Network anomaly detection through nonlinear analysis

Francesco Palmieri*, Ugo Fiore

Università degli Studi di Napoli Federico II, CSI, Complesso Universitario Monte S. Angelo, Via Cinthia, 80126, Napoli, Italy

Abstract

Nowadays every network is susceptible on a daily basis to a significant number of different threats and attacks both from the inside and outside world. Some attacks only exploit system vulnerabilities and their traffic pattern is undistinguishable from normal behavior, but in many cases the attack mechanisms combine protocol or OS tampering activity with a specific traffic pattern having its own particular characteristics. Since these traffic anomalies are now conceived as 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. In this work we present a novel approach to network-based anomaly detection based on the analysis of 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 recurrence quantification analysis, a nonlinear technique widely used in many science fields to explore the hidden dynamics and time correlations of statistical time series. Our model demonstrated to be effective for providing a deterministic interpretation of recurrence 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.

1. Introduction

Anomaly detection is becoming an increasingly vital component of any network security infrastructure. Network anomalies, circumstances when network behavior deviates from its normal operational baseline, can arise due to various causes such as malfunctioning network devices, network overload, malicious activity, denial of service (DoS) attacks and network intrusions that disrupt the normal delivery of network services. Detecting, identifying and classifying these operational hazards is very important but at the same time hard. First of all, the definition of normal network behavior depends on several factors relating to the day to day business operations. These include traffic volume, applications running on the network, and the data they process. The goal of anomaly detection is to devise techniques that will model what a normal working network should look like and report any deviations from that definition. These techniques are typically based on machine learning, data mining or statistical analysis. The ultimate aim of anomaly detection systems is to achieve an adaptive behavior that responds in "real-time", so that problematical events can be countered as quickly as possible. However, normal behavior can only be determined by learning about past events: trends take time to learn and analyze. This paradox can only be resolved by modeling future behavior, based on a statistical idealization of the past and an observation of the present (like weather forecasting) and by specifically analyzing and observing some particularly
discriminating statistical features and evolutive phenomena on the network traffic. A timely response requires rapid processing of observations, and those typically originate from network monitoring devices collecting data at high rates. Consequently, designing an effective anomaly detection system involves extracting relevant information from a voluminous amount of noisy, high-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 tampering activities and specific communication patterns 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 match stored signatures or known communication behavior. 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, and the results are compared against constant pre-assigned threshold values that are often independent from the current network utilization and the number of users. Instances where the threshold is crossed are flagged as anomalous. The trouble with existing anomaly detectors is that they take a rather narrow view of what an anomaly means, that is, they rely only on network traffic observations over a short time scale, to “explain away” observed anomalies. More specifically, the basic observation that makes traditional time-series analysis unreliable for anomaly detection is that there is an inhomogeneous pattern to human/computer resource usage, and this is clearly reflected in network resource usage. Network traffic features and hence the characteristics of probability distributions of their IP-layer packets change dynamically in the time domain (Tretyakov et al., 1998; Takayasu et al., 2000) in relation to specific network conditions. It follows that, since power laws apply to changes in traffic density, the traffic statistical characteristics change with the phase transition patterns and that their fractal-like behaviors can be affected by the packet density and its time-variation trend (Masugi and Takuma, 2007). Such a complex dynamic traffic system possesses several, almost hidden, regularities and specific statistic features that are not influenced by noise or events observable on a short time scale and hence can be more effective for investigation. Common ground between such features resides in their shared recurrence properties. That is, 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. In this paper, we propose a novel anomaly detection scheme, particularly suitable for IP networks, 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 time series. Our scheme is based on recurrence quantification analysis (RQA) (Zbilut, 1994; Marwan and Kurths, 2002) that has been used to observe and study the above non-stationary properties in such transition patterns. Such properties, and more precisely those associated to the most informative RQA descriptors, have been used as inputs to a machine learning process, driven by support vector machines (SVM) to classify anomalous events. We demonstrated the possibility of a pure operational use of concepts and techniques derived by complex systems dynamics for developing deterministic qualitative and quantitative observations that can be reliably used in detecting anomalous events characterized by measurable variations in the statistical properties of the traffic time series. Our approach differentiates from the majority of the volume-based anomaly detection schemes by its peculiar theoretical perspective: we chose a method that does not make any specific assumption on the mathematical structure of data, does not rely on assumptions of stationarity and does not need to consider the studied traffic data as the output of a linear dynamical system.

2. Related work

Anomaly detection has been studied widely and has received an increasing attention in the last years. 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 (Brutlag, 2000), adaptive thresholding, cumulative sum (Siris and Papagalou, 2004a; Blazek et al., 2001), maximum entropy estimation (Gu et al., 2005), and principal component analysis (Lakhina et al., 2004). Some of these works analyze the volume of aggregate traffic on a network link, 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 (Spade and Defense, ), ADAM (Barbar et al., 2001) and NIDES (Anderson et al., 1995) 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 (Mahoney and Chan, 2001), ALAD (Mahoney and Chan, 2002a), LEARD (Mahoney and Chan, 2002b) and NETAD (Mahoney, 2003) monitor a larger set of fields of the packet header and use more complex time-based models, in which the probability of an event depends on the time elapsed since its last occurrence. In (Talpade et al., 1999) is presented an anomaly detection framework based on path changing, packet delay and statistic inference on the packet header information called NOMAD. The (Siris and Papagalou, 2004b) 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 (Siris and Papagalou, 2004b; Garcia-Teodoro et al., 2009). In (Dainotti et al., 2006) 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 (Shin et al., 2005), the authors have proposed monitoring the number of the SYN packets and the change of the ratio of SYN packets to other type TCP packets to determine whether anomalies have taken place. In (Cheng et al., 2002), spectral analysis is used to identify legitimate TCP flows, which should exhibit strong periodicity.

In contrast to the above approaches, our detection solution uses only the nonlinear characteristics of the network traffic to reveal, through recurrence quantification 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.

3. A nonlinear approach to anomaly detection

At a first glance, anomaly detection seems straightforward. One essentially has to pick a statistical definition of an anomaly, process the measurement data by a statistical-analysis technique, and classify the statistical 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, so as to work 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 typical of 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. It runs in two phases. In the first one, the system profiles the normal behavior of network activity from some available traffic patterns and stores these profiles as a knowledge base to be used as a reference in the following analysis. In the second phase, the system compares the current network activity profile with the stored profiles and reports alerts when anything other than normal profile is seen on network. Such anomaly detection scheme can be essentially viewed as a machine learning problem, based on modeling normal and anomalous traffic from a “training set”, by constructing the “baseline” traffic profile, and then flagging all the deviations from this model. This is a classical classification task in machine learning, where only one significant class exists in the training data (the anomalous traffic class) and we have to learn the characteristics of such class and determine if any unseen instance belongs to it or not (binary classification). A preliminary study, also defined as the “supervised” learning phase, on both “normal” and pre-classified “anomalous” network traffic samples (known positives in the “training set”) determines the most discriminating or reference features among those that constitute the parameters of our traffic model. These features are the main drivers of the machine-learning-based binary classifier at the core of our anomaly detection strategy. It has been implemented by using SVMs because of their inherent capability to reliably detect the nonlinear relationships between features and their corresponding classes.

3.1. Identifying the normal traffic profile: the baseline

Interpreting the dynamics in 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 catalogued as anomalous because its degree of deviation from the baseline network behavior observed at these points is high enough. The first task in an anomaly detection process is network baselining, that 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. Our approach to network baselining recognizes 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 track persistent events like connections. Since, under normal traffic conditions, the system can be considered as approximately stable, i.e. close to a steady state, the combination of these events can be used to characterize its recent history and baseline traffic model parameters. Fluctuations can be measured as a time series and analyzed in order to provide the necessary information about the deviation from the above baseline.

3.2. Analyzing the traffic properties on multiple timescales

Paxson and Floyd (1995) showed that many types of network processes, such as the rate of a particular type of packet, their dimension or inter-arrival times have self-similar or fractal behavior with a fractal dimension that changes slowly over time (Taqqu et al., 1997; Chakraborty et al., 2004) 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 with its self-similarity features can be viewed as a stationary process. The concept of stationarity is used to describe an assumed regularity in a series of data (its statistical distribution does not change over time) (Shumway and Stoffer, 2000) and a time series is defined as non-stationary (Priestley, 1988) 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 \( i \). The anomalous traffic caused by an attack still has some distribution regularities, but these differ from the normal ones. When the two kinds of network traffic are overlapped, the distribution regularities will be broken and the 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. Traffic engineering practices regarding traffic volume analysis are tightly related with the notions of non-stationarity. Hence, detecting non-stationarity is important as 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 (Washington et al., 2003) 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, one in which no sample, no matter how short or long, can predict the rate of events for any other sample.

Anomalous events are not bound to a specific time scale. Instead, they tend to occur in bursts separated by long gaps whatever the time scale involved, from milliseconds to months. We believe this behavior is due to abrupt changes of state in the system, such as 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 clear harmonics of some period, so they cannot be eliminated by simply time-differencing the time series. Rather, they lead to apparent short-term variations that, together with noise, can lead to apparent anomalies that are to be discarded as false positives. It comes as no surprise that the major hidden pattern is a daily one, once again driven by the daily 24 hours rhythm of activity, but it is not immediately clear why it is not the fundamental period of the system. The weekly pattern can be reproduced with very low levels of noise, because the variations over many weeks of the weekly pattern are small. The daily pattern has much higher levels of uncertainty, since not all days are equivalent: weekends typically show very low activity and artificially increase the uncertainty in the expected time series. The difference between a weekend day and the variation in any day of the week over several weeks is significant; hence the working week yields the cleanest periodicity. 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 from stationarity 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 across multiple time scales need not 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.

3.3. Anomaly detection through non-linear analysis

The idea of viewing the time series of 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 SVM classification model have been built offline, all the following activities consist in...
a “quantitative” recurrence assessment that could realistically be performed on-line, by using well-known RQA techniques, even on resource-constrained network devices.

3.4. Scope and limitations

The presented approach has been explicitly conceived to 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 side, our efforts focused on detecting less evident variations in traffic intrinsic properties that emerge only by observations made on multiple time scales (e.g., variations in long-range correlation, self-similarity and “hidden” periodic structures and clustering properties that are not apparent in the original 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.

4. Recurrence quantification analysis

The RQA concept, introduced by Zbilut and Webber (1992), can be viewed as an efficient and deterministic way to easily identify non-stationarity and recurrence features in the network traffic. The power of such analysis is that it discovers time correlations between data that are not based on linear or nonlinear assumptions and cannot be distinguished through the direct study 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 \( m \). The embedding dimension can be defined as the minimum number of dynamical variables that can describe the system attractor, whereas the time delay is related to the process of reconstructing the effects of those variables. 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. These measures, showing information about the deterministic structure and the complexity of the dynamics of the observed traffic patterns, can be computed in a sliding window scheme, acting upon successive fixed-length time windows (epochs). Alignment of those statistic variables (outputs) with the original time series (input) when adjusting for the 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 (Zbilut and Webber, 1992) 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.

4.1. Characterizing the system dynamics

Before starting the analysis process, we need to acquire significant insights on the nonstationary variation patterns of the time-series data constituting the baseline samples. The main idea is to reconstruct the initially unknown system dynamics in the phase space by using time-delay embedding. According to continuous dynamical modeling, each dynamical system can be described by differential equations like:

\[
\frac{dx}{dt} = f(x, \lambda)
\]  

(1)

where the real variable \( t \) denotes the time, the vector \( X = (x_1, x_2, \ldots, x_n) \) represents the system state variables, depending on both the time \( t \) and the initial conditions, and \( \lambda \) are parameters of the system, while \( f(\cdot) = (f_1(\cdot), f_2(\cdot), \ldots, f_n(\cdot)) \) is a nonlinear function of these parameters and variables. A state of the system, described by the vector variable \( X \), corresponds to a definite point in the phase space. Each time variation of the system state can be represented as a motion in phase space, along some curve called phase trajectory. The theory of time-delay embedding is a way to shift from a temporal time series of specific observations to a new 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. Usually, a detailed analysis of a dynamic system is possible when the equations of motion and the degrees of freedom \( n \) are all known. The changing state of such dynamic system can be indeed represented by sequences of “state vectors” or vectors of state variables in the phase space (Eckmann and Ruelle, 1985). Although only a few significant quantities can be usually observed in such a system, it is possible to reconstruct its entire dynamics from a relatively small number of variables. Typically, all that is available to the analyst is a time series as the result of sampling from a single observation point. 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 (Takens, 1981; Packard et al., 1980; Sauer et al., 1991) stating that we can reliably reconstruct a phase space trajectory, by using the time series of a single observable variable, through the method of time delays. In detail, Takens demonstrated, under specific conditions, the existence of a one-to-one correspondence between the original system’s attractor dynamics and the values assumed by a limited number of variables. That is, if we do not know the equations defining a dynamical system and are not able to measure all its state space variables, we cannot access the state space of the original system. However, if we are able to determine
a one-to-one correspondence between the reconstructed state space and the original one only by measuring some variables, then we can unambiguously identify the original state space from the above measurements. Starting from the scalar time series \( \{x_i\}_{i=1}^T \), we generate a sequence of (embedded) vectors \( y_i = (x_i, x_{i+d}, x_{i+2d}, \ldots, x_{i+(m-1)d}) \). The set of all embedded vectors \( y_i, i = 1, \ldots, T - (m - 1)d \), constitutes a phase trajectory in \( \mathbb{R}^m \) where \( m \) is the embedding dimension and \( d \) is the time delay.

Each unknown point of the phase space at time \( i \) is reconstructed by the delayed vector \( y_i \) in an \( m \)-dimensional space, thus called the reconstructed phase space. The sequence of embedded vectors recreates the original dynamics only if the values of \( m \) and \( d \) are chosen properly. In particular, for the Takens theorem to hold, 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 Correlation Dimension \( r \) of Grassberger–Procaccia (Marwan et al., 2007), frequently used to distinguish between random and chaotic behavior. On the other side, 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 \( d > m \), that is, after the geometry is unfolded. The most natural question is, then, how to choose an appropriate value for the time delay \( d \) and the embedding dimension \( m \). Several methods have been developed to guess \( m \) and \( d \). As a general rule, a suitable embedding delay \( d \) has to fulfill two criteria. First, \( d \) has to be large enough so that the information we get from measuring the value of variable \( x \) at time \( t + d \) is relevant and significantly different from the information we already have by knowing the measured value of the variable at time \( t \). Only then it will be possible to gather enough information about all other variables that influence the value of the measured variable to successfully reconstruct the whole phase space with a reasonable choice of the embedding dimension \( m \). Note here that, generally, a shorter embedding delay can be compensated with a larger embedding dimension. This is also the reason why the original embedding theorem is formulated with respect to \( m \) and says basically nothing about \( d \). Second, \( d \) should not be larger than the typical time in which the system loses memory of its initial state. For a larger \( d \), the reconstructed phase space would look more or less random, since it would consist of uncorrelated points.

The latter condition is particularly important for chaotic systems which are intrinsically unpredictable and hence lose memory of the initial state as time progresses. Since the mutual information between \( x_1 \) and \( x_{i+1} \) quantifies the amount of information we have about the state \( x_{i+1} \), presuming we know \( x_1 \), Fraser and Swinney (1986) proposed to use the first minimum of the Average Mutual Information Function (AMI) as the optimal embedding delay. The average mutual information (AMI) minimum is a good estimate for \( d \), on the grounds that uncorrelated variables tend to produce uncorrelated values. In fact, the delay should be selected so as 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 \( p_{ij}(\tau) \) 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 transition in \( \tau \) time from the \( i \)-th to the \( j \)-th bin. The average mutual transition function is:

\[
S(\tau) = -\sum_{ij} p_{ij}(\tau) \ln \left( \frac{p_{ij}(\tau)}{p_i p_j} \right)
\]

A similar argument supports the proposal to use the first zero-crossing of the autocorrelation function (Marwan et al., 2007). 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 developed by Kennel et al. (1992). This method relies on the assumption that the phase space of a deterministic system folds and unfolds smoothly with no sudden irregularities appearing in its structure. By exploiting this assumption, we must come 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. If a phase space point has a close neighbor that does not fulfill this criterion, it is marked as having a false nearest neighbor. From the geometrical point of view, this occurs whenever two points in the phase space solely appear to be close, whereas under forward iteration they are mapped randomly due to projection effects. The random mappings occur because the whole attractor is projected onto a hyperplane that has a smaller dimensionality than the actual phase space and so the distances between points become distorted. In detail, to find the embedding dimension we analyze, with respect to \( m \), the number of false nearest neighbors and the \( m \) parameter is increased in integer steps until the recruitment of nearest neighbors of the dynamic under scrutiny becomes unchanging. At this particular value of \( m \) the information of the system has been maximized (and, technically speaking, the attractor has been completely unfolded). Thus, there is no need to explore higher dimensions since no new information would be gained. For example, two points on a circle can appear close to each other, even though they are not, if e.g. the circle is seen sideways (as a projection), appearing like a line segment. Increasing by one the dimension \( m \) of the reconstructed space often permits to differentiate between the points of the orbit, i.e. those which are true neighbors and those which are not. Let \( y \) be a point of the reconstructed space. Let \( y(0) \) 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 \( y \)).

\[
D^2_m(y, y^{(0)}) = \sum_{k=1}^{m-1} [y_k - y_k^{(0)}]^2
\]

Next, increase \( m \) to \( m + 1 \) and compute the new distance, i.e. \( D^2_{m+1}(y, y^{(0)}) \). The point \( y(i) \) is said a false nearest neighbor if:

\[
\frac{D^2_{m+1}(y, y^{(0)}) - D^2_m(y, y^{(0)})}{D^2_m(y, y^{(0)})} > D_{75}
\]
where \( D_{TS} \) is a predefined threshold. Note that the number of false nearest neighbors depends on \( D_{TS} \). In practice, the percentage of false nearest neighbors (FNN) is computed for each \( m \) of a set of values; the embedding dimension is said to be found for 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.

### 4.2 Exploring recurrence phenomena

Recurrent behavior is an inherent property of oscillating systems. For regular oscillators 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 (Zbilut and Giuliani, 1998):

\[
r_{ij} = \Theta(\epsilon - ||x_i - x_j||), \quad i,j = 1...N
\]

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

\[
\Theta(x) = \begin{cases} 
1 & \text{for } x > 0 \\
0 & \text{otherwise}
\end{cases}
\]

As implied by its name, the norm function geometrically defines the size (and shape) of the neighborhood surrounding each reference point. The recurrence area is largest for the \( \max \) norm, smallest for the \( \min \) norm, and intermediate for the Euclidean norm. \( \Theta(x) \) can be viewed an unbiased estimator of the correlation integral, defined as the mean probability that the states at two different times are close:

\[
C(\epsilon) = \lim_{T \to \infty} \frac{1}{T^2} \sum_{i,j=1 \atop i \neq j}^T \Theta(\epsilon - \|x_i - x_j\|)
\]

Hence, due to the close relationship between the correlation integral and the fractal dimension of the involved time series (Grassberger and Procaccia, 1983), it is natural to presume that such integral is also capable of characterizing the time-series dynamics. In other words, \( r_{ij} \) 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. The state \( j \) belongs to the neighborhood centered in \( i \) of size \( \epsilon \); this means that the state \( \cdot \) of the system at time \( i \) has some “similarity” with the state of the system at \( j \), in other words we can say that the system is staying on nearby “orbits”. 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 threshold \( \epsilon \) in the recurrence matrix \( R \) is called recurrence plot (RP). The most important features of recurrence plots are their large- and small-scale structures, often being termed typology and texture, respectively. These features of the RP relate 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.

### 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 higher percent recurrence values. \( \%REC \) is given by Marwan and Kurths as:

\[
\%REC = \frac{1}{N^2} \sum_{i=1}^N r_{ij}
\]

where \( R_{ij} \) is the recurrence estimated by the Eq. (5). 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 0 (no recurrent points) to 100 (all 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 dispersed recurrent points and those that are organized in diagonal patterns, representing strings of vectors (deterministically) repeating themselves. Periodic signals will give very long diagonal lines, chaotic signals will give very short diagonal lines, and stochastic signals (e.g. random numbers) will give no diagonal lines at all. High values of \( \%DET \) show that traffic exhibit a deterministic structure. Deterministic structures may have a high degree of complexity or the opposite.

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 (intermittency).

The \( \text{ENT} \) variable is a measure of signal complexity and is calibrated in units of bits/integer bins in the frequency distribution histogram. Individual histogram bin probabilities \( P_{bin} \) are computed for each non-zero bin and then summed according to Shannon’s equation. 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. Shannon entropy is computed according to the formula:

\[
\text{ENT} = - \sum_{bin=1}^N P_{bin} \log_b(P_{bin})
\]
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 the deterministic structure of the recurrence plot, the larger the value of the entropy becomes.

The TREND variable is the slope of the least squares regression of local recurrence as a function of the orthogonal displacement from the main diagonal in the recurrence matrix. It quantifies the degree of system stationarity. A “flat” TREND diagram indicates stationarity, whereas 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:

\[
\delta_j = a + \beta \eta_j + \epsilon_j
\]

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 not drift in a dynamical system, there is no fading of 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 mean and standard deviation that are sequence independent, recurrence quantification measurements are strongly dependent on the sequential organization of the involved time series. Random shuffling of the original sequence destroys the small-scale structuring of line segments (diagonal as well as vertical) in the recurrence matrix and alters the computed recurrence variables, but does not 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 offered by traditional statistics.

5. SVM for event classification

Support vector machines (SVMs) have been used to perform binary classification of anomalous and normal traffic patterns using the descriptors obtained from RQA analysis. SVMs’ 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 enough similar to multilayer perceptron in classical neural networks based on Vapnik’s statistical learning theory (Vapnik, 1995). In its operation, an SVM model transforms non-linearly the original input space into a higher dimensional feature space and employs a linear hyperplane in the feature space to separate points. The hyperplane vector \( w \) has a representation in terms of the training samples \((x_i, y_i)\) and their Lagrangian multipliers \( \alpha_i \):

\[
w = \sum_{i=1}^{N} \alpha_i y_i x_i
\]

SVM tackles the computational intractability problem of having to deal with a very high dimensional space by the introduction of suitable kernel functions. The use of such kernel functions also extends the class of decision functions to the non-linear case. This is done by mapping the data from the input space \( X \) into a higher dimensional feature space \( \mathbf{k} \) through a function \( \phi : X \rightarrow \mathbf{k} \) and solving the linear learning problem in \( \mathbf{k} \). The actual function \( \phi \) does not need to be known: it is sufficient to have a kernel function \( k \) which calculates the inner product in the feature space:

\[
k(x, y) = \phi(x) \cdot \phi(y)
\]

In doing this, SVM can be viewed as an alternative training method for classifiers based on multi-layer perceptron in which the network weights are determined by solving a quadratic programming problem with linear constraints, rather than by solving a non-convex, unconstrained minimization problem as in traditional training for neural networks. In detail, our training dataset is structured as \( \{(x_i, y_i)\}\) with \( i = 1, 2, \ldots, N \). Here \( x_i \) is the vector representing the input features (the RQA descriptors) for the \( i \)-th sample in the training dataset, \( y_i \) is the corresponding class label \( \text{positive:Anomalous traffic, negative:Normal traffic} \) and \( N \) is the total number of elements in the training dataset. For the two classes, \( y_i \) is assigned a value of +1 (indicating the positive class) and −1 (for the negative class). The goal of SVM modeling is the determination of the optimal hyperplane separating clustered aggregation of vectors in such a way that those associated with one category of the target variable will stay on one side of the plane and those associated to the other category will be located the other side of the plane. The vectors located closer to the hyperplane will be the support vectors. If, as in our case, the analysis process consists in separating binary target variables with two predictors, and the clusters can be divided by a straight line, life would be much easier and the achieved results more effective and reliable. Thus, in the classification model learned by SVM, positive support vectors are the support vectors which satisfy \( y_i = +1 \), and negative support vectors are the support vectors which satisfy \( y_i = −1 \). The SVM-based classification function is of the form:

\[
f(x) = \sum_{i=1}^{m} y_i a_i k(x, x) + b
\]

In the above function \( m \) is the number of input data having non-zero values of Lagrange multipliers \( (a_i) \) (usually less than \( N \)) obtained by solving a quadratic optimization problem, \( k(x, x) \) is the kernel matrix and \( b \) is a bias term. Our kernel matrix calculations are performed with a \textit{sigmoid} kernel function, originating from neural networks, and applied broadly in non-linear SVM (Lin and Lin, 2003), defined by:

\[
k(x, y) = \tanh(\rho \cdot x \cdot y + \zeta)
\]

where, \( \rho \) and \( \zeta \) are hyper-parameters of sigmoid kernel.
6. Implementation details

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) 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. 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.

We at first used two 24 hours (respectively A and B in Table 1) and one 2-weeks long (C, 336 hours) traces for presenting the basic concepts and demonstrating in a simple and effective way the fundamental features and properties of the proposed approach. Specifically, the A trace contains several anomalous events simulated through distributed SYN floods, LAND and port-scanning attacks occurring at various times (see Table 2), while the other two traces (B and C) have been properly post-processed through Snort 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 C, 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 B (certified anomaly free day) have been used for presenting the usefulness of RQA in discriminating the above sample anomalous events by using easily readable and understandable graphics drawn on one-day time scales. Then, for functional analysis and accuracy testing we worked on a longer time period by using a seven days trace (D in Table 1). It contains 50 simulated attacks located at known time positions within the 168-hours observation interval. The first two 24 hour traces A (known positives) and B (certified baseline) have been combined and used for building the training set, populated with pre-classified observations. Finally, the trace D dataset has been used for result verification and final model validation (testing set).

6.1. The traffic features of interest

We used CAIDA’s Coral Reef suite (Keys et al., 2001) to process the above packet traces and compute the traffic feature values to be used in our analysis. 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 in choosing a too large number of features was the objective that calculation should be realistically possible within a resource constrained IP network device. Thus the considered features needed to fit the following constraint-selection criteria:

- Complete packet payload independence.
- No implicit dependence from the transport layer.
- Computational Simplicity.

Accordingly, the features that demonstrated to be most selective were the overall byte rate (based on the IP length excluding link layer overhead) and, less impressively, the average inter-arrival time between packets, calculated with at least microsecond precision and accuracy. Note that recurrence quantification analysis assumes a scalar time series. The above features have thus been studied one at a time and combined into specific aggregate structures that will be used in classifying traffic anomalies. More precisely, we define a set of features describing the traffic pattern in a specific time interval.

6.2. 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

<table>
<thead>
<tr>
<th>Table 1 – General workload dimensions of the traces.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trace</td>
</tr>
<tr>
<td>Duration</td>
</tr>
<tr>
<td>Anomalies</td>
</tr>
<tr>
<td>Packets</td>
</tr>
<tr>
<td>Bytes</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 2 – The simulated attacks in trace A.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start Time</td>
</tr>
<tr>
<td>01:15</td>
</tr>
<tr>
<td>02:15</td>
</tr>
<tr>
<td>03:15</td>
</tr>
<tr>
<td>04:15</td>
</tr>
<tr>
<td>05:15</td>
</tr>
<tr>
<td>06:15</td>
</tr>
<tr>
<td>13:15</td>
</tr>
<tr>
<td>22:15</td>
</tr>
<tr>
<td>23:15</td>
</tr>
</tbody>
</table>
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 (Strozzi et al., 2007). 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.

6.3. 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 (Ruelle, 1990) where, starting from simple geometrical considerations, Ruelle determined that if the calculated dimension of our system is well below $2 \log_2 N$, 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 C (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 B (anomaly free) traces resulting in 172800 pre-classified samples.

6.4. 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 quantiative results obtained need to be processed through data mining methodologies provided by the WEKA (Hall et al., 2009) system, to extract the more selective traffic profile features and generate the anomaly detection criteria by using SVM. The software products used in our nonlinear time series analysis are the TISEAN 3.0.1 package toolset by Hegger and Schreiber (Hegger et al., 1999) and the RQA v10.1 code developed by Webber and Zbilut (RQA), 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 deterministic stationary system. Thus, in order to justify further analyses, we have to verify if the involved baseline traffic time series possesses properties typical of deterministic stationary signals. Accordingly, as a preliminary step in our nonlinear analysis our initial assumptions on the stationarity of the baseline time series have been verified by observing its space time separation plot (stp), (Provenzale et al., 1992) reporting the probability that two points in the reconstructed phase-space have distance smaller than $\epsilon$, (i.e. $||s_i - s_j|| < \epsilon$ as a function of $\epsilon$ 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 $\epsilon$ for each time separation interval $\Delta t$. In this graphic the horizontal axis represented the separation in time whereas the vertical axis represented the separation in space. Here a roughly flat profile in all the components is a clear symptom of stationarity (see Fig. 1 below).

After establishing stationarity of the baseline time series, we perform a determinism test to verify whether the system behavior is a consequence of deterministic dynamics. While the stationarity test allowed us to verify that the system parameters were held almost constant during data observation, the determinism test enables us to distinguish between deterministic chaos and irregular random behavior, which often resembles chaos. Here, if a system is really deterministic, it can be described by a set of ordinary differential equations like (1) and the vector field at every point of the phase space is uniquely determined by these equations. We apply a simple yet effective determinism test, originally proposed by Kaplan and Glass (1992) based on the idea that neighboring trajectories in a small portion of the embedding space should all point in the same direction, thus assuring uniqueness of solutions in the phase space, which is the
hallmark of determinism. Hence the above test measures average directional vectors in a coarse-grained embedding space. If the system is deterministic, the average length of all the directional vectors $k$ will be near to 1, while for a completely random system $k \approx 0$. The test on the 2-week baseline resulted in a determinism value of 0.867, that is a quite satisfactory determinism degree for our time series.

After establishing that the baseline traffic pattern originates from a deterministic stationary system, we calculate the maximal Lyapunov exponent (Kantz, 1994). 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. We can see from the stretching factors depicted in Fig. 2 that an accurate quantification of the maximum exponent value is not simple because of the combination of noise and oscillations. However, a positive slope can be appreciated in all cases, and therefore a positive exponent is present. We can thus conclude that the studied time series possesses properties typical of deterministic chaotic signals.

6.5. 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 has focused on the problem of choosing the optimal parameter values for the delay coordinates/state space reconstruction process, resulting in the considerations and heuristic estimators presented in section 4.1. Unfortunately, no rigorous way exists for determining the optimal value of such parameters and, in any case, 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 (Casdagli et al., 1991). However, when $\varepsilon$ 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 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 chosen as the first one which minimizes the average mutual information of equation (2) calculated through the “mutual” program, while the embedding dimension $m$ has been chosen as the value for which the percentage of false nearest neighbors, calculated with the “false-nearest” program, reaches its lowest value. 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 completely 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. 3 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.

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. 4), 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 $m$.

Finally, according to the “rules of thumb” proposed in (Marwan et al., 2007) we have determined our value for the threshold $\varepsilon$ as the 10% of the maximum phase space diameter.

6.6. 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 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 subseries and computed the variables in each subinterval, called an epoch. Such intervals have been regularly shifted so that if $N_e$ is the length of each epoch and $d_e$ the shift, the epoch $i$ corresponds in the time series to the interval starting in $t = (i - 1)d_e + 1$ and ending in $t = (i - 1)d_e + N_e + 1$. 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 frame) and yield lower time resolution RQA variables whereas smaller ones focus on local dynamics (shorter time frame) 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 first investigation has been devised in view of comparing global effects due to Structures in subseries, while the computation for various epochs has been made to emphasize the changes in state inside the whole time series. We used in calculations the Euclidean Norm rescaled with respect to the maximum value (due to the cutoff threshold being 10% of the maximum). Some of the most significant Recurrence Quantification results computed for inter-arrival time and average packet length are, respectively, summarized in Figs. 5 and 6. 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 warrant 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 (left side) 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 possibly 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 have characteristic shapes, and hence 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 has a minimum value of zero, when all the observations fall into a single class, and achieves its maximum value when each class is occupied equally. This provides an adaptive, relative scale that can be 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. 6 right side). 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 downturn. 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.

By comparing the lower side in Fig. 6(b) and the average and autocorrelation charts reported in Fig. 5 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. 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 computing the similarity between the original series and all its lag series. More regular data will exhibit lower fluctuations of its autocorrelation coefficients. Unfortunately, we cannot easily identify in both the Fig. 5 graphs any appreciable track of most of our generated anomalies reported in Table 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. 6(b) gives us a detailed report about all the above events, including those with the lowest packet rate and duration among all. 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. In other words, subtle dynamical departures from "steady state" occurring in time series data might be delayed or even missed by the former tools, but detected sooner and/or more accurately by recurrence tools.

6.7. Building the feature vectors for SVM classification

Construction of the discriminating feature vectors is the key step in building the knowledge base in our classification
model. Here, each recognizable traffic anomalies must be represented in terms of some RQA properties that are informative when distinguishing deviations from normal traffic behavior. The set of chosen RQA descriptors were combined, for each training set, as a pair of 5-dimensional feature vectors respectively associated to the inter-arrival time and average packet length measurements. Our main objective is to build a decision tree for traffic classification choosing the most promising RQA attribute 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 predict the traffic class best. Because irrelevant attributes are known to degrade the performance of the classification process, the RQA attribute pairs we considered have to be screened, to identify and exclude useless or redundant ones. The attribute selected generate a different set of rules, one rule for every discriminating threshold value. To determine the most informative features for binary classification, all the RQA features were subjected to selection by calculating their mutual information gain, ranking them and selecting the best ones for building the final binary classification model implemented through SVMs. The InfoGain algorithm with the ranker method (Witten and Frank, 2005) was implemented using WEKA 3.5.8 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 for validation exactly once, and all of the samples are used for both training and validation. 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 feature vectors (inter-arrival time and average packet length). This confirms our previous observation from the graphs in Fig. 5. The final classification models has been built with these three best discriminating features, calculated on the inter-arrival time feature vectors, by using the WEKA LibSVM wrapper library (Chang and Lin, 2001) and the C-SVC (regularized support vector classification) algorithm with a pure sigmoid kernel ($C = 1$, $\zeta = 0.3$, $\rho = 0$) and 10-fold cross-validation. The inter-arrival time RQA measures have been

![Fig. 6 — %REC and ENT RQA descriptors for inter-arrival times, measured in trace B for anomaly-free traffic (a) and in presence of anomalies (b) in trace A.](image)

### Table 3 — Confusion matrix (on epochs).

<table>
<thead>
<tr>
<th>Classified as →</th>
<th>Regular</th>
<th>Anomalous</th>
</tr>
</thead>
<tbody>
<tr>
<td>Regular</td>
<td>6226</td>
<td>211</td>
</tr>
<tr>
<td>Anomalous</td>
<td>144</td>
<td>132</td>
</tr>
</tbody>
</table>
selected for classification of both interactive and non-interactive traffic. It should be noted that for non-interactive traffic, packet sizes are also a very good indicator of the protocol in use, but the relatively unstructured nature of interactive traffic makes the inter-arrival time a significantly better discriminator.

7. Experimental evaluation

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 “D”. The results of the above test ran on all the 6713 feature vector instances corresponding to the 168 hours, are summarized in the confusion matrix (Table 3). 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 3 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.

Focusing our attention to events rather than of epochs, our classification scheme individuated 49 out of 50 anomalous events. We presented in Table 3 the details about epochs in order to give an insight of the real detection mechanism and to stress the importance of the epoch size as a parameter determining the speed of detectable phenomena. Thus, bearing in mind that these numbers are relative to epochs and not to events, Table 4 reports the most significant metrics that have been used to assess the effectiveness and accuracy of our technique.

The first four metrics are directly related to the classifier’s accuracy measuring the percentage of correct classification with respect to the overall data and to the associated classification errors. The accuracy metric takes into account both positive and negative instances by paying equal attention to all types of error. The recall and precision scores indicate, respectively, errors which are caused by classifying positive instances as being negative and 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 (0.385) 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 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 metric 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 of about 0.4 and beyond indicates a reasonable agreement beyond chance.

7.1. 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 (Lippmann et al., 2000) datasets, although somewhat used in the earliest works in literature, have been harshly criticized (McHugh, 2000) for the usage of synthetic simulated background data not containing the noise (packet storms, strange fragments) that characterize 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. Whereas, until

<table>
<thead>
<tr>
<th>Metric</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Correctly Classified Instances</td>
<td>94.712%</td>
</tr>
<tr>
<td>Incorrectly Classified Instances</td>
<td>5.288%</td>
</tr>
<tr>
<td>Precision (not anomalous traffic)</td>
<td>0.977</td>
</tr>
<tr>
<td>Recall (not anomalous traffic)</td>
<td>0.967</td>
</tr>
<tr>
<td>Mean absolute error</td>
<td>0.053</td>
</tr>
<tr>
<td>Root mean squared error</td>
<td>0.230</td>
</tr>
<tr>
<td>Kappa statistic</td>
<td>0.399</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Metric</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Correctly Classified Instances</td>
<td>96.884%</td>
</tr>
<tr>
<td>Incorrectly Classified Instances</td>
<td>1.151%</td>
</tr>
<tr>
<td>Precision (not anomalous traffic)</td>
<td>0.989</td>
</tr>
<tr>
<td>Recall (not anomalous traffic)</td>
<td>1.000</td>
</tr>
<tr>
<td>Mean absolute error</td>
<td>0.011</td>
</tr>
<tr>
<td>Root mean squared error</td>
<td>0.106</td>
</tr>
<tr>
<td>Kappa statistic</td>
<td>0.724</td>
</tr>
</tbody>
</table>
Detecting network anomaly is an efficient way to discover many existing malfunctions and tackle performance problems in the network. Enhancing the capability of the detection is very meaningful to improve the network availability and to guarantee the quality of services. We presented a new approach to anomaly detection, based on recurrence 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 results show that nonlinear techniques such as RQA 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. Besides, by leveraging on fundamental “hidden” nonlinear dynamics, the approach is promisingly more robust against elusion mechanisms. 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.

Direction open to further investigation include the use of Cross Recurrence Quantification (Marwan and Kurths, 2002) 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.

8. Conclusions and future work

References

Anderson D, et al. Detecting unusual program behavior using the statistical component of the Next-generation Intrusion...


Taqqu MS, Teverovsky V, Willinger W. Is network traffic self-similar or multifractal? Fractals 1997;5:53.
Witten IH, Frank E. Data mining: practical machine learning tools and techniques. 2nd ed. SF Morgan Kaufmann; 2005.

Francesco Palmieri holds two Computer Science degrees from Salerno University, Italy. Since 1989, he worked for several international companies on a variety of networking-related projects, concerned with nation-wide communication systems, network management, transport protocols, and IP networking. Since 1997 he leads the network management/operation centre of the Federico II University, in Napoli, Italy. He has been closely involved with the development of the Internet in Italy in the last years, particularly within the academic and research sector, as a member of the Technical-Scientific Advisory Committee and of the Computer Emergency Response Team of the Italian Academic and Research Network GARR. He is an active researcher in the fields of high performance/evolutionary networking and network security.

Ugo Fiore (Italian Physics degree, 1989) has been with Italian National Council for Research at the beginning of his career. He has been working for more than 10 years in the industry, developing software support systems for telco operators. He is currently with the network management/operation centre of the Federico II University, in Napoli, Italy. His research interests focus on optimization techniques and algorithms aiming at improving the performance of high-speed core networks. He is also actively investigating security-related algorithms and protocols.