

## 1.2 State of the art

Anomaly detection has been studied widely and has received an increasing attention in the last years.

The concept of automatically spotting anomalies in computer security was first proposed by Anderson in [4]. Most of the works in the recent research literature treat anomalies as deviations in the overall traffic volume and employ several statistical techniques for detection: exponential smoothing and Holt-Winters forecasting [18], adaptive thresholding, cumulative sum [19][20], maximum entropy estimation [21], and principal component analysis [22]. Some of these works analyze the volume of aggregate traffic on a network link [11] [12], others identify different flows carried on several links, and finally others look at the time series of specific kinds of packets inside aggregate traffic, restricting their focus to few kinds of attacks.

The SPADE [23], ADAM [24] and NIDES [25] systems learn a statistical frequency-based model of normal network traffic based on the distribution of most anomalous attributes like addresses and ports per transaction, and flag deviations from this model. In contrast, other anomaly detection systems like PHAD [26], ALAD [27], LEARD [28] and NETAD [29] monitor a larger set of fields of the packet header and use more complex time-based models, where an event's probability is only conditioned by the time elapsed since its last occurrence. In [30] is presented an anomaly detection framework, called NOMAD, based on changes in routing paths, packet delay and statistic inference on the packet header information. The [31] approach considered the tradeoff between the attack detection probability, the false alarm ratio, and the detection delay to estimate whether SYN-flooding attacks happened or not. With this method, the anomaly can be detected after anomalous behaviors have threatened the network, and hence the real-time detection cannot be guaranteed.

More recent works have extended the range of techniques used: state-based transition analysis, neural networks, fuzzy logic, genetic algorithms, and N-gram analysis [31][32]. In [33] is presented an automated system to detect volume-based anomalies caused by DoS attacks, combining some traditional approaches such as adaptive threshold and cumulative sum, with a novel method based on the continuous wavelet transform. In [34], the authors have proposed monitoring the quantity of the SYN packets and the change of their ratio to other types of TCP packets to determine the presence of anomalies. In [35], spectral analysis is used to identify legitimate

TCP flows, which should exhibit strong periodicity.

The idea of using machine learning techniques for anomaly detection, by developing a generalization capability from training data to correctly classify future data as normal or abnormal has been exploited in many proposals [10], based on neural networks (NNs) [5] [6], support vector machines (SVMs) [7] and data mining [8]. These techniques can be further categorized as generative or discriminative approaches. A generative approach (e.g., [9]) builds a model solely based on normal training examples and evaluates each testing case to see how well it fits the model. A discriminative approach (e.g., [7]), on the other hand, attempts to learn the distinction between the normal and abnormal classes.

In contrast to the above approaches, our machine-learning based detection solution uses only the nonlinear characteristics of the network traffic to reveal, through recurrence quantification and wavelet analysis, all the variations on hidden periodicities that can reliably and rapidly flag the occurrence of abnormal events. To the best of our knowledge, this is the first anomaly detection approach leveraging only upon non-stationary phenomena and nonlinear properties of the network traffic dynamics.

## 1.3 The Organization of this Thesis

The main goal of this dissertation is to understand and propose a novel solution to the problem of automatic network anomaly detection.

We at first motivate the anomaly detection problem by providing the fundamental background information about current available methods/solutions and their known limitations, in the context of a state-of-the-art scenario. We also briefly describe the main contributions of our approach and the basic ideas beyond it.

In the second chapter we examined the fundamental dynamics characterizing traffic time series (stationariety, determinism, linearity, distributions etc.) together

with the associated fractal properties (self similarity and long range dependence). The known complexity of the network traffic has led to the suggestion that it cannot be analyzed within the framework of available traffic models. In particular, the strong evidences of long-range correlations and self-similarities within the context of a nonlinear and non-stationary behavior, suggest that nonlinear analysis techniques can be the best available choice for gaining a deeper understanding of the main features of traffic data to be inspected for detecting anomalous events.

These techniques, examined in detail in the third chapter, provide us very interesting insights into traffic periodic structures and clustering properties that are not apparent in the original time series and can be used to characterize the involved traffic dynamics in a more effective and discriminating way. Both Recurrence Quantification and Wavelet-based Multi-resolution Analysis are presented, within the context of a new approach to anomaly detection based on the study of multi-scale patterns, as effective tools for building the feature space.

In the fourth chapter we provide an overview of the machine learning model and methodologies used for classifying the feature vectors and the associated traffic events into anomalous or not.

More precisely, from the theoretical point of view, we casted the anomaly detection problem into a Minimum Volume Set determination and practically handled it as a binary classification scheme to be handled with Support Vector Machines.

In order to provide strong evidence on the effectiveness of our contribution, all the ideas and techniques proposed in this thesis have been validated by using real traffic data from a production network. Hence, we presented in the fourth chapter a proof of concept implementation of the proposed techniques. It started from a detailed study of the nonlinear dynamics of some specific traffic flow types to show how the use of features based on multi-scale recurrence phenomena and hidden non-stationary transition patterns can be effective in flagging volume-based anomalous events, within the aforementioned machine learning context.

Additionally, in the fifth chapter, their performance was compared against well-known results in analogous experiences, showing satisfactory results in most cases.

Finally, in the last chapter we will provide some conclusions about our work together with several interesting future research hints.

# References


[1] F. Palmieri, U. Fiore, "Insights into peer to peer traffic through nonlinear analysis", *16th IEEE Symposium on Computers and Communications (ISCC 2010)*, pp. 714-720, IEEE CS Press, ISBN 978-1-4244-7755-5.

[2] F. Palmieri, U. Fiore, "A non-linear, recurrence-based approach to traffic classification", *Computer Networks* 53(6) pp. 761773, 2009, Elsevier, ISSN 1389-1286.

[3] F. Palmieri, U. Fiore, "Network Anomaly Detection Through Nonlinear Analysis", *Computers & Security*, 29(7) pp. 737-755, Elsevier, 2010, ISSN 0167-4048.

[4] J. P. Anderson. "Computer Security Threat Monitoring and Surveillance", Technical Report, James P. Anderson Co., Fort Washington, Pennsylvania, April 1980.

[5] A. K. Ghosh and A. Schwartzbard. "A Study in Using Neural Networks for Anomaly and Misuse Detection", Proceedings of the 8th USENIX Security Symposium, pp. 23-36, Washington, D.C. US, 1999.

[6] V. N. P. Dao and V. R. Vemuri. "A Performance Comparison of Different Back Propagation Neural Networks Methods in Computer Network Intrusion Detection", *Differential Equations and Dynamical Systems*, vol 10, No 1-2, pp 201-214, Jan-April, 2002.

[7] S. Mukkamala, G. I. Janoski, and A. H. Sung. "Intrusion Detection Using Support Vector Machines", *Proceedings of the High Performance Computing Symposium - HPC 2002*, pp 178-183, San Diego, April 2002.

[8] W. Lee, S. J. Stolfo and K. Mok. "Data mining in work flow environments: Experiences in intrusion detection", *Proceedings of the 1999 Conference on Knowledge Discovery and Data Mining (KDD-99)*, 1999.

[9] C. Warrender, S. Forrest and B. Pearlmutter. "Detecting Intrusions Using System Calls: Alternative Data Models." *In Proceedings of 1999 IEEE Symposium on Security and Privacy*, pp 133-145, Oakland, 1999.

[10] T. Lane and C. E. Brodley. "An Application of Machine Learning to Anomaly Detection", *Proceedings of the 20th National Information Systems Security Conference*, pp 366-377, Baltimore, MD. Oct. 1997.

[11] M. Thottan, "Fault detection in IP networks," Ph.D. dissertation, Rensselaer Polytech. Inst., Troy, NY, 2000. Under patent with RPI.

[12] H. Wang, D. Zhang, and K. G. Shin, "Detecting syn flooding attacks," *in Proc. IEEE INFOCOM*, 2002.

[13] A. Tretyakov, H. Takayasu, M. Takayasu, "Phase transition pattern in a computer network", *Physica A* 253, pp. 315-22, 1998.

[14] M. Takayasu, H. Takayasu, K. Fukuda, "Dynamic phase transition observed in the Internet traffic flow", *Physica A* 277, pp. 248–55, 2000.

[15] M. Masugi, T. Takuma, "Multi-fractal analysis of IP-network traffic for assessing time variations in scaling properties", *Physica D* 225, pp. 119–126, 2007.

[16] C. L. Webber Jr. and J. P. Zbilut, "Dynamical assessment of physiological system and status using recurrence plot strategies", *Journal of Applied Physiol.*., vol. 76, pp. 965–973, 1994.

[17] N. Marwan and J. Kurths, "Nonlinear analysis of bivariate data with cross recurrence plots", *Phys. Lett. A*, vol. 302, pp. 299–307, 2002.

[18] J. Brutlag, "Aberrant behavior detection in time series for network monitoring", *USENIX Fourteenth System Administration Conference* LISA XIV, 2000.

[19] V. A. Siris, F. Papagalou, "Application of Anomaly Detection Algorithms for Detecting SYN Flooding Attacks", IEEE GLOBECOM, pp. 2050-2054, 2004.

[20] R. B. Blazek, H. Kim, B. Rozovskii, A. Tartakovsky, "A Novel Approach to Detection of Denial-of-Service Attacks via Adaptive Sequential and Batch-Sequential Change-Point Detection Methods", *IEEE Workshop Information Assurance and Security*, pp. 220-226, 2001.

[21] Y. Gu, A. McCallum, D. Towsley, "Detecting Anomalies in Network Traffic Using Maximum Entropy Estimation", *IMC Conference*, 2005.

[22] A. Lakhina, M. Crovella, C. Diot, "Diagnosing Network-Wide Traffic Anomalies", *ACM SIGCOMM 2004*, 2004.

[23] SPADE, Silicon Defense, http://www.silicondefense.com/software/spice/.

[24] D. Barbar, N. Wu, S. Jajodia, "Detecting Novel Network Intrusions using Bayes Estimators", *First SIAM International Conference on Data Mining*, 2001.

[25] D. Anderson et. al., "Detecting unusual program behavior using the statistical component of the Next-generation Intrusion Detection Expert System (NIDES)", Computer Science Laboratory *SRI-CSL 95-06*, 1995.

[26] M. Mahoney, P. K. Chan, "PHAD: Packet Header Anomaly Detection for Identifying Hostile Network Traffic", *Florida Tech. technical report 2001-04*, 2001.

[27] M. Mahoney, P. K. Chan, "Learning Models of Network Traffic for Detecting Novel Attacks", Florida Tech. technical report 2002-08, 2002.

[28] M. Mahoney, P. K. Chan, "Learning Nonstationary Models of Normal Network Traffic for Detecting Novel Attacks ", Edmonton, Alberta: *Proceedings SIGKDD*, pp 376-385, 2002.

[29] M. Mahoney, "Network Traffic Anomaly Detection Based on Packet Bytes", *Proceedings ACM-SAC*, pp. 346-350, 2003.

[30] R. Talpade, G. Kim, S. Khurana, "NOMAD: Traffic-based network monitoring framework for anomaly detection", Computers and Communications, *Proceedings. IEEE International Symposium*, pp. 442-451, 1999.

[31] V. A. Siris, F. Papagalou, "Application of anomaly detection algorithms for detecting SYN flooding attacks", *Global Telecommunications Conference, 29 (3)* pp. 2050-2054, 2004.

[32] P. Garcia-Teodoro, J. Diaz-Verdejo, G. Macia-Fernandez, E. Vazquez, "Anomaly-based network intrusion detection: Techniques, systems and challenges", *Computers & Security*, Volume 28, Issues 1-2, pp. 18-28, 2009.

[33] A. Dainotti, A. Pescape and G. Ventre, "Wavelet-based Detection of DoS Attacks", *IEEE Global Telecommunications Conference*, 2006.

[34] S. W. Shin, K. Y. Kim, J. Jang, "D- SAT: detecting SYN flooding attack by two-stage statistical approach", *Applications and the Internet, The 2005 Symposium*, pp 430-436, 2005.

[35] C. M. Cheng, H.T. Kung, K. S. Tan, "Use of spectral analysis in defense against DoS attacks", *Proc. IEEE GLOBECOM*, pp. 2143-2148, 2002.

# Chapter 2

# Analyzing nonlinear traffic dynamics

## 2.1 Introduction

The main goal of an anomaly detection system is discriminating the occurrence of hostile activities from the normal network traffic, and such analysis must be accomplished in a sufficiently fast way to keep up with an high speed network. In doing this, it must either try to model any kind of attack or anomalous event that can affect the network (there are thousands of known ones) or simply construct a sufficiently general model describing the normal traffic.

In doing this, is very important to be able to effectively describe this kind of traffic in a statistical manner, in terms of the intrinsic characteristics and properties of a number of objects, including packets, bursts, flows, connections, depending on the time scale of relevant statistical variations. Such features need to be combined within a common model, a simple, yet representative mathematical abstraction, totally representing the involved traffic dynamics, behavior and properties.

The preferred choice for modeling purposes depends on the basic objects to be analyzed over time in order to describe the traffic dynamics. Hence, in designing the model features it is necessary to bear in mind the properties and facilities characterizing each implied traffic object. Transport layer statistics such as the total number of packets sent, the ratio of the bytes sent in each direction, the packet inter-arrival time, the duration of each connection and the average size of the packets can be used to characterize the traffic behavior.

However, there are two main challenges to be faced with when modeling normal traffic for anomaly detection. First, network traffic is very complex and unpredictable, and second, the model is subject to changes over time. There are a significant number of protocols and applications characterizing the normal traffic, each with its own specific features and properties, and new ones can arise at any time.

The complexity shown by the traffic measurements has led to the suggestion that the network traffic cannot be analyzed within the framework of the available traditional traffic models.

In particular, a very common problem to be dealt with when modeling network traffic, originates from the extreme difficulty of determining in an analytical way, the statistical behavior of many observable events (for example, their average rate in bytes per second or packets per second), independently from the duration of the sampling period. For a long time it has been assumed that network traffic could be represented by using a Poisson model where the individual events are independent between each other. In this case, by measuring specific features (e.g. packet rates) we would notice random variations over small time windows (e.g. packets per second) that would tend to be averaged out over longer periods (e.g. weekly or monthly observations).

Furthermore, if traffic data should be characterized by a Markovian arrival process, it would exhibit a typical burst length that tend to be smoothed over a sufficiently long time scale.

Unfortunately, the experiences in [15] and [1] demonstrated that this is not true for many types of events. In addition, measurements on real Internet traffic traces indicated that a significant variance in traffic behavior (burstiness) can be observed on a wide range of time scales. That is, when observing traffic trends referring to packets per second or packets per day or month, regardless from the difference in scale, they show the same behavior characterized by bursts of sustained significant traffic rates separated by intervals of very limited activity. The same kind of bursty or or intermittent behavior could be observed either for a fraction of a second or for

hours/weeks, showing that the traffic rates distribution is almost independent from the time scale.

This behavior typically characterizes the self-similar or fractal processes, where the strong evidences of long-range correlations, within the context of a nonlinear and non-stationary behavior, suggest that nonlinear analysis techniques can be the best available choice for gaining a deeper understanding of the main features of traffic data. These techniques have the potential to provide very interesting insights into traffic periodic structures and clustering properties that are not apparent in the original time series and can be used to characterize the involved traffic dynamics in a more effective and discriminating way.

## 2.2 Identifying the normal traffic profile

Interpreting the dynamics characterizing a traffic pattern is central to the problem of defining when an event is an anomaly. "Anomalousness" is a subjective judgment, made within the context of past experience, and can be codified into a rule or criterion about what is sufficiently divergent from a normal traffic behavior. Evidently, the problem of anomaly detection in any complex system implies the existence of a subjacent concept of normality. The notion of "normal" is usually provided by a formal model expressing the relations between all the fundamental variables involved in the system dynamics. Consequently, when considering network traffic flowing across one or more observation points, an event is classified as anomalous because its degree of deviation from the "normal" (also known as "*baseline*") network behavior observed at these points is high enough.

The above concepts are simply depicted in Fig. 2.1 plotting a time series with some packet observations taken at fixed time intervals. The dotted line is the baseline. The shaded zone around the baseline can be considered as a tolerance area containing all the points whose distance from the baseline falls into a given range. The size of such area can be determined by using straightforward anomaly detection methods (see Introduction) working according to a simplistic threshold-based

detection paradigm. In this case, a threshold can be used to separate the less significant variations due to noise phenomena from the large ones due to unusual events. A time interval is assumed to contain an anomalous event when the value of the associated observation exceed the threshold, and hence falls outside the tolerance interval (labels A and B), for more than a specific (and significant) portion of the interval itself.

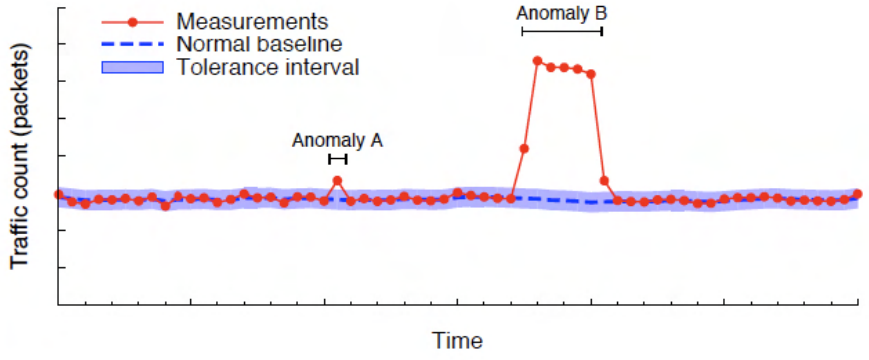



**Figure 2.1**: Anomaly detection scenario.

Thus, the first fundamental task in anomaly detection is network traffic baselining, that is the preliminary activity of measuring and rating the typical network behavior. Obtaining an effective and reliable network baseline requires profiling the normal network utilization, in terms of protocol usage, peak and average traffic rates, throughput estimation etc, collected over a significant period of time. This is a very slow and complex task requiring a lot of computing effort and human expertise, but fortunately it has to be performed only once, in the initial "knowledge construction" phase. Our approach to network baselining is based on the fact that any parametric network traffic model is an approximation to reality. We can construct our traffic model by looking at inter-arrival times of packet transmission and reception events, together with information about packet sizes, and attempting to use the memory of recent past to identify persistent events like end-to-end connections. Since, under normal traffic conditions, the system can be considered as approximately stable, i.e. close to a steady state, the occurrence of these events can be used to characterize

its recent history and hence the baseline traffic model parameters. Fluctuations can be measured as a time series, (a sequence of scalar samples $\{x_t\}_{t=1}^{T}$ measured at uniformly spaced time intervals) and analyzed in order to provide the necessary information about the deviation from the above baseline.

## 2.3 Analyzing the traffic time series

Working with real network traffic is much more complex than analyzing artificial data generated from simulation experiments. Real life traffic data is likely to exhibit periodicity (due to, for example, daily usage patterns), characteristic trends and sometimes quantization effects. Furthermore, such data is characterized by strong irregularities and a very complex chaotic behavior often described by nonlinear dynamics. Hence, the analysis of the associated time series has, among its goals, the separation of high-dimensional and stochastic dynamics from low-dimensional deterministic signals, the estimation of system parameters or invariants (characterization), and, finally, prediction and modeling. In particular, the last two issues are the most complex and critical ones since, due to the current networks' polymorphism and heterogeneity, as long as the high burstiness and long time relativity of the associated traffic, the models developed for traditional telecommunication networks (e.g. Markov and Poisson models, as already explained in the previous sections) are not suitable for traffic description and prediction [1]. Consequently, alternative traffic models need to be developed to characterize this kind of traffic with considerable confidence and over a wide range of conditions.

### 2.3.1 Verifying linearity and searching for chaos

Nonlinear analysis of traffic time series is essentially based on separating stochastic high-dimensional dynamics from low-dimensional deterministic ones, estimating system parameters through its characterization and detection of invariant properties, with the sake of modeling and predicting the underlying system behavior.

However determining with an absolute precision whether a series is stochastic or deterministically chaotic can be very complex. That's worse, the extent to which a non-linear deterministic process maintains its properties when affected by noise is also undetermined. In fact, the noise can affect a system in much different ways, either with an additive unwanted contribution or by introducing measurement errors, even though the equations describing the system remain deterministic in their nature.

However, in presence of a significant quantity of noise and when the dynamics generating the time series are not known, it is necessary to determine if the amount of nonlinear deterministic dependencies requires further analysis or if the series can be directly considered as stochastic. The main motivation behind the above idea is that linear stochastic processes can give origin to very complex signals and that not all the structures that can be observed in a data set can be associated to nonlinear system dynamics [22]. The method of surrogate data [2] may be a useful tool to identify if the irregularity of data is most likely due to nonlinear deterministic structure or rather due to random inputs to the system or fluctuations in its parameters. It consists in generating a group of "surrogate" data sets similar to the original time series, but consistent with the null hypothesis, and computing a discriminating statistic for the original and for each of the surrogate data sets. In general a linear stochastic process can be described by:

$$x_n = a_0 + \sum_{i=1}^{M_1} a_i x_{n-i} + \sum_{j=1}^{M_2} b_j \eta_{n-j} \tag{2.1}$$

where $\eta_n$ are independent Gaussian random numbers with zero mean and unit variance and $a_i$, $b_j$, $M_1$ and $M_3$ are constants. We need to compute some nonlinear observables from the original data and analyze if the resulting values suggest that the data are nonlinear by calculating the same quantity from a number of comparable linear models and verifying that they are totally different. Surrogate data can be created by taking the fast (discrete) Fourier transform (FFT) of the original data and multiplying it by a random phase parameter uniformly distributed in $[0, 2\pi[$,

then computing the inverse FFT, thus obtaining a time series with the prescribed spectrum. This process of phase randomization preserves the Gaussian properties of a distribution. The null hypothesis to be tested is that the data results from a Gaussian linear stochastic process.

### 2.3.2 Reconstructing the nonlinear dynamics

Due to the nonlinear characteristics of the traffic under examination, linear techniques cannot take into consideration all the irregular or non-sistematic behaviors that can be observed in such data, and the ability to identify a wide range of properties of the time series under scrutiny is essential for improving the understanding of the process involved and for providing an accurate approximation of the complex data structures emerging from traffic data. In order to reconstruct the underlying dynamical system, we need to estimate the embedding dimensional parameters of the traffic time series. According to continuous dynamical modeling, each dynamical system can be described by differential equations like [22]:

$$\frac{dx}{dt} = f(x, \lambda) \tag{2.2}$$

where the variable $t$ represents time, the vector $x = (x_1, x_2, \ldots, x_n)$ contains the system's status variables that depend on the time $t$ and on the initial conditions, the vector $\lambda$ contain static system parameters, whereas $f(\cdot) = (f_1(\cdot), f_2(\cdot), \ldots, f_n(\cdot))$ is a nonlinear function of the above state variables and constant parameters. Each status of the system is described by the vector $x$ with its $n$ independent components and corresponds to a specific point in the phase space. Any variation in time of the system status corresponds to a motion in the phase space along a curve known as phase trajectory. It is not always possible to obtain experimental observations of the complete system status and, in most of the dynamical systems, we can base our analysis only on few observable elements associated to the state space coordinates by:

$$s(t) = h \cdot x(t) \tag{2.3}$$

where $h$ is an unknown nonlinear function known as measurement function.

Takens demonstrated in [4] the possibility of performing state space reconstruction by starting from the observations of a sufficient long time series, $s(t)$, describing the dynamical system of interest. Accordingly, the dynamics on the attractor of the underlying original system exhibit, under certain conditions, a one-to-one correspondence with the values assumed by a limited number of variables in the so-called phase space, characterized by a greater dimension, and obtained through *time-delay coordinate embedding* techniques. Such observation opens the door to a new approach to dynamical systems analysis, based on nonlinear time series analysis and time-delay embedding.

As a general rule, a dynamic system can be analyzed in depth only when both its motion equations (see eq. 2.2) and degrees of freedom $n$ are known. The continuously evolving status of such system can be represented by a sequences of "status vectors" $x$, containing the aforementioned status variables $x_i$ and defined in the above phase space [20].

By applying time-delay-coordinate embedding, also if we don't know the equations defining the underlying dynamical system and cannot directly evaluate all the original state space variables, we can be able to determine the above one-to-one correspondence between the original state space and the reconstructed one only by observing the behavior of few variables, so that we can unambiguously identify the original state space from these measurements [22]. This allows us to gather a complete knowledge of the involved system dynamics also if we can base our analysis only on the observation of a single time series obtained as the result of sampling from a single monitoring point.

In detail, the method of *delay-coordinate embedding* makes use of past values to reconstruct a useful version of the internal dynamics. It starts from the Takens theorem [4] stating that we can reliably reconstruct a phase space trajectory, by

using the time series corresponding to the evolution of a single variable by observing it on multiple time scales through the method of time delays.

Thus, starting from the scalar time series $\{x_t\}_{t=1}^{T}$ we can generate a sequence of (embedded) vectors $y_i(\tau) \equiv \left(x_i, x_{i+\tau}, x_{i+2\tau}, \ldots, x_{i+(m-1)\tau}\right)$ constituting a phase trajectory in $\Re^m$ where $m$ is the embedding dimension and $\tau$ is the time delay. Each unknown point within the phase space at time $i$ can be reconstructed by using the delayed vector $y_i(\tau)$ in an $m$-dimensional space, that is the reconstructed phase space. The sequence of embedded vectors recreates the original dynamics only if the values of $m$ and $\tau$ are chosen properly.

Time-delay embedding can be considered as a way to shift from a temporal time series of specific observations to a new, dimensionally enhanced, state space that is "similar", in a topological sense, to that of the dynamical system under analysis. All the dynamics in this new space will be directly associated to the dynamics in the original space by a nonlinear transformation, called the reconstruction map.

The above technique is based on the consideration that according to [4], let $s(t)$ be the measure of some variable of our system, (see eq. 2.3), we can pass from the derivatives of various order $\{s(t), \dot{s}(t), \ddot{s}(t), \ldots\}$ to delay coordinates, $\{s(t), s(t+\tau t), s(t+2\tau t), \ldots\}$ where $\tau t$ is a properly determined time delay. That is, if we consider the following approximation of the derivative of s(t):

$$\frac{ds(t)}{dt} \cong \frac{s(t+\tau t) - s(t)}{\tau t} \tag{2.4}$$

$$\frac{d^2 s(t)}{dt^2} \cong \frac{s(t+2\tau t) - 2s(t+\tau t) + s(t)}{2\tau t^2} \tag{2.5}$$

it is immediately observable that the additional information carried from each new derivative is also embedded within in the delay coordinates series. Clearly, the significant gain that can be achieved by using delay coordinates instead of derivatives stems from the fact that high order derivatives, in presence of a large number of dimensions, tend to considerably amplify the noise in the observations [22].

The Takens' embedding theorem is the basis for the above embedding concepts and in general for the study of nonlinear systems since it provides a formal theo-

retical framework for the state space reconstruction problem. Anyway, it should be considered that the Takens' theorem is really valid only in presence of an unrealistic infinite number of points not affected from noise at all. In this case, the correct determination of the time delay is not relevant, and can only be useful to drive the choice of the embedding dimension.

In realistic cases, a proper choice of the time delay $\tau$ and the determination of an optimal embedding dimension $m$, assume a fundamental importance for the overall quality of the analysis process. In fact, many research efforts on state space reconstruction are focused on the problems of determining the best values of time delay and embedding dimension which become the only critical parameters for the reconstruction of the state space starting from delay coordinates[22].

In particular, for the Takens theorem to hold in real applications' environment, the choice of $m$ must assure that $m > 2d+1$, where $d$ is the original (unknown) system's dimension. A good estimation for $d$ is provided by the Grassberger-Procaccia [5] Correlation Dimension $\nu$, a measure of the dimensionality of the space occupied by a set of random points that is also frequently used to distinguish between random and chaotic behavior. It is associated to the correlation integral $C(\varepsilon)$ (that will be defined in eq. 3.3) by the relation:

$$C(\varepsilon) \sim \varepsilon^{\nu} \tag{2.6}$$

On the other hand, the most common method for determining the minimum embedding dimension $m$ is calculating some invariants of the attractor. We have to progressively increment the embedding dimension used for the above computations until the values of these invariant stops changing. Given that these invariants are geometric properties of the system dynamics, they become independent from $d$ when $d \geq m$, that is, after the geometry is unfolded. The most natural question is, then, how to determine the proper values to be assigned to the time delay $\tau$ and the embedding dimension $m$.

A lot of methods have been proposed to guess a satisfactory assignment for $m$ and $\tau$. As a general rule, a suitable embedding delay $\tau$ has to fulfill two criteria

[25]. First of all, the value of $\tau$ has to be sufficiently large to ensure that the information obtainable from the observation of the variable $x$ at the time $t + \tau$ is still relevant and significantly different from the one we already have got by the observed value of the same variable at time $t$. Only in this case it will be possible to obtain enough information about all the other variables affecting the value of the one under observation, in order to be able to successfully perform a complete phase space reconstruction with a reasonable choice of the embedding dimension $m$. It should be considered that, a too short embedding delay can be compensated with a larger embedding dimension. For the same reason the original embedding theorem has been formulated with respect to $m$, and essentially nothing is stated about $\tau$. Also, the value of $\tau$ should not be chosen to be larger than the average time in which the underlying system loses memory of its initial status. For larger values of $\tau$, the reconstructed phase space would lose its organization by assuming a random one, consisting of almost unrelated points. Such condition assumes a critical importance for chaotic systems that, being intrinsically unpredictable, lose memory about their initial status as the time passes.

Because the mutual information between $x_i$ and $x_{i+t}$ quantifies the amount of information we have about the state $x_{i+t}$ given our knowledge about the state $x_i$, according to [6] the first minimum of the Average Mutual Information Function (AMI) can be used to determine the optimal embedding delay. The *average mutual information* (AMI) minimum is a good estimate for $\tau$, assuming that uncorrelated variables tend to produce uncorrelated values. In fact, the delay should be selected in such a way to minimize the interaction between points of the measured time series. This, in effect, opens up the attractor (assuming that one exists), by projecting it across its largest profile. Suppose that the time series domain is partitioned into equiprobable bins. Let $p_i$ be the probability to find a time series value in the $i$-th bin; let $p_{i,j}(\tau)$ be the joint probability to find a time series value in the $i$-th bin and a time series value in the $j$-th bin after a time $\tau$, i.e. the probability of passing in a time $\tau$ from the $i$-th to the $j$-th bin. The average mutual information function can be formally defined as:

$$S(\tau) = -\sum_{i,j} p_{ij}(\tau) \ln\left(\frac{p_{ij}(\tau)}{p_i p_j}\right) \tag{2.7}$$

A similar argument supports the proposal to use the first zero-crossing of the autocorrelation function [21]. Other authors suggest instead that the first AMI *maximum* should be selected, since it relates to natural periods of the system.

On the other side, good values for $m$ can be found by using methods like *false nearest neighbors* (FNN) [7] relying on the assumption that the phase space of a system behaving in a deterministic way folds and unfolds in a smooth way with no unexpected irregularities emerging in its observable structure. The exploitation of this assumption, lead to the conclusion that, if two points are close in the $m$-dimensional embedding space, their correspondents in the $(m+1)$-dimensional embedding space have to stay sufficiently close. When a point in the phase space has a near neighbor not fulfilling this criterion, it is flagged as having a false nearest neighbor.

This phenomenon, considered from the geometrical point of view, takes place when two points seem to be close in the phase space, but, due to projection effects, they are mapped at random under forward iteration. Such random mappings originates from the fact that the associated attractor is projected onto a hyper-plane with a dimension lower than the one characterizing the actual phase space, so that all the distances between points in such space result to be distorted.

In detail, to find the embedding dimension $m$ we analyze the number of false nearest neighbors and the $m$ parameter is increased in integer steps until the recruitment of nearest neighbors stops changing. At this particular value of $m$ the information of the system has been maximized and exploring further dimensions is not necessary since no new information would be gained. E.g., if in a two-dimensional space a circular area is observed as a projection on its side it appears like a segment, and two points on it can resemble close to each other, even if they are not. Thus, increasing by one the dimension $m$ of the reconstructed may allow the differentiation between the points on a circular orbit, by correctly discriminating the true neighbors from the false ones. Let $y$ be a point of the reconstructed space. Let $y^{(r)}$ be the $r$-th

nearest neighbor of $y$ and compute the (squared) Euclidean distance $D^2$ between them (as usual, $y_k$ denotes the $k$-th component of $y$).

$$D_m^2\left(y, y^{(r)}\right) = \sum_{k=1}^{m-1}\left[y_k - y_k^{(r)}\right]^2 \tag{2.8}$$

Next, increase $m$ to $m+1$ and compute the new distance, i.e. $D_{m+1}^2\left(y, y^{(r)}\right)$. The point $y^{(r)}$ is said a false nearest neighbor if:

$$\frac{D_{m+1}^2\left(y, y^{(r)}\right) - D_m^2\left(y, y^{(r)}\right)}{D_m^2\left(y, y^{(r)}\right)} > D_{TS} \tag{2.9}$$

where $D_{TS}$ is a predefined threshold. Note that the number of false nearest neighbors depends on $D_{TS}$. That is, the percentage of false nearest neighbors (FNN) is determined for each $m$ within a set of values, and an acceptable embedding dimension can be considered to be found in correspondence with the first $m$ such that the percentage of FNN drops to zero. Note that with real-world, noisy data, this percentage never reaches zero so the embedding dimension providing the lowest FNN percentage is usually chosen.

Unfortunately the above method has some subjectivity in defining that a neighbor is false and consequently we prefer to use a similar method [8], which is based on evaluating the mean value of the distance between time-delay vectors, $E1(m)$ together with the quantity $E2(m)$, defined as:

$$E2(m) = \frac{E^*(m+1)}{E^*(m)} \tag{2.10}$$

where

$$E^*(m) = \frac{1}{N - m\tau}\sum_{i=1}^{N-m\tau}\left|x_{i+m\tau} - x_{n(i,m)+m\tau}\right| \tag{2.11}$$

and $n(i,m)$ is an integer such that $y_{n(i,m)}(\tau)$ is the nearest neighbor of $y_i(\tau)$ in the $m$-dimensional reconstructed state space. Here, the determination of the embedding dimension only depends on the time delay $\tau$ and its value is obtained when the values of $E1$ and $E2$ reach saturation.

### 2.3.3 Inspecting traffic stationariety and determinism

Before starting the analysis process, we need to acquire significant insights on the non-stationary variation patterns of our time-series data. The concept of stationarity is very important in traditional linear and nonlinear time series analysis, in particular when coping with time series of traffic variables that are, most times, non-stationary [9]. A time series is defined as non-stationary, [10] when, for some $q$, the joint probability distribution of $x_i, x_{i+1}, \ldots, x_{i+(q-1)}$ is dependent on the time index $i$. Traffic engineering practices regarding traffic volume analysis are tightly related with the notions of non-stationarity. Hence, detecting non-stationarity is crucial as it describes the deviation points in the statistical behavior of the underlying process that can be observed as the time goes by. In many dynamic phenomena, it is of fundamental importance to trace these points [9]. Sudden changes in the statistical characteristics of traffic variables, e.g. quantity of packets or packet size, can lead to understanding the different dynamics associated to the specific behavior of the involved end applications. Even if a precise definition of stationarity exists, there aren't comprehensive criteria of formulas for univocally assessing the stationarity of a time series. Anyway, by observing the basic statistic system properties, i.e. mean, variance, spectral components, correlations, etc. and verifying that they should not change beyond bare fluctuations, can give us almost reliable information about the stationary or non-stationary nature of a system.

Accordingly, the stationarity of traffic time series can be verified by observing its *space-time separation plot (stp)*, [11] reporting the probability that two points in the reconstructed phase space have distance smaller than $\varepsilon$, as percentile isolines. This functional dependence can be shown by displaying the number of neighbor points as a function of both the spatial distance and the time separation. In detail, for each time separation $\Delta t$ an accumulated histogram of spatial distance $\varepsilon$ is graphically reported into a plot where the separation in time is represented on the horizontal axis whereas the separation in space is represented on the vertical axis. A roughly flat profile in all the components, with saturation on the contour curves is a clear symptom of stationarity.

After positively establishing the presence of non-stationarity, the determinism properties of the traffic time series must be inspected to enforce the distinction between deterministic chaos and irregular random behavior, which often resembles chaos. This task can be easily accomplished through a simple but very effective determinism test, originally proposed by Kaplan and Glass [12] starting from the consideration that if a system is really deterministic, it can be described by a set of ordinary differential equations like (2.2) and the vector field $V_k$ (see the following eq. 2.12) is uniquely determined by these equations at every point of the phase space.

More precisely, the Kaplan and Glass test enables the construction of the system's vector field directly from the time series, and subsequently test if the reconstructed vector field assures uniqueness of solutions in the phase space. The determinism test is based on a correct reconstruction of the attractor in the embedding space [24]. The embedding space has to be partitioned into boxes of equal size. A vector is assigned to each box that is occupied by the trajectory. It will be the approximation for the vector field. The vector associated to a specific box can thus be calculated in the following steps. At each step $i$ the trajectory passes through the $k$-th box it generates a unit vector $e_i$ whose direction is determined by joining the first point in the phase space in which the trajectory goes into the box and the last point in which the trajectory exits from the box. This process results in the calculation of the average direction of the trajectory traversing the box during each pass [25]. The approximation for the vector field $V_k$ in the $k$-th box of the phase space is now simply the average vector calculated on all the passes obtained according to the equation

$$V_k = \frac{1}{P_k} \sum_{1=1}^{P_k} e_i \tag{2.12}$$

where $P_k$ is the number of all the passages through the $k$-th box. However, the described method provides no information about how rapidly the trajectory traverses a particular box. Nothing can be said in advance about the absolute lengths of the obtained vectors. The absolute magnitude of the vector field is, however, not

important for the determinism test. What is important is the fact that if the time series originated from a deterministic system, and the coarse grained partitioning is fine enough, so that the obtained vector field should consist exclusively of vectors that have unitary length (remember that each $e_i$ is a unit vector). This is due to the need for uniqueness of all the solutions in the phase space. Thus, since the solutions in the phase space must be unique, all the unit vectors inside each particular box have to point in the same direction. In other words, the trajectories inside each box cannot cross each other, since the uniqueness condition would be violated at each crossing point.

Furthermore, it should be considered that each crossing condition decreases the size of the average vector $V_k$. That is, in the case in which the crossing of two trajectories inside the $k$-th box occurs at right angles, then the size of $V_k$ would be, according to the Pythagoras theorem, $\frac{\sqrt{2}}{2} \approx 0.707 < 1$ [25]. Hence, if the system is deterministic, (i.e. no trajectory crossing a particular box occur), each average resultant vector obtained according to eq. (2.12) will be exactly of length 1. Accordingly, the average length of all resultant vectors $V_k$, which will be the reliable indicators for determinism, will be exactly 1, while for a system with a stochastic behavior it will result in a value substantially smaller than 1.

Immediately following the assessment for determinism, establishing that the involved traffic time series originate from a deterministic non-stationary system, we need to continue our analysis of chaos properties of a time series by calculating the maximal Lyapunov exponent [13]. The Lyapunov exponent or Lyapunov characteristic exponent of a dynamical system is a quantity that characterizes the rate of separation of infinitesimally close trajectories in the phase space. Quantitatively, two trajectories in phase space with initial separation $\delta \mathbf{Z}_0$ diverge (provided that the divergence can be treated within the linearized approximation) if:

$$\delta \mathbf{Z}(t) \approx e^{\lambda t} \delta \mathbf{Z}_0 \tag{2.13}$$

The Maximal Lyapunov exponent determines a notion of predictability for a

dynamical system. It can be defined as:

$$\lambda = \lim_{t\to\infty} \lim_{\delta \mathbf{Z}_0 \to 0} \frac{1}{t} \ln \frac{\delta \mathbf{Z}(t)}{\delta \mathbf{Z}_0}. \tag{2.14}$$

A positive value is usually taken as an indication that the system is chaotic (provided some other conditions are met, e.g., phase space compactness). The basic idea in the calculation of the maximum exponent is to find a pair of spatially nearby points in the attractor and follow their evolution in time, measuring the rate of divergence, until the points can no longer be considered close.

### 2.3.4 Analyzing the power spectrum

The study of the power spectrum has the sake of discovering the main frequencies in a system, since in correspondence of the dominant frequencies and their integer multiples or harmonics we can always observe sharp or broad peaks. As usual such study starts from the fourier transform of the function s(t) given in Eq. 2.3, defined as:

$$\tilde{s}(f) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} s(t) e^{2\pi i f t} dt \tag{2.15}$$

The power spectrum of a function can be represented as the squared modulus of its continuous Fourier transform, $P(f) = |\tilde{s}(f)|^2$ that when transposed in the domain of a discrete finite time series is given by:

$$\tilde{s}_k = \frac{1}{\sqrt{N}} \sum_{j=1}^{N} s_j e^{2\pi i k n/N} \tag{2.16}$$

Here, the frequencies in physical units are $f_k = \frac{k}{N\Delta t}$, where $k = \{-\frac{N}{2}, \ldots, \frac{N}{2}\}$ and $\Delta t$ is the sampling interval. It has been observed in [23] that the power spectra of many different systems diverge at low frequencies. The appearance of a scaling behavior in the power spectrum can flag the existence of a self-organization with many degree of freedom [22].

## 2.4 Detecting Fractal Behaviors

It has been demonstrated [15] that the notion of self-similarity, that is the typical fractal behavior in terms of resemblance or correspondence between scaled parts of an object and the object as a whole, can be used to explain the extreme burstiness of the real network traffic that is still present over a wide range of time scales. In fact, network traffic can be characterized by its bursts of activity that can be idealized as existing at every time scale, from milliseconds to days, and as looking similar independently of the time scale, i.e., the traffic can be idealized as self-similar and consequently long-range dependent. From a statistical point of view, network traffic exhibits also other fractal characteristics in its second-order statistics such as slowly decaying variance and long-range dependence over a wide range of time and frequency scales. This means that the variance of the sample mean decreases much more slowly than the reciprocal of the sample size, or in other terms that the distribution of the traffic process decays more slowly than exponentially (as in a Poisson distribution), and autocorrelation exhibits an hyperbolic ("long range") rather than exponential ("short range") decay. Such properties suggested that fractal and generally non-linear models are the most appropriate mathematical tools to describe certain aspects of network traffic behavior.

### 2.4.1 Assessing Self Similarity

The notion of fractal behavior, in the sense of an object that is self-similar on all the scales is translated into the definition of Self-Similar processes with stationary increments. Let $X = \{x_k : k > 0\}$ a discrete-time domain process, representing the amount of data transmitted in consecutive short time periods, and let $X(m) = \{X_k^{(m)} : k = 1\}$ its aggregate form obtained by averaging the $x_k$ over adjacent, non overlapping blocks of size $m$:

$$X_k^{(m)} = \frac{1}{m} \sum_{i=(k-1)m+1}^{km} x_i \tag{2.17}$$

The process $X$ has the *self-similarity* property, if $X_k$ and $m^{1-H}X^{(m)}$ have identical finite-dimensional distributions for all $m \geq 1$, that is:

$$X = m^{1-H}X^{(m)} \tag{2.18}$$

The parameter $H$, referred to as the *Hurst exponent*, is defined in $[0,1]$ and represents the degree of self-similarity in the observed sample. When the value of the Hurst parameter falls between 0.5 and 1 the sample is said to be self-similar (values of $H$ closer to 1 indicate a high degree of self-similarity). The Hurst exponent is directly related to the "fractal dimension", which gives a measure of the roughness of a surface. The relationship between the fractal dimension, $d$, and the Hurst exponent, $H$, is $d = 2 - H$.

Hurst exponents also quantify the correlation of a fractional Brownian motion [22]. A fractional Brownian motion (*fBm*) corresponds to a random walk with a Hurst exponent different from 0.5 and hence with a memory. $H$ is equal to 0.5 for random walk-based time series, $< 0.5$ for anti-correlated series, and $> 0.5$ for positively correlated series. The decaying of spectral density $s$ of a *fBm* has a relationship with the $H$ exponent:

$$s \propto \frac{1}{\mathrm{f}^{\alpha}} \tag{2.19}$$

where $\alpha = 2H + 1$

However, the estimation of the Hurst exponent from empirical data is not a simple task. The data must be measured at high lags/low frequencies where fewer observations are available. All estimators are vulnerable to trends and periodicity in the data and to other sources of corruption. To obtain reliable estimates, despite possible effects of non-stationariety, a variety of methods [16] are available (e.g. aggregated variance method, R/S plot, periodogram method, and wavelet-based Whittle technique etc.). In this work we tested all the above techniques and obtained the best and most reliable results by using the Whittle estimator [17], fitting a straight line to a frequency spectrum derived using wavelets, that can be

thought as akin to Fourier series but using more complex waveforms instead of simple sine waves. In detail, being $f(\lambda,\theta)$ the parametric form of the spectral density of a Gaussian stationary process described by $X_t$, where $\theta$ is the parameter to be estimated, being $G(\lambda)$ the periodogram of $N$ samples defined as:

$$G(\lambda) = \frac{1}{2\pi N}\left|\sum_{t=1}^{N} X_t e^{jt\lambda}\right|^2 \tag{2.20}$$

The approximated Whittle estimator is the value $\widehat{\theta}$ minimizing the function:

$$\widetilde{Q}(\theta) = \frac{4\pi}{N}\left\{\sum_{k=1}^{N^*}\frac{G(\lambda_k)}{f(\lambda_k,\theta)} + \sum_{k=1}^{N^*}\log f(\lambda_k,\theta)\right\} \tag{2.21}$$

where $N^* = \left\lfloor\frac{N-1}{2}\right\rfloor$ and $\lambda_k = 2\pi N^{-1}k$

This method, while computationally more expensive is not associated to graphical methods where the estimation accuracy depends on how the plot is interpreted and calculated thus it has been preferred, for reliability sakes, to the other ones.

### 2.4.2 Searching for Long-range dependence

Long-range dependence refers to the degree of dependence between samples observed at a specific time on those ones taken at an earlier time. In this case, the involved process demonstrates to have some memory of past events, which are "forgotten" as the time moves forward. The mathematical definition of long-range dependent processes is given in terms of their *autocorrelation function*. The autocorrelation of a stochastic process described by a discrete time series measures the degree of correlation between nearby and far-off events. That is, when a data set exhibits an high autocorrelation, a value $x_i$ at time $t_i$ is highly correlated with a value $x_{i+d}$ at time$t_{i+d}$, where $d$ is some time increment in the future. More formally, let $X = \{x_k : k > 0\}$ a process in the discrete-time domain with mean $\mu = E[x_k]$, variance $\sigma^2 = E[(x_k - \mu)^2]$ and normalized autocorrelation function:

$$r(k) \equiv \frac{E\left[(x_n - \mu)(x_{n+k} - \mu)\right]}{\sigma^2} \tag{2.22}$$

The process $X$ is said to be long-range dependent (LRD) if its autocorrelation function $r(k)$ is non-summable, i.e.

$$\sum_{k=0}^{\infty} r(k) = \infty \tag{2.23}$$

Since the behavior of the tail of $r(k)$ completely determines its summability, the details of how $r(k)$ decays with $k$ is of great interest to study long-range dependence properties of traffic samples. That is, for short-range dependent traffic, which is non-bursty, the autocorrelation function falls off quickly with time, usually exponentially. For long range dependent traffic, it falls off much more slowly, decaying according to a power law distribution, i.e.

$$r(k) = Ck^{-\beta} \tag{2.24}$$

where, $C$ is a constant and $r(k)$ is the autocorrelation function with lag $k$. The Hurst exponent is related to $\beta$ by:

$$H = 1 - \frac{\beta}{2} \tag{2.25}$$

To analyze the traffic's degree of LRD we used the de-trended fluctuation analysis (DFA) scheme [18] [19]. Many other techniques that measure long-range power-law correlations or LRD in a time series are based on the assumption that the involved network traffic is stationary, which is not easy to be valid in the real world. The important advantage of the DFA technique lies mainly in its applicability to non-stationary time series. That is, the scaling behaviors of non-stationary traffic can be better analyzed with this method, while conventional techniques such as R/S analysis cannot be used for non-stationary signals. According to [26], let $X$ be a single-dimensional stochastic process represented by $N$ time samples, we describe the following integrated signal $u(k)$:

$$u(k) = \sum_{i=1}^{k} (x_i - \mu) \tag{2.26}$$

where $\mu$ is the mean of $x_i$. Then, the integrated signal $u(k)$ is divided into boxes of the same length $n \ll N$ and we determine the least square lines fitting the data into each box of size $n$. Next, $u(k)$ is de-trended by subtracting to it the local trend values $u_n(k)$ as follows:

$$F(n) = \sqrt{\frac{1}{N}\sum_{k=1}^{N}(u(k) - u_n(k))^2} \tag{2.27}$$

The above calculation must be repeated on a wide range of scales to properly characterize the relationship occurring between the box size $n$ and the average root-mean-square fluctuation function $F(n)$. Typically, $F(n)$ will increase with the box size $n$. A linear relationship on a log-log plot indicates the presence of power law (fractal) scaling. Under such conditions, the fluctuations can be characterized by a scaling exponent, i.e., the slope of the line relating $\log F(n)$ to $\log n$.

The compliance to power-law is a clear symptom of a scaling relationship characterized by $F(n) = n^{\alpha}$, which indicates that the involved process evolves according to a scaling law whose trend is conditioned by the exponent $\alpha$. The fractal nature of the associated fluctuation is also described by the exponent $\alpha$, which determines the long-range power correlation or degree of long range dependence of the signal.

If the above process looks like white noise, then $\alpha$ is close to 0.5. If it is correlated or persistent, $\alpha > 0.5$; whereas if its is anti-correlated or anti-persistent, $\alpha < 0.5$. That is, growing $\alpha$ values greater than 0.5 can be associated to an increasing degree of long range dependence in the time-series signal; i.e., the $\alpha$ value corresponds to the Hurst parameter when $0.5 < \alpha < 1.0$ [26][18][19].

As with most methods that depend upon line fitting, it is always possible to find a number $\alpha$ with the DFA method, but this does not necessarily imply that the time series is really self-similar.

# References


[1] V. Paxson and S. Floyd, "Wide-Area Traffic: The Failure of Poisson Modeling", *IEEE/ACM Trans. on Networking* 3(3), pp.226-244, 1995.

[2] J. Theiler, S. Eubank, A. Longtin, B Galdrikian and J. D. Farmer, "Testing for nonlinearity in time series: the method of surrogate data", *Physica D* 58, pp. 77-94, 1992.

[3] J. P. Eckmann, D. Ruelle, "Ergodic theory of chaos and strange attractors", *Rev Mod Phys* 57, pp. 617-656, 1985.

[4] F. Takens, "Detecting strange attractors in fluid turbulence", in D. Rand, L.S. Young, eds., *Dynamical Systems and Turbulence*, Springer, pp. 366-381, 1981.

[5] N. Marwan, M. C. Romano, M. Thiel, J. Kurths, "Recurrence plots for the analysis of complex systems", *Phys. Reports* 438, pp. 237-329, 2007.

[6] A.M. Fraser, H.L. Swinney, "Independent coordinates for strange attractors from mutual information", *Physics Review A* 33-2 , pp. 1134-1140, 1986.

[7] M. Kennel, R. Brown, H. Abarbanel, "Determing embedding dimension for phase space reconstruction using a geometrical construction", *Phys Rev A* 45, pp. 3403-3411, 1992.

[8] L. Cao, "Practical method for determining the minimum embedding dimension of a scalar time series", *Physica D* 110, pp. 43-50, 1997.

[9] R.H. Shumway, D.S. Stoffer, "Time series analysis and its applications", *Springer Texts in Statistics*, Springer-Verlag, 2000.

[10] M.B. Priestley, *"Nonlinear and Non-Stationary Time Series"*. Academic Press, New York, 1988.

[11] A. Provenzale, L. A. Smith, R. Vio, G. Murante, "Distinguishing between low-dimensional dynamics and randomness in measured time series", *Physica D 58*, pp. 31-49, 1992.

[12] D. T. Kaplan, L. Glass, "Direct test for determinism in a time series", *Physical Review Letters* 68, pp. 427–430, 1992.

[13] H. Kantz, "A robust method to estimate the maximal Lyapunov exponent of a time series", *Physical Letters A* 185, pp. 77–87, 1994.

[14] E. Aurell, G. Boffetta, A. Crisanti, G. Paladin, and A. Vulpiani, "Predictability in the large: an extension of the concept of Lyapunov exponent", *J. Phys. A* 30, 1, 1997.

[15] W.E. Leland, M.S. Taqqu, W. Willinger, and D.V. Wilson, "On the self-similar nature of Ethernet traffic" (extended version), *IEEE/ACM, Trans. Networking* 2, pp. 1-15, 1994.

[16] M.S. Taqqu, V. Teverovsky, W. Willinger, *"Fractals 3"*, 785, 1995.

[17] S. Molnar and T. D. Dang, *"Pitfalls in long range dependence testing and estimation"*, Proceedings of GLOBECOM, 2000.

[18] K. Ivanova and M. Ausloos, "Application of the detrended fluctuation analysis (DFA) method for describing cloud breaking", *Physica A* 274, pp. 349-354, 1999.

[19] K. Hu, P. Gopikrishnan, P. Cizeau, M. Meyer, C.K. Peng and H.E. Stanley, "Effect of trends on detrended fluctuation analysis", *Physical Review E* 64, pp. 1114-1119, 2001.

[20] J. P. Eckmann, D. Ruelle, "Ergodic theory of chaos and strange attractors", *Rev Mod Phys*, pp. 617-656, 1985.

[21] N. Marwan, M. C. Romano, M. Thiel, J. Kurths, "Recurrence plots for the analysis of complex systems", *Phys Reports*, 438, pp. 237-329, 2007.

[22] F. Strozzi, E.Gutirrez, C. Noc, T. Rossi, M.Serati and J.M. Zaldvar, "Application of non-linear time series analysis techniques to the Nordic spot electricity market data", *LIUC Paper 200*, 2007.

[23] I. Procaccia and H. Schuster, "Functional Renormalization Group Theory of Universal 1/f-noise in Dynamical Systems", *Phys. Rev. A* 28 p. 1210, 1983.

[24] S. Kodba, M. Perc and M. Marhl, "Detecting chaos from a time series", *Eur. J. Phys.* 26, pp. 205215, 2005.

[25] M. Perc, "The dynamics of human gait, *Eur. J. Phys.*, 26(3), pp. 525-534, 2005.

[26] M. Masugi, "Recurrence Plot-Based Approach to the Analysis of IP-Network Traffic in Terms of Assessing Nonstationary Transitions Over Time", *IEEE Transactions on Circuits and Systems* 53(10), pp. 2318-2326, 2006

# Chapter 3

# A nonlinear approach to anomaly detection

## 3.1 Introduction

At a first glance, anomaly detection seems straightforward. One essentially has to pick a statistical definition of an anomaly, process the measurement data by using a statistical-analysis technique, and classify the outliers as anomalies. Unfortunately, things are much more complicated. There are many ways to represent traffic and pinpoint anomalies, each with its own set of design choices, assumptions, limitations, and tunable parameters that significantly affect the results. First of all, the whole system should be designed to be protocol and service independent, to work correctly on the broadest possible range of applications. The detection mechanism should also be proactive, to dynamically and rapidly accommodate for changes in network activity and attacks of unknown type. Furthermore, a real-time anomaly detection mechanism needs to be efficient enough to scan the huge amount of traffic characterizing very high bandwidth networks, with a satisfactory degree of accuracy in detecting truly anomalous events (minimum false positive rate). Our approach, centered on the inspection of volume-based traffic features, has been conceived to fulfill all the above architectural requirements.

## 3.2 Scope, features and limitations

Fast and reliable/accurate detection are the most important requirements of a real-time anomaly detection system. Accordingly, a fundamental question to be answered when developing a new anomaly detection strategy is: how much data is necessary to classify a traffic flow as anomalous with a sufficiently high degree of confidence? In many known Internet-based attacks, several and often independent, malicious flows concur to the involved hostile activity. For example, in the case of port and/or host-based scanning activities we can observe multiple independent probing flows towards a (often very large) set of destinations, and each single compromised host infected by a worm or virus can generate a huge amount of these flows in its blind vulnerability exploitation activity. Thus, getting more traffic data (and hence as many traffic flows as possible) can greatly improve the detection accuracy when coping with the above hostile phenomena, since the combination of a sufficiently large number of individual evidences can improve our confidence in deciding about the malicious or legitimate nature of the traffic generated by a single host or a set of apparently unrelated ones.

The presented approach has been explicitly conceived to mainly cope with noisy anomalous phenomena, explicitly involving measurable variations in the statistical properties of the traffic time series. Hence it cannot detect hostile behaviors affecting only packet payload (e. g. buffer overflow attempts) or designed to be undistinguishable from regular user-originated network activities such as click frauds or other profit-oriented and/or malware-driven actions. On the other hand, our efforts focused on detecting less evident variations in intrinsic traffic properties that emerge only by observations made on multiple time scales (e.g. variations in long-range correlation, self-similarity, "hidden" periodic structures and clustering properties that are not clearly evident in the original traffic time series). Our anomaly detection strategy relies on detecting the "manifestations" of each attack or suspicious network activity rather than the explicit mechanism behind it, which is clearly unknown for zero-day attacks. For these reasons such strategy can reveal itself to be effective in

particular against zero-day attacks exhibiting a "noisy" behavior that can be evidenced at least at one time scale. Therefore, our proposed scheme should ideally be complemented by a companion system/method targeting, through proper rule and/or signature matching, the analysis and discovery of suspicious payloads and protocol transactions.

## 3.3 Analyzing the traffic properties on multiple timescales

Paxson and Floyd [2] showed that many types of network-related properties, such as the packet rate associated to a particular type of traffic, its dimension or inter-arrival times are characterized by self-similar or fractal behavior with a fractal dimension that changes slowly over time [3] [4] and become apparent at varying timescales. Although it is not always well identified, the notion of scale of analysis is implicit in every anomaly detection method proposed until now. In order to achieve our objective of obtaining an accurate model of normality in a network, a deep comprehension of the phenomena involved in its dynamics is required. The importance of the scale of analysis in anomaly detection methods lies in the fact that certain anomalies/attacks are only observable at certain scales. The normal network traffic, due to its self-similarity is usually characterized by a certain degree of regularity that can only be observed on multiple time scales. The anomalous traffic caused by an attack still presents some distribution regularities, but these may significantly differ from the typical ones, associated to normal traffic.

When the normal and anomalous network traffic are overlapped, the regularity in their individual distributions is usually lost and the associated data become unsystematic. Such a complex dynamic system may show large fluctuations of intensive quantities on long time scales, which cause the system to exhibit the characteristic of non-stationarity and nonlinearity. That is, the mean, the standard deviation, and all higher moments are variable under time translation from a scale to another. Traffic

engineering practices regarding traffic volume analysis are tightly related with the notions of non-stationariety. Hence, detecting non-stationariety is important since it describes the shifting points in the temporal statistical behavior of the underlying process. In many dynamic phenomena, it is of fundamental importance to trace these points [7] and several complex non-equilibrium systems can be described by a superposition of different dynamics, each associated with its own time scale. In our specific case, sudden changes in the statistical characteristics of traffic variables, for example volume or packet size, can lead to understanding the different dynamics associated to the specific behavior of the involved traffic sources. This behavior can only be represented by a non-stationary model where no sample independently on how short or long it can be, can be used for prediction of events associated to any other sample.

Anomalous events are not bound to a specific time scale. Instead, they tend to occur in independent bursts separated by sufficiently long gaps whatever the time scale involved, from milliseconds to days or weeks. Such behavior can be directly associated to sudden state changes occurring in the underlying system, due to malicious agents being started, sections of network becoming unreachable due to denial of service on communication lines or network devices, and so on. We must finally observe the existence of several, almost hidden, recurrence patterns within a single period of traffic observation. These patterns are not easily identifiable harmonics of some specific period, so they cannot be simply filtered out through time-differentiation of the time series. Instead, they may be associated to short-term variations that, combined with noise, can result in apparent anomalies that should be discarded as false positives. Clearly, the most significant hidden pattern that can be observed is the daily one, driven by the 24 hours rhythm of activity. The weekly pattern is typically much tolerant to noise phenomena, since the variations that can be observed over many days of activity are significantly limited. In fact, the daily pattern is affected by higher levels of uncertainty, because not all days are equivalent and traffic behavior can be very different in different days (e.g. a very low activity can be observed during weekends or vacations and this phenomenon artificially increases the entropy

in the associated time series [8]. Any method for detecting anomalies which is based on analysis of observations at a single specific time scale must explicitly take into account and filter out the normal cyclic fluctuations, i.e. it must accurately phase out only those deviations that cannot be justified as periodic oscillations. To do so, one may either characterize and discard unwanted anomaly notifications or preprocess and "flatten" the data. In addition, single time scale techniques carry the effort of choosing, among all the possible time scales, the ones which are able to provide the most discriminating information. Besides, the detected events can only include, by definition, those observable within the time scales considered for the analysis. In contrast, nonlinear techniques that operate simultaneously across multiple time scales do not need to use such filtering and can provide a wider range of anomaly signals. The only thing that needs to be ensured is that the baseline is built over a period that is long enough to encompass the normal behavior at all relevant time scales.

## 3.4 Non-linear analysis for Anomaly Detection

The idea of viewing the time series composed by overlapping normal and anomalous traffic flows as a non-stationary process, together with the analysis of the variation of their hidden non-linear properties on several time scales, provides us with a novel way of differentiating between a "normal" network activity and something other than "normal". Nonlinear analysis of traffic time series has, among its goals, the estimation of the most discriminating parameters through modeling and characterization, achieved by separating the high-dimensional and stochastic dynamics from the low-dimensional deterministic components. For these reasons, it can be considered a very powerful tool for the observation of anomalous patterns in traffic volume data, since linear methods cannot account for all the irregular phenomena observed in such data, and the ability to identify a wide range of properties of the time series under scrutiny is essential for improving the understanding of the process involved and for providing an accurate approximation of complex traffic data structures. Our

detection strategy starts from the observation of some specific nonlinear characteristics, such as recurrence phenomena and hidden non-stationary transition patterns (order-chaos or chaos-chaos) in the time series in which we want to explicitly distinguish anomalous events. These characteristics provide us very interesting insights into periodic structures and aggregation properties that are not immediately evident in the original time series and can be used to characterize the involved traffic profile in a more effective and discriminating way. Such strategy demonstrated to be particularly effective in identifying all the anomalies due to attacks that generate an unusual amount of packets or unusual fluctuations in packet rate or size respect to the baseline traffic layout, resulting in satisfactory detection times with a limited false positive rate. Once the "qualitative" network baseline schemes and the binary (i.e. anomalous vs not-anomalous) classification model have been built offline, all the following activities consist in a "quantitative" recurrence collection/assessment that could realistically be performed on-line, by using well-known Recurrence Quantification Analysis techniques, that, for their limited computational burden, can behave satisfactorily even on resource-constrained network probing devices.

### 3.4.1 Recurrence Quantification Analysis

The RQA concept, introduced by Zbilut and Webber [9], can be viewed as an efficient and deterministic way to easily identify non-stationarity and recurrence features in the network traffic. The great strength of such analysis is that it reveals time correlations between traffic data that are not based on assumptions of linearity or non-linearity and cannot be determined through the direct analysis of one-dimensional series of traffic volumes. Moreover, traffic patterns, implying the manner in which traffic states propagate through time, can be revealed by the study of the evolution of some recurrence statistics in time. The standard first step in the analysis is the choice of the embedding parameters: the time delay $\tau$ and the embedding dimension $m$. Then, the core of the RQA is the computation of several statistics that provide the identification and the quantification of transient recurrent patterns

characterizing the behavior of the time series under investigation. The resulting RQA feature extraction will be accomplished by continuously analyzing the traffic time series and computing the above recurrence-related statistics according to a sliding window scheme, structured upon successive fixed-length time intervals (in the following referred to as "epochs").

### 3.4.2 Exploring recurrence phenomena

Recurrent behavior is a basic property of systems characterized by an oscillating behavior. In typical oscillators, any pair of time-distinct states in the phase space can be arbitrarily close, whereas for chaotic systems this distance is always finite. Formally stated, in the time series $\{x_i\}_{i=1}^{T}$ the recurrence of a state $x$ from the time $i$ in a different time $j$ is given by the following equation [10]:

$$r_{i,j} = \Theta\left(\varepsilon - \|x_i - x_j\|\right), \qquad i, j = 1 \ldots N \tag{3.1}$$

where, $r_{i,j}$ is an element of the recurrence matrix $R$, $N$ is the number of states $x_i$ in the time window of study, $\varepsilon$ is a threshold determining the number of recurring points, $\|\cdot\|$ is a norm, and $\Theta(x)$ is the Heaviside step function, defined as:

$$\Theta(x) = \begin{cases} 1 & \text{for x} \geq 0 \\ 0 & \text{otherwise} \end{cases} \tag{3.2}$$

According to its name, the norm function gives a geometrical definition of the size and shape of the neighborhood area that surrounds each reference point. The area in which recurrence can be observed will be the largest for the *max norm*, the smallest for the *min norm*, and assume intermediate sizes for the *Euclidean norm* [1][11]. $\Theta(x)$ can be viewed an unbiased estimator of the correlation integral, defined as the mean probability that the states at two different times are close:

$$C(\varepsilon) = \lim_{T\to\infty} \frac{1}{T^2} \sum_{\substack{i,j=1 \\ i\neq j}}^{T} \Theta\left(\varepsilon - \|x_i - x_j\|\right) \tag{3.3}$$

Hence, by considering the close relationship between the correlation integral and the fractal dimension of the involved time series [12], it is a direct consequence to assume that this integral is also a very good indicator that can be used for characterizing the time-series dynamics. In other words, $r_{i,j}$ assumes a value of 1 if the interaction between the observed quantities in the instant $i$ is almost the same as in the instant $j$ (i.e., the interaction is recurring), and a value of 0 otherwise. Each recurrent point indicates an isolated recurrence of the phase relationship between the time series. If the time series is deterministic, the orbit in the phase space will revisit some points sometime in the future, forming a picture of the system's *attractor*.

The graphical representations of the elements at or below the aforementioned threshold $\varepsilon$ in the recurrence matrix $R$ is called *recurrence plot* (RP). Essentially, a Recurrence Plot is a two-dimensional graphical representation of the matrix $D = d_{i,j}$, reporting the mutual distances between embedded vectors, where the pixel located at coordinates $(i, j)$ is shaded according to the distance between the $i$-th and $j$-th vectors. More precisely, if the distance $d_{i,j}$ is lower than the fixed cutoff value $\varepsilon$ (the two points are sufficiently close to each other) a dot is plotted in $(i, j)$. Since each coordinate $i$ represents a point in time, the Recurrence Plot gives us information about the correlation in time between the points in the phase space. In fact, each horizontal coordinate $i$ in the Recurrence Plot is associated to the system state at the time $i$ and each vertical coordinate $j$ is associated to the state at the time $j$. Thus, a recurrent point in $(i, j)$ means that the interaction between the observed quantities in the instant $i$ is almost the same as in the instant $j$, i.e., the interaction is recurring. So, in presence of a recurrent point $(i, j)$, the system state at the time $j$ will be part of the neighborhood area of the system state observed at the time $i$, whose size is determined by a maximum distance $\varepsilon$; this means that the state of the

system at time $i$ has some "similarity" with the state of the system at the time $j$, in other words we can say that the system will keep its states on nearby "orbits".

The most important features of recurrence plots are their large and small-scale structures, known as typology and texture, respectively. These features of the Recurrence Plot are strictly related to properties of the system. A homogenous typology suggests the presence of a stationary process, whereas a non-homogenous typology is a sign of non-stationarity. Information regarding the deterministic (versus stochastic) origin of the data can be gained by the observation of texture. Lack of texture, i.e. the tendency of recurrence points to stay isolated, often indicates a stochastic origin of the examined time series, while diagonal lines hint at deterministic oscillations. Those oscillations can be classified into simple, complex or chaotic oscillations, depending on the pattern shape, size and complexity.

### 3.4.3 Quantitative recurrence evaluation

Patterns of recurrence in traffic time series necessarily have mathematical underpinnings that readily become apparent by studying the evolution of some properly chosen variable parameters.

The first variable, *%REC*, corresponding to the correlation sum, measures the percentage of recurrent points in the phase space. Embedded processes that are periodic have an higher percentage of recurrence. %REC is given by Marwan and Kurths [1] as:

$$\%REC = \frac{1}{N^2} \sum_{i,j=1}^{N} r_{i,j} \tag{3.4}$$

where $r_{i,j}$ is the recurrence estimated by the Eq. (3.1). It closely corresponds to the definition of the correlation integral with the only difference that the points of the main diagonal are not included. This variable can range from a value of 0 (absence of recurrent points) to a maximum of 100 (all the points are recurrent).

The second variable, *%DET (determinism)*, measures the proportion (in percentage) of recurrent points forming diagonal line structures in the recurrence matrix.

It allows distinguishing between sparse recurrent points and those that are instead organized in diagonal patterns, representing strings of vectors (deterministically) repeating themselves. Periodic signals will result in long diagonal lines, chaotic signals will exhibit very short diagonal lines, whereas stochastic signals (e.g. random numbers) will not present diagonal lines. High values of %DET show that traffic exhibit a deterministic structure. Deterministic structures may have a very high or very low degree of complexity.

Analogously, the *%LAM* (*Laminarity*) parameter measures the percentage of recurrent points comprising vertical, rather than diagonal, line structures. This measure evidences chaotic transitions, and is related with the amount of laminar phases in the system, representing intermittency phenomena.

The *ENT* variable is a measure of signal complexity, calibrated in units of bits or integer bins in the histogram of the frequency distribution. The probabilities $P_{bin}$, associated to the individual histogram bins are determined for each bin assuming a nonzero value and then summed according to the well-known Shannon's formula (see eq. 3.5). More precisely, the entropy gives a measure of how much information is needed in order to recover the system. A low entropy value indicates that little information is needed to identify the system, in contrast, high entropy indicates that much information are required [11]. Shannon entropy is computed according to the formula:

$$ENT = -\sum_{bin=1}^{N} P_{bin} \log_2 (P_{bin}) \tag{3.5}$$

where $N$ is the number of bins. As the logarithms are in base 2, the entropy can be interpreted as number of bits. Entropy reflects "disorder" and is connected to decreased predictability – simply stated, a low entropy is typical of periodic behavior, while high entropy indicates chaotic behavior. The more complex is the structure of the recurrence plot, the larger will be the value of the entropy observed.

The *TREND* variable is the slope of the least square regression of the local recurrence computed as a function of the orthogonal displacement from the main

diagonal in the recurrence matrix [11]. It quantifies the degree of system stationarity. A "flat" TREND diagram indicates stationarity, whereas a drift in the signal will result in an overall increase or reduction of distances as we move away from the main diagonal. *TREND* is calculated as follows:

1. at first compute the percentage of recurrent points in diagonals parallel to the central line,

2. fit by least squares the relationship:

$$\delta_j = \alpha + \beta\eta_j + u_j \tag{3.6}$$

where $\delta_j$ is the percentage of recurrent points, and $\eta_j$ is the distance away from the central diagonal. The trend is the value of $\beta$. If there is no drift in a dynamical system, there is no fading on the recurrence plot away from the central diagonal, leading to low values (near zero) of $\beta$; however, large values (positive or negative) of $\beta$ is an evidence of a system exhibiting drift.

Finally it should be noted that, in contrast with standard statistical measures such as average and standard deviation that are sequence-independent, recurrence quantification measurements are harshly dependent on the sequential structure of the involved time series. Any random shuffling operation performed on the original sequence is able to destroy the small-scale structure of line segments (both diagonal and vertical) in the recurrence matrix, altering the values of the recurrence variables, but cannot change the mean and standard deviation. Thus the sensitivity of these recurrence parameters to transitions and random variations due to the overlapping of anomalous traffic patterns to normal ones is much greater than the one presented by traditional statistics [11].

## 3.5 Wavelet Analysis for Anomaly Detection

Also Wavelet-based methods exploit very well the traffic properties characterized by inherent self-similarity and hence recurring on multiple time scales since they are

able to simultaneously analyze signals at various levels of decomposition.

Multi-scale wavelet analysis is a promising methodology for this purpose since the associated wavelet decomposition/reconstruction process exhibits very versatile features, such as time-frequency localization properties together with the ability to perform multi-resolution analysis at different scales that can be particularly effective in extracting local "feature" information from non-stationary time-series.

By decomposing a time series on different scales, we may expect to obtain a better understanding of the data generating process as well as of the most characterizing and discriminating traffic dynamics hidden behind long time series.

### 3.5.1 Basic Concepts

Wavelet analysis is a relatively recent mathematical framework, taking its origin in both traditional and Short-Time Fourier analysis, which has captured a wide range of interests in both theoretical and applied research over the last years. Fourier analysis is based on breaking down a signal of interest into several component sinusoids of different frequencies and hence it transposes a signal from the time domain into the frequency domain representation. Unfortunately, in doing this, all the time domain information is lost, and it becomes impossible to associate each particular event with its time of occurrence. To cope with the above problem Short Time Fourier Analysis (STFT) introduces windowing methods to map a signal into a two-dimensional function of time and frequency. This allows keeping time and frequency domain information together, but with the accuracy limited by the window size. Wavelet analysis can be viewed as a further improvement allowing variable window sizes at different layers on a multi-layer stratified model. Such model provides a time and frequency (called scale in wavelet terminology) decomposition technique describing the input signal as a hierarchy of component signals, each one maintaining the time as its independent variable. The lower layers contain very sparse filtered information that can be considered as sophisticated aggregations of the original observations. Such part of the representation can be referred as the low-frequency

one. In contrast, the topmost layers in the hierarchy capture fine-grained details of the data, such as spontaneous variations. These are referred to as the high-frequency layers. More specifically, at lower layers, the use of a smaller window size gives us higher-frequency information. At higher layers, a larger window size can be used to obtain more detailed information on lower-frequency components [13].

The concept of wavelets is associated to a set of functions, forming an orthonormal basis, obtained by dilation and translation of a scaling function $\varphi$, or father wavelet, and a mother (or original) wavelet $\psi$. Thus Wavelet analysis can be viewed as a mechanism that allows breaking up a specific signal into shifted (shifting means delaying or hastening it) and scaled (scaling means stretching or compressing it) versions of the mother wavelet.

Any generic signal described by a continuous function $f(x)$ may be considered as the linear combination (wavelet series expansion) of properly chosen scaling and wavelet functions which are associated with low pass and high pass filters, respectively.

$$f(x) = \sum_{k} c_{j_0}(k)\varphi_{j_0,k}(x) + \sum_{j=j_0}^{\infty}\sum_{k} d_j(k)\psi_{j,k}(x) \tag{3.7}$$

where the two functions $\varphi(x)$ and $\psi(x)$ are respectively shifted and dilated in both time and frequency domain, the value $k$ is associated to the function relative position and the value $j$ defining its scale.

The choice of the above function plays an important role in ensuring the quality of time and frequency localization in the reconstruction process. If we choose scales and positions based on powers of two (dyadic scales and positions) then the analysis will be much more efficient and accurate. Hence, the wavelet function can be defined as:

$$\psi_{j,k}(x) = 2^{j/2}\psi(2^j x - k) \tag{3.8}$$

For each scaling value $j > j_0$ a better fine-grained resolution function is added to introduce more details to transform the input signal into a set of approximation (or

scaling) and detail (or wavelet) coefficients $c$ and $d$ by which it can be reconstructed.

Since the wavelet and scaling functions are chosen to form an orthonormal basis, the above coefficients can be determined as follows:

$$c_{j_0}(k) = \langle f(x), \varphi_{j_0,k}(x) \rangle = \int f(x)\varphi_{j_0,k}(x)dx \tag{3.9}$$

$$d_j(k) = \langle f(x), \psi_{j,k}(x) \rangle = \int f(x)\psi_{j,k}(x)dx \tag{3.10}$$

Many different wavelet families can be used (Haar, Daubechies, coiflets, symmlets), providing compact support with various degrees of smoothness and several of vanishing moments.

### 3.5.2 Discrete wavelet transform

The same consideration hold when dealing with a finite sequence of samples structured into a time series. Let $X = (x_0, \cdots, x_N)$ be an $N$-samples vector of observations from a stochastic process. In this domain the same decomposition/wavelet series expansion technique can be applicable by using the Discrete Wavelet Transform (DWT), which is better suited to decomposing the input signal over a limited set of scales, each characterized by a minor number of coefficients, with the goal of to making the original one reliably reconstructable from them. This is typically done in a way that avoids redundancy. In the discrete domain the coefficients and the expansion formula are defined as follows:

$$W_\varphi(j_0, k) = \frac{1}{\sqrt{M}} \sum_x f(x)\varphi_{j_0,k}(x) \tag{3.11}$$

$$W_\psi(j, k) = \frac{1}{\sqrt{M}} \sum_x f(x)\psi_{j,k}(x) \tag{3.12}$$

$$\begin{aligned} f(x) = {} & \tfrac{1}{\sqrt{M}} \textstyle\sum_k W_\varphi(j_0, k)\varphi_{j_0,k}(x) + \\ & \tfrac{1}{\sqrt{M}} \textstyle\sum_{j=j_0}^{\infty} \sum_k W_\psi(j, k)\psi_{j,k}(x) \end{aligned} \tag{3.13}$$

where the independent variable $x$ can only assume integer values and the term $\sqrt{M}$ is a normalization factor.

### 3.5.3 Fast discrete wavelet transform

The Fast DWT is an efficient implementation of the DWT exploiting the relations between the coefficients on different scales. It is based on an orthogonal transformation of the data operating through the use of recursive filters according to a pyramidal algorithm proposed by Mallat [15]. That is:

$$\begin{aligned} & W_\psi(j,k) = \tfrac{1}{\sqrt{M}} \sum_x f(x) 2^{j/2} \\ & \left[ \sum_m h_\psi(m-2k)\sqrt{2}\varphi(2^{j+1}x-m) \right] \\ & = \sum_m h_\varphi(m-2k) \left[ \tfrac{1}{\sqrt{M}} \sum_x f(x) 2^{(j+1)/2} \varphi(2^{j+1}x-m) \right] \end{aligned} \tag{3.14}$$

where the term between square brackets is the decomposition of the function $f(x)$ at the scale $j_0 = j+1$, and the relations between adjacent scales can be evidenced by:

$$W_\psi(j,k) = \sum_m h_\psi(m-2k) W_\varphi(j+1,m) \tag{3.15}$$

$$W_\varphi(j,k) = \sum_m h_\varphi(m-2k) W_\varphi(j+1,m) \tag{3.16}$$

In most of the implementations, the decomposition of the signal into different frequency bands is simply obtained by successive high pass and low pass filtering of the time domain signal. The resolution of the signal, which is a measure of the amount of detail information in the signal, is changed by the filtering operations, and the scale is changed by performing up-sampling and down-sampling operations.

In detail, the input signals are processed according to a multi-stage approach by using a low pass and an high pass filter at each stage. Clearly, these filtering operations introduce redundancy by duplicating (up-sampling) the input signal

(specifically the number of available samples). In order to remove these redundancies and reduce the size of data the filtered signals have to be down-sampled by half, at each stage.

Down-sampling a signal implies reducing the sampling rate, or removing some of the samples of the signal. This process, obviously, introduces a change in the scale. However, the down-sampling operation after filtering does not affect the overall resolution, since removing an half of the spectral components from the whole signal makes half the number of samples redundant in any case, so that half of the samples can be discarded without any loss of information. After filtering out all the low level details, the remaining coefficients constitute an high level summary of the signal features and hence can be used to model a specific profile characterizing the expected behavior of an input traffic flow during all the sampling period.

### 3.5.4 Discrete wavelet packet transform

The wavelet packet transform (DWPT) is a generalization of the Fast Discrete Wavelet Transform, realized by extending the aforementioned fast pyramidal algorithm, that allows a more sophisticated and flexible time series analysis. In simple words, Wavelet packet decomposition (WPD) is a wavelet transform where the signal is passed through more filters than the DWT, since both approximations as well as detailed components are decomposed. In detail, also the Wavelet packet transform operates by splitting, through the use of two high-pass and low-pass filters, the frequency content of a signal into two distinct components: respectively a low-frequency and a high-frequency ones.

However, in traditional wavelet decomposition the high-frequency component is left unaffected and the decomposition process continues to split only the low-frequency one. Conversely, in wavelet packet decomposition, also the high-frequency component is split in turn into a low-frequency part and a high-frequency one, so that the resulting process hierarchically divides the frequency space into various parts, resulting into a tree-shaped structure that allows easier localization of frequency

components.

The root of the tree is the original time series. The next level results from a single step of the wavelet transform and all the subsequent levels are constructed by recursively applying the discrete wavelet transform to the results from the previous step passed through a low and high pass filter. A typical binary tree schema resulting from a DWPT analysis is shown in Figure 3.1 where $h(n)$ and $g(n)$ are the two impulsive responses of the high-pass and low-pass filters, corresponding respectively to the wavelet and scaling functions.

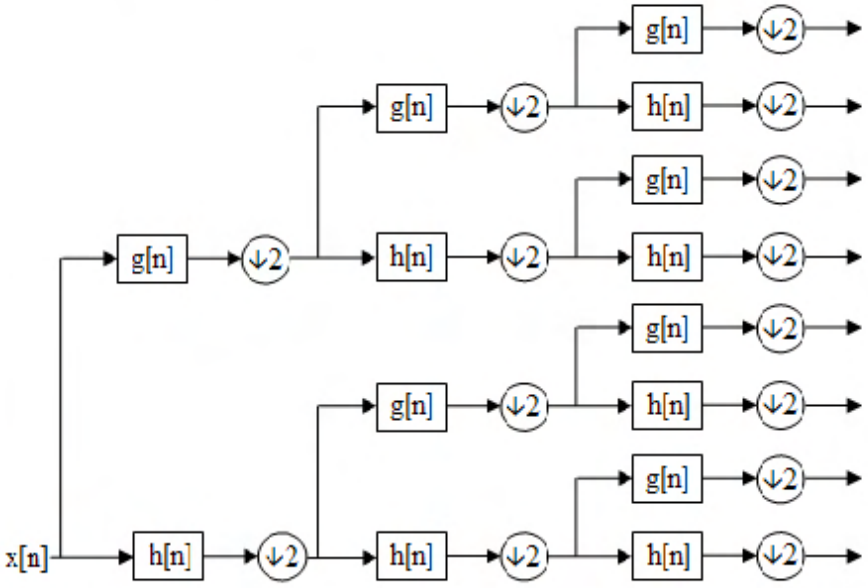



**Figure 3.1**: The DWPT decomposition tree structure.

Consequently, Wavelet packets can be viewed as particular linear combinations of wavelets which retain many of the localization properties of their parent wavelets. The binary tree structure resulting from DWPT yield a library of orthonormal bases from which the most appropriate ones can be selected to represent the input signal. The selection process is based on a search algorithm which itself is usually based on a minimization process of an information cost function, measuring the distribution of signal energy. More precisely, cost function aims at selecting the basis that expresses most of the energy in the signal in as few transform coefficients as possible.

### 3.5.5 Traffic features extraction through Wavelet Analysis

To model the behavior of a traffic time series with wavelet multi-resolution analysis the different signals resulting from discrete wavelet decomposition are examined, with the goal of inferring from them information characterizing the specific properties of the involved traffic, and hence its profile. The input time series signal is observed at different frequencies with different resolutions. The basic idea beyond this multi-resolution analysis technique is starting from a father wavelet and deriving an orthonormal mother wavelet (e.g. the Haar one) and a sequence of wavelet subspaces suitable to approximate the input function with a progressively increasing resolution.

This can be achieved by using a good time resolution and a poor frequency resolution at high frequencies with a good frequency resolution and a poor time resolution at low frequencies. The above choices are due to the fact that the typical network traffic-related signals exhibit high frequency components for short time slice and low frequency components for long ones.

This kind of feature extraction/filtering process, performed at different levels, allows for observation of traffic on many different perspectives, by removing some components of the original signal at each level, and transforming it into a set of wavelet approximation coefficients representing an approximate summary of the original signal, since all the more specific details have been removed during filtering.

In detail, the signals corresponding to traffic flows under scrutiny, are split into different components at multiple ranges of frequencies. Low frequency components are associated to patterns observable over a long enough period, such as many hours or some days; medium frequency components characterize daily variations in the flow data and high frequency ones are directly related to short term variations [17]. More precisely we can consider that the low and medium frequency bands correspond to the daily and weekly cycles, typically representing the normal traffic patterns, whereas the high-frequency ones may refer to extemporaneous events due to anomalous events or unexpected traffic peeks.

In particular, the local maxima emerging from the different components can be

very useful for detecting the presence of regularities and conversely, to spot the location of irregular structures and singularity points in the input signal [16].

Since we are interested in spotting anomalous events characterized by a change in traffic regularity that become evident on some time scale, the most interesting feature information that can be used to distinguish this kind of traffic from the other ones can be related to the measure of disorder in the individual wavelet component signals. Such information can be easily obtained from an entropy analysis on each wavelet component. The presence of peaks/local maxima and irregularities in the signal structures clearly corresponds to an increased entropy and/or Kurtosis in most of the components.

To quantify the diversity, uncertainty, sparseness or randomness properties of the individual wavelet components we used the Renyi entropy defined by:

$$H_\alpha(F) = \frac{1}{1-\alpha} \log(\sum_{i=1}^{n} p_i^\alpha) \tag{3.17}$$

where the $p_i$ are the probabilities associated to the individual observation $f_i$ in the component series $F = (f_1, f_2, \ldots f_n)$. Analogously, to explicitly spot the presence of peaks and local maxima we measured the Kurtosis of each component as:

$$K(F) = \frac{\mu_4}{\sigma^4} - 3 \tag{3.18}$$

where $\mu_4$ is the fourth momentum about the mean of the time series and $\sigma$ is its standard deviation.

The above properties estimated over all the component signals resulting from the decomposition can be used for defining a set of features describing the interesting traffic properties, and hence constitute the feature space, built from multi-dimensional feature vectors.

Also the wavelet packet transform can be a suitable mean to extract additional features that can be used in the classification process. Compared with the traditional multi-resolution wavelet analysis, it has the potentiality to achieve a better

discrimination power by allowing a more complete analysis of the higher frequency domains of a signal. All the frequency domains divided by the wavelet packet can be easily selected and classified according to the characteristics of the analyzed signal.

However, when decomposing the input signal with the Wavelet Packets Transform, it is crucial to choose from the resulting components the packets that best highlight the traffic dynamics and peculiarities and use it as the classification criteria. This implies choosing the most efficient or best basis for our classification task, i.e. the one that better shows both energy and frequency changes. The idea is to select a suitable orthogonal subset basis from the general wavelet packets according to the objectives of the specific analysis. The process starts from the WPT decomposition binary tree yielding the wavelet packets that build a complete spatial description of the source input signal. This is clearly a redundant tree consisting in a set of subspaces, and to determine the best basis and hence implicitly suppress the less representative coefficients having little effect on the overall signal integrity, a cost function must be chosen to drive the process towards its goal. The commonly used criterion for choosing the most efficient or best basis for a given signal is the minimum entropy criterion [18], based on a dynamic programming approach for selecting the best wavelet packet basis functions that can compactly represent the original signal and that can be used to develop additional elements in the feature vectors for each sample at the last level of decomposition.

## 3.6 Building the feature space

A fundamental preliminary task in a network anomaly detection process is baselining, which can be defined as the act of measuring and rating the performance of a network. Providing a network baseline requires evaluating the normal network utilization, protocol usage, peak network utilization, and average throughput of the network usage over a significant period of time. This is a very slow and complex task requiring a lot of computing effort and human expertise, but fortunately it has to be performed only once, in the initial "knowledge construction" phase where

the most significant network utilization patterns, describing the fundamental traffic dynamics, should be described in terms of specific features gathered from traffic observations.

In building our baseline we start from the assumption that the fundamental dynamics describing the traffic originating from a single host result from the linear composition of many independent streams, each associated to a specific instance of a protocol/application (e.g. web browsing, mail activity, file sharing etc.).

For this sake, the "candidate training" set packet traces have been organized into traffic flows and the corresponding time series have been computed for each the native input feature values. A traffic flow can be considered as a set of packets characterized by some common traffic features (e.g., protocol and/or origin and destination network addresses and ports).

Such aggregated traffic flows are one of the most important information sources for analyzing network anomalies since most of the abnormal behaviors that can be observed over the Internet (malware spreading, network resource abuse and outbreaks) would cause changes of the normal flow patterns. Specifically, end-to-end traffic flows have been identified by source IP and source port, destination IP and destination port and protocol. Flows are bidirectional and their first packet determines the forward direction. They are also characterized by a limited duration: UDP flows are terminated by a flow timeout. TCP flows are terminated upon proper connection tear-down (TCP state machine) or after a timeout (whichever occurs first). We only consider UDP and TCP flows that have at least one packet in each direction and transport at least one byte of payload. This excludes flows without payload (e.g. failed TCP connection attempts) or "unsuccessful" flows (e.g. requests without responses commonly found in scans). The resulting flows have been aggregated and categorized according to some application groups of interests proposed (see table 3.1).

More specifically, for each native traffic feature associated to a specific traffic group, such as number of flows, per-flow packet size or inter-arrival time, the above RQA and Wavelet Analysis measures can be computed in a sliding window scheme,

| Application Group | Protocols |
|---|---|
| WWW | HTTP, HTTPS, GOPHER, PROXIES |
| Network_Infrastructure | BGP, BIG_BROTHER, BONJOUR, BOOTP, DNS, HDL_SERVER, ICMP, IPERF, MADCAP, MULTICAST_DNS, NETWORKLENS_EVENT, NETWORKLENS_SSL_EVENT, NTP, OSUNMS, RIP, SLP, SNMP, TRACEROUTE |
| File_Transfer | BBCP, BBFTP, FTP_CONTROL, FTP_DATA, GSI_FTP, RSYNC |
| Mail/News | SMTP, SMTPS, NNTP, NNTPS, POP3, POP3S, IMAP, IMAP-SSL |
| Terminal_Emulation | SSH, DAMEWARE, FIREWALL-1, NETWARE, PCANYWHERE, REXEC, RLOGIN, RSHELL, RADMIN, TELNET, TIMBUKTU, VMWARE, VNC, WINDOWS_RDP, X11, LINUXCONF, NCP |
| Streaming | ABACAST, BACKBONE_RADIO, CAMARADES, ITUNES, LIQUIDAUDIO, MS_MEDIA, POINTCAST, REALPLAYER, RTSP, SHOUTCAS |
| Chat | CUSEEME, CONVERS, DIALPAD_CTRL, DIALPAD_DATA, DIGICHAT, ICHAT, ICU_II, IRC, ITALK, IVISIT, MS_NETMEET, NET2PHONE, PGPFONE, QNEXT, ROGER_WILCO, SIP, SKYPE, TALK, THE_PALACE, VIRTUAL_PLACES, VOCALTEC, WINDOWS_MESSENGER, XMPP_JABBER, YAHOO_MESSENGER, IVISIT |
| P2P | BITTORRENT, BLUBSTER, DIRECT_CONNECT, EDONKEY, FASTTRACK, GNUTELLA, GOBOOGY, GROUPER, HOTLINE, IMESH, NAPSTER_CONTROL, NAPSTER_DATA, ROMNET, SCOUR_EX, SHARE_DIRECT, SORIBADA, SOULSEEK, WASTE, WINMX |
| Other | anything else |

**Table 3.1**: Taxonomy of Network Applications used for building the feature space.

acting upon successive fixed-length time windows (epochs). An alignment in those statistic variables (outputs) with the original time series (input) when adjusting to achieve a better embedding dimension might reveal details not obvious in the 1-dimensional input data. This allows us to study their evolution in time and can be used for the detection of transitions [9] that we need to reveal in our approach. Note that such strategy also allows the identification of malicious behavior whose progress is gradual, provided that the epoch size is chosen in such a way that there is an appreciable variation across epochs.

The same concept can also be extended to the aggregate traffic associated to a group of hosts or an entire network, where each stream can be viewed as the aggregation of homogeneous traffic belonging to the component hosts. We can easily note that under normal network conditions the traffic time series associated to each aggregation of homogeneous streams tend to systematically follow a specific trend/distribution when observed on a sufficiently large network dimension and timescale.

The simultaneous analysis of the features associated to cumulative traffic trends, together with those ones related to the variations in each traffic class or host aggregates, introduces great control granularity in the detection process by adding new points of observation, and hence new dimensions in the feature vectors, that can ease correlation and inference activities in the machine-learning based binary classification process. The availability of several different observations associated to the individual traffic components may also be helpful in spotting and describing the nature and behavior of the observed anomalous phenomena (e.g. protocols affected, transport facilities used, traffic volumes distribution, etc.). This last issue can reveal to be of fundamental importance in the development of countermeasures or reaction strategies, that is however out of the scope of the present dissertation.

However, the statistical traffic patterns at the foundation of our analyisis can be affected by unpredictable network "noise" phenomena largely due to the randomness and burstiness of the traffic behavior, that can adversely affect the anomaly detection process. Hence, to enhance the performance of feature extraction and

subsequent detection phase we tried to improve the overall "signal-to-noise ratio" through signal pre-processing by limiting the influence the "noisy" components and unwanted dependencies. For this sake a wavelet-based de-noising technique known as nonparametric regression can be applied to the original per-flow or aggregated time series prior to any RQA or wavelet feature calculation. This technique provides a very effective and simple way of discriminating "pure" signal structures in data sets without the imposition of a parametric regression model (as in linear or polynomial regression). Nonparametric regression is based on qualitative details about the regression function, that is the functions shape being established by the input data. In this phase, a new reference (de-noised) signal is assembled from the various components resulting from the wavelet decomposition (through an inverse discrete wavelet transform) by properly altering some of the values associated to the derived signals that carry information that we would like to ignore, and hence that has to be suppressed (e.g. noise or misleading spontaneous events) to ease traffic profile identification.

This can be done, for example, by using thresholding techniques to cut specific set of frequencies, with the aim of suppressing some useless variations in the signal introduced by noise phenomena. The involved thresholding techniques act essentially by keeping or killing certain wavelet coefficients depending on their real contribution to the signal information content. The coefficients which survive exhibit a certain pattern, corresponding to fundamental traffic dynamics: a coefficient at a finer scale (presumably due to noise effect) never survives to thresholding unless its parent also survives. An example plot showing the effect of the de-noising process over a portion of the baseline signal is reported in Fig. 3.2.

The construction of the knowledge base from the aforementioned features, determined according to the previously presented methods, is the next step in building a working anomaly detection system.

For this sake, a sufficiently large number of pre-classified feature vector samples has been aggregated into homogeneous "training" data sets to be used, with the aid of traditional data mining and machine learning techniques, to determine the most

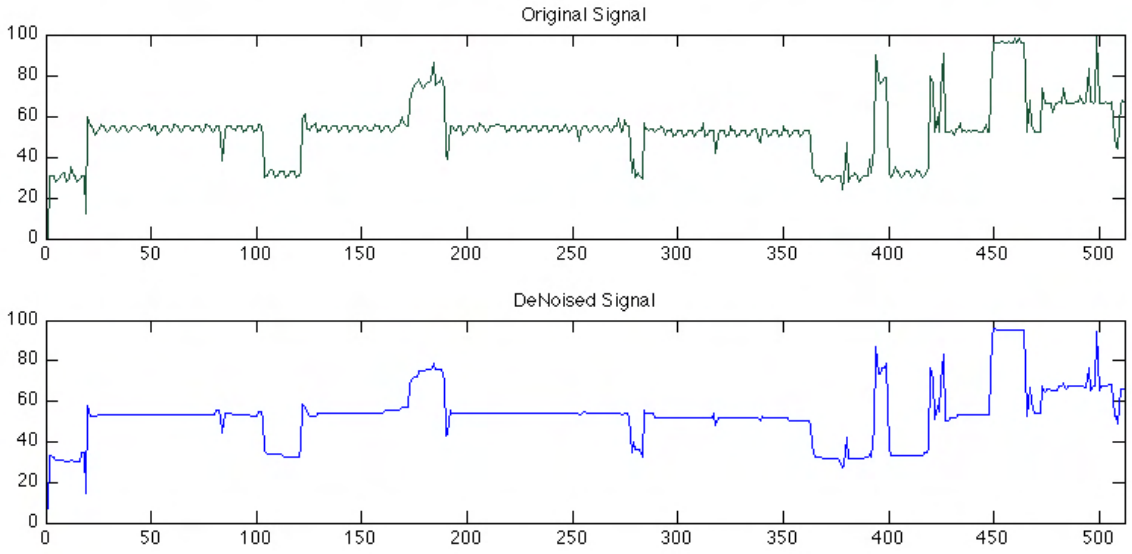



**Figure 3.2**: The de-noising effect, baseline trace.

discriminating features associated to the interesting type of traffic.

More formally, the training data is a set $S = (s_1, s_2, \ldots s_n)$ of $n$ already classified samples. Each sample $s_i = (f_1, f_2, \ldots f_m)$ is an $m$-dimensional vector where the $x_i$ represent the features of the sample. The training data is augmented with an additional feature set $C = (c_1, c_2, \ldots c_n)$ where $c_i$ represent the class (i.e. anomalous, not anomalous) to which each sample $c_i$ belongs.

# References


[1] N. Marwan and J. Kurths, "Nonlinear analysis of bivariate data with cross recurrence plots", *Phys. Lett. A*, vol. 302, pp. 299–307, 2002.

[2] V. Paxson, S. Floyd, "The Failure of Poisson Modeling", *IEEE/ACM Transactions on Networking* (3) pp. 226-244, 1995.

[3] M.S. Taqqu, V. Teverovsky, W. Willinger, "Is network traffic self-similar or multifractal?", *Fractals* 5:63, 1997.

[4] D. Chakraborty, A. Ashir, T. Suganuma, et. al., "Self-similar and fractal nature of Internet traffic", *International Journal of Network Management* 14, pp. 119-129, 2004.

[5] R.H. Shumway, D.S. Stoffer, "*Time series analysis and its applications*", *Springer Texts in Statistics*, Springer-Verlag, New York, 2000.

[6] M.B. Priestley, "*Nonlinear and Non-Stationary Time Series*", Academic Press, New York, 1988.

[7] S.P. Washington, M.G. Karlaftis, F.L. Mannering, "*Statistical and Econometric Methods for Transportation Data Analysis*", Chapman and Hall/CRC Press, 2003.

[8] M. Burgess. "*Probabilistic anomaly detection in distributed computer networks*", Sci. Comput. Program. 60:1, pp. 1-26, 2006.

[9] J. P. Zbilut, C. L. Webber, "Embeddings and delays as derived from recurrence quantification analysis", *Physics Letters A*, 171, pp. 199-203, 1992.

[10] J. P. Zbilut, A. Giuliani, C.L. Webber, "Recurrence quantification analysis and principal components in the detection of short complex signals", *Physics Letters A 237*, pp. 131–135, 1998.

[11] C. L. Webber, J. P. Zbilut, "*Recurrence quantification analysis of nonlinear dynamical systems*", In Tutorials in contemporary nonlinear methods for the behavioral sciences, pp. 26-94, 2005.

[12] P. Grassberger, I. Procaccia, "Characterization of strange attractors", *Physical Review Letters*, 50(5), 1983.

[13] M. Misti and Y. Misti and G. Oppenheim and J.-M. Poggi, *"Wavelet Toolbox Users' Guide"*, The MathWorks, 2 ed., 2010

[14] W. E. Leland and M. S. Taqqu and W. Willinger and D. V. Wilson, "On the self-similar nature of Ethernet traffic (extended version)", *IEEE/ACM Transactions on Networking*, 2, pp. 1–15, 1994.

[15] S. Mallat, "A theory of multiresolution signal decomposition: The wavelet representation", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11, pp. 674–693, 1989.

[16] S. Mallat and W. L. Hwang, "Singularity Detection and Processing with Wavelets", *IEEE Transactions on information theory*, 38(2), 1992.

[17] P. Barford, J. Kline, D. Plonka and A. Ron, *"A signal analysis of network traffic anomalies"*, Proceedings of the 2nd ACM SIGCOMM Workshop on Internet Measurment (IMW 2002), pp. 1–12, 2002.

[18] Coifman, R.R. and Wickerhauser, M.V., Entropy-based algorithms for best basis selection, *IEEE Transactions on Information Theory*, 38(2), pp. 713 -718, 1992.

# Chapter 4

# Modeling the Detection Process Through Machine Learning Techniques

## 4.1 Introduction

Machine learning can be defined as the ability of a fully automated system to learn by example how to perform a specific task or group of tasks and improve its performance over time with experience, as more examples (or training data) are provided in its knowledge base.

This approach focuses on understanding the process generating the data, and is very useful to ease our modeling task since it allows the machine to learn about the occurrence of a specific statistical phenomenon via inductive inference based on observing the empirical data that represent incomplete information about the phenomenon itself.

Classification is a fundamental task in Machine Learning, by which machines "learn" to automatically recognize complex patterns, to distinguish between exemplars based on their different patterns, and to make intelligent decisions.

Supervised learning is the machine learning task of inferring a function from supervised training data. The training data consist of a set of pre-classified training examples. In supervised learning, each example is a pair consisting of an input object (typically a vector) and a desired output value (also called the supervisory

signal). A supervised learning algorithm analyzes the training data and produces an inferred function, which is called a classifier (if the output is discrete) or a regression function (if the output is continuous). The inferred function should predict the correct output value for any valid input object. This requires the learning algorithm to generalize from the training data to unseen situations in a "reasonable" way.

An anomaly detector can be modeled as a supervised learning system analyzing the deviation from normal traffic profiles that are determined from historical data. Thus, the basic prerequisite for an effective anomaly detection system is the ability of isolating only the really significant information from a huge amount of noisy, highly-dimensional data to build a consistent and comprehensive model and then flagging all the really relevant deviations from this model. Any deviance in either traffic type, or amount could then be detected and considered a potential incident. However, separating the traffic flows associated to abnormal behaviors from those related to normal traffic patterns is not a trivial task, since different kind of anomalies can occur in traffic statistics in very different ways, so that formulating sufficiently general models of normal network activity and of anomalies can be extremely difficult.

Early detection schemes were designed to discover previously known attacks from the above data by fitting specific pre-trained models (i.e. event chains or traffic templates pre-classified as an attack or not). They were also able of flagging network anomalies due to abnormal behavior changes (such as DDOS attacks) in the overall traffic profile.

The former approach cannot be quickly adapted to handle new and unknown (zero day) types of intrusion while the latter one demonstrated to be only useful for detecting large-scale network attacks, but it was not effective in coping with a large class of abnormal behaviors that did not cause obvious changes in traffic volumes.

Modern machine learning technologies enable the development of non-parametric anomaly detection schemes that are more adaptive to variations in the characteristics of normal network behavior and hence more effective in handling any kind of phenomena that can occur on the network. Thus, nonparametric adaptive detection

schemes based on more sophisticated machine learning techniques are strongly desirable since they are able to learn the intrinsic nature of normal traffic observations and autonomously adapt themselves to the possible variations that can occur in such structure of "normality".

The main advantages of this technique are that in line of principle it is not restricted to any specific environment, and that it can enable the detection of any type of unknown attacks. Its detection capability is directly associated with the correctness of the underlying traffic model, since the use of accurate methodologies for traffic description is a critical factor for the effectiveness of anomaly detection techniques. The more realistic is the traffic source model, the more accurate is the estimate of network behavior and so, more appropriate will be the detection response. On the other hand, if the traffic model does not accurately represent the actual network traffic, one may overestimate or underestimate anomalous phenomena. However, the non-parametric nature of the above techniques can simultaneously become their main strength and Achille's heel: a nonparametric method is not based on any precise form or distribution of the sample observations and consequently it is more robust. At the same time being non-parametric means that no prior knowledge about specific anomalous phenomena is available to be incorporated into the detection model, greatly reducing its effectiveness.

Starting from the above considerations, we modeled our anomaly detection strategy according to a more sophisticated two stage "split" classification and machine learning problem, based on a traditional Minimum Volume Set/binary classification scheme, that takes the best features from non-parametric and parametric models. In such scheme, the first phase, referred as "supervised" learning, focuses on feature extraction to build effective traffic profiles to be used in the second unsupervised classification phase.

In doing this the system profiles the normal behavior of network activity from some available traffic patterns, by extracting, from both "normal" and pre-classified "anomalous" network traffic samples (known positives in the "training set"), a con-

sistent set of RQA and Wavelet based features, and stores these profiles as a knowledge base to be used as a reference in the following analysis.

The most discriminating reference features among those that constitute the implicit parameters of our traffic model, become the main drivers of the machine-learning-based binary classifier, driven by cost-sensitive SVMs, at the core of the second stage of the proposed anomaly detection strategy. In this phase the system compares the actual network activity profile with the stored ones and generates an alarm when any event corresponding to a deviation from the normal profile is observed on the network.

Such "split" learning process presents several advantages over traditional non-parametric methods. A great deal of unlabeled data, corresponding to different traffic features can be used in an exploratory first step, and then a more sophisticate classifier can be quickly built by using only the really significant (in terms of information content) labeled training samples. The supervised phase introducing the necessary a-priori knowledge about the normal traffic model (the baseline, where a limited set of parameters is available in form of most discriminant traffic features), can use either a generative or diagnostic method without any difficulty.

## 4.2 Modeling Anomaly Detection as a Machine Learning Problem

The main goal in anomaly detection is classifying the samples in a collection of observation related to specific features describing the involved phenomenon as being either normal/typical or abnormal/anomalous. Specifically, we would like to find a set in our collection such that points inside the set correspond to typical data while points outside are anomalies. This set, that is the aforementioned baseline, must be "*learned*" from a collection of training samples gathered under "*certified*" normal conditions.

Clearly, even when we are able to observe anomalous data, it is often difficult or somewhat impossible to identify the anomalous training samples a priori. In other words, we must be able to detect anomalies without knowing what they look like. Such anomaly detection scheme can be essentially viewed as a machine learning problem, based on modeling normal and anomalous data from a "*training set*", and then flagging all the deviations from this model. This is a classical classification task, where only one significant class exists in the training data (the anomalous traffic class in our case) and we have to learn the characteristics of such class and determine if any unseen instance belongs to it or not (*binary* classification).

From the theoretical point of view our network anomaly detection problem can be formulated as follows [2].

A collection of traffic data measurements is described by a scalar time series $\{x_t\}_{t=1}^{T}$ governed by a probability distribution $p$. Although all these measurements are associated to the occurrence of specific events within the event space $S$, the correspondence between them may not be known in advance. We are interested in partitioning the event space $S$ into two sub-spaces corresponding to the normal and the anomalous network traffic conditions. Also, we need to infer the membership of a particular event in one of the above subspaces starting from the corresponding time series values. To accomplish this task, since the probability distribution $p$ describing the behavior of the time series is unknown, we can use a mechanism enabling the reconstruction of its volumetric representation from the collection $\{x_t\}_{t=1}^{T}$. A general approach to the problem of identifying this representation is based on building a Minimum Volume Set (MVS) characterized by a probability mass $0 < \beta < 1$ associated to the distribution $p$ for a volume measure $\mu$ [5], that is:

$$G_{\beta}^{*} = \arg\min\{\mu(G) : p(G) \geq \beta, \quad G \text{ measurable}\} \tag{4.1}$$

In the most common case $\mu$ can be chosen to be the *Lebesgue* measure, although such technique extend easily to other measures. The parameter $\beta$ can be chosen by the user to reflect a desired false alarm rate of $1 - \beta$.

These Minimum volumes summarize the regions of greatest probability mass of the distribution $p$, and are useful for detecting anomalies and constructing confidence regions. Hence, online evaluation of minimum volume sets satisfying (4.1) inherently allows the identification of the highest density regions where the mass of $p$ is concentrated. All the points falling outside these regions and the associated events can be declared as anomalous.

It can be observed that, if $p(\cdot)$ is a multivariate Gaussian distribution and $\mu$ is the Lebesgue measure, then the Minimum volume sets are represented as ellipsoids (see fig. 4.1).

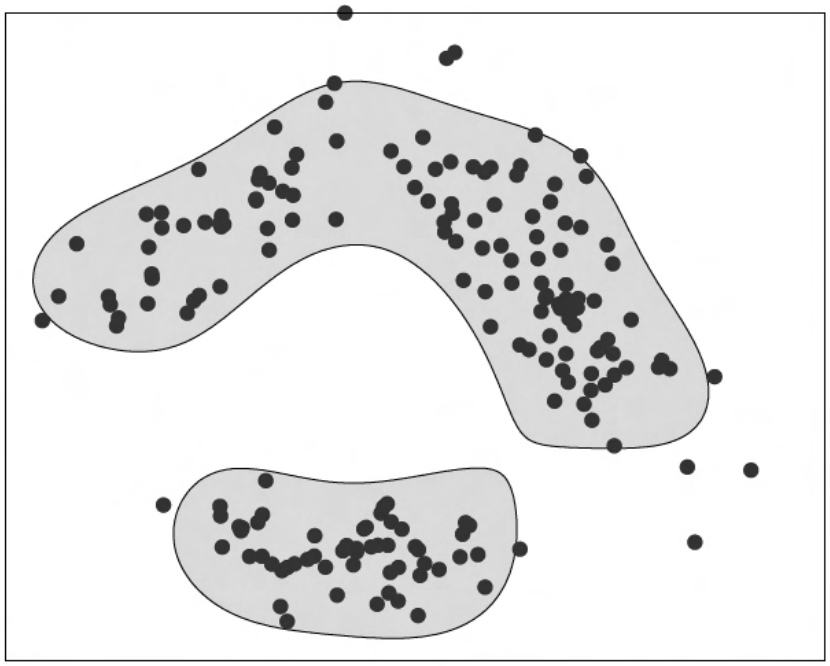


**Figure 4.1**: Minimum volume sets example with $\beta = 0.9$.

The most common available methods for estimating $G$ are presented in [6] and require the use of SVMs or, more specifically, cost-sensitive SVMs. In particular, if the anomalous class is structured according to a uniform distribution, constructing this set is actually equivalent to finding a Neyman-Pearson [7] classifier (a cost sensitive classifier that tries minimize false negatives while constraining false positives

to be below a certain significance level).

That is, we can use the above model to classify our sample data $\{x_t\}_{t=1}^{T}$ into the two classes (*positive*:Anomalous traffic, *negative*:Normal traffic), according to [6]. In detail, In Neyman-Pearson, the goal is to design a binary classifier $c_\alpha$ that minimizes the miss rate while constraining the false positive rate to not exceed some user-specified significance level $\alpha$.

We assume that two unknown probability measures $\pi$ and $\upsilon$ are associated respectively to positive and negative events. To estimate the Minimum volume using Neyman-Pearson classification, we can think of setting $\upsilon = 1 - p$ and $\pi = \mu$. In this case the Minimum volume and Neyman-Pearson classification solutions fully coincide. To implement this idea we initially label all the observed samples in $\{x_t\}_{t=1}^{T}$ as normal traffic (negative) data. We then simulate (and verify) a certain number of points from the reference measure $\mu$, and labels them as positive. Keeping the false positive rate constrained, ensures that the probability mass of the set of negative/normal observations is at least $\beta$, and since we draw the positively labeled class from the reference measure $\mu$, the minimization of false positive rate is equivalent to minimizing $\mu$. From this perspective, the reference measure $\mu$ is a prior on the distribution for anomalies. Taking $\mu$ to be the Lebesgue/uniform measure can be viewed as assuming a non-informative prior on anomalies. In summary, by taking our minimum volume set to be $G^* = \{x : c_\alpha(x) = \mathit{negative}\}$ we can estimate the Minimum Volume set of our data.

Accordingly, it is possible to use the SVM algorithms described in [6] for anomaly detection by simply describing anomalous events through artificial training sets structured according to a uniform distribution. While effective in some cases, this simple approach to generating the uniform data suffers from dimensionality problems due to the large number of features available from observations.

More specifically, the number of features resulting from the previous analyses can be quite large, many of which can be irrelevant or redundant. Also the amount of pre-classified training data that we need to examine to select the best features

may be very large, so that discriminating them by hand is practically impossible.

Furthermore, real multi-dimensional data exhibit distributions that are highly sparse. The different features describing the observed samples may have a disordered structure, presenting overlapping and mutual dependencies, and since high-dimensional data tends to be highly redundant and cannot be effectively separated among the condition of faults, this kind of structure cannot be directly processed into the classifier because it will degrade its performance. Furthermore, the raw data may lack of invariance in their statistical distributions with respect to the associated generating events. That is, the events belonging to the same region in $S$ may be mapped by $f$ in completely different regions of $\Re^m$.

Achieving an appreciable reduction in the dimensionality of raw data by using some feature extraction mechanism described by the function $g : \Re^m \rightarrow \Re^k$ with $k < m$, that is sufficiently robust to both sparsity and transition-related variance, it is generally highly desirable. The purpose of such selection process is to arrive at a smaller number of features which are more meaningful and to discard irrelevant information. This results in an increase of the computational efficiency without loss of accuracy.

Feature reduction is often used as a useful pre-processing task prior to classification by determining only the really discriminating features and discarding less relevant or redundant ones (i.e., those that can be excluded from the feature set without loss of classification accuracy). In fact, correlated or non-informative features can behave like noise in the data, thereby degrading the classifier's performance, as performance and the cost of classification are sensitive to the choice of the features used to construct the classifier.

We then can construct a minimum volume set more easily from the reduced feature set than from the complex raw data. Feature reduction also plays an important role in improving the performance of anomaly detection systems by reducing the computational complexity and hence delivering timely responses to minimize the security compromise.

At the state of the art several approaches for determining the best features are available. One of them is known as feature selection, whose goal is using different techniques to determine a subset containing the most relevant features that can be used for building robust learning models. Such approach can greatly improve the classification performance by removing irrelevant and redundant features from the available data. Another approach for reducing the number of feature to be used for classification is feature extraction. It transforms through several methods the input data by originating a new reduced set of features containing most of the relevant information from the original input data.

### 4.2.1 The feature reduction phase

A simple and sustainable approach for the really discriminating features is trying to gather the characteristic properties of the most significant traffic components from the available observations, by using data mining techniques on the pre-classified feature vectors belonging to the training set, that means, extract a minimum set of highly discriminating (or "primary") vector components (corresponding to the individual traffic features measured) on which the various traffic profiles can be reliably built. More specifically, each decision within the classification process should result from the combined observation of the "primary" traffic features.

To do this the whole feature space must be searched, vector component by vector component, for the subset that is most likely to best characterize the normal and anomalous traffic classes, by identifying and excluding useless or redundant features adversely affecting the performance of the detection/classification process.

The most informative features can be determined according to the C4.5 statistical classification algorithm [3] whose pseudo-code formulation is reported in Algorithm 1. Such algorithm provides an automatic feature selection strategy based on an initial learning/mining phase operating on a sufficiently large quantity of pre-classified data.

Such strategy is based on the construction of a decision tree containing al the

features needed to perform classification, where the branches corresponding to less useful features are automatically pruned (see Fig. 4.2). It also provides the basic mechanisms for ranking the features present in the decision tree to determine the relative importance of any individual feature.

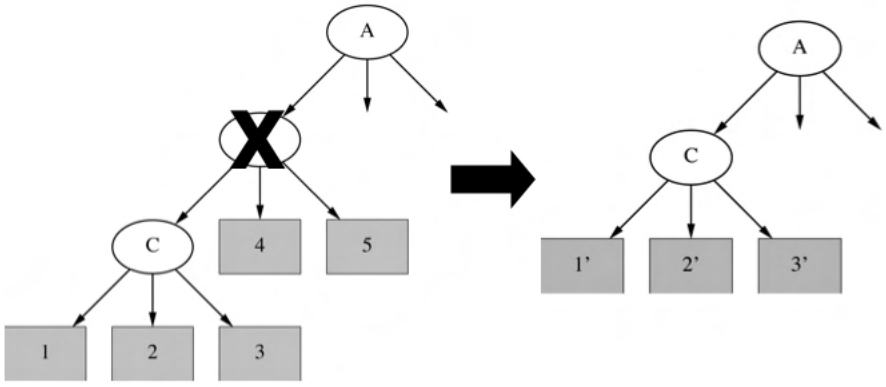



**Figure 4.2**: Pruning on the decision tree example $\beta = 0.8$.

In detail, the algorithm constructs its decision tree by using a divide and conquer strategy. At the beginning only the root node is present. Then, at each node of the tree, C4.5 chooses one attribute of the data that most effectively splits its set of samples into subsets enriched in one class or the other (the locally best choice). Its criterion is the normalized information gain (difference in entropy) that results from choosing an attribute for splitting the data. The attribute with the highest normalized information gain is chosen to make the decision. The C4.5 algorithm then recurs on the smaller sublists.

The aforementioned information gain is the change in information entropy from a prior state to a state that takes some information as given:

$$IG(X,a) = H(X) - \sum_{v \in values(a)} \frac{|\{x \in X | value(x,a) = v\}|}{|X|} \cdot H(\{x \in X | value(x,a) = v\}) \tag{4.2}$$

where $A$ is the set of all features and $X$ the set of all training samples, the $value(x,a)$ with $x \in X$ defines the value of a specific sample $x$ for the feature $a \in A$,

**Algorithm 1** C4.5-Tree($T$)

**Input:**

*$T$: current decision tree root*

**Output:**

*$N$: resulting decision tree root*

```
1: Check for base cases
2: N ← new decision node
3: maxig ← 0
4: abest ← nil
5: for each attribute a do
6:    Find the normalized information gain IG(a) from splitting on a
7:    if IG(a) ≥ maxig then
8:       maxig ← IG(a)
9:       abest ← a
10:   end if
11: end for
12: Create a decision node that splits on abest
13: for each U in the splitting of T do
14:   if U is not empty then
15:      add C4.5 − Tree(U) as a child of N
16:   end if
17: end for
18: return N
```

$H$ specifies the entropy, and $|X|$ is the number of elements in the set $X$.

The above concept can be used to define a preferred set or sequence of features, structured in the form of a C4.5 tree, where the C4.5 ranker method [1] can be used for ranking each feature and identify those ones that work best. The final results can be post-processed through 10-fold cross validation on all the training set constituents, to ensure the maximum reliability of the whole process.

Thus, when we are able to correctly separate the fundamental traffic components, we can obtain a description of traffic "profiles" that, considered with reference to a consolidated baseline, help us to isolate previously unseen, and hence possibly anomalous, behaviors. Specifically, we transform the aggregated input data into a set of relevant features so that the reduced representation contains most of the relevant information from the original signal dynamics.

The "primary" vector components/features resulting from the above transformation, being provably necessary for a correct detection, inherently minimize any unwanted redundancy and noise, and hence they provide the maximum information content in describing the traffic behavior. Consequently, they constitute the ideal input for a machine learning process aiming at detecting anomalous phenomena.

### 4.2.2 SVM-based binary classification

Support vector machines (SVMs) have been used to perform binary classification of anomalous and normal traffic by using the traffic feature descriptors obtained from the above feature extraction process. SVMs' popularity has greatly grown as a highly versatile data-mining tool for classification, since they have very good generalization capabilities and converge effectively towards optimal solutions.

A really promising property of SVMs is that they can be considered as an approximate implementation of the Structured Risk Minimization principle based on statistical learning theory rather than the Empirical Risk Minimization method, in

which the classification function is derived by minimizing the Mean Square Error over the training data set.

Our goal was separating the two classes by using a function induced from available examples and then producing a classifier that will work well on unseen examples, i.e. it generalizes well. The theory of learning classifiers based on the Neyman-Pearson criterion has recently begun to emerge and their popularity has greatly grown as a highly versatile data-mining tool for classification, since they have excellent generalization capabilities and the ability to converge to a single globally optimal solution. Support Vector Machine (SVM) models are a close cousin to classical multilayer perceptron neural networks based on Vapnik's statistical learning theory [4].

The basic idea of applying SVM to classification problems can be simply stated as follows: first, map the input vectors into a new feature space, either linearly or non-linearly, depending on the kernel function. In this space, a decision surface is constructed with special properties that ensure high generalization ability of the network. Then, within the above feature space seek an optimized linear division which separates the points within the space into two or more classes. In its operation, an SVM model transforms the original input space into a higher dimensional feature space and makes use of a linear hyper-plane in such space to separate points. The hyper-plane vector $\vec{w}$ has a representation in terms of the training samples $x_i$, $y_i$) and their Lagrangian multipliers $\alpha_i$, with $0 \leq \alpha_i \leq C$:

$$w = \sum_i \alpha_i y_i x_i \tag{4.3}$$

Here $C$ is a positive constant, which can be viewed as the cost of the misclassification error. Support Vector Machines tackle the computationally intractable problem of dealing with a very highly dimensional space by introducing suitable kernel functions. The use of such kernel functions, providing a unifying framework for most of the commonly employed model architectures, also extends the class of decision functions to the non-linear case. Such activity is accomplished by mapping

the input data space $X$ into a higher dimensional feature space $\aleph$, by using a function $\Phi : X \rightarrow \aleph$ and solving the linear learning problem in $\aleph$. It is not necessary to know the function $\Phi$: it is sufficient to have a kernel function $k(\cdot, \cdot)$ which is a symmetric positive definite function that satisfies Mercer's conditions and represents a legitimate inner product in the feature space.

$$k(x, y) = \sum_{i}^{\infty} a_i \Phi(x) \cdot \Phi(y), a_i \geq 0 \tag{4.4}$$

$$\int \int k(x, y) g(x) g(y) dx dy > 0, \quad g \in L_2 \tag{4.5}$$

In doing this, SVM can be viewed as an alternative training method for multilayer perceptron classifiers in which the weights of the network are found by solving a quadratic programming problem with linear constraints, rather than by solving a non-convex, unconstrained minimization problem as in standard neural network training. In detail, our training data set is structured as $\{(x_i, y_i)\}$ with $i = 1, 2, \ldots, N$. Here $x_i$ is the vector representing the input features for the $i$-th sample in the training data set, $y_i$ is the corresponding class label (*positive*:Anomalous traffic, *negative*:Normal traffic) and $N$ is the total number of elements in the training data set. For the two classes, $y_i$ is set to a value of $+1$ (representing the positive class) or -1 (for the negative one). The training samples $x_i$ with $\alpha_i > 0$ are called Support Vectors (SV) univocally determining the decision boundary and providing a description of the significant data for classification.

The goal of SVM modeling is to find the optimal hyper-plane that separates clusters of vector in such a way that cases with one category of the target variable are on one side of the plane and cases with the other category are on the other size of the plane. The vectors nearest to the hyper-plane are the support vectors. If, as in our case, all analysis consists of two-category target variables with two predictor variables, and the cluster of points could be divided by a straight line, life would be easier and the results more effective and reliable. Thus, in the classification model learned by SVM, the positive support vectors are those ones satisfying $y_i = +1$,

and conversely, the negative support vectors are the ones satisfying $y_i$ = -1. The SVM-based classification function (that is the output hyper-plane decision function) is of the form:

$$f(x) = \sum_{i=1}^{m} y_i \alpha_i k(x_i, x) + b \tag{4.6}$$

In the function $f(x)$ reported in the previous equation (4.6) $m$ is the number of input values characterized by non-zero values in their (usually no more than $N$) Lagrange multipliers $\alpha_i$ that can be determined by solving a quadratic optimization problem, $k(x_i, x)$ is the kernel matrix and $b$ is a specific bias term. In general, when $|f(x)|$ is high enough, the corresponding prediction confidence will be high. Meanwhile, a low $|f(x)|$ of a given pattern means that $x$ is close to the decision boundary and its corresponding prediction confidence will be low.

SVM training always seek a global optimized solution and avoid overfitting, so it has the ability to deal with a large number of features. By using a nonlinear kernel function, it is possible to compute a separating hyper-plane with a maximum margin in the feature space.

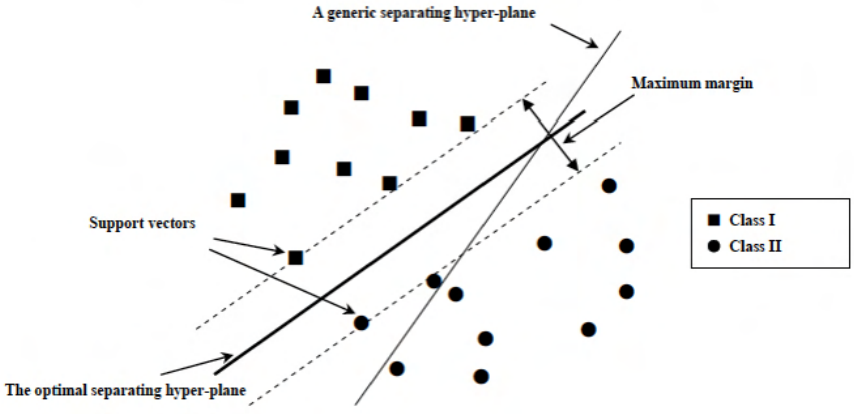



**Figure 4.3**: SVM Operating scheme.

However, we have in a typical baseline many more training samples from the *negative*: Normal class than from the positive one, and hence, since the traditional

SVM classification model aims at minimizing the error probability, it will tend to place less emphasis on the smaller class, performing much better only on discovering the points belonging to the Normal class.

In network anomaly detection a miss corresponds to failing to detect an anomalous event occurring on the actual traffic, while a false alarm corresponds to erroneously flagging the occurrence of an anomalous phenomenon that really has not took place. Clearly, the cost associated with a miss is much different than that associated with a false alarm. Here we might only be able to tolerate a certain level of false alarms, in which case we would have the lowest miss rate possible, provided that the false alarm rate satisfies some constraint.

More specifically, we have specific error rates we would like to achieve in the overall classification process, i.e., we wish to operate at a specific point of the Receiver Operating Characteristics (ROC) curve (universally used to choose the best operating point in a classifier as a trade off between selectivity and sensitivity) [8]. In order to obtain these performances the underlying classification schema has to behave according to Neyman-Pearson (NP) criteria, based on setting a target false alarm rate $\gamma$, and ensuring that the whole classification process is constrained at minimizing the miss rate subject to the condition that the false alarm rate will be never greater than $\gamma$.

There are two immediate options for adapting SVMs to the NP criterion. Since SVM-based classifiers can be interpreted as hyper-planes in an appropriate feature space, one option is to simply shift the hyper-plane to achieve the desired trade-off between false positives and false negatives.

Another approach is to use a "cost-sensitive" SVM, which introduces class-specific weights, penalizing training errors from one class more than the other. In traditional SVM, the same cost $C$ is assigned to any of misclassification error, while, in cost-sensitive SVM, different costs can be given to different types of misclassification seeking to minimize the number of high cost errors and total misclassification cost. More precisely in equation (4.3) $0 \leq \alpha_i \leq C_i$ where $C_i$ is the cost if the $i$-th sample is misclassified.

The key challenge in this approach lies in appropriately setting the free parameters in the SVM (in particular those which determine the relative costs for the two error types). Thus, training an SVM for NP classification can be easily accomplished using a cost-sensitive SVM by tuning the operating parameters to achieve the desired error constraints.

In [9] it was shown that we can use the $2\nu$-SVM to achieve the desired false alarm rate by adjusting $\upsilon$, $\rho$ and $\gamma$ appropriately. The concept of $2\nu$-SVMs has been presented in [10] as a cost-sensitive SVM having the primal formulation:

$$\min_{w,b,\zeta,\rho} \frac{1}{2}\|w\|^2 - \upsilon\rho + \frac{\gamma}{n}\sum_{i\in I_+}\zeta_i + \frac{1-\gamma}{n}\sum_{i\in I_-}\zeta_i \tag{4.7}$$

subject to:

$$y_i(k(w,x_i)+w) \geq \rho - \zeta_i; \quad \zeta_i \geq 0; \quad \rho \geq 0 \quad for i = (1,\ldots,n) \tag{4.8}$$

where $I_+$ is the class of positive events ($\{i : y_i = +1\}$) whereas $I_-$ refers to negative, or not anomalous ones ($\{i : y_i = -1\}$).

# References


[1] I. H. Witten and E. Frank, "*Data Mining: Practical Machine Learning Tools and Techniques*", 2nd edition, Morgan Kaufmann, 2005.

[2] T. Ahmed, B. Oreshkin, and M. J. Coates, "Machine learning approaches to network anomaly detection", *in Proc. SysML*, 2007.

[3] R. O. Duda, P. E. Hart and D. G. Stork, "*Pattern Classification*", 2nd edition, John Wiley and Sons, 2001.

[4] V. Vapnik, "*The Nature of Statistical Learning Theory*", 1st Ed. NY Springer, 1995.

[5] M. Davenport, R. Baraniuk, and C. Scott, "Learning minimum volume sets with support vector machines," in Proc. IEEE Int. Workshop on Machine Learning for Signal Processing (MLSP), Maynooth, Ireland, Sep. 2006.

[6] C. Scott and R. Nowak, "Learning minimum volume sets," *J. Machine Learning Research (JMLR)*, vol. 7, pp. 665–704, Apr. 2006.

[7] C. D. Scott and R. D. Nowak, "A Neyman-Pearson approach to statistical learning", *IEEE Trans. on Information Theory*, vol. 51, no. 11, pp. 38063819, 2005.

[8] A. P. Bradley, "The use of the area under the ROC curve in the evaluation of machine learning algorithms", *Pattern Recognition*, vol. 30, no. 7, pp. 1145–1159, 1997.

[9] M. A. Davenport, R. G. Baraniuk, and C. D. Scott, "Controlling false alarms with support vector machines", *in Proc. Int. Conf. on Acoustics, Speech, and Signal Proc. (ICASSP)*, 2006.

[10] M. A. Davenport. The 2nu-SVM: A cost-sensitive extension of the nu-SVM. Technical Report TREE 0504, Rice University, Dept. of Elec. and Comp. Engineering, 2005. See http://www.ece.rice.edu/~md/

[11] M. A. Davenport, Error control for support vector machines, MS thesis, Department of Electrical and Computer Engineering, Rice University, Houston, TX, 2007, http://dsp.rice.edu/software/2nu-svm

[12] Chih-Chung Chang and Chih-Jen Lin, LIBSVM : a library for support vector machines, 2001. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm

# Chapter 5

# Proof of Concept Implementation

## 5.1 Introduction

To demonstrate the effectiveness of our anomaly detection model, we present a simple proof of concept implementation built on publicly available tools and working offline on real traffic traces previously captured on the 1 Gbps link to the Internet of the "Via Claudio" Campus of the Federico II University through port mirroring from the Cisco 6509 border node to an HP DL380 Dual Processor (Intel Xeon 2.5 GHz) promiscuous mode monitoring server running the FreeBSD operating system. We saved the first 64 bytes of each packet, which includes the IP and TCP/UDP headers needed to extract timing and volume information in single flows or aggregated traffic. The whole traffic capture scenario is sketched in Fig. 5.1.

We collected incoming traffic only both for storage space availability and traffic cleaning and purity reasons (filtering undesired anomalous events is much easier since the expected traffic pattern is known). Several sample traces have been captured within multiple different time intervals, properly chosen to cover some typical cases such as the noticeable differences in network usage between morning and evening hours, and between weekdays and weekends.

The first two 24 hours traces (respectively A and B in table 5.1) contain several anomalous events simulated through distributed SYN floods, LAND and port-

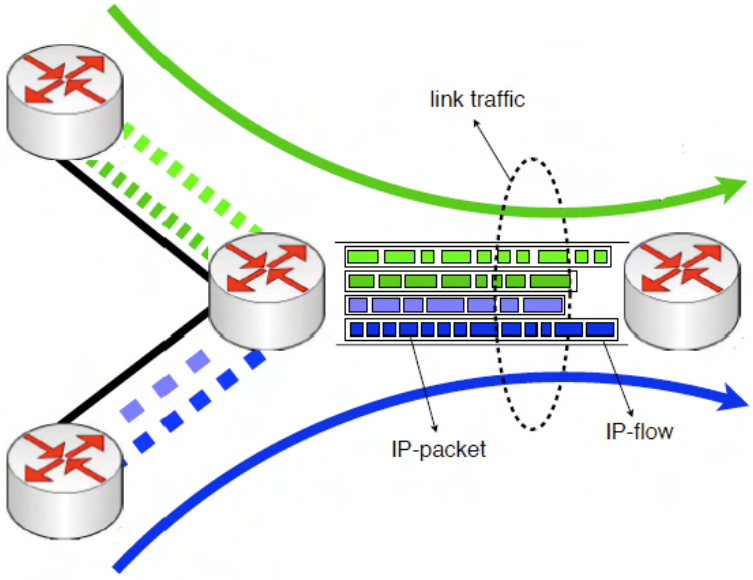



**Figure 5.1**: Data collection/traffic capture scenario.

scanning attacks occurring at various times (see table 5.2 for the anomalies in trace A), while the other two traces (C and D) have been properly post-processed through Snort [1] by detecting and filtering suspicious events to make them reasonably anomaly-free (baseline "normal" traffic). The above three types of attacks have been chosen both for duration and specific characteristics (i.e. their explicit "noisy" behavior) to be a sufficiently consistent and representative sample for a volume-based analysis. Most of the anomalous traffic patterns that can be currently observed on the Internet (inbound distributed denial of service attacks, bandwidth floods, single and multiple scans) can be associated to these attacks types.

The 2-weeks anomaly-free trace D, covering a period sufficiently long to capture all the traffic recurrence dynamics, has been used only for determining the optimal embedding parameters, while the 24 hour traces A (known attacks with their time location during the day) and C (certified anomaly free day) have been used for building the training set. Then, Finally, the trace B has been used for result verification and model validation (testing set).

| Trace | A | B | C | D |
|---|---|---|---|---|
| **Duration** | 24 hours | 24 hours | 24 hours | 336 hours |
| **Anomalies** | 9 | 6 | 0 | 0 |
| **Packets** | $5.82 \cdot 10^8$ | $6.67 \cdot 10^8$ | $5.97 \cdot 10^8$ | $6.72 \cdot 10^9$ |
| **Bytes** | $2.27 \cdot 10^{11}$ | $2.18 \cdot 10^{11}$ | $2.21 \cdot 10^{11}$ | $2.51 \cdot 10^{12}$ |

**Table 5.1**: General workload dimensions of the traces.

## 5.2 The traffic features of interest

We used the CAIDA Coral Reef suite [2] to process the above packet traces and compute the traffic *feature* values to be used in our analysis. More precisely *crl-rate* and *crl-flow* have been used respectively for aggregate and per-flow time series extraction, and *t2-convert* was the tool of choice for organizing groups of flows into homogeneous traffic classes.

On the other hand the Wavelab framework [15] running within the Matlab environment has been used for Multi-resolution wavelet analysis, Discrete Wavelet Packet Transform and best-basis calculation.

When choosing the main features to be analyzed from our traffic volume data, the "kitchen-sink" method of using as many features as possible was eschewed in favor of a constraint-based approach. The main limitation of an approach choosing a too large number of features becomes evident by considering the fundamental objective according to which calculation should be realistically possible within a resource constrained IP network device. Thus the considered features needed to fit the following constraint-selection criteria:

1. Complete packet payload independence.
2. No implicit dependence from the transport layer.
3. Computational Simplicity.

| Start Time | Duration | Attack | Packet rate |
|---|---|---|---|
| 01:15 | 60s | SYN flood | 500/s |
| 02:15 | 300s | SYN flood | 500/s |
| 03:15 | 600s | SYN flood | 500/s |
| 04:15 | 60s | SYN flood | 250/s |
| 05:15 | 300s | SYN flood | 250/s |
| 06:15 | 600s | SYN flood | 250/s |
| 13:15 | 600s | portscan | 250/s |
| 22:15 | 30s | LAND | 500/s |
| 23:15 | 15s | LAND | 500/s |

**Table 5.2**: The simulated attacks in trace A.

Note that both recurrence quantification and wavelet analysis assume the presence of a scalar time series to be processed. Thus the fundamental features have to be studied one at a time and combined into specific aggregate structures that will be used in classifying traffic anomalies. More precisely, we defined a set of features describing the traffic pattern in a specific time interval (the epoch). Let $F$ denote the feature space of the actual network traffic. We use a multi-dimensional feature vector $f \in F$ built from the fundamental traffic observations (e.g. the overall byte rate, the average inter-arrival time between packets) by computing on each epoch the needed RQA and Wavelet statistics (e.g. %recurrence, determinism, entropy, %laminarity and trend, per-wavelet-component Renyi entropy/kurtosis and DWPT best basis etc.).

Clearly, for simplicity sake, only a limited subset of features has been exploited in this proof of concept (essentially the RQA ones, that from a preliminary analysis seemed to exhibit a greater discriminating power), whose main goal was demonstrating the potential of the proposed approach. The use a large set of features,

particularly if determined by different points of observation such as the aggregate traffic sub-components (e.g. web, infrastructure, P2P etc. flows) can add greater granularity to the detection power, introducing the ability of distinguishing between anomalous events that manifest themselves by using specific network protocols/facilities as transport vectors. This gives us additional information that can be very useful in the determination of countermeasures (filtering rules etc.). A more sophisticated analysis, exploiting more complete combinations of the large portfolio of available feature will be the object of future research activities.

## 5.3 Choosing the sampling rate

Another very important design choice strongly characterizing the effectiveness of the anomaly detection process is the sampling rate. Let us start by considering the worst case in which we are sampling from a chaotic system. These systems, like the stochastic ones, are unpredictable in the long run. Such behavior is related to the speed at which nearby trajectories diverge in the phase space, which in turn is related to the Lyapunov exponents of the system under analysis [3]. Thus, if the sampling interval goes beyond the predictability window, even if the underlying system is chaotic, we will observe a stochastic behavior. In this case, if we suspect to be in presence of a deterministic underlying system, the best option is repeating the experiment by increasing the sampling rate. Interpolating between data points would not be useful since no new information would be introduced. Starting from the above considerations, we used a 1 second sampling interval. The choice of such interval results from a tradeoff between sensitivity on one side and accuracy and memory usage on the other side. A short sampling interval increases the sensitivity to transient phenomena. On the other hand, slightly larger sampling intervals increase the profiling accuracy while reducing, at the same time, the memory footprint. Nevertheless, the self-similarity properties imply that traffic characteristics exist across many time scales, i.e., aggregated traffic does not necessarily get steadier. Consequently, the chosen 1 second sampling interval resulted in the best

compromise for our analysis.

## 5.4 The baseline and training set size

It should be noted that, to achieve satisfactory results, both the baseline used for delay-coordinate embedding and the training set needed for binary classification of anomalous events should be built on a sufficiently large number of samples, taking into consideration the widest possible spectrum of traffic features, so that specific non–stationary properties and hidden transition patterns in traffic can be detected and quantified in a reliable way. In line of principle, the number of samples needed for state space reconstruction is strictly related to the original problem dimension. In order to properly characterize the dynamics associated to the observed time series, we need to adapt the chosen sampling to the phase space in which the dynamical system of interest lies.

As the dimension of the underlying system becomes higher, a greater number of samples are needed. This problem has been discussed in [4] where, starting from simple geometrical considerations, Ruelle determined that if the calculated dimension of our system is well below $2 \cdot \log_{10} N$, where $N$ is the total number of elements in the original time series, then we are using a sufficient number of points.

Of course, having a sufficient number of observations is a necessary but not a sufficient condition for reliable nonlinear time series analysis. We also need a sample that is sufficiently large to include most of the possible periodic features and recurrence phenomena occurring at the various available timescales. Accordingly, we have built our baseline traffic profile working on the 2 weeks trace D (1209600 total samples) that covers satisfactorily some typical traffic features such as the noticeable differences in usage between morning and evening hours, and the differences in usage between weekdays and weekends. We also constructed our SVM training set by combining the two 24 hours A (known positives) and C (anomaly free) traces resulting in 172800 pre-classified samples.

## 5.5 Prerequisites for nonlinear analysis

Once the training set has been chosen to contain a sufficiently large number of samples, needed to completely describe the network traffic characteristics, the previously described nonlinear analysis based on Recurrence Quantification has to be performed on this set to complete the initial "knowledge construction" phase. The quantitative results obtained need to be processed through data mining methodologies provided by the WEKA [5] system, to extract the more selective traffic profile features and generate the anomaly detection criteria used by the SVM-based classifier.

The software products used in our nonlinear time series analysis are the TISEAN 3.0.1 package toolset by Hegger and Schreiber [6] and the RQA v10.1 code developed by Webber and Zbilut [7], freely available on the web, providing a convenient set of command-line utilities, easily suitable to integration with other software for further processing and hence for automated on-line or offline implementation of our anomaly detection paradigm. However, we underline that all the RQA quantities/features of interest would be meaningless if the studied time series did not originate from a system characterized by a certain degree of determinism.

As a preliminary step, we applied the method of surrogate data, described in the previous chapter, to establish if the presence of nonlinear deterministic dependencies suggests us to perform further analysis or whether the involved time series can be directly considered as stochastic.

In doing this we verified the consistency of surrogate data with the null hypothesis (the data can be assumed as generated by a Gaussian linear stochastic process), by considering the nonlinear time series prediction error, and hence we verified that the null hypothesis could be rejected at the 95% level of significance, since the prediction error of all the surrogates exceeds that of the original data. Such a complex and irregular behavior usually implies large fluctuations of intensive quantities on long time scales, which are clear indicators of nonlinearity and (in most cases) non-stationariety.

Thus, in order to justify further steps in nonlinear analysis, we have to verify if the involved baseline traffic time series has the characteristic properties of the deterministic non-stationary signals. Accordingly, our initial assumptions on the non-stationarity of the baseline time series have been verified by observing its *space time separation plot (stp)* [8], reporting the probability that two points in the reconstructed phase-space have distance smaller than $\varepsilon$, (i.e. $|s_i - s_j| < \varepsilon$ as a function of $\varepsilon$ and of the time $t$ elapsed between the points), as percentile iso-lines. To detect this functional dependence we plotted the number of neighbor points as a function of two variables, the spatial distance and the time separation. In detail, we generated an accumulated histogram of spatial distance $\varepsilon$ for each time separation interval $\Delta t$.

We examined the space time separation plot for both the aggregate traffic and some individual traffic flow classes of interest. In these graphics the horizontal axis represented the separation in time whereas the vertical axis represented the separation in space.

For example, the plot in Fig. 5.2, represents the STP for aggregated inter-arrival time on the baseline trace, where we can observe several irregularities in the profile of some components that is a clear symptom of non-stationarity or at least of limited stationarity.

We also show two space-time separation plots, one for specific peer-to-peer traffic classes, namely the aggregated Bittorrent (Fig. 5.3) and EMule (Fig. 5.4) flows. The plot for the other features, e.g. WWW or E-Mail flows, are similar and have been omitted to save space.

In both the graphics, the absence of contour saturation is a clear indication that the time series under analysis is non-stationary. We can also observe that it exhibits significant power in the low frequencies, as for $\frac{1}{f}$ noise, or Brownian motion. In this case, all the points in the series are correlated in time and determining an attractor dimension from the traffic sample is impossible.

Further confirmation has been obtained by examining the space-time separation plot for the surrogate time series whose results are very similar to the ones obtained

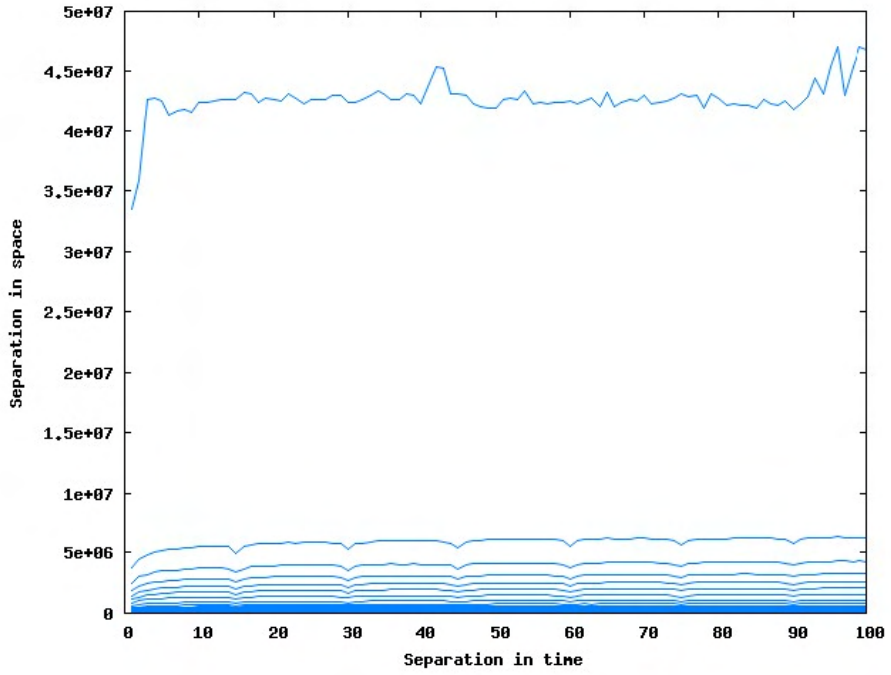



**Figure 5.2**: STP for inter-arrival time, baseline trace.

for the real time series.

To verify the extent to which the differences noticed in the space-time separation plots were intrinsic to the protocols, we ran a cross-correlation test between the two time series. The results (see Fig. 5.5) show a relatively low level of cross-correlation, thus we can conclude that the observed differences are likely to be intrinsic.

After observing the stationarity properties of the baseline time series, we perform a determinism test to verify whether the system behavior is a consequence of deterministic dynamics. For this sake we applied the well known test proposed by Kaplan and Glass [9] evaluating the average directional vectors $k$ observed in a coarse-grained embedding space. In presence of a deterministic system, the estimated average length of all the vectors $k$ will be close to 1, whereas a completely random system is characterized by $k$ vector lengths approaching to 0.

The test on the 2-week baseline aggregated traffic resulted in a determinism value of 0.867, whereas the determinism on the individual peer to peer traffic class series resulted in value of 0.831. Thus, both the evaluations resulted in a quite satisfactory

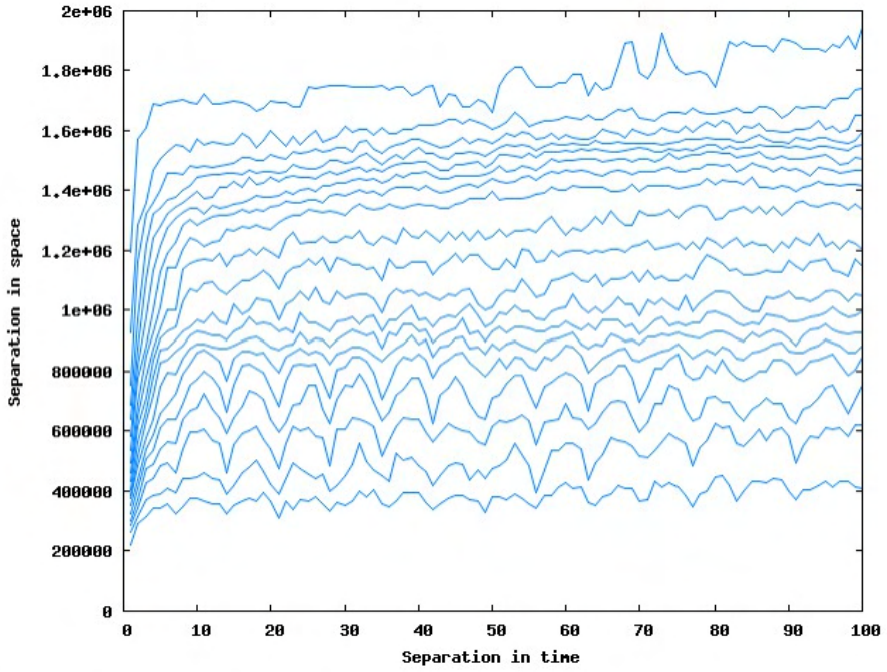



**Figure 5.3**: Space-time separation plot for Bittorrent (packet size), baseline trace.

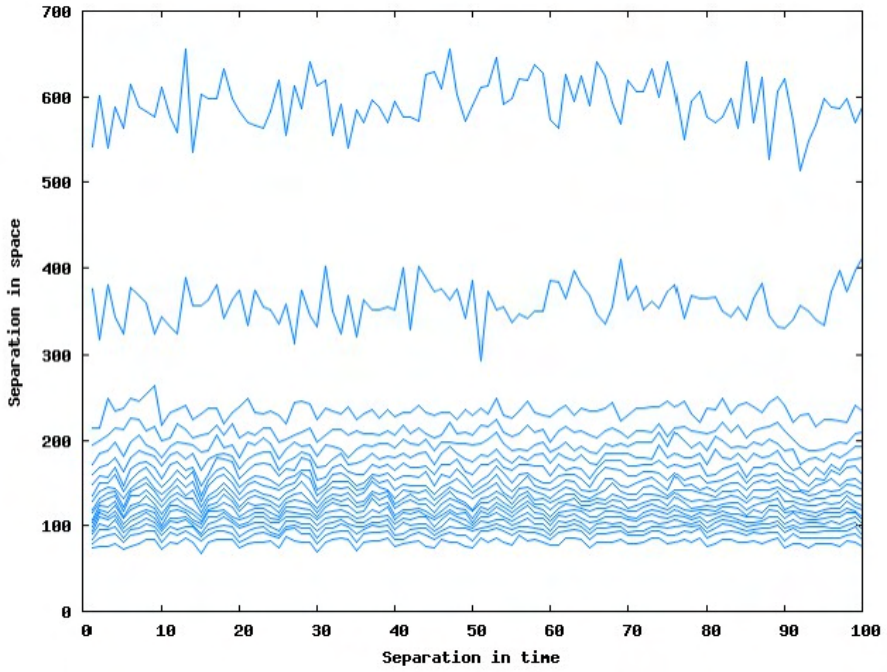



**Figure 5.4**: Space-time separation plot for EMule (packet size), baseline trace.

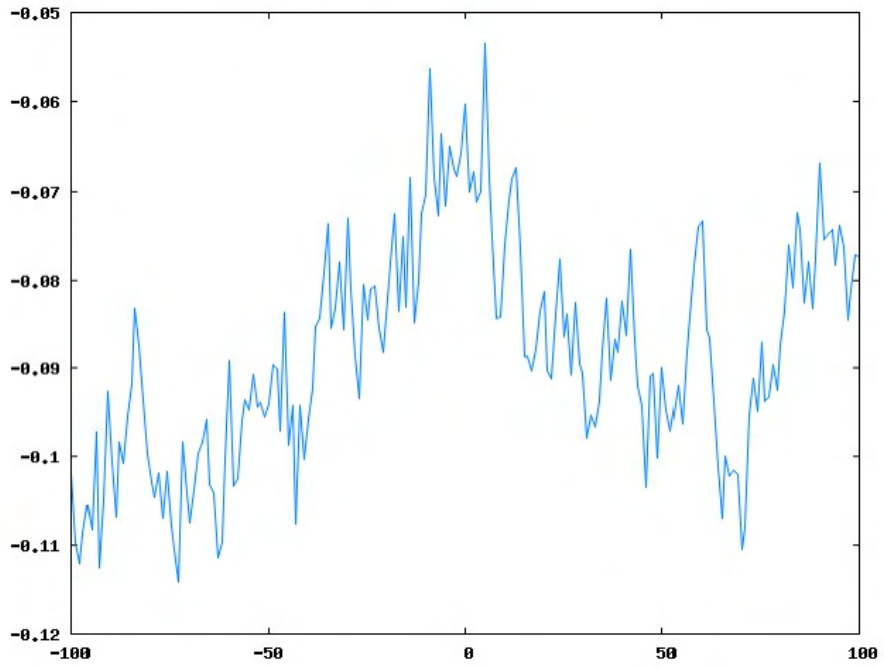



**Figure 5.5**: Cross-correlation between Bittorrent and eMule flows (packet size), baseline trace.

determinism degree.

After establishing that the baseline traffic patterns for both aggregate and individual traffic classes originates from a deterministic non-stationary system, we calculate the maximal Lyapunov exponent [10]. The fundamental idea for the calculation of the maximum exponent is to select a pair of points within the attractor that are sufficiently close in space and observe the evolution in time of their divergence, until they can no longer be considered close. We can see from the stretching factors depicted in Fig. 5.6, relative to the aggregate traffic, and in Fig. 5.7, referring to the peer-to-peer traffic class, that an accurate quantification of the maximum exponent value (estimated approximately as the slope of the straight line, i.e. 1.5) is not simple because of the combination of noise and oscillations.

A positive slope can be appreciated in both the cases, and the estimate is further confirmed by the value 1.46 obtained using another method [14] and therefore a positive exponent is present, which indicates that although the system has finite

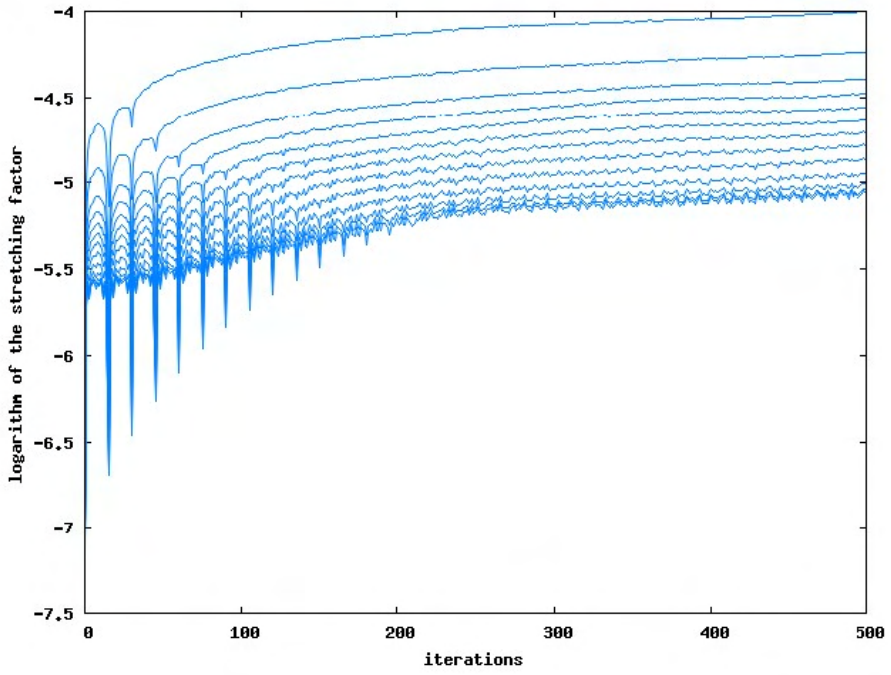



**Figure 5.6**: Logarithmic stretching factors for Lyapunov exponent estimation - aggregate traffic.

degrees of freedom, the time sequence changes dramatically and the system is almost unpredictable. We can thus conclude that the studied time series has the typical properties characterizing deterministic chaotic signals.

The presence of Self-similarity has been assessed by estimating the Hurst exponent for the aggregated baseline, resulting in a value of 0.98, and the individual traffic classes (where the obtained values varied slightly around an average value of 0.88, for the different examined classes). Notably, these values did not change across all the considered originating features (packet sizes, inter-arrival times, etc.). However, the above values are very close to 1, indicating a very high degree of self-similarity, particularly evident with the cumulative traffic trend.

Finally, to explore the presence of long-range dependencies we performed de-

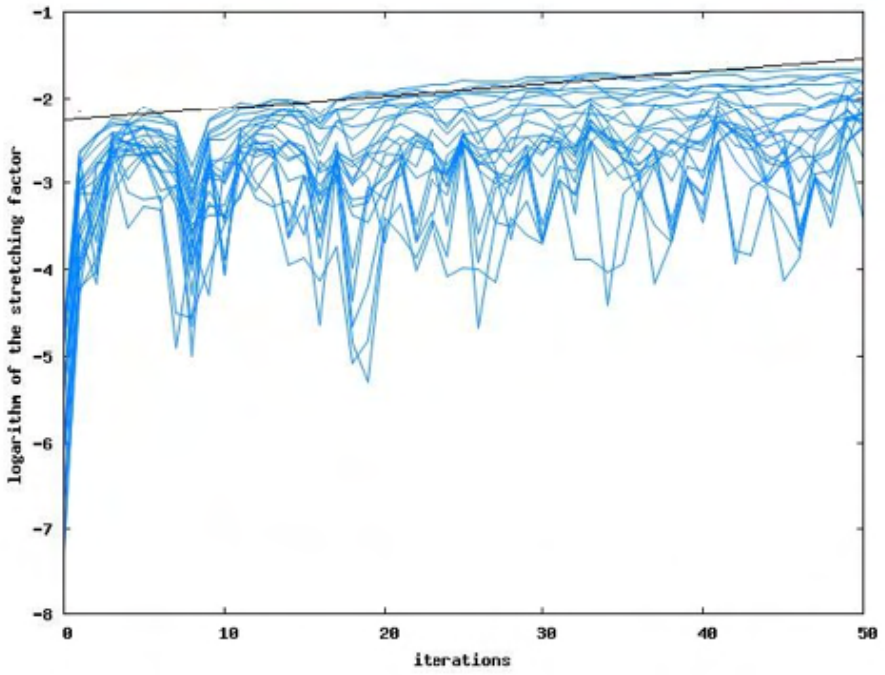



**Figure 5.7**: Logarithmic stretching factors for Lyapunov exponent estimation - P2P traffic class.

trended fluctuation analysis on the available baseline time series. An example log-log plot of the average root-mean-square fluctuation function $F(n)$ varying with the box size $n$ is shown in Fig. 5.8 for the peer-to-peer traffic class. The plot exhibits well-defined trend lines for ranges of $0.6 \leq \log_{10} n \leq 2.5$ for each measurement, with an $\alpha$ value approaching 1 for both the packet and byte rates. It can also be noted that the points on the log-log graph are sufficiently collinear across a wide range of window sizes, that is another clear symptom of self-similarity.

## 5.6 Determining the RQA parameters

The proper choice of the time delay $\tau$, and threshold $\varepsilon$ together with the correct estimation of an embedding dimension $m$, is fundamental for achieving satisfactory results from our analysis. Consequently, a lot of research efforts focused on the

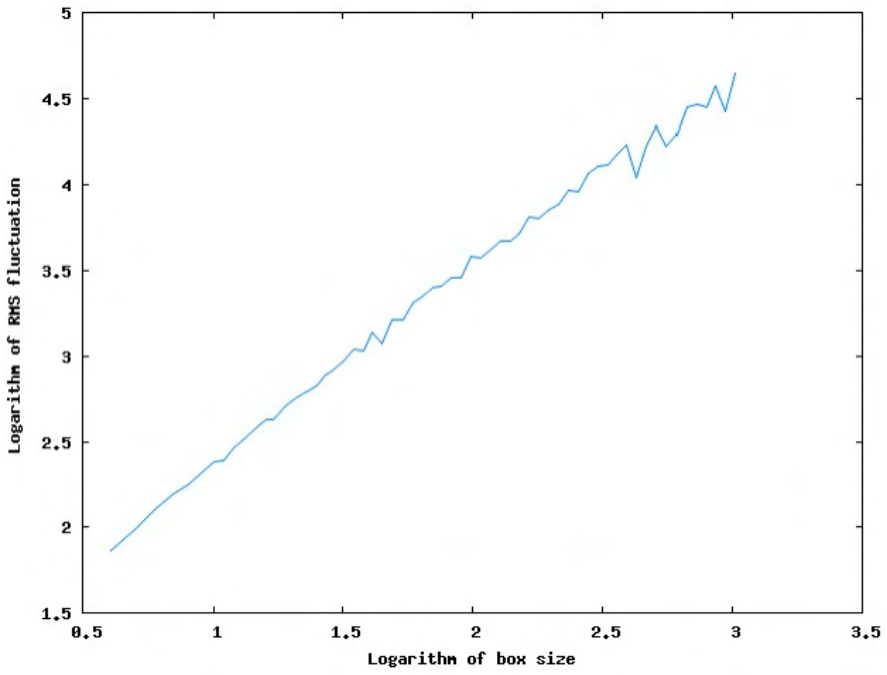



**Figure 5.8**: De-trended fluctuation analysis for P2P traffic class.

problem of choosing the optimal parameter values for the delay coordinates/state space reconstruction process, resulting in the considerations and heuristic estimators presented in the second chapter.

Unfortunately, no rigorous way exists for determining the optimal value of such parameters and, in any case, the interpretation of results produced by the above techniques requires some degree of subjectivity and expertise. Hence, the effectiveness of the final choice is strongly influenced from the analyst's own experience.

For example, if the chosen time delay is too small, there is almost no difference between the delay vector elements, since all points are accumulated around the bisectrix of the embedding space according to a phenomenon called *redundancy* [11]. However, when $\tau$ is too large, the different coordinates become uncorrelated and in this case the reconstructed trajectory can be very complicated, even if the underlying trajectory is simple: this phenomenon is called *irrelevance*. Furthermore, larger values (and hence longer time intervals) induce less sensitivity to changes on shorter time scales.

Analogous considerations can be done for the embedding dimension $m$. Large dimensions introduce to too much requirements in terms of the number of data points and consequently increments the computational time needed for invariants prediction, calculation, etc. Furthermore, since, by definition, noise has an infinite embedding dimension, it will tend to occupy the additional dimensions of the embedding space where no real dynamics are effective and, hence, it will increment the occurrence of errors in all the following calculations. Vice versa, if we select an embedding dimension that is lower than the optimal one, then the underlying dynamics cannot be unfolded, and consequently the calculations will lead us to wrong results since we are not using an effective embedding.

We face similar problems also when choosing the cutoff threshold $\varepsilon$. If $\varepsilon$ is too small, the number of available recurrence points will be insufficient to enable us to reconstruct the underlying system's recurrence structure. On the other hand, if $\varepsilon$ is too large, almost every point will be a neighbor of every other point, and also, points which are only simple consecutive points on the trajectory will tend to be included into the neighborhood. Moreover, in presence of noise we can choose a larger threshold, since noise usually can damage any existing structure in the recurrence matrix and, by using higher thresholds, these structures may be preserved. Nevertheless, the choice is strongly dependent on the particular system under analysis. Hence, we have to find a sustainable compromise also for the $\varepsilon$ value.

Starting from the above considerations, our search for the best $\tau$ and $m$ values has been accomplished by using the TISEAN tools on the entire 2 week baseline. The optimum time delay $\tau$ has been determined as the first one which minimizes the AMI defined in equation (2.7) calculated through the "*mutual*" program, while the embedding dimension $m$ has been selected as the value for which the percentage of false nearest neighbors, calculated with the "*false-nearest*" program, should reach its minimum. In detail, the appropriate embedding dimension has been calculated by

increasing the tentative value for $m$ until no significant change in the percentage of false nearest neighbors is observed. At this particular value of $m$, the attractor has been totally unfolded, and no further information about the system can be gained by exploring higher dimensions.

Finally, to confirm that the chosen values give interesting hints about the underlying dynamics, we also visually inspected phase portraits by looking at the different RPs obtained by incrementally increasing time delay and dimension, verifying that the "critical" parameter values are those at which we could observe marked changes in the diagram structure. Fig. 5.9 details the AMI calculations for the average packet length. The AMI results for the other features have not been shown because they are very similar to those reported below.

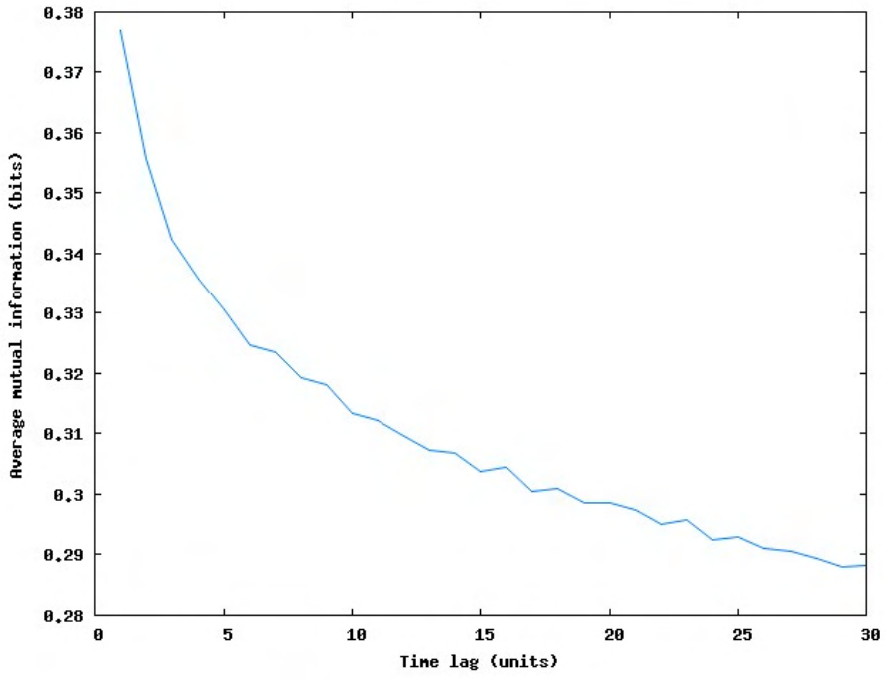



**Figure 5.9**: AMI for average packet length, baseline trace.

From the above figure it can be seen that the AMI value rapidly decreases for all the considered protocols. We can locate the first AMI minimum corresponding to 15 time units, that can be selected as a trustworthy tentative value for the time

delay $\tau$.

Similarly, the baseline FNN plots (one of which is shown in Fig. 5.10), reveal, besides some differences in curve sharpness, a nearly common FNN plateau near dimension 20, thus suggesting the use of that value as a good tentative estimate for the common embedding dimension $m$.

Both the FNN and E1/E2 measurements yielded a value around 20, suggesting that this embedding dimension may be intrinsic in the observed traffic dynamics.

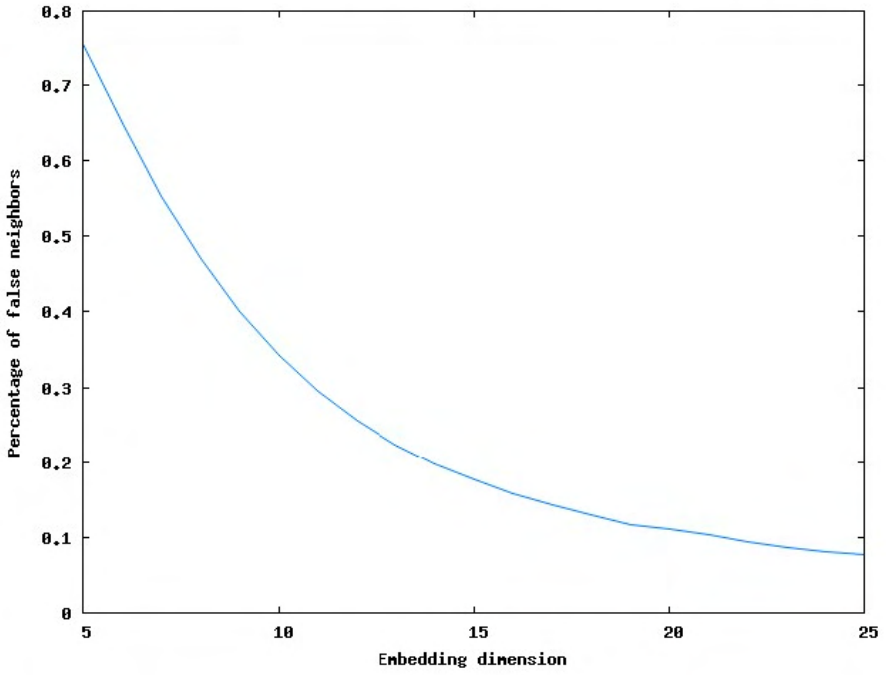



**Figure 5.10**: FNN for average packet length, baseline trace.

The effectiveness of the determined embedding dimensions has also been verified on all the pre-classified training sets by analyzing the corresponding RPs generated by using Visual Recurrence Analysis (VRA) version 5.01 software, freely available on the web. Visual recurrence analysis, performed by observing the occurrence of colors and structures in recurrence plots, is a powerful descriptive technique that revealed to be immediate, sufficiently fast and reliable for analyzing and characterizing at a first glance the traffic time series. It can be considered a satisfactory empirical methodology that can be used to quickly set up and verify some hypotheses in

nonlinear analysis (e.g. embedding parameters choice) that can be successively tested more rigorously with the available RQA tools.

We used ideas from the theory of smooth dynamical systems to identify the type of patterns that the recurrence plots should and should not contain, and we distinguished between acceptable and low-quality embeddings by observing their corresponding plots. Cleaning a recurrence plot from non horizontal patterns is a first step in the determination of the correct embedding parameters, but usually is not sufficient. Two other undesirable features are isolated points, or very short lines and gaps interrupting the observable segments. We would expect then that a "clean" RP such as represents a better reconstruction of the phase space dynamics that can be used for a reliable Quantification Analysis. Thus, once proper time delays and embedding dimension values have been determined, the associated RP reveal slight regularities, that can be clear indications of a deterministic behavior characterizing the underlying system.

The plots reported in the figures 5.6 and 5.6 show some of the RPs corresponding to the most used protocols present in our baseline training samples. A massive presence of hot colors (red, yellow, orange) denotes small distances between vectors. Visual inspection of these plots immediately reveals the presence of small-scale structures. The presence of a very fine-grained organization into the second plot reflects high level of burstiness and randomness.

Finally, according to the "rules of thumb" proposed in [14] we have determined our value for the threshold $\varepsilon$ as the 10% of the maximum phase space diameter.

## 5.7 Recurrence Quantification Analysis for anomaly detection

Once the most suitable embedding dimensions common to all the sampled normal traffic features and interesting flow types have been determined, it is time to perform the quantification measurements and analyze the results in order to determine the

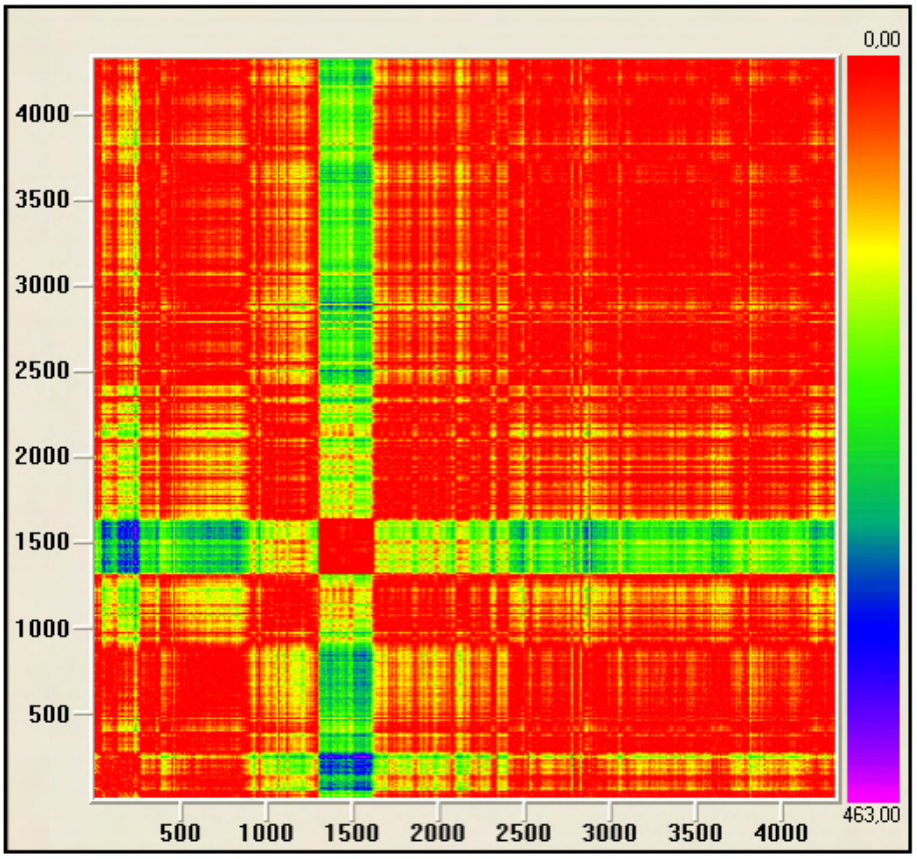



**Figure 5.11**: RP for aggregate P2P traffic, average packet length, baseline trace.

best discriminating properties of each traffic class. Since our main objective is to timely detect variations occurring in traffic patterns, we need to perform RQA within a series of sliding windows instead of analyzing the data as a whole.

Accordingly, the RQA variables have been computed by dividing the whole time series into sub-series and performing recurrence estimation in each sub-interval, defined as an epoch. Such intervals have been regularly shifted so that if $l_e$ is the duration of each epoch and $o_e$ the shift (or offset), the epoch $i$ corresponds in the time series to the interval starting at the time $t_{i-i} = (i-1)o_e + 1$ and ending in $t_i = (i-1)o_e + l_e + 1$. The epoch sizes have been empirically determined according to a compromise between quickness in detecting drifts from normality (shorter epochs) and effectiveness gained from smoothing data (longer epochs).

Precisely, larger epochs focus on global dynamics (longer time frames) and yield

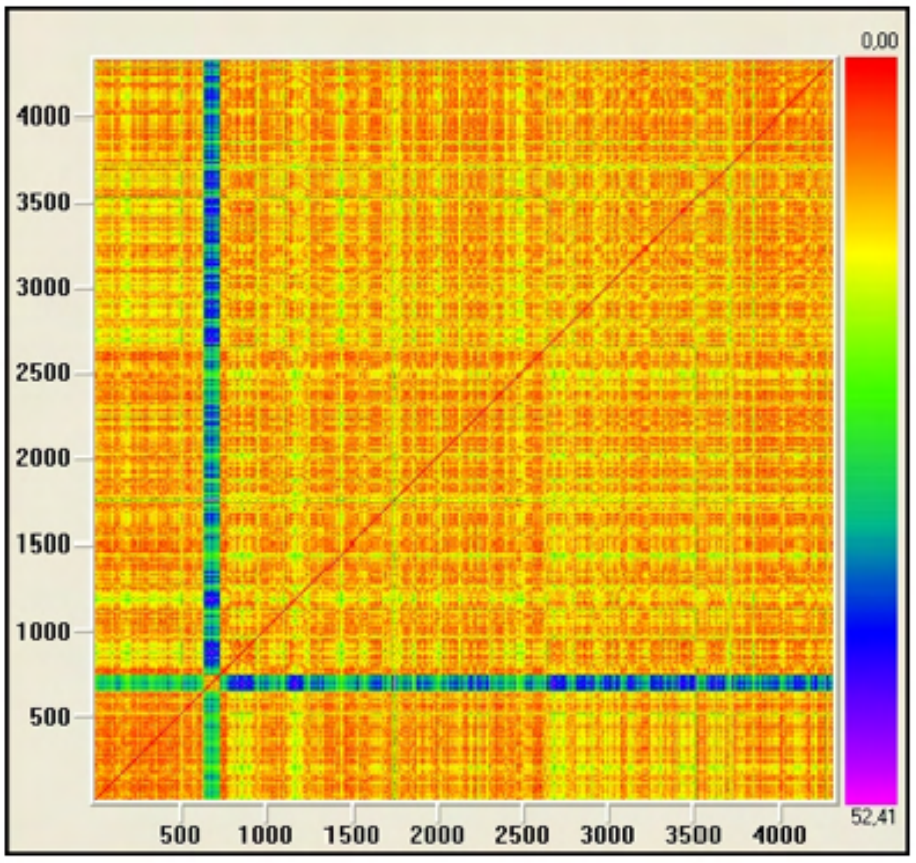



**Figure 5.12**: RP for aggregate DNS traffic, average packet length, baseline trace.

lower time resolution RQA variables whereas smaller ones focus on local dynamics (shorter time frames) and yield higher time-resolution variables. Here, the time shift between consecutive epochs has been empirically chosen to be 25% of the epoch length. Thus in our tests each epoch is 6 minutes long and regularly shifted by 90 seconds, so that each epoch overlaps the next one by 270 seconds.

The calculation for various epochs has the goal of making the state changes more evident within the whole time series. We used in all the calculations the Euclidean Norm rescaled with respect to the maximum value (due to the cutoff threshold being 10% of the maximum).

Two immediate examples of average (on all the epochs) quantification results computed for inter-arrival time variance and packet length by using the baseline data set and differentiating on some interesting groups of pre-classified flows, are

respectively summarized in Tables 5.3 and 5.4.

| | HTTP | P2P | Infrastructure | E-Mail | Terminal Emulation |
|---|---|---|---|---|---|
| REC | 15,464 | 3,764 | 1,804 | 1,414 | 16,501 |
| DET | 35,349 | 8,929 | 2,270 | 1,184 | 85,030 |
| ENT | 3,390 | 2,277 | 1,860 | 1,864 | 5,757 |
| DIV | 0,012 | 0,038 | 0,063 | 0,056 | 0,001 |
| RATIO | 2,286 | 2,430 | 1,259 | 0,837 | 5,154 |
| LAM | 1,746 | 0,009 | 0,000 | 0,000 | 77,358 |
| TREND | -6,940 | 0,351 | -0,107 | 0,063 | -11,213 |

**Table 5.3**: Average RQA measurements for inter-arrival time variance.

| | HTTP | P2P | Infrastructure | E-Mail | Terminal Emulation |
|---|---|---|---|---|---|
| REC | 19,505 | 2,343 | 41,577 | 5,677 | 18,560 |
| DET | 29,359 | 5,005 | 48,092 | 5,329 | 83,578 |
| ENT | 3,115 | 2,046 | 3,621 | 2,018 | 5,957 |
| DIV | 0,023 | 0,045 | 0,017 | 0,053 | 0,001 |
| RATIO | 1,505 | 2,136 | 1,157 | 0,939 | 4,503 |
| LAM | 0,867 | 0,000 | 2,938 | 0,000 | 76,805 |
| TREND | 0,181 | -0,180 | -1,441 | -0,075 | -11,979 |

**Table 5.4**: Average RQA measurements for packet lengths.

Analogously, other highly explicative RQA observations, computed for inter-arrival time and average packet length are, respectively, summarized in figures 5.7, 5.16, 5.7 and 5.18. Interpreting the meaning of these results is central to the problem of defining when an event is an anomaly.

The *Anomalousness* concept can be viewed as a subjective judgement, built within the context of past experience, and can be codified into a "policy", made of a set of rules and criteria, defining what is sufficiently anomalous to guarantee a positive response from the detection logic. Thus, in order to practically characterize what anomalies are, we must have a straightforward classification policy based on these criteria.

From the observation of the RQA results depicted in fig. 5.13 we can immediately evidence how the Recurrence percentage variable is the strongest discriminator for anomalous events since it exhibits noticeable increase in presence of all the simulated anomalies. Here we can observe significant peaks corresponding to denial of service attacks characterized by a high packet rate and an acceptable sensitivity to scans and moderate Denial of Service attacks. This is clearly due to the close relationship between the %REC variable and the correlation integral (or the fractal dimension of the time series) that is known to be capable of characterizing the involved traffic dynamics. Consequently we argue that an anomalous traffic pattern, especially a DoS attack, would be able to change the structure of the correlation integral of the normal traffic, and hence the %REC variable can be leveraged to detect abnormal traffic.

However, recurrence alone is not sufficient for a univocal and reliable interpretation of the complex properties characterizing anomalous events and we need to acquire more information about the deterministic chaotic process describing the involved traffic, from the observation of the distributions of values which are described by characteristic shapes, and thus have characteristic uncertainties.

The value of the entropy variable ENT embodies the above uncertainty information and provides a convenient scalar measure for building classification policies associated to the traffic process. The entropy assumes a minimum value of zero, when all the observations fall into a single class, and reaches its maximum value when they are equally distributed between the available classes. This results in an relative scale that can be adaptively applied to any interval of observation. If we measure entropy as a percentage of the maximum attainable value, then we can

define a threshold located near to the half of the scale which may then be used as a filtering criterion in our classification policy. For the normal time series observation, the entropy sequence has a relative steady fluctuation except for some short occasional events (fig. 5.7 and 5.18). But when the attack starts, the entropy sequence begins a significant uptrend. When the attack finishes, the entropy sequence gets back to steady fluctuation by exhibiting a rapid downtrend.

Finally, the significance of the above measurements is strictly related with a sufficiently high degree of determinism in the time series describing the traffic process, that can be detected from the observation of the %DET variable.

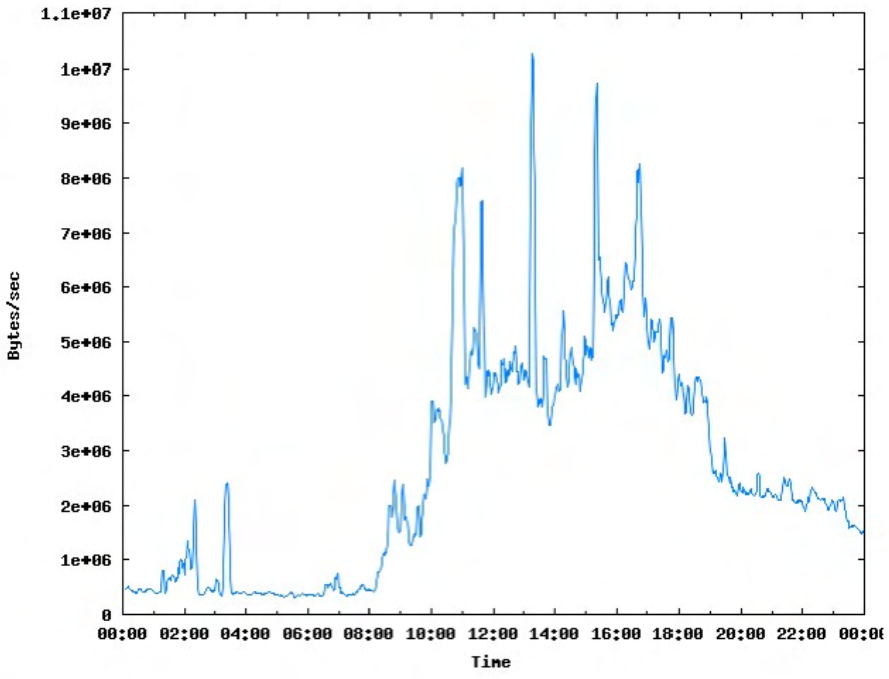



**Figure 5.13**: Average for inter-arrival times, measured in presence of anomalies on trace A.

By comparing the RQA observations in fig. 5.16 and fig. 5.18 with the *average* and *autocorrelation* charts reported in fig. 5.14 we can notice how the RQA features used in our model exhibit a much higher sensitivity in spotting anomalous phenomena respect to traditional linear statistic-based methods.

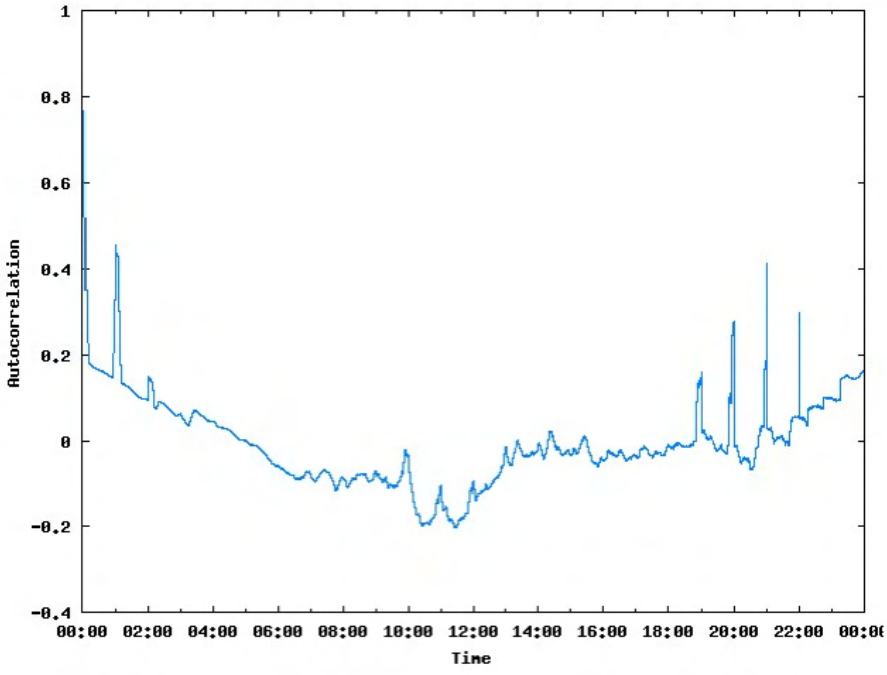
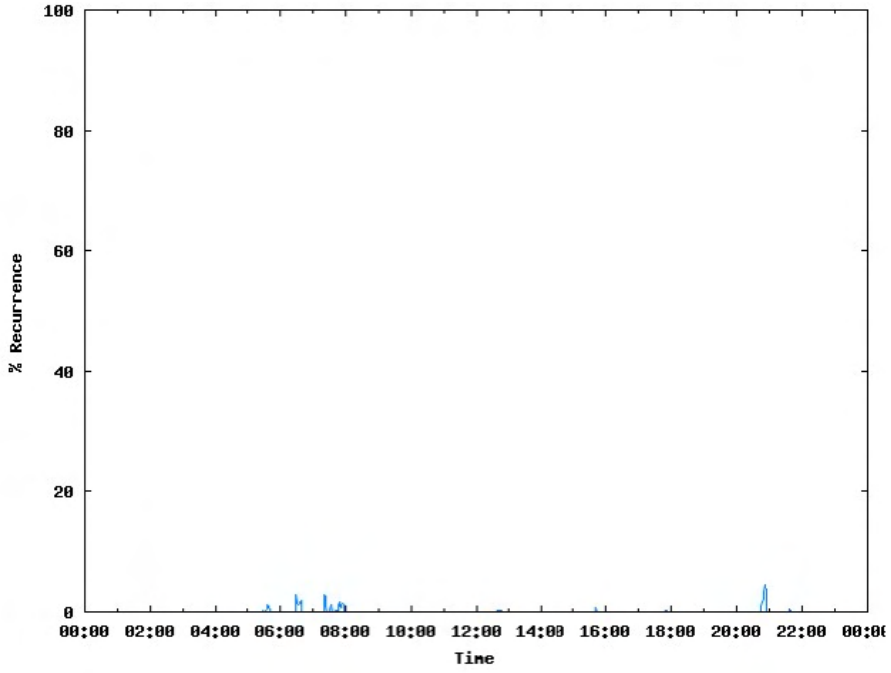



**Figure 5.14**: Autocorrelation coefficients for inter-arrival times, measured in presence of anomalies on trace A.

The averaged (on epoch windows) time series representation gives us a direct and immediate evidence of the aggregate traffic pattern, while the autocorrelation can be used for measuring the regularity degree of the time series by estimating the similarity between the original ones and their associated lag series. Highly regular data will be characterized by lower fluctuations in autocorrelation coefficients.

Unfortunately, we cannot easily identify in both the fig. 5.13 and fig. 5.14 graphs any appreciable track of most of the artificially generated anomalies reported in table 5.2. More precisely, the average chart allows us to reliably detect only the 600s portscan around 13:15 that is however scarcely distinguishable from several other peaks due to normal traffic. In the autocorrelation chart we only can see the two events characterized by the highest packet rate (60s flood at 1:15 and 30s LAND at 22:15). On the other side the combined examination of the %REC and ENT charts in fig. 5.16 and fig. 5.18 gives us a detailed report about all the above events, including those with the lowest packet rate and duration.

Figure 5.15: %REC for inter-arrival times, measured in trace B for anomaly-free traffic.



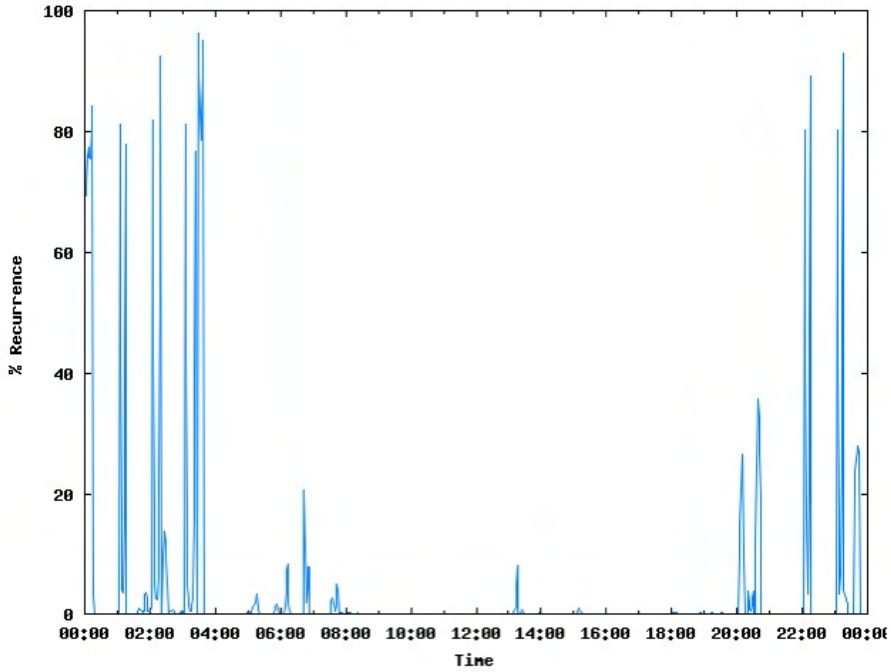
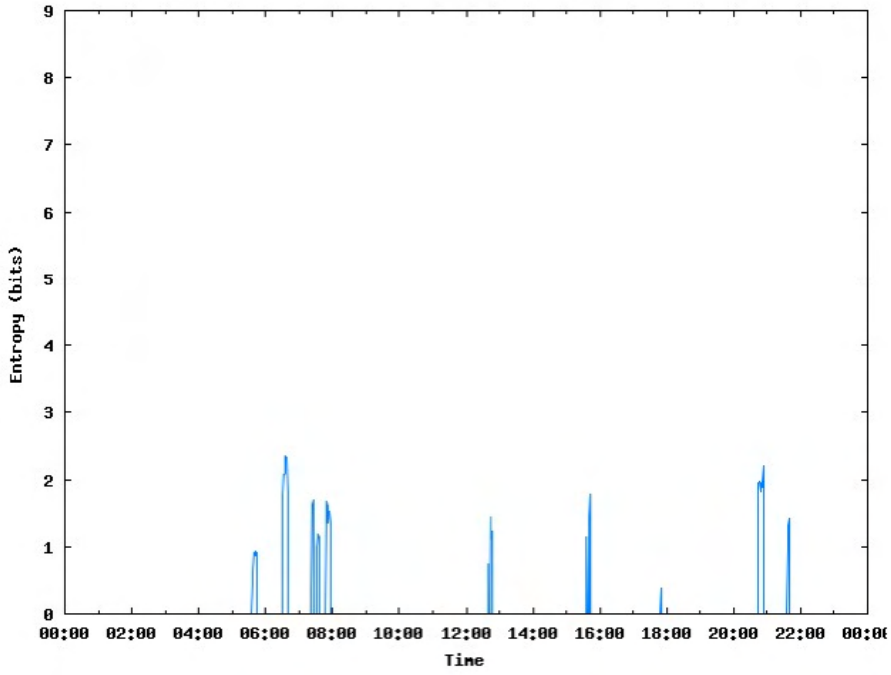




Figure 5.16: %REC for inter-arrival times, measured in presence of anomalies on trace A.

Figure 5.17: ENT for inter-arrival times, measured in trace B for anomaly-free traffic.



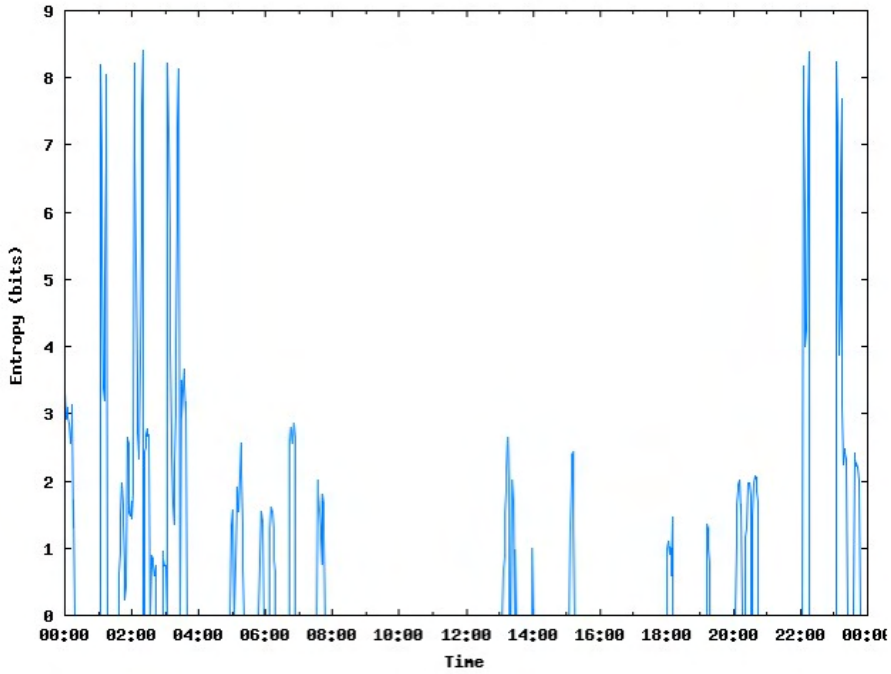




Figure 5.18: ENT for inter-arrival times, measured in presence of anomalies on trace A.

It can be concluded that whereas traditional statistic-based strategies are mainly responsive to sharp variation and more "isolated" changes in traffic pattern (such observations greatly suffer from the aggregation of the high number of components occurring in traffic volume), recurrence quantification is substantially more sensitive. Accordingly, subtle changes from "steady state" occurring in time series data might be delayed or even missed by the former tools, but detected rapidly and effectively by using recurrence-aware techniques.

## 5.8 Wavelet Analysis for anomaly detection

The study of wavelet-based features through multi-resolution analysis has been performed by using the Haar function as the mother wavelet, working with dyadic scales. It can be easily observed that such a simultaneous time and frequency representation of the signal provides easy isolation capability for the local features (shown by peaks and edges on the appropriate scale) exhibited from each signal. This can be extremely useful to discriminate the properties characterizing the interesting traffic (spotted by the absence of peaks or edge-related phenomena over a certain threshold) and consequently can be a strong differentiator, operating simultaneously on multiple scales and hence more immune to noise and transient phenomena, against the other traffic classes.

Some example plots referring to multi-resolution analysis results on WWW and Terminal Emulation traffic classes are respectively reported in Fig. 5.19 and Fig. 5.20 where the differences in traffic properties/features observable on the different isolated components can be easily appreciated.

Wavelet packets can also be used in feature extraction for the purpose of making the binary classification process easier. The Discrete wavelet packet (DWP) decomposition and the associated decomposition tree of an the collection (packet rate) of several highly regular traffic flows (time synchronization, very difficult to be characterized in an aggregate form, by using traditional statistical observations) is

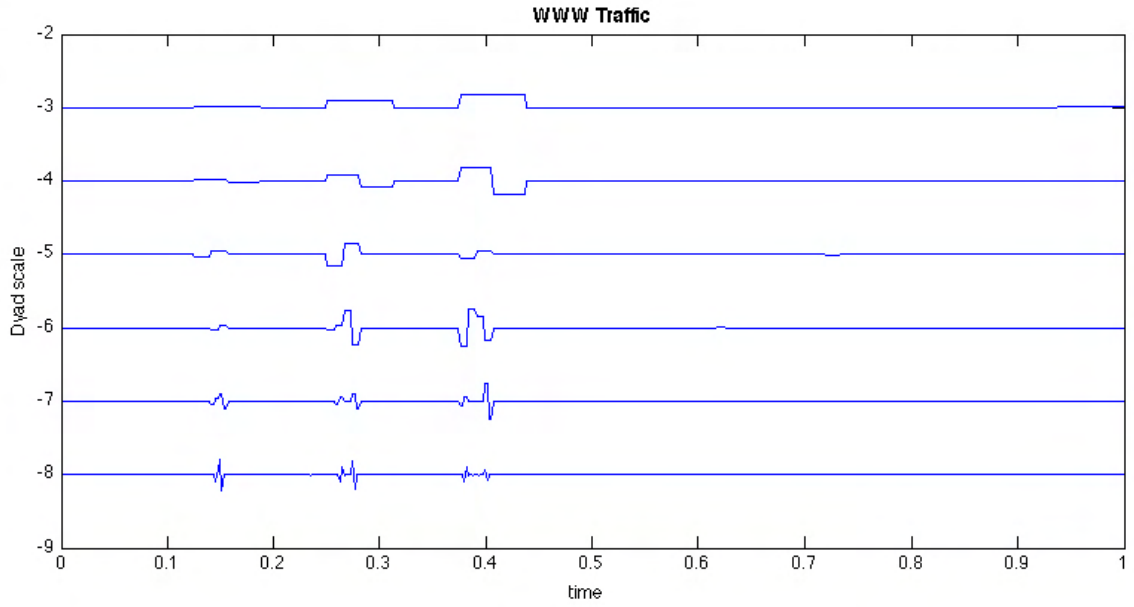



**Figure 5.19**: MRA decomposition plot for WWW traffic flows.

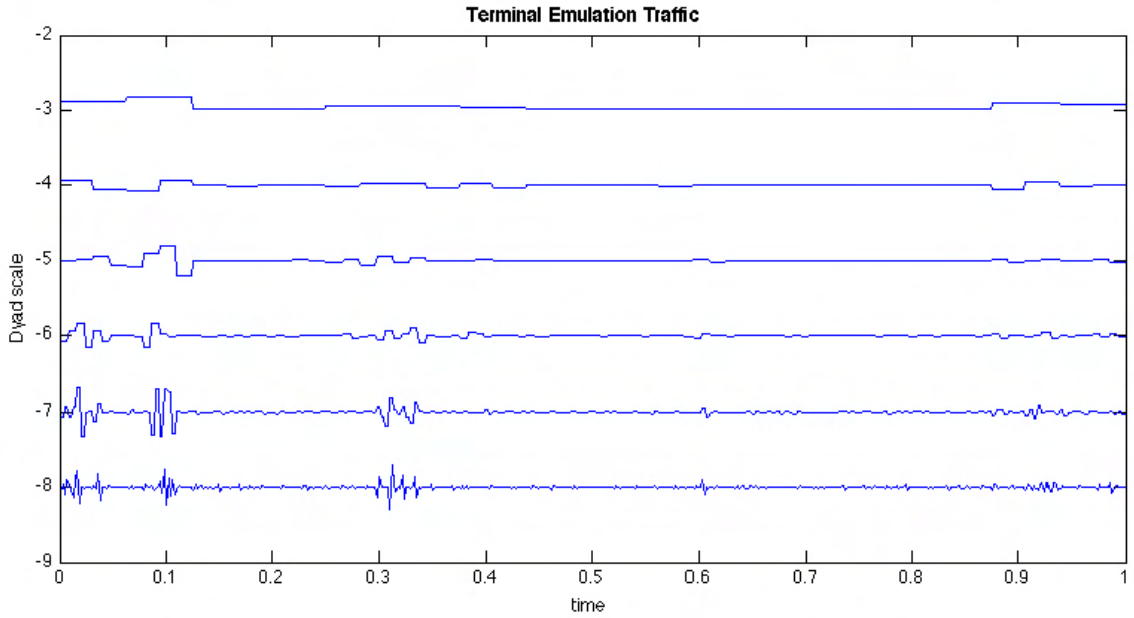



**Figure 5.20**: MRA decomposition plot for terminal emulation traffic flows.

shown in Fig. 5.21 and Fig. 5.22, where a best basis is associated with the most discriminating wavelet. The decomposition graphic also depicts the interesting DWPT properties as a transient hunter, capturing the significant differences in signal contents through the variations between the multiple resolutions, and highlighting that the most characterizing wavelet are those most matching the variations in the data itself.

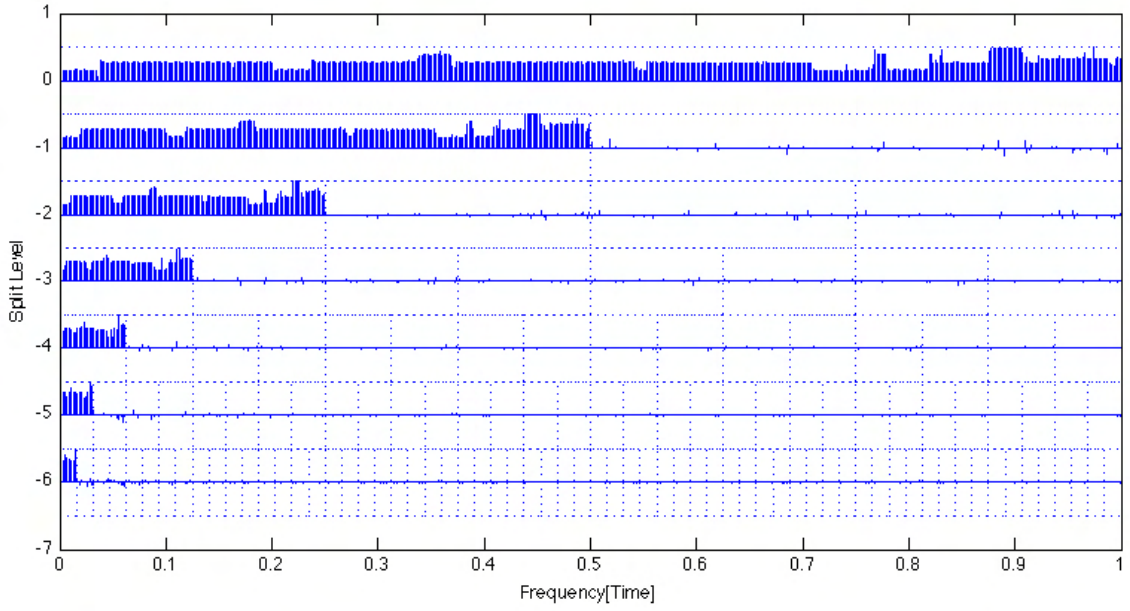



**Figure 5.21**: DWPT decomposition plot for several aggregated synchronization flows.

## 5.9 Building the SVM-based classifier

The construction of the discriminating feature vectors, according to the C4.5-based methodology described in the previous chapter, is the first step in building the knowledge base in our sample anomaly/normal event classification model, to be used in proof-of-concept evaluation.

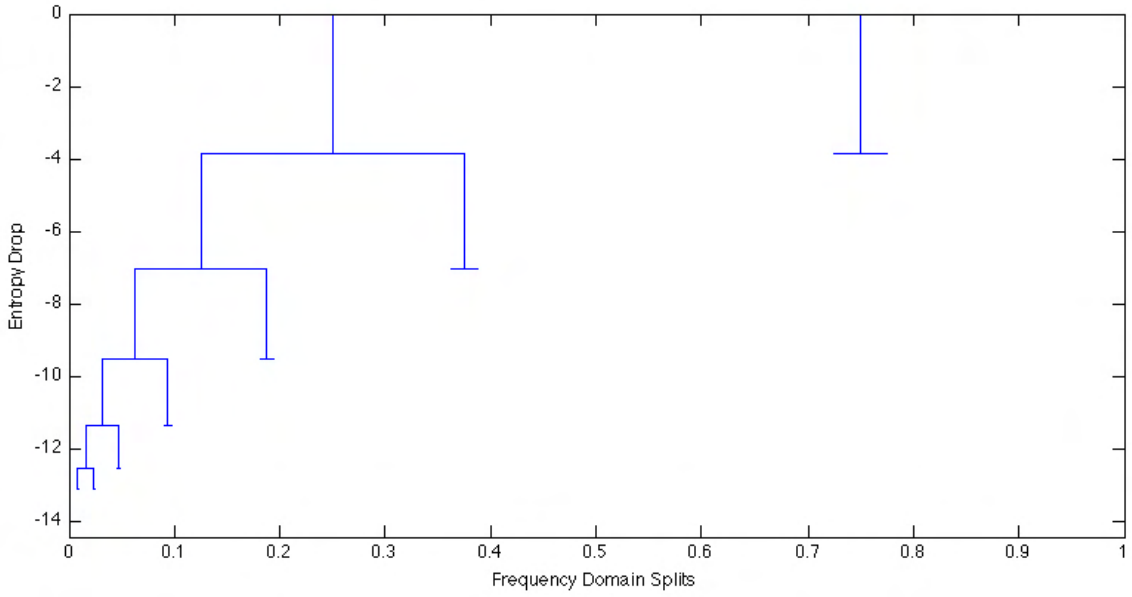



**Figure 5.22**: DWPT decomposition tree for several aggregated synchronization flows.

Here, recognizable traffic anomalies must be represented in terms of some non-linear traffic properties that are useful to distinguish deviations from normal traffic behavior.

Due to the very large number of available features resulting from the combination RQA and wavelets-based measurements the above process may result in an unacceptable complexity and thus, in order to keep our analysis within the scope of a proof of concept, merely demonstrating the effectiveness of the proposed approach, in all the following steps, we restricted our test classification model to only RQA attributes.

Our main objective is to build a decision tree for discriminating anomalous traffic choosing the most promising RQA attribute, enabling us to reliably reveal the deviations from a baseline model, to split on at each point in our decision process and branch accordingly. To do this, we have to search the attribute space for the subset that is most likely to best predict the normal traffic class. Because irrelevant attributes are known to degrade the performance of the classification process, the

RQA attributes considered have been screened, to identify and exclude useless or redundant ones. The final set of attributes selected implicitly generate a different set of cut-off rules, one for every discriminating threshold value.

To determine the most informative features for binary classification, all the RQA features were subjected to selection by determining their mutual information gain (by using the well known InfoGain algorithm, ranking them and selecting the most promising ones for building the final binary classification model implemented through SVMs.

The Information Gain estimation, combined with the ranker method [12] has been implemented by using WEKA and 10-fold cross validation. In 10-fold cross validation (10 is a standard value for the number of folds in Weka), each sample is used exactly once for the sake of validation, and all the available samples are used for both validation and initial training. For each sample in the training set, different models were built for the best 3 and the best 4 features, in addition to the model built using all the features. The attributes resulting in a best InfoGain ranking score are, in order, %REC, ENT and %DET for both the type of basic features (inter-arrival time and average packet length). This confirms our previous observation from the graphs in in figures 5.7, 5.16, 5.7 and 5.18.

The final proof-of-concept classification model has been built with these three best discriminating features, calculated on the all the feature vectors, by using the LIBSVM Support Vector machine implementation [13] for classification and training.

We used, for simplicity sake, a multi-layer perceptron kernel model, where kernel matrix calculations are performed with a *sigmoid* kernel function, originating from neural networks, defined by:

$$k(x, y) = \tanh(\rho \cdot \langle x, y \rangle + \zeta) \tag{5.1}$$

where, $\rho$ and $\zeta$ are parameters of the sigmoid kernel.

The resulting classifier will be used for the performance evaluation experiments reported in the following chapter.

# References


[1] Snort: The open-source network intrusion detection system, http://www.snort.org/.

[2] K. Keys, D. Moore, R. Koga, E. Lagache, M. Tesch, and K. Claffy, "The architecture of the CoralReef: Internet Traffic monitoring software suite", *PAM Conference*, 2001.

[3] F. Strozzi, E.Gutirrez, C. Noc, T. Rossi, M.Serati and J.M. Zaldvar, "Application of non-linear time series analysis techniques to the Nordic spot electricity market data", *LIUC Paper 200*, 2007.

[4] D. Ruelle, "Deterministic chaos: the science and the fiction", *Proc. R. Soc. Lond. A 427*, 241-248, 1990.

[5] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I. H. Witten, "The WEKA Data Mining Software: An Update", SIGKDD Explorations, Vol. 11, Issue 1, 2009.

[6] R. Hegger, H. Kantz, T. Schreiber, "Practical implementation of non linear time series method: TISEAN package", *Chaos* 9 413-435, 1999.

[7] RQA 10.1. http://homepages.luc.edu/~cwebber.

[8] A. Provenzale, L. A. Smith, R. Vio, G. Murante, "Distinguishing between low-dimensional dynamics and randomness in measured time series". *Physica D* 58, 31-49, 1992.

[9] D. T. Kaplan, L. Glass, "Direct test for determinism in a time series", *Physical Review Letters* 68 427–30, 1992.

[10] H. Kantz, "A robust method to estimate the maximal Lyapunov exponent of a time series", *Physical Letters A* 185 77–87, 1994.

[11] M. Casdagli, S. Eubank, J. D. Farmer, J. Gibson, " State space reconstruction in the presence of noise", *Physica D* 51, 52-98, 1991

[12] I. H. Witten, E. Frank, "*Data Mining: Practical Machine Learning Tools and Techniques*", 2nd edition. SF Morgan Kaufmann, 2005.

[13] C. C. Chang, C. J Lin, "LIBSVM – A Library for Support Vector Machines", http://www.csie.ntu.edu.tw/~cjlin/libsvm/, 2001.

[14] N. Marwan, M. C. Romano, M. Thiel, J. Kurths, "Recurrence plots for the analysis of complex systems", *Phys Reports*, 438, pp. 237-329, 2007.

[15] J. Buckheit, S. Chen, D. Donoho, I. Johnstone, and J. Scargle, "About WaveLab" http://www-stat.stanford.edu/wavelab/Wavelab850/AboutWaveLab.pdf, accessed July, 2011.

# Chapter 6

# Experimental evaluation

## 6.1 Scheme Validation and Detection Performance

In this section we present the results of the anomaly detection model's validation and its main performance features evaluated by running the SVM-based binary classifier, built as detailed in the previous paragraphs, on the pre-classified trace "B".

The most significant metrics that can be used to assess the effectiveness and accuracy of our detection scheme are defined by the entries in the confusion matrix (see fig. 6.1), described in the following.

Let us consider a two-class prediction problem (binary classification), in which the outcomes are labeled either as positive ($P$) or negative ($N$) class. There are four possible outcomes from a binary classifier. If the outcome from a prediction is $P$ and the actual value is also $P$, then it is called a true positive ($TP$); otherwise if the actual value is $N$ then we have a false positive ($FP$). Conversely, a true negative ($TN$) has occurred when both the prediction outcome and the actual value are $N$, and a false negative (FN) occurs when the prediction outcome is $N$ while the actual value is $P$.

The *True Positive Rate* ($TPR$) or *Sensitivity*, and the *True Negative Rate* ($TNR$) or *Specificity* represent the percentage of elements that were correctly identified to, respectively, belong or not belong to traffic class $X$ (Anomalous or Regular traffic). The *False Negatives Rate* ($FNR$) is the percentage of the members of a class $X$

classified as not belonging to the class $X$. Correspondingly, the *False Positives Rate* ($FPR$) is the percentage of members of other classes classified as belonging to class $X$, expressing, in some sense, the trustworthiness of the binary classifier in flagging the events as anomalous or not. The ideal confusion matrix is, therefore, a multiple of the identity matrix. A good traffic classifier aims to minimize the FNR and FPR, although the relative importance of each of these metrics heavily depends on the intended use of the classification results. A low FNR guarantees that only a small fraction of class $X$ flows will be discarded, whereas a low FPR means that the set of flows classified as belonging to traffic class $X$ will not contain non-$X$ flows.

| **Classified as** | $X$ | $\overline{X}$ |
|---|---|---|
| $X$ | $TPR$ | $FPR$ |
| $\overline{X}$ | $FNR$ | $TNR$ |

**Table 6.1**: A taxonomy of the accuracy metrics.

The results of the above test, ran on all the 953 feature vector instances corresponding to the 24 hours, are summarized in the confusion matrix reported in table 6.2. To effectively evaluate how well our model works and, at the same time, how and when it fails, it should be considered that the entries of table 6.2 refer to *epochs* and not to *events*. Recalling that epochs overlap, note that longer events are associated to more than one epoch, whereas short events correspond to a single epoch. Also, since an event may span over more than one epoch, not all of which are necessarily flagged as anomalous, we consider an event as individuated if at least one epoch within the event time span is flagged as anomalous.

| **Classified as** | **Regular** | **Anomalous** |
|---|---|---|
| **Regular** | 873 | 32 |
| **Anomalous** | 28 | 20 |

**Table 6.2**: Confusion matrix (on epochs).

Focusing our attention on events rather than on epochs, our classification scheme individuated almost all the anomalous events. However we presented in table 6.2 the confusion matrix details determined on epochs in order to give an insight of the real effectiveness of the detection mechanism and to stress the importance of the epoch size as a parameter determining the minimum duration of detectable phenomena and the detection speed.

Thus, bearing in mind that these numbers are relative to epochs and not to events, the table 6.3 reports the most significant metrics that have been used to assess the effectiveness and accuracy of the proposed technique.

| Metric | Value |
|---|---|
| **Correctly Classified Instances** | 93.704% |
| **Incorrectly Classified Instances** | 6.296% |
| **Precision (not anomalous traffic)** | 0.969 |
| **Recall (not anomalous traffic)** | 0.965 |
| **Mean absolute error** | 0.063 |
| **Root mean squared error** | 0.251 |
| **Kappa statistic** | 0.367 |

**Table 6.3**: The detection performance metrics (on epochs)

The first four metrics are directly related to the classifier's accuracy measuring the percentage of correct classifications with respect to the overall data and to the associated classification errors.

The accuracy metric ($\frac{tp+tn}{tp+tn+fp+fn}$) takes into account both positive and negative instances by paying equal attention to all the types of error.

The recall ($\frac{tp}{tp+fn}$) and precision ($\frac{tp}{tp+fp}$) scores indicate, respectively, the errors which are caused by classifying positive instances as being negative and the errors which are caused by classifying negative instances as being positive.

We can observe a significant classification accuracy associated to a very high

precision in identifying traffic that is not affected by anomalies. On the other side we can see from the confusion matrix a limited precision in identifying with absolute certainty epochs that correspond to anomalous events. This is a common problem in volume-based anomaly detection systems, often characterized by low detection efficiency for positive events, and is amplified by the number of overlapping epochs to which an event may correspond. However, this can be not considered a real problem in our solution, since almost all anomalous events are recognized as such, while at the same time the system shows a high efficiency in identifying traffic patterns that can be considered "normal", and we are essentially interested in distinguishing the occurrence of suspicious events (and eventually flagging them for further analysis) deviating from the "normal" or baseline traffic behavior.

Finally we also analyzed the Kappa statistics as an alternative to the traditional accuracy metrics for evaluating our classifier. In machine learning, Kappa is used as a measure to assess the improvement of a classifier's accuracy over a predictor employing chance as its guide. The Kappa coefficient has a range between -1 and 1, where -1 corresponds to total disagreement (i.e., total misclassification) and 1 to perfect agreement (i.e., a 100% accurate classification). Usually, a kappa score near to 0.4 indicates a reasonable agreement beyond chance.

## 6.2 Results comparison

Comparing our results with alternative techniques is not immediate, in the absence of a general framework for assessment and validation. The constantly changing nature of real network traffic prejudices the isolation of aspects of the "anomalous" behavior so that is very difficult to build a common reference framework to be universally used for classification and comparison. To start with, publicly available data sets and taxonomies for benchmarking anomaly detection systems are generally considered to be scarcely significant and error-prone.

For instance, the well-known DARPA [1] data sets, although somewhat used in the earliest works in literature, have been harshly criticized [2] for the usage of syn-

thetic simulated background data not containing any form of noise (packet storms, strange fragments) that usually characterize the real data. That's worse DARPA data do not refer to complete weekly or monthly traffic periods, but only contains specific days (Monday to Friday) and time intervals (8am – 6am of the next day). These data cannot thus be simply concatenated to reconstruct a complete traffic view on a sufficiently large timescale, that is a fundamental prerequisite to exploit the real strengths of the proposed non-linear detection model.

Whereas, until now, no technique has emerged as a reference standard that can be used for the assessment of reliable result and since the various results available in literature cannot be easily compared with each other, we performed a tentative comparison on the final accuracy of the results achieved by our scheme and those obtained on the DARPA 1999 data set (since these are the only reference data available) by some distinguished anomaly detection methods such as the k-Nearest Neighbor outlier mining algorithm (K-NN), fixed-width clustering (Cluster), support vector machines (SVM), the modified clustering-TV algorithm (Modified Cluster-TV), pfMAFIA, and the HDG-Clustering algorithms [3][4][5][6].

Accordingly, as a reference for the comparison, we collected a set of performance results available in literature [4] for the above alternative approaches, concerning the area under the Receiver Operator Characteristic (ROC) curve that can be viewed as a good approximate measure of accuracy. These results are comparable, since all the above techniques have been evaluated by running them against approximately the same data sets (that are however different from ours), and are reported in table 6.4.

The ROC curve is a graphical plot of the sensitivity, or true positive rate, vs. false positive rate ($1 - TNR$), for a binary classifier system as its discrimination threshold is varied. In more detail, the ROC space is defined by using $FPR$ and $TPR$ as $x$ and $y$ axes respectively and each prediction result or single instance of the confusion matrix represents a point in such space. The Area Under the ROC Curve corresponds to the probability that a classifier will rank a randomly chosen

positive instance higher than a randomly chosen negative one. It can be shown that the area under the ROC curve can indicate whether positives are ranked higher than negatives.

| Technique | Value |
|---|---|
| **K-NN** | 0.895 |
| **Fixed-width Clustering** | 0.940 |
| **SVM** | 0.949 |
| **Modified Clustering-TV** | 0.973 |
| **pfMAFIA** | 0.867 |
| **HDG-Clustering** | 0.976 |

**Table 6.4**: Area under the ROC curve (DARPA data set).

Since we have chosen to base our performance evaluation on real traffic data, and as we do not have access to a detailed implementation for all of the above listed techniques, we may refer to the data in table 6.4 to establish an estimated ranking between the different methods. Since our approach has been evaluated on real data, that is clearly far more complex from the artificially generated DARPA ones, we can consider such comparison as a worst case performance analysis.

We can then compare the ROC curve results evaluated on our datasets (0.691), and hence relative to the proposed technique with those reported in table 6.4.

The ROC data are generally lower than the reference values, since they have been obtained on real traffic, therefore under different, and much harder, conditions. By the same reason, the significance of these results is much higher.

Furthermore, the comparison is not so straightforward, because also the measurement units used are different: the other data are expressed in terms of connections or flows, ours is based on the evolution over time of the aggregate traffic data, segmented in epochs, as already explained in the previous section. Therefore, there are

two fundamental differences. First, for a single detected event spanning multiple overlapping epochs, some of these epochs may not be flagged as anomalous due to temporal shift and timescale superposition effects. Second, a connection is either flagged as anomalous or it deemed normal, and this classification holds for the connection in its entirety: a connection that behaves normally for 95% of its duration and is anomalous for the remaining 5% is completely anomalous. As we are interested in the instantaneous evolution of aggregate traffic over time, we preferred to take a more time-oriented approach.

As already evidenced previously, our volume-based approach is characterized by a high efficiency in identifying the "normal" traffic and a more limited precision in defining with absolute certainty the presence of an anomalous event. In fact, the constantly changing nature of real network traffic prejudices the isolation of aspects of the "anomalous" behavior so that is very difficult to build a common reference framework to be universally used for classification and comparison. However, we believe that this work addresses also the last issue, since non-linear characteristics should be closer to the inherent dynamics of traffic, and hence less amenable to fluctuations.

In conclusion, even though neither ours nor the other available performance results have been obtained within a standard validation framework, given the empirical run time results analysis performed on our nonlinear schema, together with the above comparison and observed performance metrics we can consider that the proposed non-linear detection framework can be seen as reasonably successful and effective in detecting deviation from normal behavior, also with respect to the other existing techniques, and hence is a promising subject for further research.

# References


[1] R. Lippmann, J. Haines, D. Fried, J. Korba and K. Das, "Analysis and results of the 1999 DARPA off-line intrusion detection evaluation", *Computer Networks* 34 (4), pp. 579–595, 2000.

[2] J. McHugh, "The 1998 Lincoln Laboratory IDS Evaluation (A Critique)", *Proceedings of the Recent Advances in Intrusion Detection*, pp. 145-161, Toulouse, France, 2000.

[3] E. Eskin, A. Arnold, M. Prerau, L. Portnoy, S. Stolfo, "A geometric framework for unsupervised anomaly detection: Detecting intrusions in unlabeled data", in *Applications of Data Mining in Computer Security*, 2002.

[4] J. Oldmeadow, S. Ravinutala, C. Leckie, "Adaptive clustering for network intrusion detection", *Proceedings of the Third International Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 2004.

[5] K. Leung, C. Leckie, "Unsupervised anomaly detection in network intrusion detection using clusters", *Proceedings of the 28th Australasian conference on Computer Science*, Vol. 38 pp. 333-342, 2005.

[6] C.F. Tsai, C.C. Yen, "Unsupervised anomaly detection using HDG-Clustering algorithm", *Lecture Notes in Computer Science*, Vol. 4985 pp. 356-365, 2008.

# Chapter 7

# Conclusions and Further Research

Identifying anomalous events is an efficient way to discover a lot of existing malfunctions and handle most of the security and performance problems that may occur in the network. Enhancing the detection capability is a fundamental step to improve the network availability and ensure the required quality of service. We presented a new supervised machine learning approach to anomaly detection, based on the combination of recurrence and wavelet analysis and performed a detailed study of the nonlinear dynamics of network traffic behavior, to observe recurrence phenomena and hidden non-stationary transition patterns in the time series associated to different phenomena that we would like to detect. The experimental evaluation of machine learning techniques to support nonlinear methods in network anomaly detection indicates their potential and highlights the areas where improvement is required.

The non-stationary nature of the network measurements impose that the algorithms used for detection and classification must able to dynamically and efficiently adapt over time.

The rich set of features that can be simultaneously evaluated, associated to cumulative traffic trends or related to the variations in each traffic class or host aggregate, introduces great control granularity in the detection process by adding multiple points of observation, and hence new dimensions in the feature space, that

can ease correlation and inference activities in the machine-learning based binary classification process.

Having multiple different observations associated to the individual traffic components may also be helpful in spotting and describing the nature and behavior of the observed anomalous phenomena (e.g. protocols affected, transport facilities used, traffic volumes distribution, etc.). This last issue can reveal to be of fundamental importance in the development of countermeasures or reaction strategies, that can be a very interesting subject for further research.

The results show that nonlinear techniques such as RQA or multi-resolution wavelet analysis can be valuable for gaining insights into the hidden statistical characteristics of network traffic, and that those techniques can, together with SVMs machine-learning aptitudes, be reliably used for anomaly detection.

Because both these techniques have been formerly conceived for nonlinear analysis and chaos theory, they naturally demonstrate to be particularly effective for traffic flow time series, due to the inherent fractal behavior of network traffic data.

Besides, by leveraging on fundamental "hidden" nonlinear dynamics, the approach is promisingly more robust against elusion mechanisms.

Direction open to further investigation include the use of Cross Recurrence Quantification [1] analysis as a nonlinear method for measuring the degree of coupling between multiple combined traffic features. We also are looking for ways to extend out analysis to a subset of non-noisy anomalous events, namely those involving perceivable traffic volumes characterized by patterns that recur over time in a way that significantly diverges from ordinary user activity.

Other future research directions originate from the consideration that different wavelet basis functions could be applied for extracting more sophisticated traffic features, that is, analyzing whether the different wavelet families (e.g a Coifflet better than a Daubechies) have different or more specific sensitivity to specific classes of

anomalous events. Additionally, multiple wavelet functions could be employed in parallel, to extend the feature space and the overall discriminating power.

# References

[1] N. Marwan and J. Kurths, "Nonlinear analysis of bivariate data with cross recurrence plots", *Phys. Lett. A*, vol. 302, pp. 299–307, 2002.

# Appendix A

# Some data manipulation templates

## A.1 The Coral `t2convert` pre-classification table

The figure A.1 reports the `t2convert` classification table, used for naive port and protocol-based aggregation of the traffic flows into sub-component classes to be used in building feature vectors.

## A.2 The `weka` arff header/template

The figure A.2 reports an example WEKA arff header/template that can be used for features ranking between RQA and Wavelet-based estimators

```
description:    World Wide Web traffic
name:   WWW  group:  WWW
sport:  80,70,443,1080,3218,8080
dport:  *
sym:    1  protocol:       6
# ------------------------------------------------------------
description:    Unfrastructure services (DNS, routing etc.)
name:   Network_Infrastructure  group:  Network_Infrastructure
sport:  3,43,53,67,68,113,389,689,111,161-162,123,179,520,427,135,137-139,445,568,569,1512,33434-33524
dport:  *
sym:    1  protocol:       1,6,17
# ------------------------------------------------------------
description:    File Transfer traffic (data stream)
name:   File_Transfer  group:  File_Transfer
sport:  20,989,21,990,2811,5020-5022,5031-5039,873
dport:  *
sym:    1  protocol:       6
# ------------------------------------------------------------
description:    Mail and News traffic
name:   Mail/News  group:  Mail/News
sport:  25,465,587,109-110,995,143,220,585,993,119,563
dport:  *
sym:    1  protocol:       6
# ------------------------------------------------------------
description:    Telnet, rloginn, ssh, etc.
name:   Terminal_Emulation  group:  Terminal_Emulation
sport:  22,23,992,512,513,514,3389,5800-5806,5900-5906
dport:  *
sym:    1  protocol:       6
# ------------------------------------------------------------
description:    Video and Audio streaming services
name:   Streaming  group:  Streaming
sport:  7070-7071,6970-7170,554,8554,1935,18888,18889,1755,7007,8000,9000,2048,7070,8080,2050,
1972,2047-2048,3689,90,4000-4100,4500,9000-9100
dport:  *
sym:    1  protocol:       6,17
# ------------------------------------------------------------
description:    Online Chat traffic
name:   Chat  group:  Chat
sport:  522,1503,1720,1731,194,529,994,6665-6669,7000,33033,2001-2120,6801,6901
dport:  *
sym:    1  protocol:       6,17
# ------------------------------------------------------------
description:    Peer to peer traffic servers
name:   P2P group:  P2P
sport:  411-412,1214,1337,1863,2234,2705,3531,4444,4661-4662,4665,4672,4711-4712,5335,5500-5501,5555,6257,
6346-6347,6574,6666,6677,6688,6697,6699,6699,6881-6999,7128,7144,7243,7244,7674-7675,7777,8038,8090-8091,
8311,8888,9493,10376,10377,22321-22322,32285
dport:  *
sym:    1  protocol:       6,17
# ------------------------------------------------------------
description:    Catchall
name:   Other group:  Other
sport:  *
dport:  *
sym:    1  protocol:       *
```

**Figure A.1**: The Coral `t2convert` pre-classification table

```
@relation anomaly

@attribute rec NUMERIC
@attribute det NUMERIC
@attribute ent NUMERIC
@attribute trend NUMERIC
@attribute lam NUMERIC
@attribute wle NUMERIC
@attribute wme NUMERIC
@attribute whe NUMERIC
@attribute wlk NUMERIC
@attribute wmk NUMERIC
@attribute whk NUMERIC
@attribute label { NO, YES }

@data
0.012,0.000,-1.000,-0.126,0.000,0.000,0.000,0.000,0.000,0.000,0.000,NO
0.009,0.000,-1.000,-0.095,0.000,0.000,0.000,0.000,0.000,0.000,0.000,NO
0.003,0.000,-1.000,-0.032,0.000,0.000,0.000,0.000,0.000,0.000,0.000,NO
... ... ...
```

**Figure A.2**: The WEKA arff header