SEVENTH FRAMEWORK PROGRAMME	**SEVENTH FRAMEWORK PROGRAMME** **Trustworthy ICT**

Project Title:

Enhanced Network Security for Seamless Service Provisioning in the Smart Mobile Ecosystem

NEMESYS

Grant Agreement No: 317888, Specific Targeted Research Project (STREP)

DELIVERABLE

D5.1.1 Correlation Analysis and Abnormal Event Detection Module - First Version

Deliverable No.	**D5.1.1**		
Workpackage No.	**WP5**	Workpackage Title	**Root Cause Analysis of Attack Phenomena Targeting Mobile Devices**
Task No.	**T5.1**	Task Title	**Correlation Analysis for the Detection of Abnormal Events**
Lead Beneficiary	**ICL**		
Dissemination Level	**PU**		
Nature of Deliverable	**R**		
Delivery Date	**30 April 2014**		
Status	**F**		
File name	**NEMESYS_Deliverable_D5.1.pdf**		
Project Start Date	**01 November 2012**		
Project Duration	**36 Months**		

Authors List

Author's Name	Partner	E-mail Address
Leading Author / Editor		
Omer H. Abdelrahman	ICL	o.abd06@imperial.ac.uk
Co-Authors		
Vasilios Mavroudis	CERTH	mavroudis@iti.gr
Stavros Papadopoulos	CERTH	spap@iti.gr
Anastasios Drosou	CERTH	drosou@iti.gr
Boris Oklander	ICL	b.oklander@imperial.ac.uk

Reviewers List

Reviewer's Name	Partner	E-mail Address
Ravishankar Borgaonkar	TUB	ravii@sec.t-labs.tu-berlin.de
Konstantinos Demestichas	COSMOTE	cdemest@cn.ntua.gr
Gokce Gorbil	ICL	g.gorbil@imperial.ac.uk

Contents

List of Figures

List of Tables

Abbreviations

AoCI	Advice of Charge (Information)
AuC	Authentication Centre
BAIC	Barring of All Incoming Calls
BAOC	Barring of All Outgoing Calls
BIRO	Barring of Incoming calls when Roaming
BOIC	Barring of Outgoing International Calls
BOIH	Barring of Outgoing International calls except Home country
BORO	Barring of Outgoing calls when Roaming
BS	Basic Service
CDR	Charging Data Record
CFB	Call Forwarding on mobile subscriber Busy
CFC	Conditional Call Forwarding
CFNA	Call Forwarding on No Answer
CFNR	Call Forwarding on subscriber Not Reachable
CFS	Correlation-based Feature Selection
CLIP	Calling Line Identification Presentation
CLIR	Calling Line Identification Restriction
COLP	Connected Line Identification Presentation
CS	Circuit Switched
CT	Call Transfer
CW	Call Waiting
CFU	Call Forwarding Unconditional
DGA	Domain Generating Algorithm
DoS	Denial of Service
DDoS	Distributed Denial of Service
FIFO	First In First Out
GED	Graph Edit Distance
GPRS	General Packet Radio Service
HLR	Home Location Register
HSS	Home Subscriber Server
IP	Internet Protocol
KL	Kullback-Leibler
KLDA	Kullback-Leibler Discriminant Analysis
KNMF	Kernel Non-negative Matrix Factorisation
KPCA	Kernel Principal Component Analysis

LR	Location Registration
LU	Location Update
LU_G	LU from Group members
LU_R	LU from home subscribers Returning from other PLMNs
LU_T	LU from all home subscribers in Total
LU_V	LU from home subscribers Visiting other PLMNs
MAP	Mobile Application Part
MCS	Maximum Common Subgraph
MDS	Multi-Dimensional Scaling
MM	Mobility Management
MMS	Multimedia Messaging Service
MNO	Mobile Network Operator
mRMR	Minimum Redundancy Maximum Relevance
MSC	Mobile Switching Centre
MPTY	Multi Party Service
OCCF	Operator Controlled Call Forwarding
ODB	Operator Determined Barring
PC	Personal Computer
PCA	Principal Component Analysis
PLMN	Public Land Mobile Network
QR	Quick Response
RNC	Radio Network Controller
SGSN	Serving GPRS Support Node
SMS	Short Message Service
SOM	Self-Organising Maps
SS	Supplementary Services
TCP	Transmission Control Protocol
UDR	Usage Detail Record
UE	User Equipment
USSD	Unstructured Supplementary Service Data
VLR	Visitor Location Register

Abstract

This deliverable explores spatio-temporal information correlation techniques for the detection of abnormal events and attacks in mobile networks. To this end, a variety of expressive features are defined and extracted from signaling traces collected from operational 3G/4G mobile networks as well as synthetic billing-related records. Appropriate proximity measures that can be used to characterise the similarity of different aspects of spatio-temporal data are also defined for each set of data. These features and proximity measures are then used for the discriminant analysis of the datasets in the presence of anomalous instances. A model-based approach using multi-class queueing models is also developed, which allows to conduct quick what-if analysis to determine whether an observed behaviour is normal or malicious in order to perform signaling-based anomaly detection. The deliverable is supplemented by a discussion on the correlations between attack strategies observed in wireline and mobile networks, and provides a review of existing tools for correlation-based anomaly detection.

1. Introduction

One of the key characteristics of mobile communications is that mobile traffic is continuously monitored by the mobile service providers for billing and accounting purposes. Therefore, many malicious activities will have an evident impact on the accounting and billing information of the misbehaving or attacking users, as well as their signaling behaviour. In this respect, abnormal network traffic and event detection requires methods to characterise and extract principal statistics from the observed traffic, such as frequencies, correlations and times between events, and possible spatio-temporal tendencies over different timescales. These mechanisms, which are the subject of this deliverable, should be able to utilise data from multiple and heterogeneous sources, including Charging Data Records (CDR) for the users and control-plane protocol data, combined with security alerts that may be available from external entities, e.g. through the data collection infrastructure being developed as part of the NEMESYS project.

1.1. Objectives and Scope of the Deliverable

This deliverable is part of Task 5.1, which (i) explores different data processing and representation techniques that will be used by the *network-based* anomaly detection algorithms being developed in NEMESYS, and (ii) realises a prototype of the detection module based on these algorithms. This deliverable focuses on the first objective since the anomaly detection algorithms are currently being evaluated, and the design of the module will be specified in the final version of the deliverable, due in month 24 of the project.

Correlation analysis is a well-known technique used in security in order to discover relationships between entities so as to improve the effectiveness of anomaly detection. This involves analysing the dependency, relevance, and redundancy between different traffic variables to:

- select expressive features that can distinguish malicious behaviour from normal activity changes in order to reduce the number of variables to be monitored by the detection algorithms, and
- identify normal *correlations* between the attributes and use them as input features

to the anomaly detection algorithms. In this context, deviations from reference relationships between the traffic variables indicate anomalies.

In the scope of this deliverable, we explore a number of frameworks for data representation and analysis. First, we conduct a correlation study of time series data comprising signaling traces collected from operational 3G/4G mobile networks. Second, we present a graph-based correlation analysis of CDR for a large number of mobile users, and we show how the approach can reveal abnormal communication patterns in the social graphs of malicious users. Finally, we develop a novel model-based approach which allows us to conduct quick "what if" analysis to determine whether an observed behaviour is normal or malicious. In all of these cases, we use a number of proximity measures to characterise the similarity of different aspects of spatio-temporal data.

1.2. Coupling Between Attacks in Wireline and Mobile Networks

In this section we provide a high-level analysis of the coupling between attack strategies observed in wireline networks and their cellular counterparts. We evaluate such possible coupling between the two systems, both in terms of the types of threats encountered, and interdependencies that cause security incidents in the "wired" Internet to affect the availability and security of mobile network services and infrastructure. This analysis constitutes the most basic form of correlation, in the sense of discovering relationships between entities to improve the threat identification process, and attempts to answer two basic questions related to the objectives of this deliverable:

1) Can we use traditional network-based correlation and anomaly detection methods in cellular networks?

2) How can correlation analysis help the threat identification and elimination processes, even when performed at the highest level (i.e. between mobile and wireline networks)?

Smart mobile devices are open to both traditional and mobile-specific threats due to the multiple roles these devices play and the heterogeneity of mobile communication technologies and networked services [15]. Among the traditional threats that mobile devices face, there are physical attacks that require physical access to the device, device-independent attacks such as eavesdropping on the wireless medium or man-in-the-middle attacks, email-based spam and phishing, and IP-based attacks. Current IP-based attacks encountered on mobile devices [109] have been found to be largely similar to those on

non-cellular devices, but there are also a number of traits of attacks that are tailored specifically for mobile devices, as discussed below. With the growing popularity of smart devices, mobile-specific threats have evolved from SMS/MMS-based denial-of-service (DoS) attacks [71, 104] to more sophisticated attacks that usually come in the form of malware and target both the core network [41,105] and the mobile users [71]. The ability of smart devices to install and run apps not only from official markets but also from unknown sources exposes them to malware [33, 120], and while mobile malware is currently a real but small threat compared to desktop malware in Europe and the USA, it is clearly evolving and growing as attackers experiment with new business models by targeting mobile users (cf. NEMESYS deliverables [12, 65] and references therein). The year 2012 has also seen the emergence of the first mobile botnets [67]. A botnet is a collection of Internet-connected devices acting together to perform tasks, often under the control of a command and control server. In wireline networks, malicious botnets are used to generate various forms of spam, phishing, and distributed denial-of-service (DDoS) attacks. Mobile botnets extend such capability to cellular networks, give cybercriminals the advantages of control and adaptability, and pose a significant threat to the mobile core network as they could be used to launch debilitating signaling-based DDoS attacks [56, 72, 73, 105].

To answer Question 1 above, the preceding discussion indicates that mobile malware is not very different from its non-cellular counterpart in the sense that they both rely on the same Internet infrastructure to support their illicit operations [60]. In particular, both types of malware use (i) download sites and social engineering tricks (e.g. phishing), although mobile malware also employs novel social engineering methods such as app repackaging and QR codes; (ii) command and control servers which are used, for instance, by premium SMS diallers to regularly change destination numbers in order to avoid and circumvent blacklisting; and (iii) data transfer for uploading credentials, user's location or other stolen personal information. Furthermore, attacks on both mobile and personal computers (PCs) have been shown to share many behavioural characteristics, including host changes, growth patterns, etc. Thus, traditional network-based correlation techniques which have been applied successfully in the wired domain could be also used to detect mobile threats that follow the behavioural patterns of their traditional counterparts.

The coupling between attacks in mobile and wireline networks is not limited to the types of threats involved or the use of common hosting infrastructure, but also includes other *interdependencies* that cause security incidents in wireline networks to affect the availability and security of cellular networks. First, the Internet carries a lot of unwanted traffic [89] which includes backscatter traffic associated with remote DoS attacks, scanning probes, spam, exploit attempts, etc. If such traffic reaches a mobile network, it

may cause a *signaling storm* due to the frequent but perhaps small amounts of data generated, requiring repeated signaling to allocate and deallocate radio channels and other resources, and therefore have a negative impact on the control plane of the network. Second, some mobile apps go haywire when an unexpected event occurs in the Internet, such as loss of connectivity to a popular cloud service, causing a signaling storm. For example, an important feature of smartphones is the ability to receive "push notifications" from cloud services in order to notify the user of an incoming message or VoIP call. This feature is enabled by having the mobile device send periodic keep-alive messages to a cloud server. In normal operation, this keep-alive period is a large value, e.g. 5 minutes. However, if the cloud service becomes unavailable, e.g. due to a DDoS attack, then the mobile device will attempt to reconnect more frequently, generating significantly higher signaling load than normal in the process as has recently been reported [86, 87]. In this case, the storm will have very little impact on the volume of data handled by the mobile network, and thus it will not visible along this spatial dimension which may nevertheless be useful in detecting other attacks. On the other hand, a large number of mobile devices will attempt to recover connectivity within a short time period, causing significant increase in the number of TCP SYNC packets directed to the server [8], as well as higher delays for those users served by the signaling-overloaded network component(s). This example illustrates that security incidents in mobile networks often exhibit both spatial and temporal locality that could be exploited for detection purposes, hence the answer to Question 2 above, namely correlation and analysis of events that span different time periods across multiple spatial dimensions (e.g. traffic variables, users, network elements, services, etc.) can improve the accuracy of detection and assist in identifying the root cause of an anomaly.

1.3. Outline of the Deliverable

The rest of this deliverable is organised as follows:

Chapter 2 provides a review of common tools used by anomaly detection algorithms, including correlation metrics, feature selection and reduction techniques, and deviation measures for both time series and graph-based representations of raw data.

Chapter 3 presents analysis of signaling traffic traces, collected from commercial 3G/4G networks, in order to provide relevant features that can be used by the anomaly detection algorithms being developed in NEMESYS. To this end, time series decomposition and dimensionality reduction techniques are utilised to enable efficient processing of the data and improve the performance of the anomaly detection algorithms. The extracted features are further reduced by examining the amount of information they

convey, and the correlations between them under both normal and attack conditions.

Chapter 4 develops a graph-based approach for correlation analysis and abnormal event detection in CDR data. First, the data records are transformed into graph representations, and graph matching techniques are applied in order to quantify their spatio-temporal dissimilarities. Then the calculated distances between graphs are mapped into 2D space, allowing the visualisation of high dimensional data for both feature selection and anomalous graph detection. The efficiency of the proposed approach in correlating anomalous events that span different time periods across multiple users is demonstrated on the VAST2008 CDR dataset [1].

Chapter 5 presents a quantitative model-based framework for correlating mobile user activities with their impact on different network components so as to perform anomaly detection more effectively and reduce false alarms. The approach incorporates modelling of the mobile network at different levels of abstraction to properly represent the components and the processes that are prone to anomalous behaviour, and it allows anomaly detection algorithms to obtain quick estimates of the impact of a suspected set of users so that abnormal but non-performance-impacting behaviours are not incorrectly flagged as malicious. Chapter 6 concludes the report with a summary and directions for future work.

2. Background

This chapter provides background information and an overview of the available tools for correlation analysis in mobile networks. The chapter is organised around the main topics addressed by this deliverable: approaches for correlation analysis, feature selection and reduction techniques, and deviation measures for anomaly detection. While special emphasis is given to the methods that are used in the deliverable, we also briefly review other techniques that are currently being investigated and which may be included in the final version of the abnormal event detection module.

2.1. Correlation Metrics

One approach to perform anomaly detection is to model the correlations between different attributes under normal network conditions, and then identify anomalous correlations that may characterise specific attacks. For instance, the signaling load generated by a user is usually positively correlated with its data volume and mobility, and a deviation from this normal relationship therefore indicates anomalous behaviour that may lead to a signaling storm.

2.1.1. Pearson Product-Moment Correlation Coefficient

The Pearson correlation coefficient is one of the most commonly used correlation metrics, which measures the strength of the linear relationship between two variables X and Y as the ratio of their covariance to the product of their standard deviations:

$$r(x,y) = \frac{cov(x,y)}{\sigma_x \sigma_y} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{2.1}$$

where $\bar{x}, \bar{y}$ are the sample means, and $-1 \leq r \leq 1$ which takes positive values when the two variables increase/decrease concurrently, negative values when one variable tends to decrease as the other increases, and 0 for independent variables.

If the relationship between the variables is nonlinear, the raw data can be converted to ranked variables [108], whereby numerical or ordinal values are replaced by their rank when the data are sorted, and correlation is then computed as in (2.1) from these ranks,

giving *Spearman's rank correlation coefficient* [4,44] which we denote by $\rho(x, y)$. Thus ρ is a *non-parametric* measure of the similarity of the orderings of the data, in the sense that it assesses how well the relationship between two variables can be described using any monotonic function, instead of being restricted to linear relationships.

2.1.2. Kendall's τ Rank Coefficient

Kendall's τ [119] is a non-parametric measure of statistical dependence between two ranked variables x and y, each of size n, and is given by:

$$\tau(x, y) = \frac{n_c - n_d}{\frac{1}{2}n(n-1)} \tag{2.2}$$

where the denominator is the number of possible pairings of x with y observations, while n_c is the number of concordant pairs and n_d is the number of discordant pairs which are defined as follows. A pair of observations (x_i, y_i) and (x_j, y_j) are said to be concordant if $x_i - x_j$ and $y_i - y_j$ have the same sign, otherwise they are called discordant. In the presence of rank ties, there are other variants of Kendall's τ which make adjustments for the ties so that the coefficient remains in the range $[-1, 1]$.

It should be noted that a previous study [80] comparing rank-based correlation methods showed that Kendall's τ is superior to Spearman's coefficient, but this may not always be the case. Thus, the time series analysis presented in Chapter 3 we utilise, among other methods, the three correlation coefficients described above in order to extract meaningful statistics and other characteristics of signaling traces from core mobile components.

2.1.3. Mutual Information

While linear methods of correlation analysis can be useful in many cases of interest, in general it is essential to also consider nonlinear relationships between variables. Hence the motivation for considering information theoretic metrics is their capability to quantify a general dependence between random variables, without having to convert the data to ranked variables.

Entropy is an important concept in information theory which measures the uncertainty or unpredictability in a dataset X as follows:

$$H(X) = -\sum_{x} p(x) \log p(x) \tag{2.3}$$

where $p(x)$ is the probability of a data item $x \in X$. Entropy is typically interpreted as the minimum number of bits required to encode the classification of a data item;

for example, if there is one class then entropy is 0, while a dataset containing multiple classes has entropy greater than 0. *Mutual information* is a measure of the dependence between two variables, and therefore it can be used to detect anomalous values based on the fact that during an abnormal event the mutual information of the variable with the rest of the variables deviates from normal [23, 93]. Formally, the mutual information of two random variables X and Y is defined as:

$$I(X;Y) = \sum_{y \in Y} \sum_{x \in X} p(x,y) \log \frac{p(x,y)}{p(x)p(y)} \tag{2.4}$$

where $p(x, y)$ is their joint probability mass function, and $p(x)$ and $p(y)$ are their marginal distributions.

2.2. Feature Selection and Representation

Given a dataset in which data items are defined by a set of attributes, the objective of anomaly detection algorithms is to partition the data into distinct classes (e.g. normal and abnormal) based on the values of the features. Feature selection is the process of selecting a subset of the attributes that best expresses the different classes in the dataset, and discarding irrelevant features that provide no useful information. Further processing may also be applied to the selected features in order to analyse the dependency, relevance, and redundancy between them and to find an intrinsic dimensionality that is much smaller than the original dimensionality [10, 114, 115]. This section reviews a number of feature selection and reduction techniques that could be used for anomaly detection in mobile traffic.

2.2.1. Expert Knowledge

Features that are relevant to a specific attack model can be selected using expert knowledge which is gained either from previous experiences or through modelling, simulation and analysis of novel attack scenarios. The use of the latter model-based approach in mobile networks involves representing how the communication system functions at the level of each mobile connection, and explicitly describing the main internal resources of the network architecture, as well as the external resources that the mobile user may access during its call. Then by conducting a "what if" analysis, and observing the impact of attacks on various metrics and counters collected at different resolutions and aggregation levels, it is possible to identify the features that are relevant for detecting the attacks.

2.2.2. Information Theory

In this section we discuss how information theoretic concepts can be used to extract relevant features that embody important information about the different classes in a dataset. The *information gain* of a feature A which can take values $v \in V_A$, is defined as the reduction of entropy when the dataset X is partitioned according to the feature values, i.e.:

$$IG(X, A) = H(X) - \sum_{v \in V_A} \frac{|X_v|}{|X|} H(X_v)$$

where X_v is the subset of X where A has value v. A feature with high information gain is therefore desirable since it produces highly regular partitions (i.e. subsets with low entropy), which is better for classification purposes [57]. Alternatively, one may use mutual information (2.4) to extract relevant features. In particular, since mutual information quantifies the shared information between two random variables, it can be taken as a measure of the relevance between features and the class labels. In this context, features with high predictive power will have larger mutual information [63, 83, 97, 110], while less important features will be uncorrelated to the classification variable. Also, features can be selected to be mutually independent with each other in addition to having strong correlation with the class labels, as in the *Minimum Redundancy Maximum Relevance* (mRMR) algorithm [5, 84]. In general these approaches are referred to as correlation-based feature selection (CFS) [11,42] and can be based on any of the methods discussed in Section 2.1.

2.2.3. Wavelet Analysis

Wavelet analysis has been applied in the literature for the selection of relevant time series features via a combined time-frequency representation. Specifically, wavelet transforms decompose a time series data into a low frequency signal representing the long term trend, and a high frequency signal capturing short term variations. It is then possible to construct a new signal from the two (or possibly more) components in the decomposition, such that only features with high variations (i.e. spikes) from the long term trend are selected [112]. Such sudden changes are sometimes difficult to identify in the original signal because they may be obscured by an overall trend of the data [14,21,118]. Decomposition techniques are applied in Chapter 3 for the analysis of signaling traffic traces collected from specific mobile core network components.

2.2.4. Dimensionality Reduction

Multi-dimensional Scaling

Multi-dimensional Scaling (MDS) [16] is a linear dimensionality reduction method used in order to compute optimal mapping of the given multi-dimensional data to a lower dimensional space, based on the distances of the data points. MDS is often the initial approach for dimensionality reduction, and its flexibility lies in the fact that the selection of the distance function directly affects the mapping results [31, 43, 98, 116].

MDS is the approach used in Chapters 3 and 4 for visualising high dimensional datasets in the 2D plane. It takes as input a distance matrix $\mathbf{D}_{N\times N} = [d_{ij}]$, where the element d_{ij} represents the distance between the i-th and j-th objects, and outputs a matrix $\mathbf{X}_{N\times M}$, where M is the number of dimensions in the projected space, e.g. $M = 2$ for 2D visualisations. The mapping in MDS is performed in such a way that the distance of each data instance to the other instances is proportional to the distances in $\mathbf{D}$. The matrix $\mathbf{X}$ contains the coordinates of each data instance in the projected 2D space:

$$\mathbf{X} = \begin{bmatrix} x_{11} & \cdots & x_{1M} \\ x_{21} & \cdots & x_{2M} \\ \vdots & \ddots & \vdots \\ x_{N1} & \cdots & x_{NM} \end{bmatrix} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_N \end{bmatrix}$$

where $\mathbf{x}_i$ is a vector that contains the coordinates of point i. The relation between the distance d_{ij} and the vectors $\mathbf{x}_i$, $\mathbf{x}_j$ is given by:

$$\begin{aligned} d_{ij}^2 &= (\mathbf{x}_i - \mathbf{x}_j)^T(\mathbf{x}_i - \mathbf{x}_j) \\ &= \mathbf{x}_i^T\mathbf{x}_i - 2\mathbf{x}_i^T\mathbf{x}_j + \mathbf{x}_j^T\mathbf{x}_j \end{aligned}$$

where $\mathbf{x}^T$ denotes the transpose vector of $\mathbf{x}$. From the equation above, we need to compute a new matrix $\hat{\mathbf{D}}$ which contains the squared distances between the instances:

$$\hat{\mathbf{D}} = \begin{bmatrix} d_{11}^2 & d_{12}^2 & \cdots & d_{1N}^2 \\ d_{21}^2 & d_{22}^2 & \cdots & d_{2N}^2 \\ \vdots & \vdots & \ddots & \vdots \\ d_{N1}^2 & d_{N2}^2 & \cdots & d_{NN}^2 \end{bmatrix} = \begin{bmatrix} 0 & d_{12}^2 & \cdots & d_{1N}^2 \\ d_{12}^2 & 0 & \cdots & d_{2N}^2 \\ \vdots & \vdots & 0 & \vdots \\ d_{1N}^2 & d_{2N}^2 & \cdots & 0 \end{bmatrix}$$

The matrix $\hat{\mathbf{D}}$ can be expressed in terms of $\mathbf{X}$ as follows:

$$\hat{\mathbf{D}} = \mathbf{c}\mathbf{b}^T - 2\mathbf{X}\mathbf{X}^T + \mathbf{b}\mathbf{c}^T, \text{ where } \mathbf{b} = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}, \ \mathbf{c} = \begin{bmatrix} \mathbf{x}_1^T\mathbf{x}_1 \\ \mathbf{x}_2^T\mathbf{x}_2 \\ \vdots \\ \mathbf{x}_N^T\mathbf{x}_N \end{bmatrix} \tag{2.5}$$

Now multiplying $\mathbf{J} = \mathbf{I} - \frac{1}{N}\mathbf{b}\mathbf{b}^T$ by the left and right hand sides of (2.5) we obtain further simplifications:

$$\mathbf{X}\mathbf{X}^T = -\frac{1}{N}\mathbf{J}\hat{\mathbf{D}}\mathbf{J} \tag{2.6}$$

where the terms on the right hand side of (2.6) can be factorised by eigenvalue decomposition yielding:

$$\mathbf{X}\mathbf{X}^T = -\frac{1}{N}\mathbf{J}\hat{\mathbf{D}}\mathbf{J} = \mathbf{Q}\mathbf{L}\mathbf{Q}^T$$

where the matrix $\mathbf{L}$ is a diagonal matrix containing the eigenvalues l_i sorted such that $l_1 \geq l_2 \geq \ldots \geq l_N \geq 0$. By selecting the first two largest eigenvalues for the 2D projection and setting the rest to zero, the required matrix $\mathbf{X}_{2D}$ can be obtained as follows:

$$\begin{aligned}
\mathbf{X}\mathbf{X}^T &= \left(\mathbf{Q}\mathbf{L}^{\frac{1}{2}}\right)\left(\mathbf{Q}\mathbf{L}^{\frac{1}{2}}\right)^T \Rightarrow \\
\mathbf{X} &= \mathbf{Q}\mathbf{L}^{\frac{1}{2}} \Rightarrow \\
\mathbf{X} &= \mathbf{Q}\begin{bmatrix} \sqrt{l_1} & 0 & \cdots & 0 \\ 0 & \sqrt{l_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sqrt{l_N} \end{bmatrix} \Rightarrow \\
\mathbf{X}_{2D} &= \mathbf{Q}\begin{bmatrix} \sqrt{l_1} & 0 \\ 0 & \sqrt{l_2} \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \end{bmatrix}
\end{aligned} \tag{2.7}$$

which contains the 2D coordinates, where the projected distance of one object to the others is indicative of its difference with the rest. In the context of anomaly detection, points that are far from the point cloud's centre may indicate an irregularity with respect to their network activity.

Other Dimensionality Reduction Techniques

While MDS is the main dimensionality reduction approach used in this deliverable, there are several other techniques that have been proposed in the literature. Principal Component Analysis (PCA) uses a statistical procedure that transforms a set of possibly correlated variables into a smaller set of linearly uncorrelated variables (i.e. principal components). Since the number of principal components is smaller than or equal to the

number of original variables, the transformation reduces the dimensionality of the data. Furthermore, using this procedure meaningful underlying variables may be identified [22, 46, 48, 58, 81].

Isomap is a manifold modelling algorithm and is often used as an alternative approach to MDS. In a first step, the distances of the k-nearest neighbours are computed and the pairwise Geodesic distances are estimated. Subsequently, MDS is applied on the resulting distance matrix [68, 102, 111]. Isomap is applied in cases where manifold modelling is required to accurately project the data in lower-dimensional spaces, e.g. where points on the underlying manifold with distant geodesic distances may have small Euclidean distance.

Finally, another approach to dimensionality reduction, the kernel trick [70], can be applied to extend linear dimensionality reduction algorithms to nonlinear ones. This has been a common practice in the literature and there are numerous such extensions of popular algorithms such as KPCA [69], KLDA [92], KNMF [117], among others. Typically, different kernel based techniques are examined in order to determine which one best fits the problem [113].

It should be noted that in the majority of the experiments conducted in [51, 106], linear models outperformed manifold learning approaches, when used for visualisation purposes. Based on these results, MDS was selected as the initial approach for visualising mobile network signaling and CDR data. However, manifold-based dimensionality reduction methods will be examined in the final version of the deliverable.

2.3. Measures for Anomaly Detection

The choice of an appropriate deviation metric is an important step in the development of any anomaly detection algorithm. These metrics are typically mathematical constructs that provide an approximation of the distance of an observation from a normal profile to determine its degree of abnormality, but they do not necessarily satisfy all the properties that a proper distance measure should have, such as non-negativity and symmetry. There are many different ways in which dissimilarity between data attributes can be measured, and in this section we briefly review some of the more important ones.

2.3.1. Information Divergence

The Kullback-Leibler (KL) divergence, also known as relative entropy, measures the difference between two probability distributions. Let p and q be the probability mass functions of two data samples representing training and test datasets, then the KL

divergence of q from p is defined as:

$$d_{KL}(p,q) = \sum_x p(x) \ln \frac{p(x)}{q(x)} \tag{2.8}$$

where $d_{KL}(p,q) \geq 0$ with $d_{KL}(p,q) = 0$ if and only if $p = q$. Note that KL divergence is not a distance metric since it does not satisfy the triangular inequality and is not symmetric, i.e. $d_{KL}(p,q) \neq d_{KL}(q,p)$. This lack of symmetry presents some challenges when some events are rare in only one of the two distributions, causing $d_{KL}(p,q)$ and $d_{KL}(q,p)$ to take very different values. To overcome this limitation, several proposals for "averaging" the two divergence values have been considered, including entropy-normalisation [52] as follows:

$$d_{EN}(p,q) = \frac{1}{2}\left[\frac{d_{KL}(p,q)}{H(p)} + \frac{d_{KL}(q,p)}{H(q)}\right]$$

which has an information-theoretic interpretation: $d_{KL}(p,q)$ gives the expected number of extra bits required to encode samples from p when using a code based on q, while $H(p)$ represents the number of bits required to encode p. Thus the ratio $d_{KL}(p,q)/H(p)$ is the relative overhead in bits caused by replacing p with q. This metric was used in [26] to detect changes in feature distributions of individual mobile users at multiple timescales.

2.3.2. Distance Metrics

There are a number of metrics to measure the distance of an observation $\mathbf{x}$ from a dataset $\mathbf{X}$ described by a set of d-dimensional feature vectors whose mean is μ and covariance matrix is $\mathbf{\Sigma}$. One such metric is the Mahalanobis distance which generalises Euclidean distance by taking into account the correlations of the dataset:

$$d_M(\mathbf{x},\mathbf{X}) = \sqrt{(\mathbf{x}-\mu)^T \cdot \mathbf{\Sigma}^{-1} \cdot (\mathbf{x}-\mu)}$$

When the features are independent so that the covariance matrix is diagonal, the resulting distance measure is known as a normalised Euclidean distance:

$$d_M(\mathbf{x},\mathbf{X}) = \sqrt{\sum_{i=1}^{d}\left[\frac{x_i-\mu_i}{\sigma_i}\right]^2}$$

where $(x_i - \mu_i)/\sigma_i$ is the z-score for the i-th attribute, which measures the number of standard deviations σ_i an observation is above its expected value. For anomaly detection in mobile networks, a mapping of the z-scores for radio measurements and other

performance indicators to a real number in $[0, 1]$ was used in [100] in order to describe how well the behaviour of a cell matches its normal profile.

Clearly, Euclidean distance is a special case of Mahalanobis distance with the identity matrix as the covariance matrix. It belongs to a class of metrics that measure distance between vectors in the d-dimensional space defined by feature variables. An alternative way to turn dot products into distances is the *cosine distance* between two vectors $\mathbf{x}$ and $\mathbf{y}$ defined as:

$$d_{\cos}(\mathbf{x}, \mathbf{y}) = \frac{1}{\pi} \cos^{-1} \left(\frac{\mathbf{x} \cdot \mathbf{y}}{|\mathbf{x}| \ |\mathbf{y}|} \right)$$

which ranges from 0 when the two vectors are identical, to 1 when they are exactly opposite, while $d_{\cos} = 1/2$ indicates independence. The *Canberra distance* is another numerical measure of the distance between pairs of points in a vector space. It is a weighted version of Manhattan distance [49] and has been used as a metric for comparing ranked lists and for intrusion detection in computer security [32]:

$$d_C(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{d} \frac{|x_i - y_i|}{|x_i| + |y_i|} \tag{2.9}$$

Finally, the *Hamming distance* between two vectors is a simple metric that counts the number of non-matching attributes in the vectors:

$$d_H(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{d} \mathbf{1}[x_i \neq y_i]$$

where the indicator function $\mathbf{1}[A]$ takes the value 1 if the condition A is true and 0 otherwise. This metric is useful for comparing strings, as when correlating domain names [13] to discover malware that employs domain generating algorithms (DGAs) to evade blacklisting.

2.3.3. Density-based Measures

Density-based anomaly detection techniques map data instances to spatial regions, and estimate an *anomaly score* for a new instance by measuring its proximity to other instances from the same neighbourhood. Since instances that belong to low density regions are relatively distant from their neighbours, they can be considered anomalous.

k-Nearest Neighbours

This method uses the k-nearest neighbours to calculate an outlier score. The density around an object is the inverse of the average of the distance from the point to each of its k-nearest neighbours:

$$\text{density}_k(\mathbf{x}) = \left[\frac{\sum_{\mathbf{y} \in N_k(\mathbf{x})} d_X(\mathbf{x}, \mathbf{y})}{|N_k(\mathbf{x})|} \right]^{-1} \tag{2.10}$$

with some distance measure $d_X(\mathbf{x}, \mathbf{y})$, $N_k(\mathbf{x})$ being the set of k-nearest neighbours of $\mathbf{x}$ and $|N_k(\mathbf{x})| \geq k$ its size. If the distance between a point and its neighbours is small, the density will be high and the anomaly score should therefore be small.

Local Outlier Factor (LOF)

The LOF method measures the "outlierness" of each instance by examining its sparsity compared to other *normal* instances. The method is applied on the d-dimensional Euclidean space that is defined by the selected feature variables. For an object $\mathbf{x}$ the outlier score [17] is computed by finding its k-nearest neighbours, then computing a *reachability distance* rd of $\mathbf{x}$ with respect to other instances:

$$rd_k(\mathbf{x}, \mathbf{y}) = \max\{k\text{NN-}dist(\mathbf{x}),\ d_X(\mathbf{x}, \mathbf{y})\}$$

where kNN-$dist(\mathbf{x})$ is the distance between $\mathbf{x}$ and its k-nearest neighbour. Hence if $\mathbf{y}$ is one of the k-nearest neighbours of $\mathbf{x}$, then the reachability distance between the two is kNN-$dist(\mathbf{x})$, otherwise it is the actual distance. This smoothing operation reduces the statistical fluctuations of $d_X(\mathbf{x}, \mathbf{y})$ for all $\mathbf{y}$ close to $\mathbf{x}$. The anomaly score $LOF(\mathbf{x})$ is then estimated as:

$$LOF(\mathbf{x}) = \frac{\sum_{\mathbf{y} \in N_k(\mathbf{x})} \frac{lrd(\mathbf{y})}{lrd(\mathbf{x})}}{|N_k(\mathbf{x})|} \tag{2.11}$$

where $lrd(\mathbf{x})$ is the local reachability density of $\mathbf{x}$ which is the inverse of its average reachability distance based on its k-nearest neighbours, i.e. $lrd(\mathbf{x}) = \text{density}_k(\mathbf{x})$ when $d_X(\mathbf{x}, \mathbf{y}) = rd_k(\mathbf{x}, \mathbf{y})$. LOF is typically used for unsupervised classification, when there are no training datasets that contain anomalies. A more detailed description of the LOF method and its theoretical foundations can be found in [17].

2.3.4. Graph-based Measures

When data is represented by a graph, detection of abnormal events is performed using *graph matching* methods which determine how similar or dissimilar two graphs are.

Although there are many flexible modelling structures in the literature, most of them lack the well-established mathematical framework and some important operations [64]. For example, the similarity between two real vectors is well-defined, while the similarity between two graph is not an easy task. To tackle the problem of graph matching, three main methods have been proposed in the literature [64]:

- Graph Edit Distance: These methods operate directly in the graph domain, and define the cost of edit operations necessary to make the graphs equal.
- Graph Kernels: These methods utilise graph kernels to define similarities between graphs. The definition of graph kernels also permits the application of kernel machines in the graph domain.
- Graph Embedding: These methods transform the graphs to be compared into vector representations, thus allowing the application of various well-known distance measures and analysis techniques.

These methods are described in detail in the sequel.

Graph Edit Distance

Graph Edit Distance (GED) can be considered as an application of the well-known string edit distance, e.g. Levenshtein distance [59], in the domain of graphs. It measures the cost of transforming the first graph until it is equal to the second. This transformation is comprised of both structural and label distortions. The GED measure is defined directly on the graph domain $\mathbb{G}$ and returns a real positive number:

$$d : \mathbb{G} \times \mathbb{G} \to \mathbb{R}^+ \tag{2.12}$$

For the direct calculation of such a measure the cost of basic edit operations must be established. These edit operations are defined on the graph nodes and edges, and include the following [64]:

- Substitution of an existing node (edge) with a new node (edge)
- Deletion of an existing node or edge
- Insertion of a new node or edge

Other basic edit operations have also been suggested in the literature, such as merging or splitting [7]. Afterwards, a cost is associated with each edit operation using a function c as follows:

$$c : \mathbb{O} \to \mathbb{R}^+ \tag{2.13}$$

where $\mathbb{O}$ is the set of edit operations.

Utilising the aforementioned definitions, the GED measure is the minimum edit cost required to transform the first graph into the second:

$$d_{GED}\left(G_1, G_2\right) = \arg\min_{O_s \subseteq \mathbb{O}} \sum_{o_i \in O_s} c\left(o_i\right) \tag{2.14}$$

where $c\left(o_i\right)$ is the cost of operation $o_i \in O_s$, and $O_s \subseteq \mathbb{O}$ is a subset of all the possible operations $\mathbb{O}$.

An important factor in the calculation of the GED metric is the definition of the cost function $c(.)$. An intuitive way to define it is by considering the Euclidean distances between the weights of the two graphs [78]. It should be noted that labels are distinct characteristics of each edge or node taken from a set of symbols, while weights represent quantitative values that characterise each edge or node and belong to the $\mathbb{R}^n$ space, typically with $n = 1$. An unsupervised method for the estimation of $c(.)$ is proposed in [74] in order to estimate the unknown probability distribution of the edit operations in a sample set of labelled graphs. Another unsupervised method for the estimation of $c(.)$ based on self-organising maps (SOM) is presented in [75].

After the definition of the cost function, the GED measure is found as a solution to the optimisation problem in (2.14). One of the most widely known algorithms for solving this optimisation problem is the A* algorithm [18]. This algorithm employs a search tree to model the edit paths, which is constructed by considering the nodes of the first graph. It uses a best-first search and finds a least-cost path from a given initial node to a goal node. As A* traverses the tree, it follows a path of the lowest expected total cost or distance, keeping a sorted priority queue of alternate path segments along the way. Neuhaus et al. [79] formulated the problem as a quadratic programming problem, which is based on the definition of a fuzzy edit path between two labelled graphs, so that nodes and edges from one graph can be assigned to multiple nodes and edges of another graph. Riesen et al. [90] formulated the problem as an assignment problem, in which the aim is to find the lowest cost assignment between objects of two different sets.

A simple metric for evaluating the GED measure, under the assumption that the cost of all the edit operations is equal to 1, i.e. $c(.) = 1$, and that there are only two operations: insertions and deletion, is the following [19]:

$$d_{GED}(G_i, G_j) = |V_i| + |V_j| - 2\,|V_i \cap V_j| + |E_i| + |E_j| - 2\,|E_i \cap E_j| \tag{2.15}$$

which measures the number of edit operations necessary to make the two graphs equal: the first graph is denoted as $G_i(V_i, E_i)$ and the second as $G_j(V_j, E_j)$. This measure is directly applicable to undirected graphs without weights.

Graph Kernels

Kernel functions have been widely used in Statistics and machine learning to compute in an efficient manner the similarities between different objects. While there are many well-known kernels that operate on the vector space, including Linear, Polynomial, and Gaussian, in the domain of the graphs, the available kernels are very limited and are known as graph kernels.

Most of the graph kernels operate on a new graph which is a direct combination of the two graphs that are being compared with respect to their similarity. The most widely used method to combine two graphs is the tensor or direct product graph [64]. Then, various characteristics are extracted from the combined graph and used by the kernel function for the calculation of the similarity value. For example, Gaertner et al. [34] utilised the adjacency matrix of the direct product of two graphs, by calculating a weighted sum kernel from the entries of the adjacency matrix as follows:

$$k(G_1, G_2) = \sum_{i,j=1}^{|V_x|} \left[\sum_{n=0}^{\infty} \lambda_n A_x^n \right]_{i,j} \tag{2.16}$$

where $\lambda_n \in \mathbb{R}$ is the n-th weight, and A_x the adjacency matrix. This was extended in [103] by defining a different transition matrix in which each entry is computed by a simple gaussian kernel that measures the similarities between individual nodes in the adjacency matrix of the direct product graph.

Neuhaus et al. [77] utilised GED measures to reduce the size of product graphs, and ease the computation of the graph kernel. In particular, the adjacency matrix is defined by utilising the optimal one-to-one node substitution calculated by a GED-based algorithm. Additional graph kernels proposed in the literature can be found in [50, 76].

Graph Embedding

Graph embedding techniques consist in mapping an input graph to a vector space using a mapping function $\phi : \mathbb{G} \rightarrow \mathbb{R}^n$, where $\mathbb{G}$ is the domain of graphs and $\mathbb{R}^n$ is the n dimensional space of real numbers. This mapping allows us to apply well-developed machine learning methods, such as PCA, to the transformed graphs.

Riesen et al. [91] describe a graph embedding scheme which produces a dissimilarity representation of each input graph. Multiple dissimilarity measures can be applied without changes in the method, but the authors used GED based measures. Specifically, given a set of weighted graphs $G = \{G_1, G_2, .., G_t\}$, where the graph nodes and edges have weights associated with them, and a set of prototype graphs $P = \{P_1, P_2, ..P_m\},\ P \subseteq G$,

the emending vector in the $\mathbb{R}^m$ space, i.e. the vector representation of graph $G_i \in G$ is defined as follows:

$$\phi^P(G_i) = [d_X(G_i, P_1), ..., d_X(G_i, P_m)]^T \tag{2.17}$$

where $d_X(.)$ is the selected distance function between two graphs. Such an embedding depends on the selection of the set of prototypes P. These issues are discussed in depth in [91], where different prototype selection approaches are presented. Examples are the *Random Prototype Selector*, which randomly selects m prototypes, and the *Spanning Prototype Selector*, which aims to cover the set G with equally distanced prototypes.

Another approach called *symbolic histograms* for graph embedding is proposed by Vescovo et al. [28, 29], which consists in identifying frequent subgraphs in the initial set of graphs G. Given a set of input graphs $G = \{G_1, G_2, .., G_t\}$ and a set of subgraphs $S = \{S_1, S_2, .., S_m\}$, the graph embedding function $\phi^S : G \to \mathbb{R}^n$ is defined as follows:

$$\phi^S(G_i) = [occ(S_1), ..., occ(S_m)]^T, \ \forall G_i \in G \tag{2.18}$$

where $occ : S \to \mathbb{N}$ counts the number of occurrences of the input subgraph for a given graph in G. The occurrence function is evaluated utilising a weighted GED measure. The set S is called the alphabet, and its selection influences the final result of the graph embedding. An iterative incremental method is proposed to automatically determine the alphabet set, S, that uses clustering to find recurrent graph structures in G.

Jain et al. [45] propose a graph embedding method based on structure spaces. This embedding method provides a representation of the graphs in the $\mathbb{R}^n$ space, taking into account their weighted adjacency matrices. The main method takes the rows of the adjacency matrix and stacks them in order, creating a 1D vector representation. This embedding allows for standard operations like inner product, Euclidean distance, and norm, to be applied on the graph domain. Since every vector representation depends on the order of the rows of the adjacency matrix, each operation is defined as an optimisation problem, which searches the space of all possible row orderings to find the optimum. For example, the Euclidean distance between two graphs in the embedded space is defined as:

$$d_E(G_1, G_2) = \min_{\mathbf{x} \in A(G_1), \mathbf{y} \in A(G_2)} \|\mathbf{x} - \mathbf{y}\|^2 \tag{2.19}$$

where $\mathbf{x} \in A(G_1)$ is a possible vector representation of graph G_1, and $A(G_1)$ is the set of all possible vector representations of graph G_1 taking into account the ordering of the rows of the adjacency matrix.

Other Measures of Graph Distance

Additional distance metrics between graphs are presented in [19], which differ from the metrics discussed above in that they operate directly on the graph domain, similar to GED, but without transforming the graphs. The Maximum Common Subgraph (MCS) metric $d_{MCS}(G_1, G_2)$ of two graphs G_1 and G_2 is based on their isomorphism, and is defined as follows:

$$d_{MCS}(G_1, G_2) = 1 - \frac{|V_{MCS}|}{\max\{|V_1|, |V_2|\}} \tag{2.20}$$

where $|V_{MCS}|$ is the number of vertices of the MCS of G_1 and G_2, and $|V_i|$ is the number of vertices of G_i, $i \in \{1, 2\}$. The same distance metric can also be defined for the edges of the graphs.

The second distance metric takes into account the graph spectrum, i.e. the eigen decomposition of the adjacency matrix, in order to find dissimilarities in the structures of the graphs. Defining the spectrum of graph G_i as $\sigma(G_i) = \{\lambda_1^i, ..., \lambda_m^i\}$ in descending order, the distance measure of the eigenvalues is then:

$$d_\sigma(G_1, G_2) = \sqrt{\frac{\sum_{j=1}^{k} \left(\lambda_j^1 - \lambda_j^2\right)^2}{\min\left\{\sum_{j=1}^{k} \left(\lambda_j^1\right)^2, \sum_{j=1}^{k} \left(\lambda_j^2\right)^2\right\}}} \tag{2.21}$$

where k is number of highest eigenvalues that are taken into account.

2.4. Summary

In this chapter we presented a survey of existing tools for correlation-based threat identification and analysis. First we reviewed a number of correlation metrics that operate either on the raw data, or after preprocessing the data on its rank or information content. We then discussed different approaches for feature selection and extraction, which aim to reduce the number of variables to be monitored by the anomaly detection algorithms. Finally, deviation measures that characterise the similarity of different aspects of spatio-temporal data have been covered, including metrics based on distance, density and graphs. There is usually no clear winner when it comes to selecting a method or metric since the decision is dictated by factors related to the application domain, data representation framework and other factors. The justifications for applying some of these tools in the subsequent chapters will be provided therein.

3. Time Series Correlation

3.1. Introduction

This chapter presents a time series analysis of two distinct datasets collected from operational mobile networks. The results obtained from the study will be utilised by the anomaly detection algorithms developed within NEMESYS (WP4). The raw network traffic is represented by a set of traffic variables and subsequently various methods are utilised in order to extract useful information and study the correlation and the characteristics of the data. In particular, time series decomposition methods, dimensionality reduction techniques and correlation coefficients are applied on the data and their results are presented in this chapter, which is organised as follows. First, the core network architecture and the network traffic model are outlined. Then, analysis of the traffic variables is performed for each of the two datasets, and the correlations between the variables are examined using various metrics. Finally, some conclusions are drawn from the results.

3.1.1. The Core Network Architecture

This section provides an overview of 3G/4G mobile networks, as shown in Figure 3.1. Special emphasis is given to two components: (i) the Home Location Register (HLR) or Home Subscriber Server (HSS), and (ii) the Authentication Centre (AuC). The HLR is a component of 3G networks and the HSS is its equivalent in 4G deployments, but they both perform the same functionalities. Thus we will use *HLR* in the rest of this chapter.

The HLR supports the network control layer by performing subscription and session handling and providing capabilities for mobility management, user security, user authentication, access authorisation, and service authorisation. In large 3G/4G networks, there are usually more than one HLR/HSS databases, each serving millions of subscribers depending on the HLR/HSS manufacturer and configuration. Consequently, a failure in the operation of the HLR will cause service outages for all subscribers that are paired with it. Moreover, the AuC is a network component used to authenticate subscribers that request to attach to the network. Once authentication is successful, the subscriber gains access to the services provided by the network. In practice, the AuC is physically attached to or co-located with the HLR. The signaling traffic of the HLR and the AuC

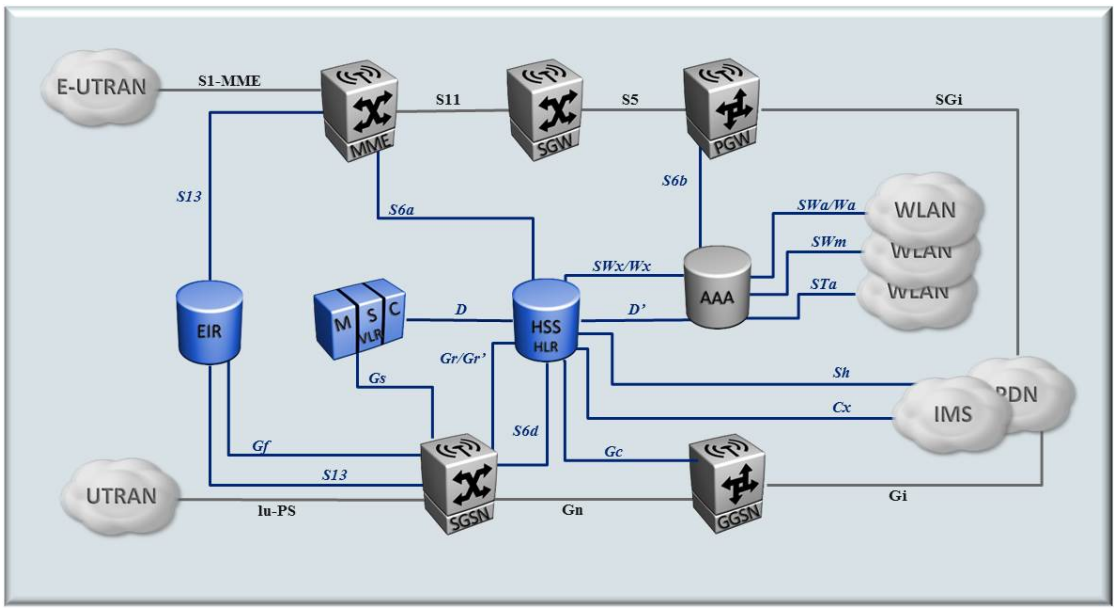


Figure 3.1.: Overview of a 3G/4G network [30]. In this diagram the central role of the HLR component becomes apparent, as it is connected with all core network components.

```
1 0.000000  810        820        GSM  103 invoke sendAuthenticationInfo
2 0.000001  820        810        GSM  201 returnResultLast sendAuthenticationInfo
3 0.000002  810        820        GSM   60 invoke sendAuthenticationInfo
4 0.000003  820        810        GSM  236 returnResultLast sendAuthenticationInfo
5 0.000004  810        820        GSM  116 invoke updateGprsLocation
6 0.000005  820        810        GSM  105 returnError
```

Figure 3.2.: Raw network traffic from the Gr interface between the HLR and the SGSN. Packets 1 and 3 correspond to authentication requests from mobile subscribers. These packets can be exploited for authentication attacks, as described later.

is initiated by the mobile subscribers, and reaches them after being routed by other core network elements, such as the Mobile Switching Centre (MSC), the Serving GPRS Support Node (SGSN) and the Visitor Location Register (VLR). The communication protocol that carries these signaling messages is the MAP protocol, and an excerpt of a packet capture of signaling-related communications between the SGSN and the HLR can be seen in Figure 3.2. More specifically, Figure 3.2 illustrates two successful authentication transactions initiated by the SGSN, and one GPRS location update which resulted in an error because the subscriber was not allowed to roam on the network.

3.1.2. The Network Traffic Model

This section introduces the traffic model which represents the signaling load by a set of attributes extracted from the activity in the control-plane of the mobile network. More specifically, the traffic is modelled as a set of counters indicating the number of signaling requests received by the component within a predefined time period. This method of modelling the network traffic has the following merits:

- Offers an abstract overview of the network traffic and thus significantly decreases the amount of raw data that needs to be processed to extract useful information.
- Is in line with the current monitoring infrastructure of the majority of the network carriers. In particular, most network carriers collect statistics from each network component, rather than raw traffic data.
- Preserves the privacy of the subscribers. This is important, as the legislations of many European countries (e.g. Italy, Greece) prohibit the collection and analysis of data on a per-user basis. Hence, the proposed traffic model processes data on a per-message-type basis.

Similar approaches that use aggregate statistics have been applied also in computer networks [9, 81, 95, 96, 101].

In order to define the HLR/AuC signaling traffic model more formally, let matrix Q be:

$$Q = \begin{pmatrix} q_{1,1} & \cdots & q_{1,Y} \\ \vdots & \ddots & \vdots \\ q_{K,1} & \cdots & q_{K,Y} \end{pmatrix} \tag{3.1}$$

where K is the number of traffic variables and Y is the number of observations of network traffic. An element q_{ij} of Q is the j-th observation for traffic variable i. We denote a row of Q by $Q_{row}(i) = \{q_{ij}, \forall j \in [1, Y]\}$, where $i \in [1, K]$ and a traffic instance as $Q_{col}(j) = \{q_{ij}, \forall i \in [1, K]\}$, where $j \in [1, Y]$. Each row in Q corresponds to a *traffic variable* and each column to an observed *traffic instance*. Using Q, the correlation between the different traffic variables and traffic instances is studied for the detection of signaling anomalies in mobile networks.

3.2. Analysis of the HLR dataset

This dataset contains traces collected from the 3G/4G mobile network of the Greek telecommunication provider COSMOTE during a period of five days. In this dataset, the network traffic is represented by a set of variables, as described in Section 3.1.2, from which time series are formed as shown in Figure 3.3.

A time series is composed of the values of the traffic variables over time, and it measures the network traffic activity over a given period of time. In other words, each variable is merely a counter of signaling requests that were received by the monitored HLR within a specific time frame. The traffic variables can be divided into three categories:

Basic Services (BS): These traffic variables are related to basic services that are provided by the network such as voice calls, SMS and data.

Supplementary Services (SS): These variables correspond to supplementary service provided by the network such as call forwarding or call barring. These services are usually handled by the subscribers with the use of Unstructured Supplementary Service Data (USSD) codes.

Mobility Management (MM): These traffic variables relate to one of the major functions of any mobile network, namely tracking the location of the subscribers in the network, so as to enable seamless delivery of mobile services to the users.

Table 3.1.: The traffic variables from COSMOTE dataset along with their category and description.

Category	Name	Description
BS	B16	Data transfer with circuit duplex asynchronous 9600 bps
	B17	General data transfer with circuit duplex asynchronous
	B1E	Data transfer with circuit duplex synchronous 9600 bps
	B1F	General data transfer with circuit duplex synchronous
	T11	Teleservice, normal speech service
	T21	Teleservice, Receiving short messages from other subscriber
	T22	Teleservice, Sending short message to another subscriber
	T61	Teleservice, Facsimile transmission with alternative speech
	T62	Teleservice, Facsimile transmission
SS	AoCI	Advice of charge (information)
	BAIC	Barring of all incoming calls
	BAOC	Barring of outgoing calls
	BIRO	Barring of mobile terminated calls when roaming
	BOIC	Barring of international calls
	BOIH	Barring of international calls except home country
	BORO	Barring of mobile originated calls when roaming
	CFB	Call forwarding on mobile subscriber busy
	CFC	Conditional call forwarding
	CFNA	Call forwarding on no answer
	CFNR	Call Forwarding on subscriber not reachable
	CFU	Call forwarding unconditional
	CLIP	Calling line identification presentation
	CLIR	Calling line identification restriction
	COLP	Connected line identification presentation
	CT	Call transfer
	CW	Call waiting
	HOLD	Call hold
	MPTY	Multi party service
	OCCF	Operator controlled call forwarding
	ODB	Operator determined barring
MM	LR	Location registration from GPRS subscribers
	LU_G	LU from group members
	LU_R	LU from home subscribers returning from other PLMNs
	LU_T	LU from all home subscribers in total
	LU_V	LU from home subscribers visiting other PLMNs

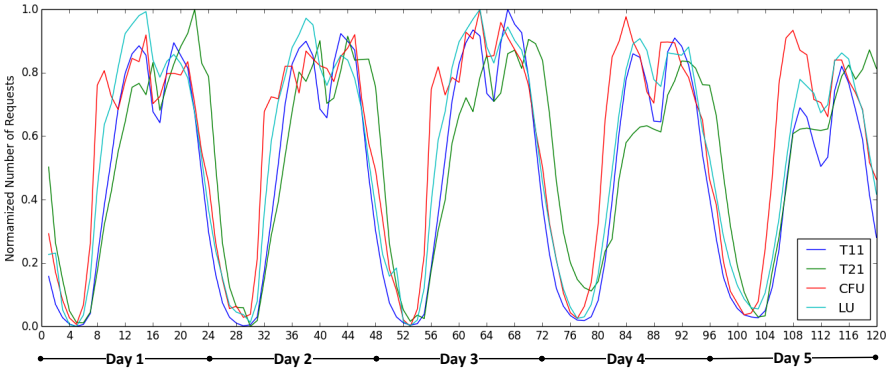


Figure 3.3.: Signaling traffic variables collected from one HLR of COSMOTE's network for a period of 5 days (20–25 Nov 2013) with granularity of 1-hour. The traffic variables are: Teleservice 11 (T11), Teleservice 21 (T21), Call Forwarding Unconditional Activation (CFU) and Location Updates (LU) from home subscribers.

In order to design and develop efficient algorithms for the analysis of the traffic variables, a deep understanding of the normal traffic profile is required. In addition, the features that contain significant information and are useful for anomaly detection purposes need to be identified. Hence, methods for time-frequency representation of the data, dimensionality reduction, and extracting the information content of the data will be presented in the rest of this section.

3.2.1. Decomposition

As a first step in the analysis, the time series of all the traffic variables are decomposed in order to deconstruct them into their notional components. In particular these components are:

- The trend component that reflects the long term progression T_t.
- The seasonal component that describes the seasonality of the data S_t.

- The residuals which describe the random non-regular influences R_t.

In this work additive decomposition [24] is applied so that each time series y_t is represented as a sum of the above three components, i.e. $y_t = T_t + S_t + R_t$. Additive decomposition was selected because the series exhibit no exponential growth and the amplitude of the seasonal component remains almost constant over time. Due to space constrains, however, only the analysis of two traffic variables from each category is presented in detail. In particular, the two variables with the highest entropy values were selected, since they provide more information than the other variables as discussed in Section 2.2.2.

Note that all the figures of this section use a *scale bar* in order to indicate the relative magnitude of the variations in each of the components and how this variation affects the original data. Hence, smaller scale bars correspond to components that highly affect the original data, whereas larger scale bars indicate components with less significance.

Basic Services

First we present the analysis of the two traffic variables that exhibit the highest entropy values. These variables are T11 and T21 (cf. Table 3.1), which represent the number of requests for the most commonly used circuit switched (CS) communication services in mobile networks, namely voice calls and incoming SMS, respectively. Other variables with relatively high entropy values are bearer services (e.g., B16, B17) which are involved in the transmission of signaling information that specify network attributes such as the minimum quality level for a voice call and the automatic re-establishment of a bearer service after the service has been disconnected due to interference.

The results are presented in Figures 3.4 and 3.5, from which the following observations can be made: both time series are smooth and the fluctuations within each day are clearly visible. In the case of T11 no spikes are observed in any of the five days, while for T21 there is a spike on the first day, which is also the highest observed value for this traffic variable. The seasonal component clearly shows traffic volume variations that occur within each day and is very similar for the two variables. Our analyses of all the variables in this category reveal that most of them have similar seasonal components. The trend component is very interesting since in both cases it illustrates activity differences between different days of the week. More specifically, for T11 the trend component is stable during all the workdays while during the weekend there is a major decrease, which becomes more pronounced on Sunday. On the other hand, the trend component for T21 is not so significant but it shows that there is a minor decrease in volume during the weekend. The remainder component, which is the difference between the data and the

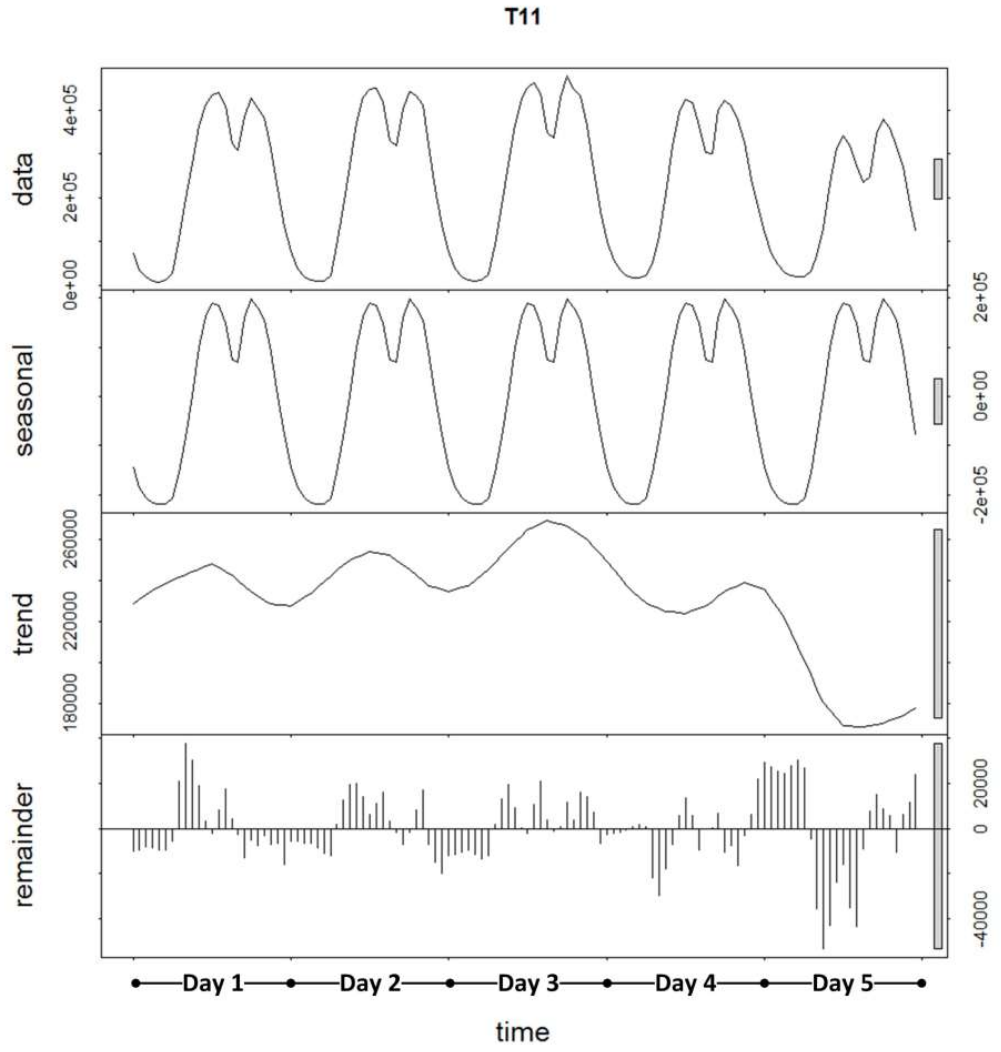


Figure 3.4.: The time series of the traffic variable T11 along with its seasonal, trend and remainder components.

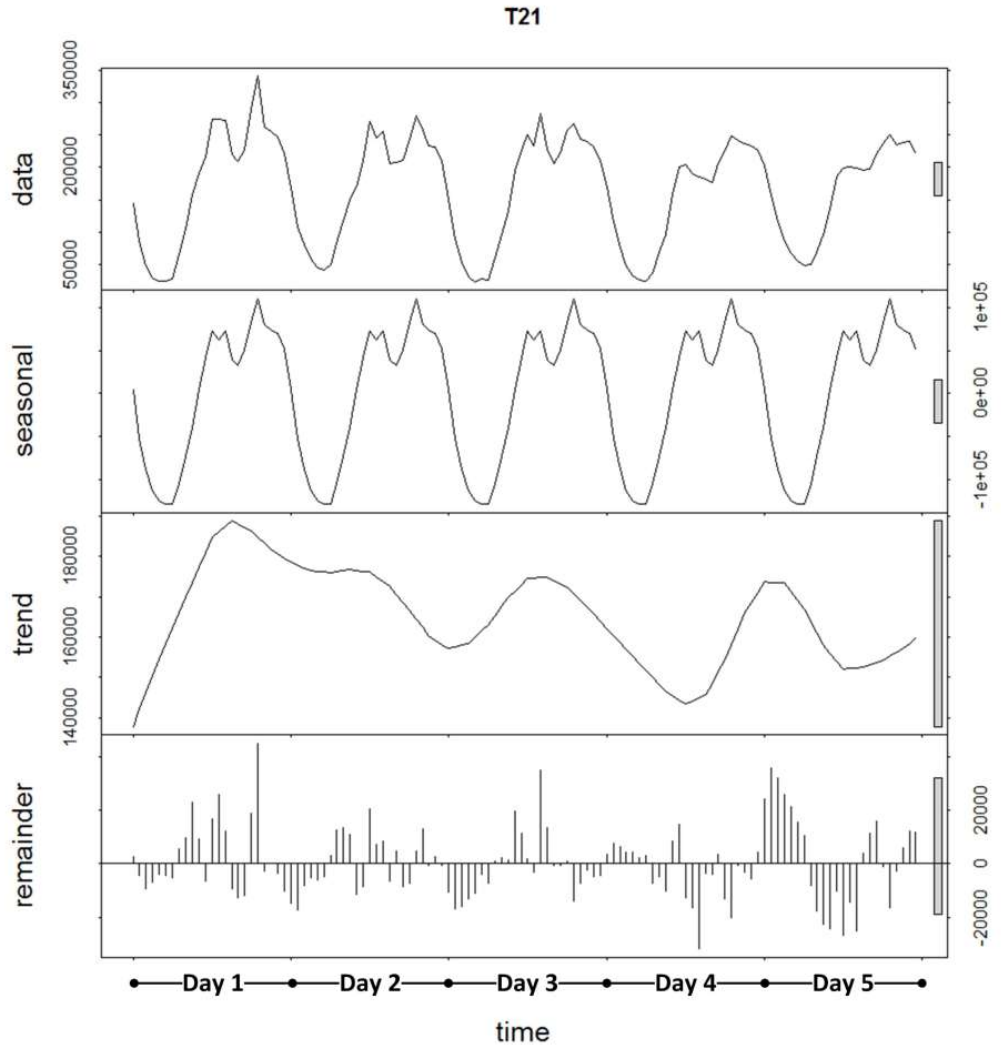


Figure 3.5.: The time series of the traffic variable T21 along with its seasonal, trend and remainder components.

estimated (seasonal + trend), captures short-term variations and thus can be used in order to detect anomalies that cause sudden changes in traffic volume, since it magnifies any differences from the reference profile. These results are in accordance with previous studies of computer and mobile network traffic [20, 25, 27, 54, 66, 88].

Supplementary Services

The two traffic variables from SS which exhibit the highest entropy values are CFU (unconditional call forwarding) and CFNR (call forwarding when the subscriber is unreachable), which reflect the fact that call forwarding functions are among the most commonly used supplementary services in mobile networks. Their additive decomposition is shown in Figures 3.6 and 3.7, from which we observe that the CFU time series is smooth, while the CFNR is quite noisy. More specifically, in the case of CFU no spikes are observed in any of the five days and in the case of CFNR there are various spikes, with the largest being in the third day. Despite these, the fluctuations within each day are clearly visible in both time series. Hence, the seasonal component clearly depicts the traffic volume variations that occur within each day. However, not all of the SS variables appeared to be seasonal. This is mostly because certain services, such as BOIC, are very rarely used and thus there is no seasonality to be observed.

For CFU and CFNR, the trend component appears to be less significant compared to the other components, as shown with the scale bar. It illustrates some activity differences between different days of the week in both time series. In particular, for CFU, after being stable for the first two days, it reaches its peak on the third and then descends during the weekend. On the other hand, the trend component for CFNR is stable during the work week and stays close to its highest values, and in the weekend it gradually decreases. Finally, the remainder component appears to be in both variables random and does not have spikes. As indicated previously, the remainder could be utilised to detect sudden changes in the traffic volume.

Mobility Management

In this section, we present the decomposition results for the two MM traffic variables exhibiting the highest entropy values. These are LU_T and LR, which correspond to the total volume of location update requests and the total volume of location registration requests from GPRS subscribers, respectively. From Figures 3.8 and 3.9 we see that both time series are smooth. However, it should be noted that in the case of LR there is a sudden increase in the volume of requests in a specific time of the day. Additionally, the fluctuations within each day are clearly visible in both time series. As a result, the

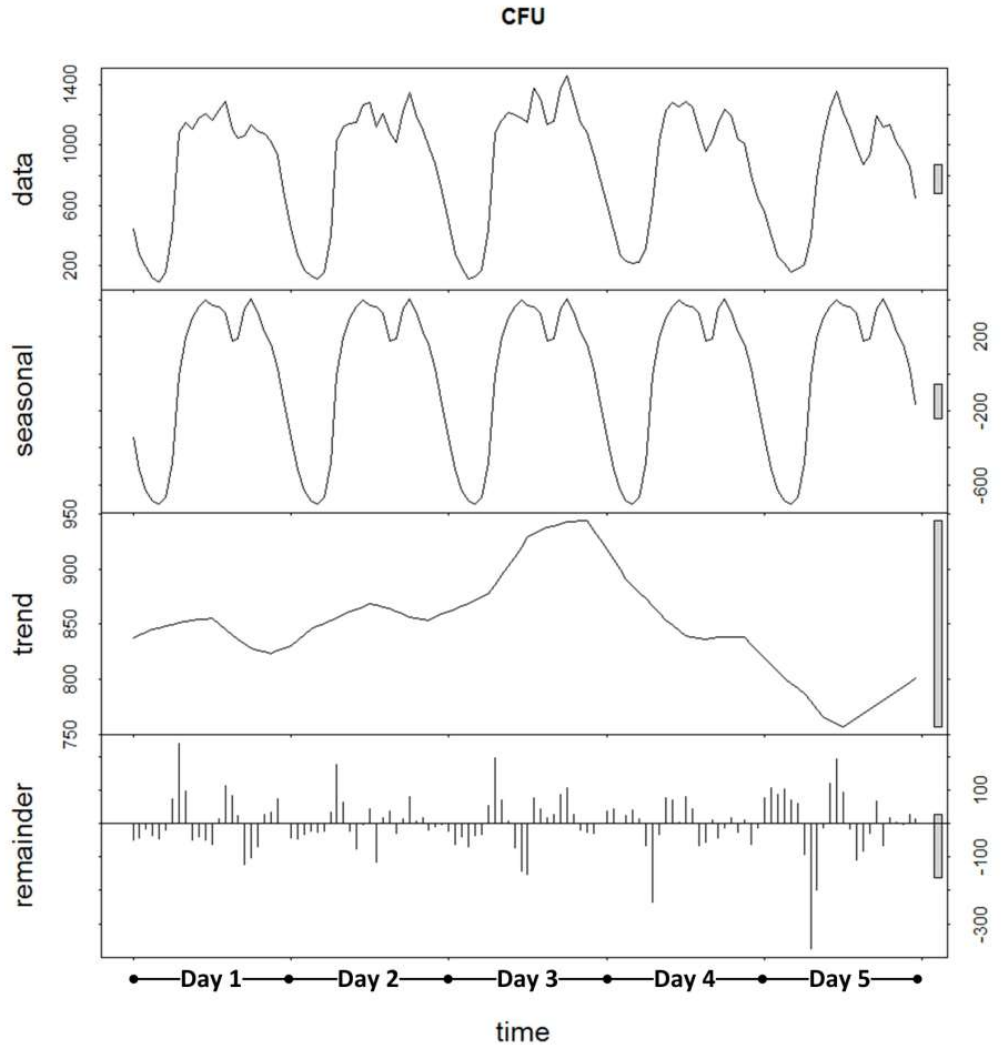


Figure 3.6.: The time series of the traffic variable CFU along with its seasonal, trend and remainder components.

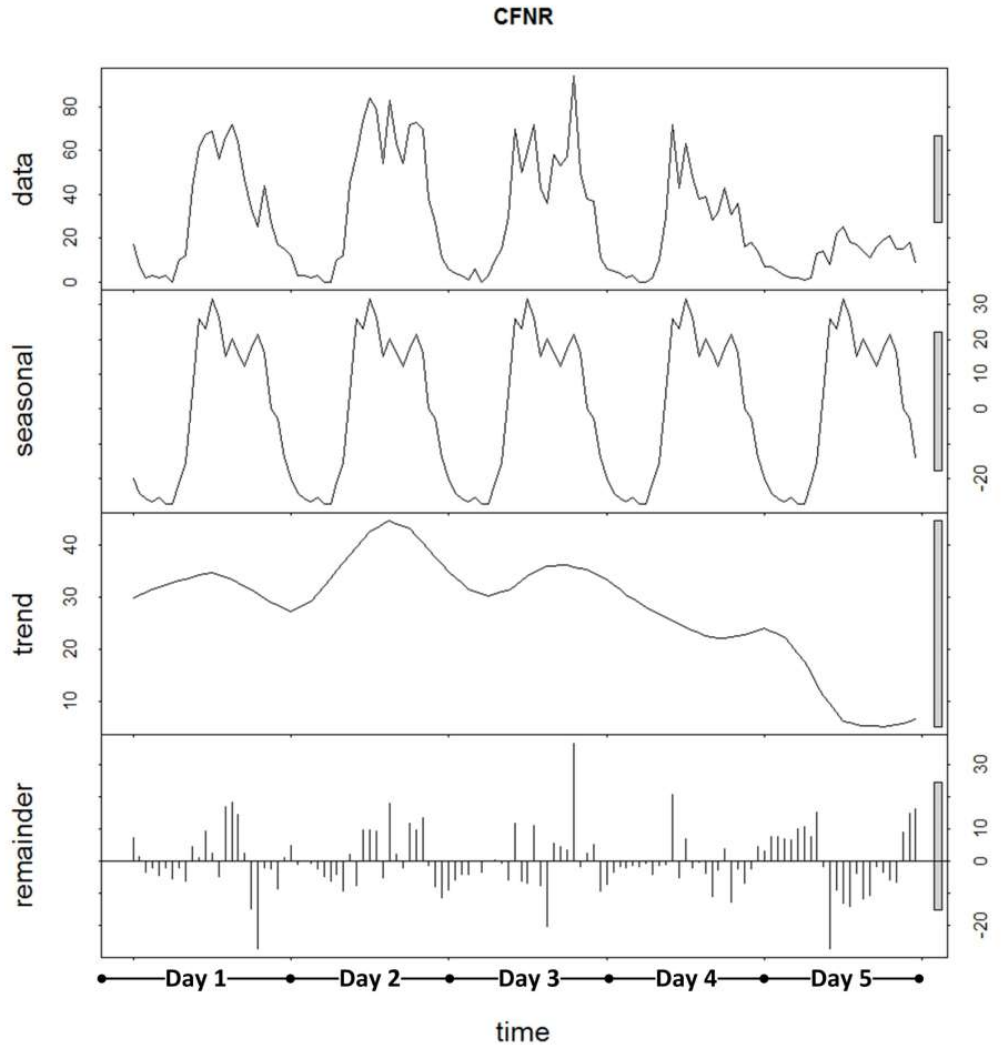


Figure 3.7.: The time series of the traffic variable CFNR along with its seasonal, trend and remainder components.

seasonal component clearly depicts the traffic volume variations that occur within each day.

Furthermore, similar to the previous results, the trend component illustrates activity differences between different days of the week. For both LU_T and LR time series, we can observe a significant decrease in the trend component during the last two days, signifying the difference in profiles between weekdays and weekends. Finally, the remainder component of each variable shows random short-term variations and does not contain abnormally high values.

3.2.2. Multidimensional Scaling

As a second step in the analysis of the available data, we applied various dimensionality reduction techniques in order to test how they behave under normal traffic conditions and in the presence of anomalies. Here we present results obtained using multidimensional scaling (MDS) with the Canberra distance metric defined in (2.9). In particular, MDS was applied so as to reduce the number of dimensions (~ 35) to a single set of 2D coordinates. In Figure 3.10 we show the "expected" shape of the 2D projection of the network activity within a period of 24 hours. The shape may appear rotated but its basic geometrical properties (e.g. curvature) remains the same under normal traffic conditions.

As observed in Figure 3.11, the 2D projection of the traffic variables forms a shape similar to the one shown in Figure 3.10. Moreover, it becomes apparent that the traffic is periodic, with 24 hours periodicity. This verifies the results of the decomposition analysis from the previous section: all days follow a similar path and this becomes even more pronounced in Figure 3.11(f) where we plot the results for all the 5 days combined. Note also that Figures 3.11(a), (b), and (c) appear to be closely similar, as (d) and (e). This can be attributed to the fact that the former are weekdays and the latter are weekend days.

Figure 3.12 shows the MDS results when an anomaly is artificially inserted in day 4 to represent a DDoS attack exploiting CFU signaling requests. We see that the overall shape of day 4 in Figure 3.12(a) and the combined days in Figure 3.12(b) are again similar to the expected one, except that now the synthetic anomaly appears as an outlier in the shape and can be easily identified either visually or through a detection mechanism that operates on the projected data.

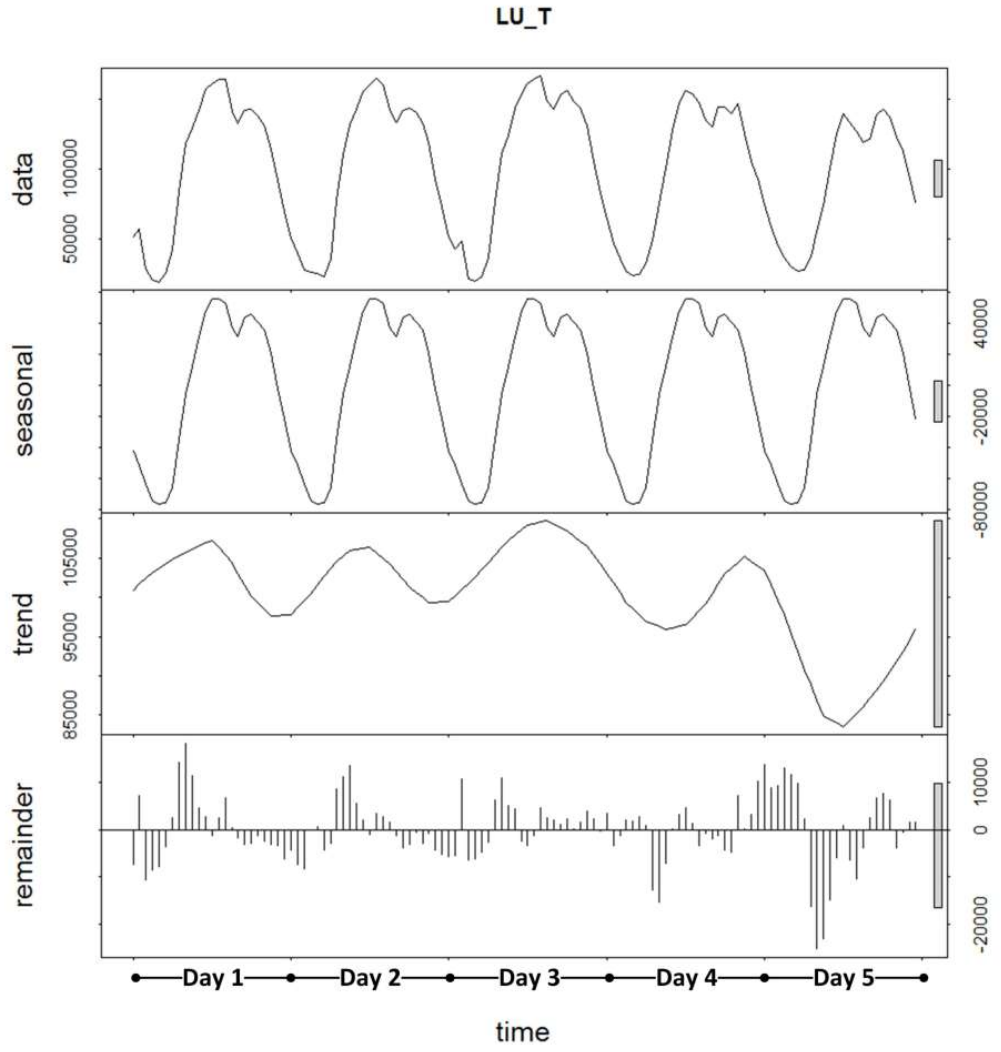


Figure 3.8.: The time series of the traffic variable LU_T along with its seasonal, trend and remainder components.

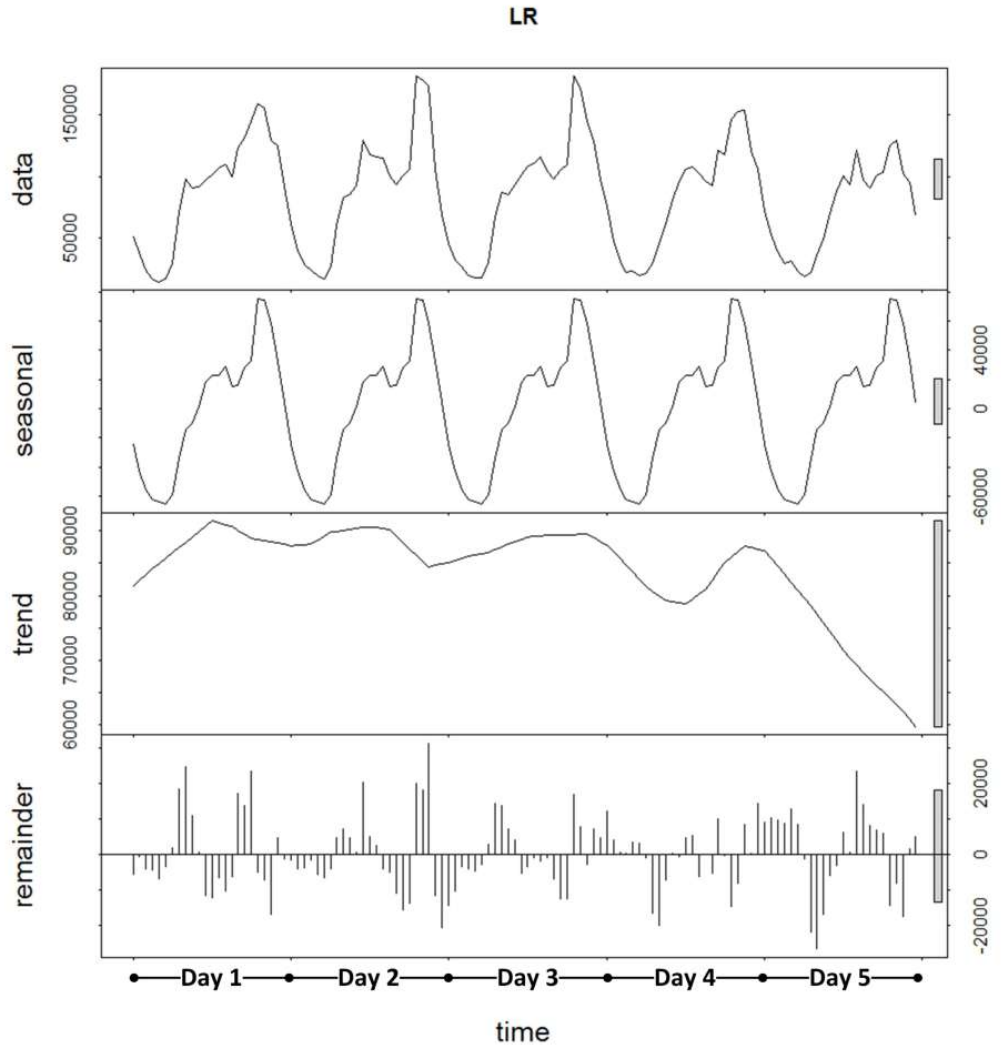


Figure 3.9.: The time series of the traffic variable LR along with its seasonal, trend and remainder components.

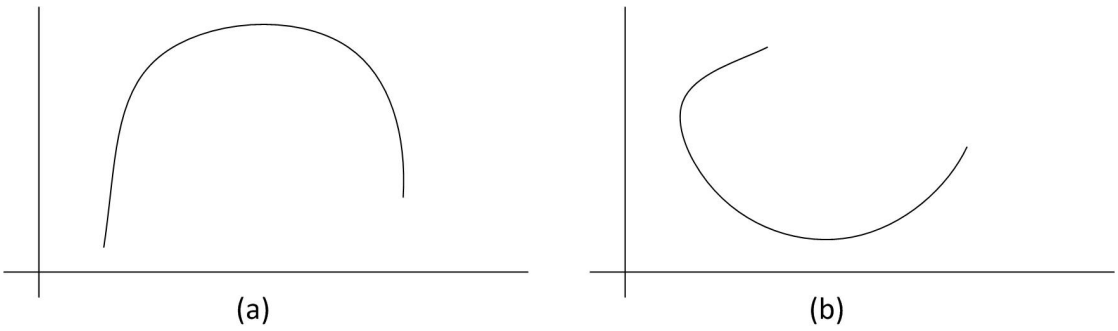


Figure 3.10.: Two examples (a) and (b) of the expected shapes of the 2D projection of the network traffic within a 24 hour period. The shape does not differ significantly between different days, and significant deviations indicate the presence of anomalies in network activity.

3.2.3. Entropy of Traffic Variables and Features

Next the information content of the traffic variables and the extracted features is examined. This, along with the correlation analysis presented in Section 3.2.4, helps to determine which features contain useful information and which ones should be discarded. For measuring the information content of the variables, we apply entropy $H(.)$ as defined in (2.3) to the original variables O, their derivatives D, as well as their trend T, seasonal S and remainder R components, and the results are depicted in Figure 3.13.

The features whose entropy is zero hold no information at all and can be safely discarded. Thus, these variables are not examined in our subsequent correlation analysis. Furthermore, the traffic variables whose $H(O)$ and $H(D)$ values are mostly less than 1 carry very low information content. Despite this, it is possible that some of these variables will exhibit strong correlation with other low-information-content variables and hence might be useful for the anomaly detection, where deviations from these normal correlations may indicate anomalies. Examples of such traffic variables are BOIH and BOIC. On the other hand, information-rich variables such as T11, T21, LU_T and CFU exhibit high entropy values in all the features and thus should certainly be considered for anomaly detection.

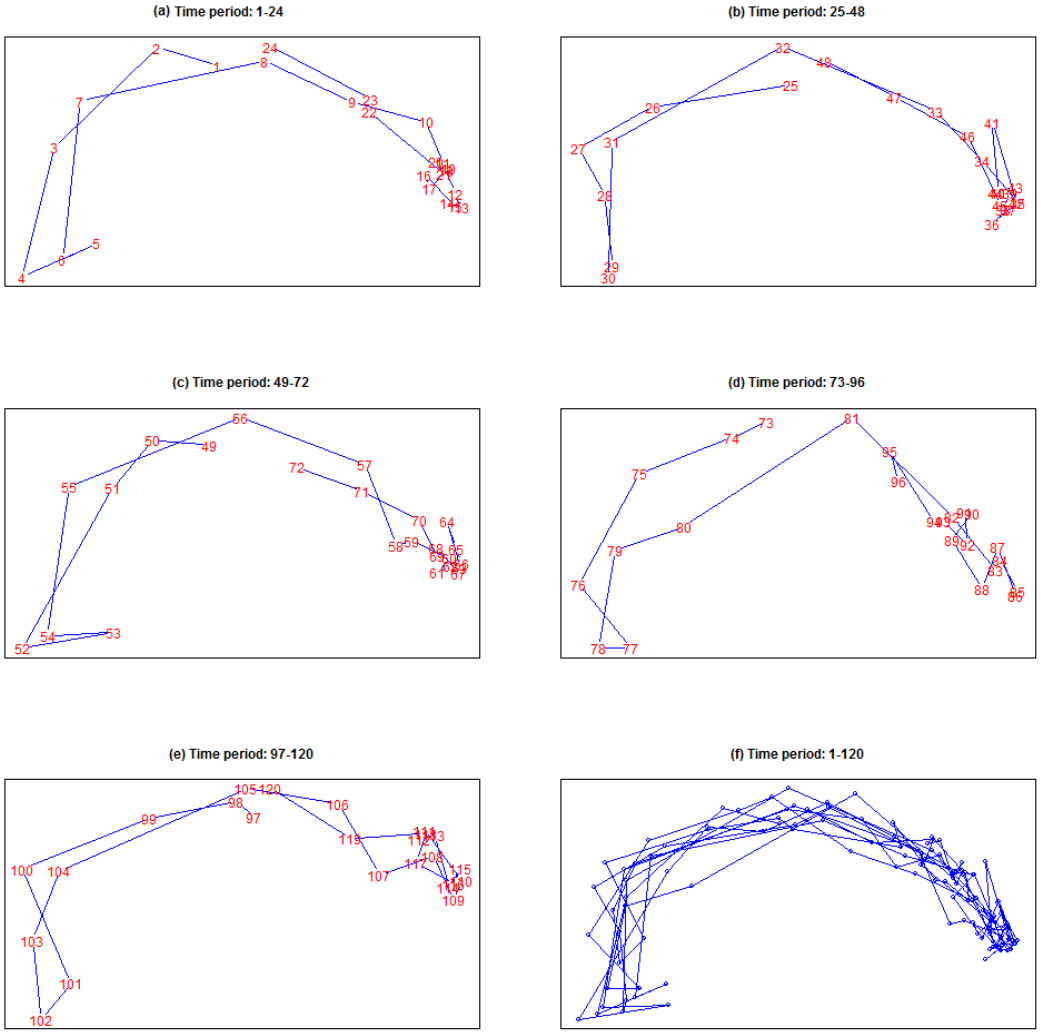


Figure 3.11.: Projection of all the traffic instances from the COSMOTE dataset in the 2D space using MDS with the Canberra distance metric: (a)-(e) correspond to the five days contained in the dataset, and (f) depicts all the days combined.

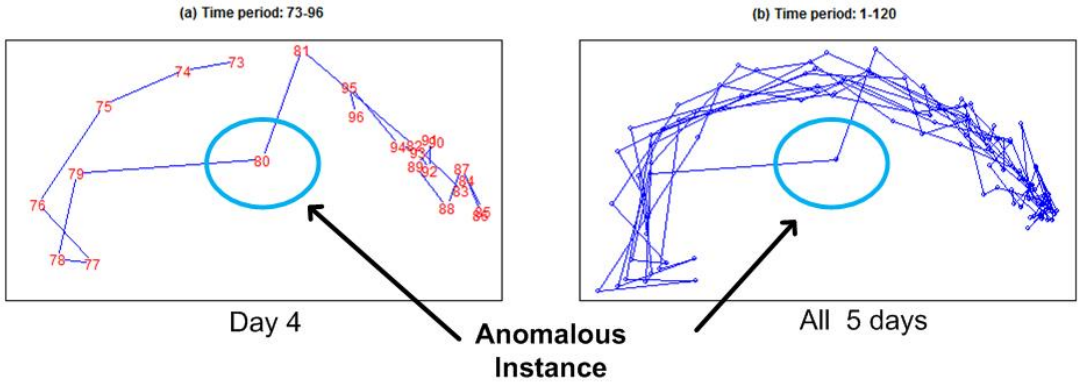


Figure 3.12.: MDS projection of all the traffic instances for (a) day 4 and (b) all the days combined, when an anomaly representing a DDoS attack with CFU requests is artificially injected on day 4.

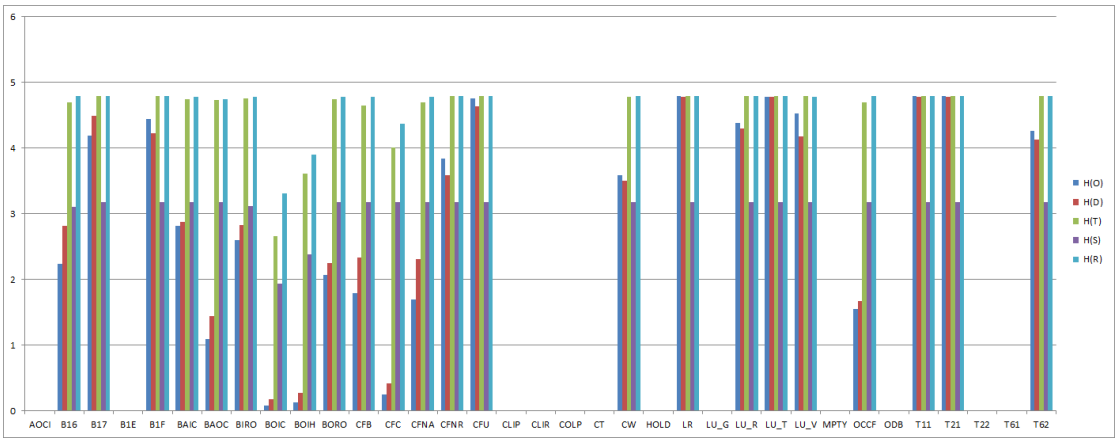


Figure 3.13.: Entropy charts for all the traffic variables and the extracted features of the COSMOTE dataset.

3.2.4. Correlation Analysis

In this section, the correlations between the traffic variables are studied in order to examine which ones are strongly correlated and which are independent. Understanding the relationships between the variables, and how they influence each other, is important for the development of anomaly detection algorithms. For instance, traffic variables that are highly correlated under normal conditions are likely to exhibit different correlation profiles under abnormal circumstances. The results of this study will also be utilised in order to select the features that are more descriptive and to discard those that hold little useful information. To this end, three different correlation metrics are used: Pearson's correlation coefficient is applied to measure the linear relationship between pairs of variables, while Spearman's and the Kendall's correlation are applied to measure relationships that may not be linear, since they both use the ranks of the data (cf. Section 2.1).

Figure 3.14 shows a visualisation of the correlation matrix of the traffic variables. An initial observation is that certain variables appear to be largely independent of other variables, namely CFC and B16. This is mostly because very few subscribers use these services and hence there are no specific usage patterns. Furthermore, other variables appear to be strongly correlated with only one other variable. These correlated pairs include CFNA-CFB, BOIC-BOIH and BAIC-BIRO, which are expected when considering the purpose of these signaling messages. More specifically, CFNA and CFB are both related to call forwarding, BOIC and BOIH to barring international calls, and BAIC and BIRO to incoming calls. However, apart from these correlated pairs, the aforementioned variables are independent of all other signaling variables. This can be attributed to the fact that the services they correspond to are used only sporadically (e.g., when a subscriber travels abroad), unlike with the use of more popular network services such as voice calls. Another correlated pair is T11-T21, which is of high significance as the two messages correspond to the voice calls and SMS sent by subscribers, respectively. The correlation between these two services is due to the fact that users utilise SMS and voice call regularly within a day. It should be noted that any observed correlation between two or more variables is useful for the anomaly detection algorithms. On the other hand, anomalies in traffic variables that are independent can be detected using other statistical properties, e.g. maximum value of first derivative of the variable.

Apart from the correlated pairs, there are also some *correlation clusters* that can be observed in the matrix in Figure 3.14. A correlation cluster contains a group of traffic variables that are strongly correlated with each other:

- The first cluster includes the Location Registration (LR) variable which appears to be correlated with the CW, T21, B1F, CFU, LU_T and T11. The correlation

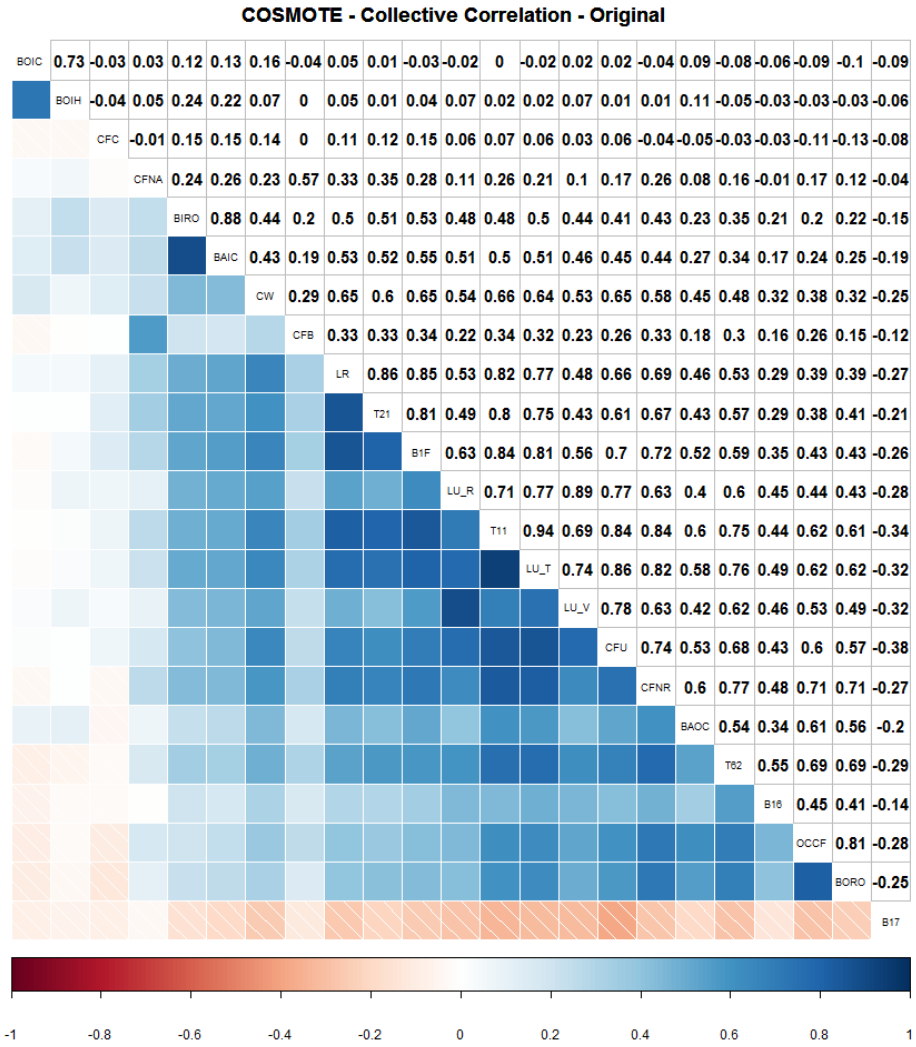


Figure 3.14.: Visualisation of the correlation matrix of the HLR traffic variables constructed using the mean value of the Pearson's, Spearman's, and Kendall's coefficients. The blue and red colours correspond to higher positive and negative correlations, respectively. The lower triangular part of the matrix presents visually the correlation strength, while the upper part shows the actual correlation value.

values vary from strong (~ 0.7) to very strong (~ 0.9) depending on the coefficient used. An intuitive explanation of this correlation cluster is that when subscribers use one of the network services, they are likely to trigger a location registration procedure in the network if they have moved to a new location area.

- Another cluster, which is the most prevalent, is the one that includes the most commonly used services of the network. The variables that correspond to these services are: T11, CFU, T62, B1F, T21, and CW. This implies that the majority of the subscribers use more than one service within the day, and the number of service requests they trigger depends on the time of the day (e.g. the volume of requests is low late at night). Additionally, LU_T appears also to be correlated with most of these variables, indicating that when subscribers use their mobile devices they tend to move between different location areas.

- There is also a cluster that includes all the traffic variables related to location updates. This can be explained because by the facts that LU_T is influenced by all other LU variables, and that mobility patterns are very similar for home, roaming and visiting users. All the correlation metrics indicate that there is strong correlation between these variables.

Figure 3.15 shows the correlation matrix for the derivatives of the traffic variables. We generally observe weaker correlations than those in Figure 3.14, but there are specific variables whose growth rates show strong correlation. More specifically, BIRO-BAIC and BOIC-BOIH pairs exhibit similar growth patterns, while the growth of LU_T appears to correlate with that of T11, T21, B1F and CFU. This can be explained by the fact that when a mobile attempts to use one of these services, it is likely to trigger a location update request due to mobility.

One the other hand, the seasonal components of the traffic variables exhibit higher correlations than the original time series, as illustrated in Figure 3.16, indicating that they have similar periodic behaviour. One such highly correlated pair is BORO-OCCF which can be attributed to the network configuration which redirects incoming calls when call barring has been enabled. Moreover, the seasonality of the CFU and CFNR signaling requests appears to be similar and can be attributed to the similarity of their functionality. There is also a strong correlation between LU_R and LU_V, implying that subscribers have similar mobility patterns when roaming. Finally, an interesting finding is the high correlation between the seasonality of T21 and location registrations (LR). This can be explained if we consider that most mobile networks provide SMS notifications (e.g. for missed calls) to their subscribers when they become available again, thus coinciding with location registration requests.

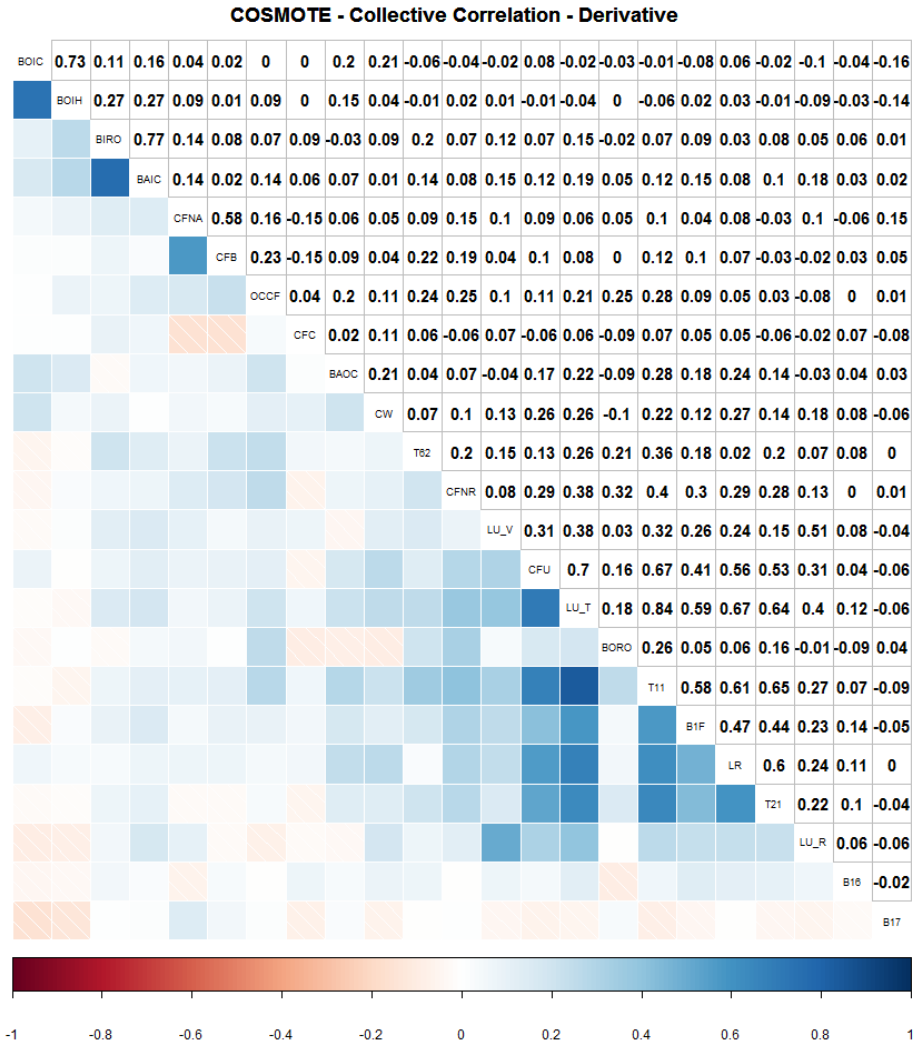


Figure 3.15.: Visualisation of the correlation matrix of the derivatives of the HLR traffic variables constructed using the mean value of the three correlation coefficients. The blue and red values correspond to higher positive and negative correlation respectively. The lower part of the matrix visually presents the correlation strength, while the upper part shows the actual correlation value.

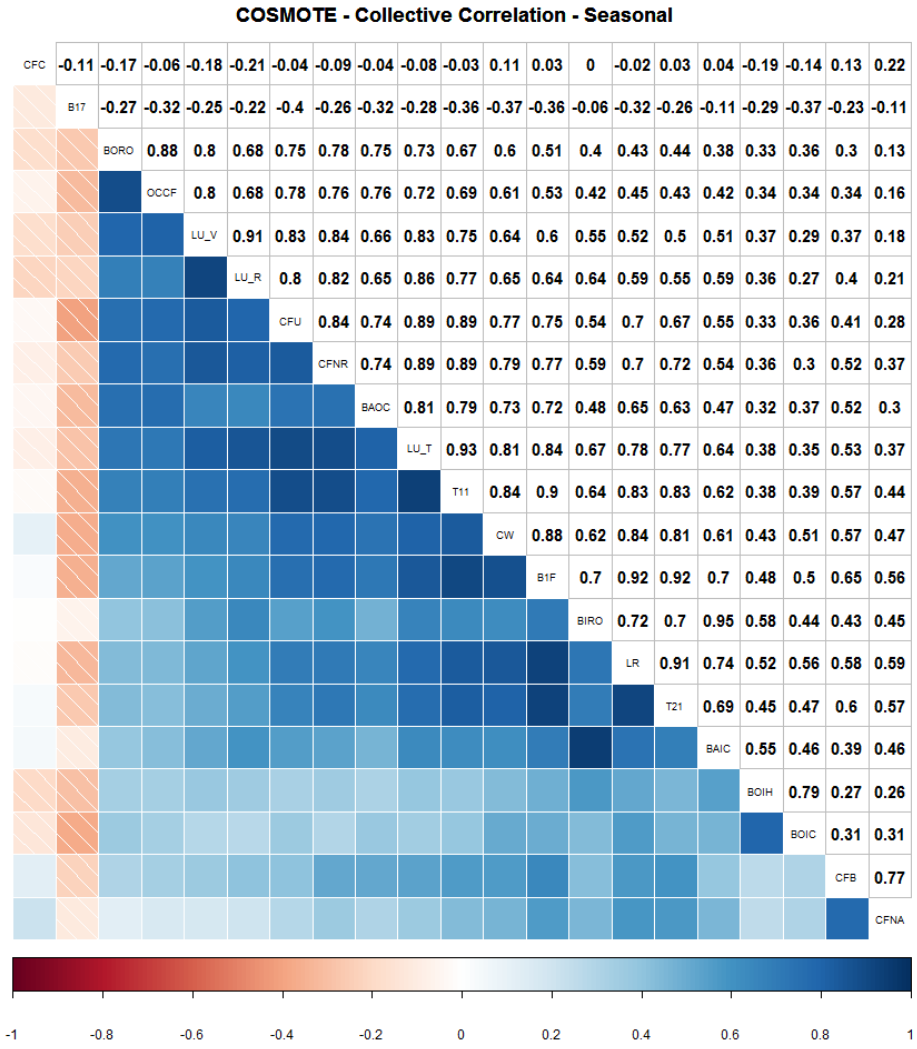


Figure 3.16.: Visualisation of the correlation matrix of the seasonal components of the HLR traffic variables constructed using the mean value of the three coefficient. The blue and red values correspond to higher positive and negative correlation respectively. The lower part of the matrix visually presents the correlation strength, while the upper part shows the actual correlation value.

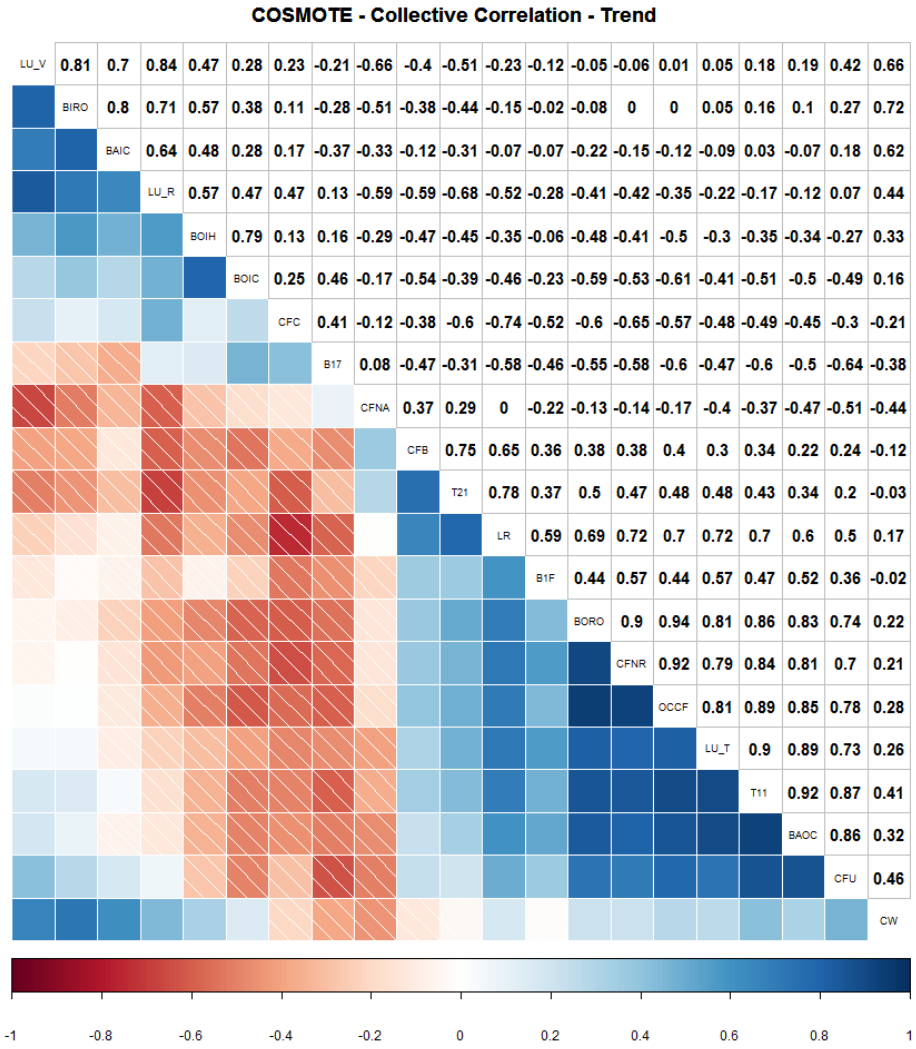


Figure 3.17.: Visualisation of the correlation matrix of the trend components of the HLR variables, constructed using the mean of the three coefficients. The blue and red values correspond to higher positive and negative correlation respectively. The lower part of the matrix visually presents the correlation strength, while the upper part shows the actual correlation value.

Similarly, the trend correlation matrix in Figure 3.17 illustrates that there is a cluster which contains LU_V, LU_R, BIRO and BAIC, due to the use of these signaling requests when roaming in other networks and countries. BORO, CFNR, OCCF, LU_T, T11, BAOC and CFU form another cluster of very strong correlations, which can be attributed to the fact that when users are actively using their mobile device (T11, LU_T) they are more likely to enable/disable other services of the network (i.e. BORO, CFNR, OCCF, BAOC and CFU) as well. Interestingly enough, the trends of T11 and T21 are weakly correlated, implying that users may prefer to use one service over the other in certain days of the week.

The remainder correlation matrix in Figure 3.18 shows very few correlations between the variables. This is because the remainders are the result of minor random deviations from the normal profile and hence they do not follow a specific pattern. Despite this, there are two strongly correlated pairs, namely BOIC-BOIH and BAIC-BORO, indicating that the majority of users use these services together. This could be the result of a configuration in the network that provides a unified command to handle both services. Finally, there were some correlation pairs which were not outlined in the above analysis. These pairs either demonstrated mediocre correlation values or were strongly correlated with other variables/features and thus their relationship was deemed less significant.

3.3. Analysis of the MSC Dataset

This dataset was provided by TIIT and contains 6 traffic variables that relate to users' registration and authentication procedures for a period of three days with 15 minutes granularity. The dataset contains signaling traces from ~ 60 MSC components of TI's mobile network whose details are as follows: 3 of the traffic variables measure user initiated requests sent from each MSC to its corresponding HLR, and the other 3 variables measure the volume of the requests that were successfully served by the HLRs. The major difference between this dataset and the one from COSMOTE is that it allows us to process the data on a per region basis, since each MSC serves a specific geographical area, and typically each network region is served by a small number of MSCs ($\sim 1-10$). The traffic variables contained in the dataset are described in Table 3.2, and the time series of four of them are depicted in Figure 3.19.

3.3.1. Decomposition

As with the previous dataset, the individual components of the time series are computed and analysed. We present here results for the two variables with the highest entropy values, which are NAutReqTot and NLocNRgTot. From Figures 3.20 and 3.21 one can

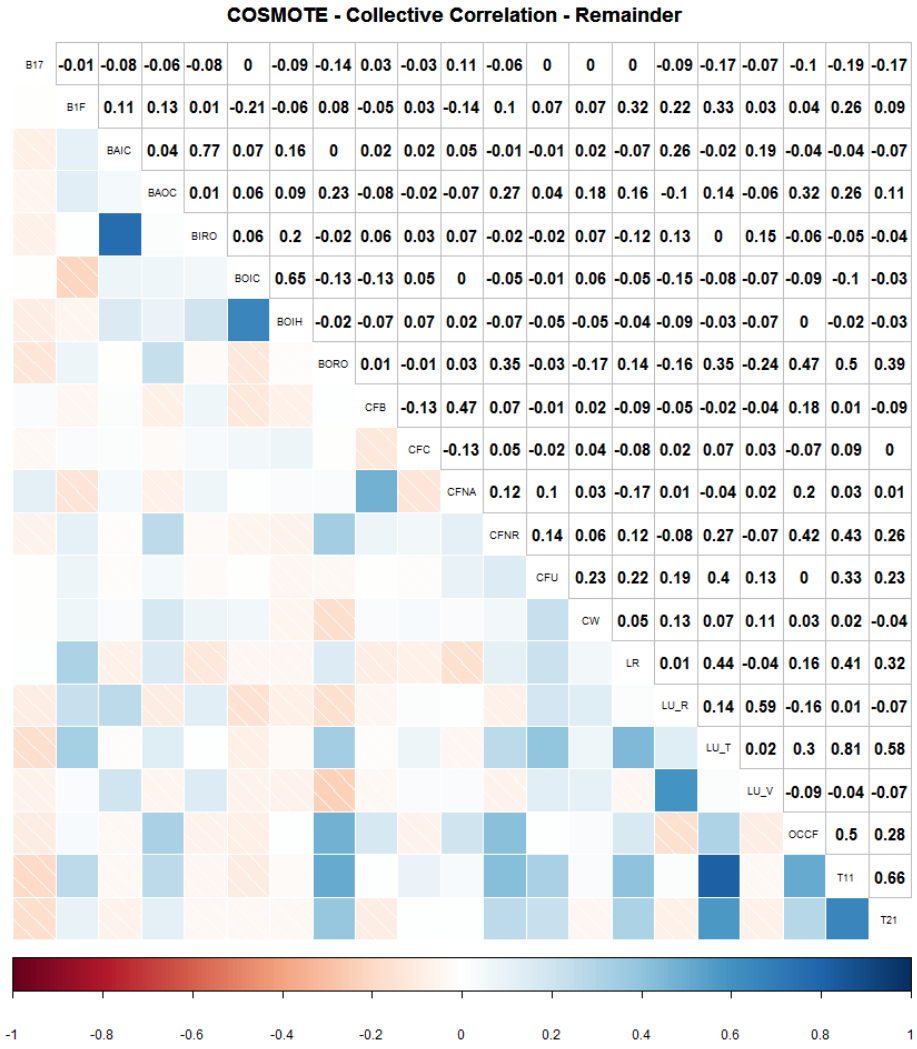


Figure 3.18.: Visualisation of the correlation matrix of remainder components of the HLR variables, constructed using the mean value of the three correlation coefficients. The blue and red values correspond to higher positive and negative correlation respectively. The lower part of the matrix visually presents the correlation strength, while the upper part shows the actual correlation value.

Table 3.2.: Traffic variables from TI dataset along with their description.

Name	Description
NLocNRgTot	Total number of location update attempts from non-registered users.
NLocNRgSucc	Successful location update attempts from non-registered users.
NLocOldTot	Total number of location update attempts from registered users.
NLocOldSucc	Successful location update attempts from registered users.
NAutReqTot	Total number of authentication requests sent.
NAutReqSucc	Successful authentication requests sent.

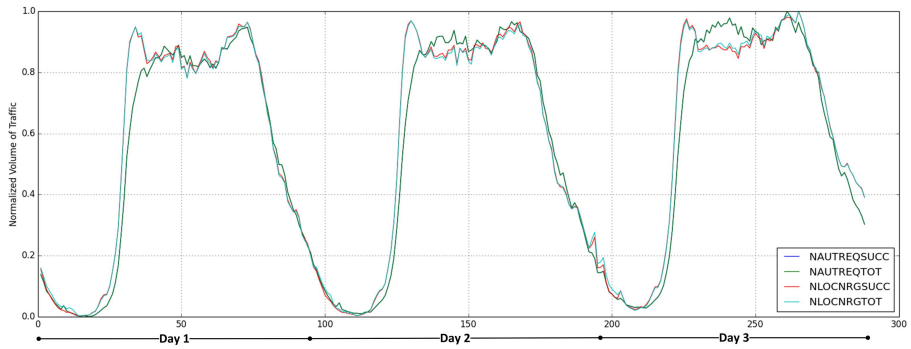


Figure 3.19.: Time series of the signaling traffic variables NLocNRgTot, NLocNRgSucc, NAutReqTot, and NAutReqSucc (Table 3.2) for a period of 3 days with granularity of 15 minutes.

observe that the time series of both variables are smooth, showing the time-dependent variations which become more pronounced in the seasonal component. On the other hand, the trend components show the long-term increase or decrease in the data, showing for example a gradual increase from the middle of the first day in Figure 3.20 and from the last day in Figure 3.21. Finally, the remainder component shows the short-term variations and can be used to detect anomalies that cause sudden changes in traffic volume.

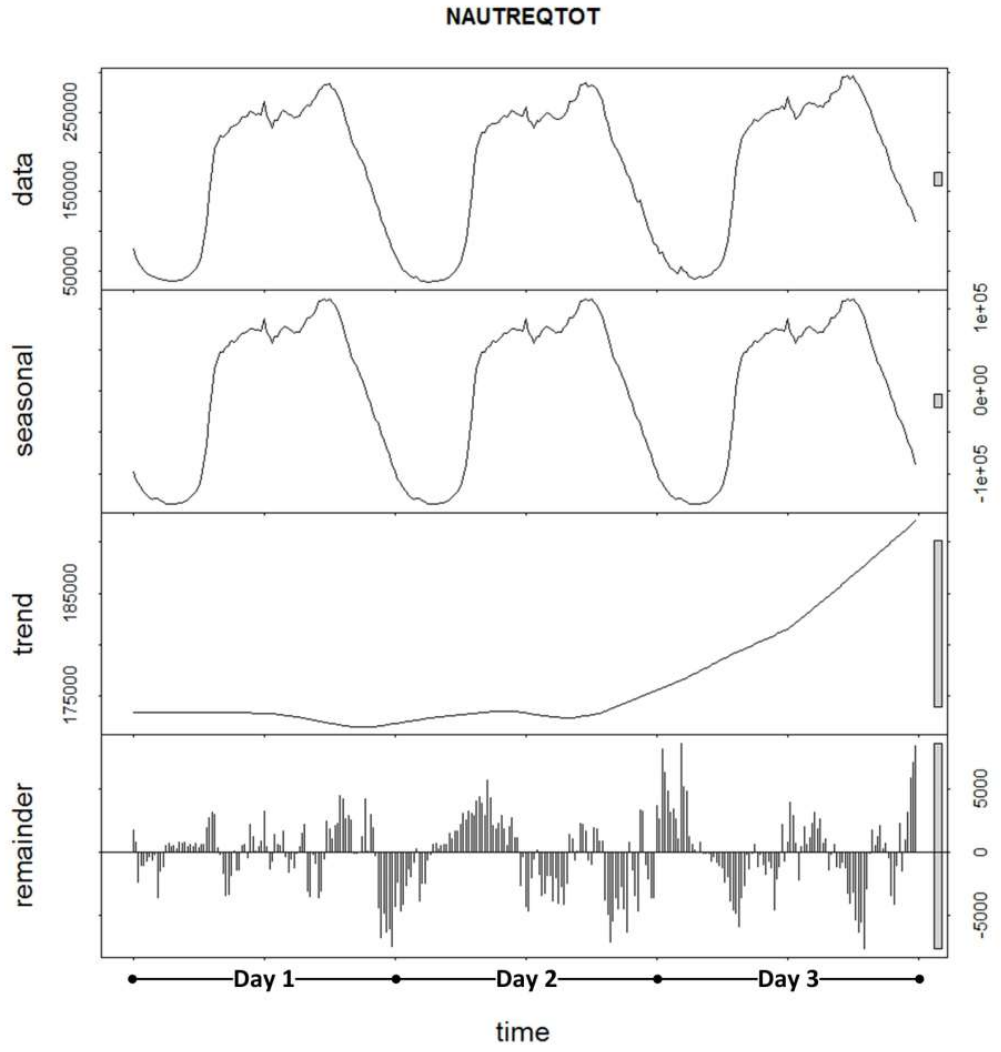


Figure 3.20.: Time series of the traffic variable NAutReqTot, from the TI Dataset, along with its seasonal, trend and remainder components.

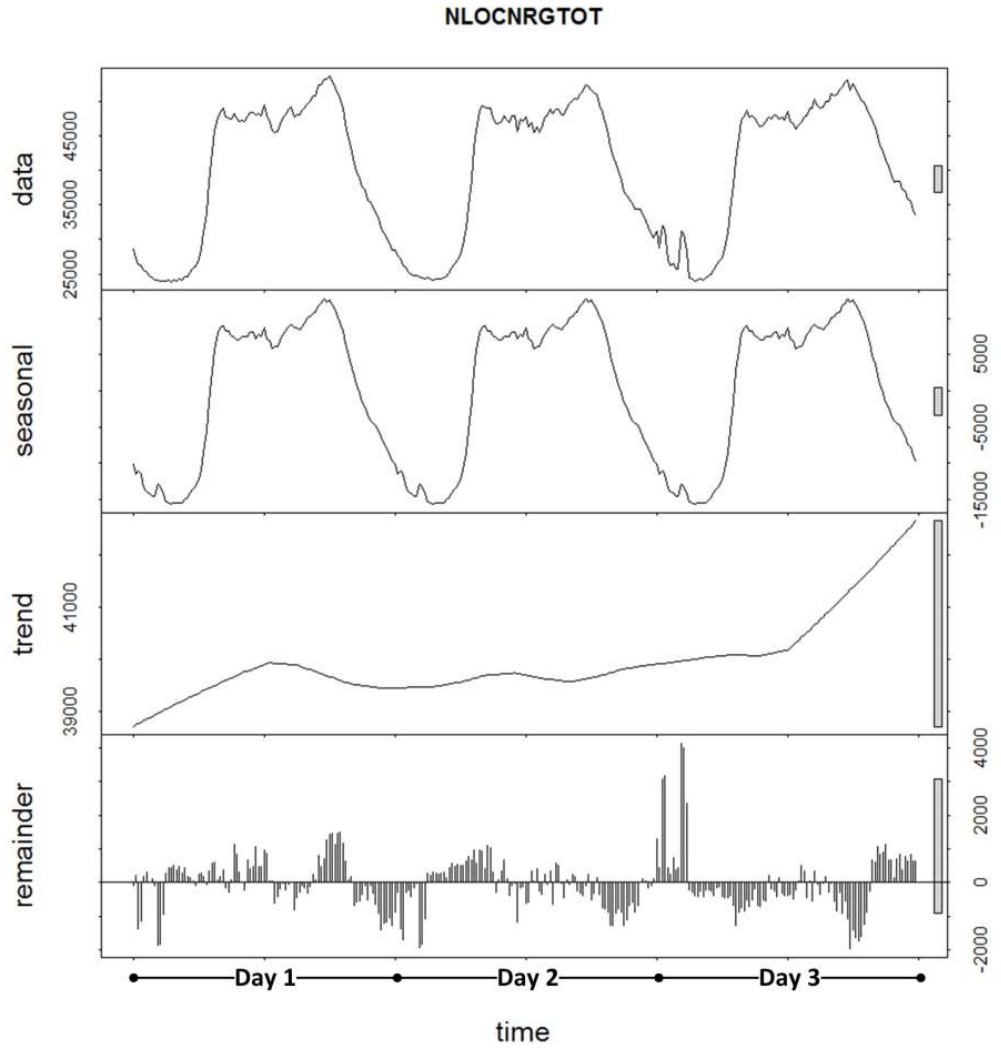


Figure 3.21.: Time series of the traffic variable NLocNRgTot, from the TI Dataset, along with its seasonal, trend and remainder components.

Multidimensional Scaling

Next we project the data into the 2-dimensional space, using MDS with the Canberra distance, in order to examine whether the dataset will produce a consistent shape that could be used as a reference model for the anomaly detection algorithms. This is confirmed in Figure 3.22, where it can be observed that the dataset exhibits very strong periodicity, showing almost identical shapes for the three days except for a small deviation on day 1.

Entropy

Examining the information content of the traffic variables in Table 3.3 shows that all the variables have very large entropy values. Also, the entropy of the *seasonal* and *trend* components are almost identical, suggesting that the time series are very similar. This is also verified by the correlation matrix, shown in Figure 3.25. All the features appear to be suitable for anomaly detection, but clearly they contain redundant information which is better identified through the correlation analysis presented in the following section.

Table 3.3.: Entropy of all traffic variables and extracted features from TI dataset, where O denotes the original time series, D their derivatives, and T, S, R are respectively the trend, seasonal and remainder components. The values range from 0 to 5.663.

Name	**H(O)**	**H(D)**	**H(T)**	**H(S)**	**H(R)**
NLocNRgTot	5.663	5.645	5.663	3.178	5.663
NLocNRgSucc	5.653	5.587	5.663	3.178	5.663
NLocOldTot	5.658	5.631	5.663	3.178	5.663
NLocOldSucc	5.663	5.635	5.663	3.178	5.663
NAutReqTot	5.663	5.65	5.663	3.178	5.663
NAutReqSucc	5.663	5.64	5.663	3.178	5.663

3.3.2. Correlation Analysis

Figure 3.23 visualises the correlation matrices of the original traffic variables using Pearson's, Spearman's and Kendall's coefficients. In all cases, there is very strong correlation between the attempt-success pairs, which provides a useful feature for the detection of signaling DoS attacks that generate service requests (i.e. attempts) but do not complete the transactions (i.e. successes). In this case, correlation does imply causation, since an

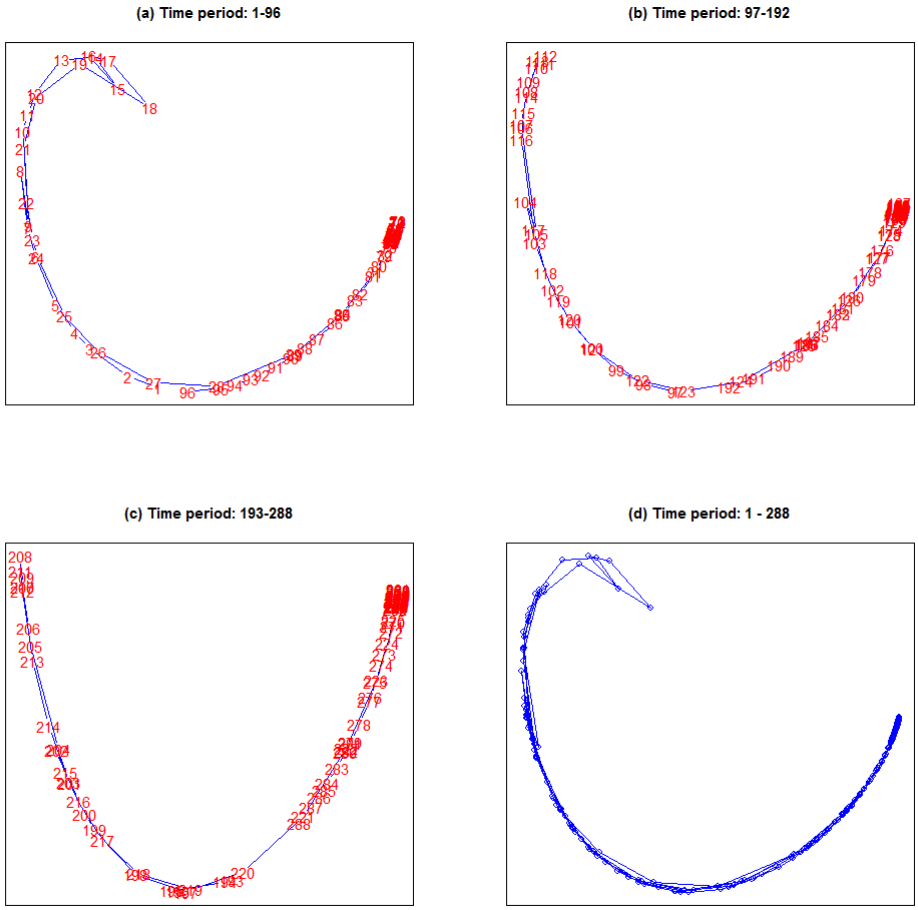


Figure 3.22.: Projection of all the traffic instances from TI dataset in the 2-dimensional space using MDS with the Canberra distance metric: (a)-(c) correspond to the three days contained in the dataset, and (d) shows all the days combined.

increase in the number of successfully served requests (e.g., authentication requests) is strongly influenced by the number of attempts made. Furthermore, both Pearson's and Spearman's coefficients show a strong coupling between the authentication procedure and the location updates sent by registered users, both between attempts and successes. Interestingly, the same level of correlation is not observed for non-registered users. Moreover, NLocNRgTot and NLocNRgSucc appear to be correlated with the authentication attempts and successes. However, this is significant only when using Spearman's coefficient, and less with Pearson's, while Kendall's yields the smallest value. This is expected since Kendall's coefficient is more suitable for ordinal-level variables.

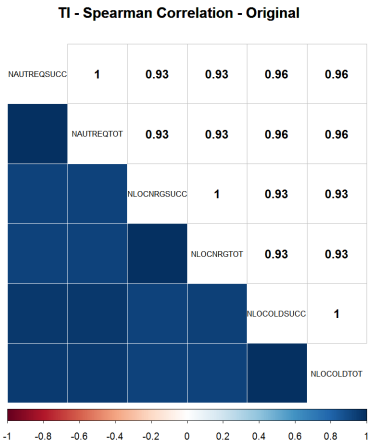
The correlation matrix of the derivatives of the time series is shown in Figure 3.24, where we can see that the growth rate of the attempt-success pairs follows a linear relationship, since Pearson's coefficient is 1. The reason for this linearity is that the volume of successfully served requests is merely a proportion of the volume of the attempted transactions. Figures 3.24(a) and (b) also show that the growth of all the traffic variables is strongly associated, which is due to the fact that both registered and unregistered subscribers trigger location registrations and authentication attempts at similar rates during different times of the day. This is an important observation that will be utilised in Section 3.3.3 where artificial anomalies are injected in the traffic.

Apart from the correlated pairs (attempt-success) that are seen in most of the features, Figure 3.25 reveals a very strong correlation (~ 1) between the seasonal components of NLocNRgTot, NLocNRgSucc, NLocOldTot and NLocOldSucc. These features form a correlation cluster which is justified considering the mobility habits of the network subscribers (e.g. commuting to work). More specifically, the volume of location registration requests is proportional to the mobility of subscribers, and both registered and unregistered users exhibit similar mobility patterns. In Figure 3.26 we see that all pairs of variables exhibit a strong correlation ~ 1 in the trend component, which is consistent with Figures 3.19, 3.20 and 3.21, where the variables appear to follow similar daily trend; since they all mandate very similar functions of the network, their trend is controlled by the same subscriber habits (e.g. increased usage in specific days of the month). Similarly, Figure 3.27 shows very strong linear correlations between all the remainder components of the traffic variables. This indicates that a benign *usage spike* in the network traffic will affect all the variables, whereas a malicious one may affect only few of them.

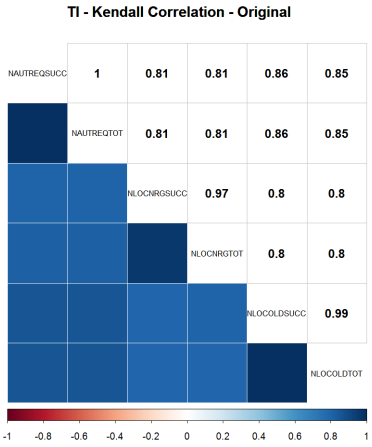
3.3.3. Correlation Results During Abnormal Incidents

We expect that signaling anomalies in mobile networks will affect a limited number of traffic variables, and this observation could assist in the identification of such anomalies. Specifically, we expect that during a signaling DoS attack, the traffic variables that are

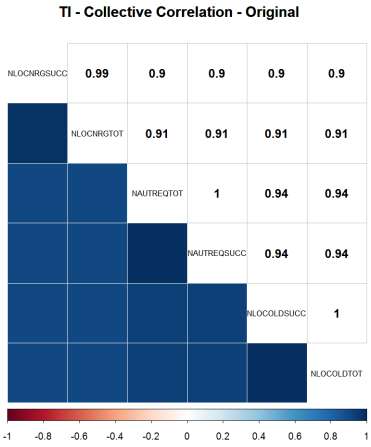
(a) Pearson's Correlation



(b) Spearman's Correlation

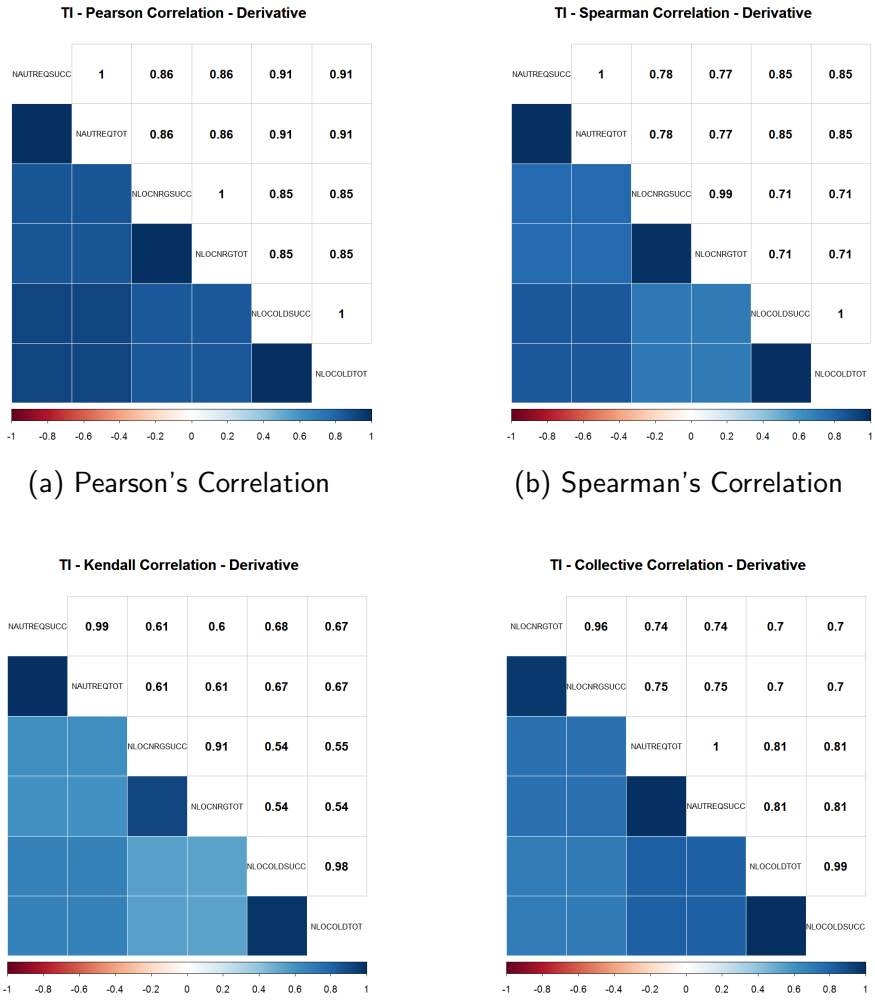


(c) Kendall's Correlation



(d) Collective Correlation

Figure 3.23.: Visualisation of the correlation matrices for the traffic variables of the TI dataset. The blue and red values correspond to high positive and negative correlations respectively. The lower triangle of the matrix visually presents the correlation strength, while the upper part shows the actual values. The collective correlation is the mean value of the three correlation coefficients.



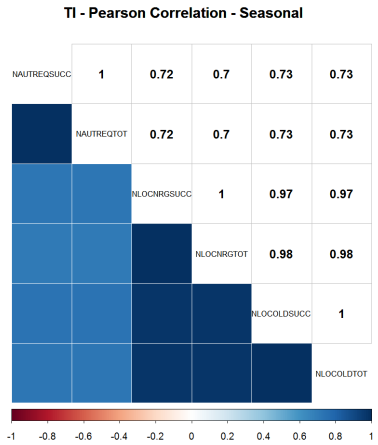
(a) Pearson's Correlation

(b) Spearman's Correlation

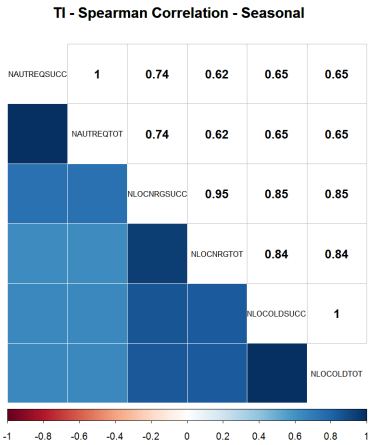
(c) Kendall's Correlation

(d) Collective Correlation

Figure 3.24.: Visualisation of the correlation matrices for derivative of the traffic variables of the TI dataset. The blue and red values correspond to high positive and negative correlations, respectively. The lower triangle of the matrix visually presents the correlation strength, while the upper part shows the actual correlation values. The collective correlation is the mean value of the other three correlation coefficients.

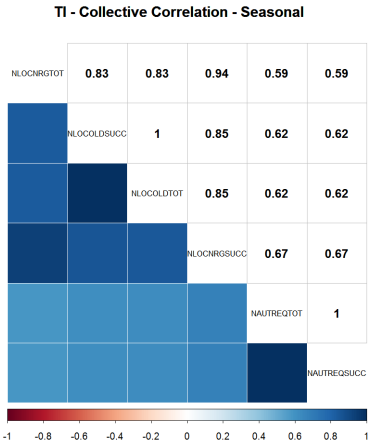


(a) Pearson's Correlation



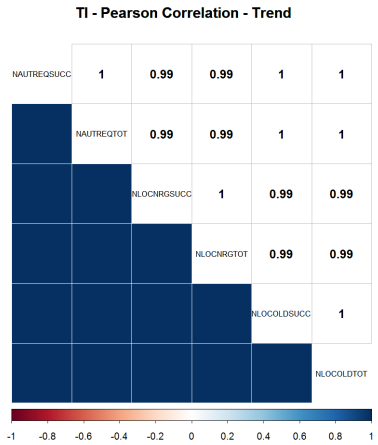
(b) Spearman's Correlation

(c) Kendall's Correlation

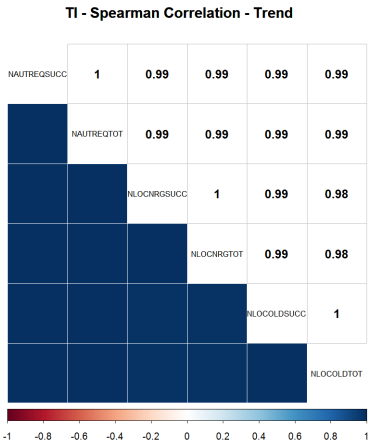


(d) Collective Correlation

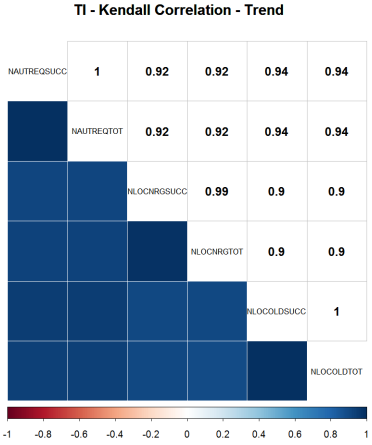
Figure 3.25.: Visualisation of the correlation matrices for the seasonal components of the traffic variables of the TI dataset. The blue and red values correspond to high positive and negative correlations, respectively. The lower triangle of the matrix visually presents the correlation strength, while the upper part shows the actual correlation values. The collective correlation is the mean value of the other three correlation coefficients.



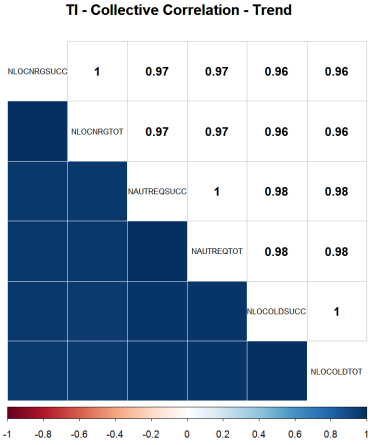
(a) Pearson's Correlation



(b) Spearman's Correlation

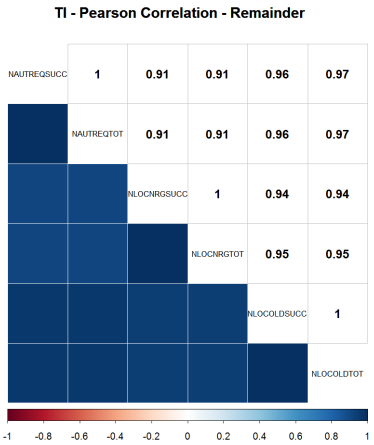


(c) Kendall's Correlation

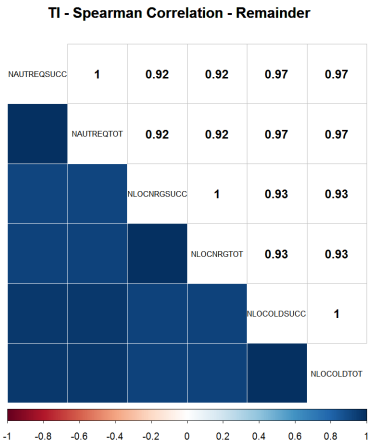


(d) Collective Correlation

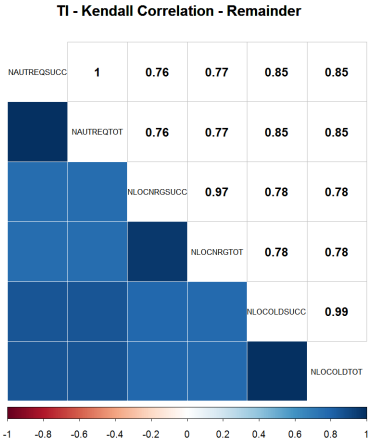
Figure 3.26.: Visualisation of the correlation matrices for the trend components of the traffic variables of the TI dataset. The blue and red values correspond to high positive and negative correlations, respectively. The lower triangle of the matrix visually presents the correlation strength, while the upper part shows the actual correlation values. The collective correlation is the mean value of the other three correlation coefficients.



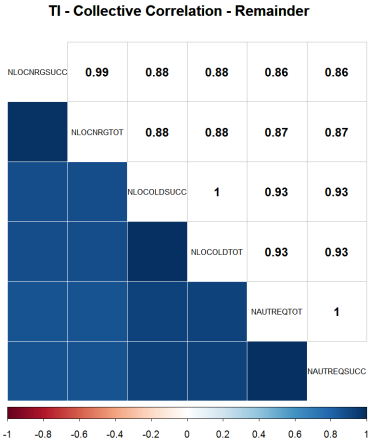
(a) Pearson's Correlation



(b) Spearman's Correlation



(c) Kendall's Correlation



(d) Collective Correlation

Figure 3.27.: Visualisation of the correlation matrices for the remainder components of traffic variables of the TI dataset. The blue and red values correspond to higher positive and negative correlation respectively. The blue and red values correspond to high positive and negative correlations, respectively. The lower triangle of the matrix visually presents the correlation strength, while the upper part shows the actual correlation values. The collective correlation is the mean value of the other three correlation coefficients.

exploited by the attacker will deviate from their normal correlations with other variables, hence providing useful features for anomaly detection algorithms. In order to validate this hypothesis, we conducted a number of experiments, one of which is presented here, where the original TI dataset was modified to include three anomalous instances. The values of these instances were selected based on the literature [2, 6, 41, 47, 105].

The synthetic attack dataset was created assuming that an attacker generates excessive number of requests for the traffic variable NAutReqTot. The results in Figure 3.28 shows that the correlations between NAutReqTot and the other variables are significantly affected. Further, the deviations from the expected correlations are more pronounced in the *seasonal component* of the time series and the *derivative*. These two features are more sensitive to respectively long and short term changes and hence should be used as inputs to the anomaly detection algorithms. Finally, we see in Figure 3.29 that the anomalous instances 100, 101, 102 deviate from the normal U shape that demonstrates a pattern of periodic behaviour with a 24 hour period, and they appear as outliers.

3.4. Summary and Future Work

This chapter presented and analysed features extracted from signalling traces collected from 3G/4G operational mobile networks and to be used by the anomaly detection algorithms in WP4. Towards this goal, the attributes of the original traffic variables and the extracted features were examined to gain an understanding of their reference behaviour and identify the information they provide about the network. The features were extracted using techniques such as time series decomposition and dimensionality reduction. Furthermore, we examined the correlations between the extracted features, and identified some underlying patterns in the signaling traffic. This was achieved by visualising the relationships between the variables, based on different correlation coefficients, under normal conditions as well as in the presence of anomalous traffic instances. Moreover, our analysis of anomalous traffic instances revealed that the *derivatives* and *seasonal component* features, in addition to the MDS technique are the most capable of revealing deviations from standard behaviour.

The work presented in this chapter will be extended in several directions, and the results will be included in the final version of deliverable D5.1.2. First, the decomposition analysis of the traffic variables will be performed on data traces over longer periods of time when they become available. This will reveal, for instance, if there is seasonality hidden within larger periods of time (e.g. week, month, etc.) apart from the 24-hour periodicity observed in the current dataset. In turn, this allows anomaly detection algorithms to identify both short and long term anomalies that may otherwise be hidden

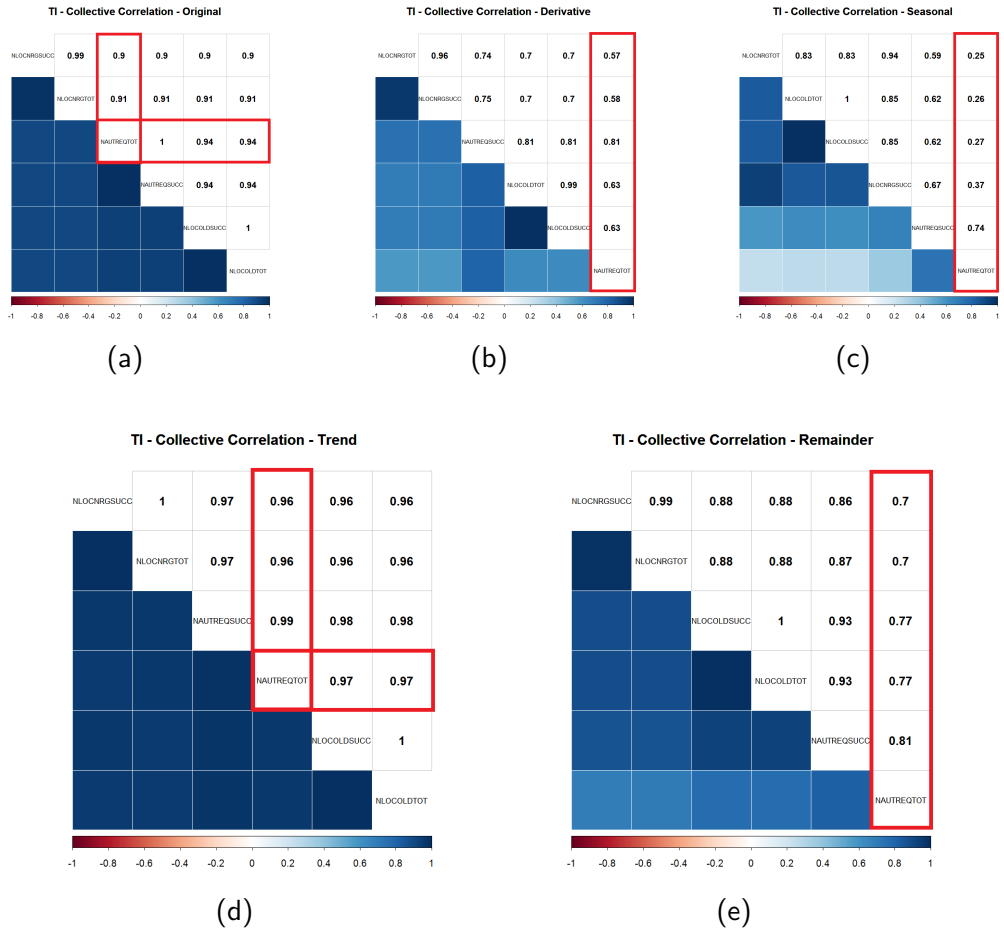


Figure 3.28.: Visualisation of the collective correlation matrices for the MSC dataset when 3 anomalous instances are injected: (a) original variables, (b) their derivatives, and (c), (d) and (e) are the seasonal, trend and remainder components, respectively. The red box indicates the correlation vector of the affected traffic variable.

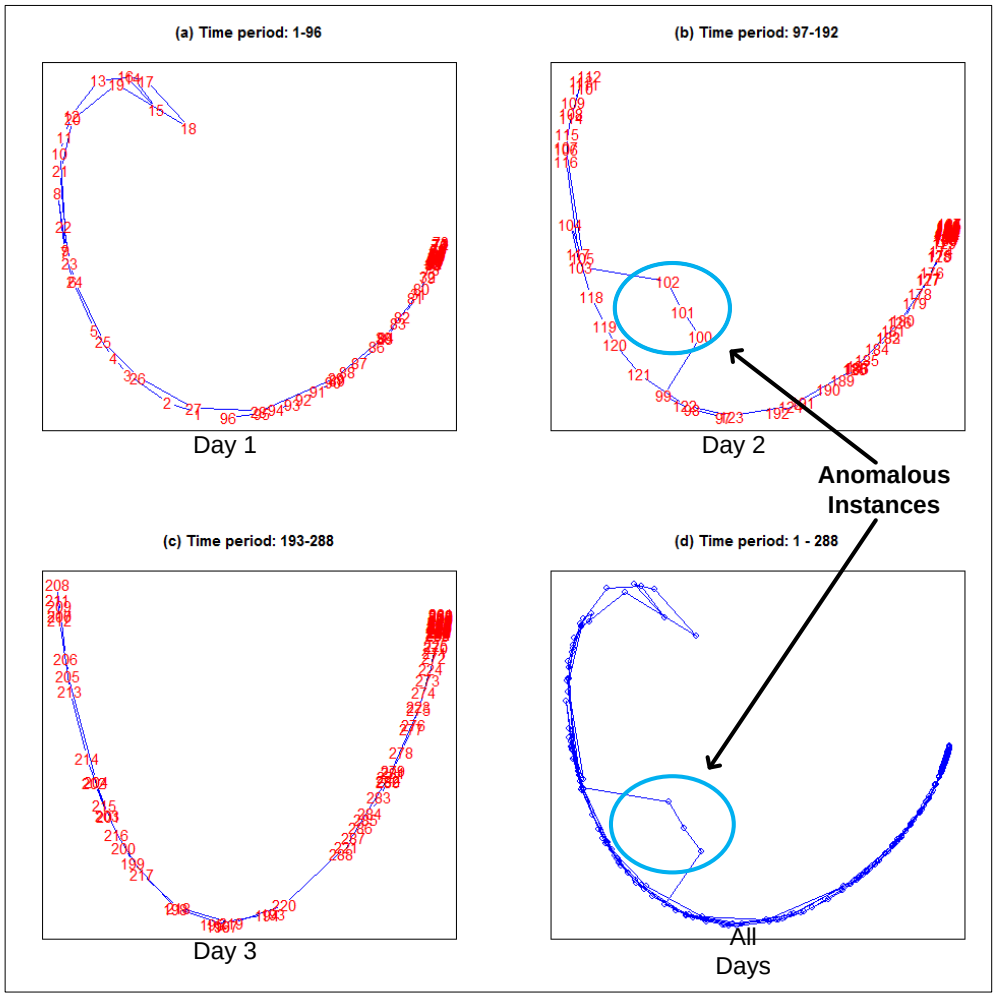


Figure 3.29.: Projection of the MSC-attack dataset in 2D using MDS with Canberra distance. The anomalies are observed on the second day in (b) and (d).

in the raw data. We also plan to combine the results of the correlation analysis with the entropy measurements in order to build upon and improve existing dimensionality reduction techniques. The planned work also includes the comparison of the developed dimensionality reduction method with popular techniques such as mRMR. The results presented in this report will be utilised by the anomaly detection algorithms developed in WP4 in order to reduce the monitored variables, process the data in a timely manner and accurately detect signaling anomalies.

4. Graph-based Correlation

4.1. Introduction

This chapter presents a graph-based approach for abnormal event detection using billing related data, which are known as call detail records (CDR) in 3GPP standards. The main methodology proposed in the context of NEMESYS is the transformation of the raw input data into graph representations using social and k-partite graphs:

Social graphs: Each vertex of this graph represents a user, while the edges represent user interactions, e.g. voice calls, data transfers and SMS. Figure 4.7 shows an example social graph that shows two different user clusters and the interactions between them.

K-partite graphs: Each vertex of this graph represents a different entity from a selected set of the data attributes. These attributes can be users, dates, and communication types, e.g. SMS, voice calls, and Internet traffic. There are multiple sets, and therefore multiple attributes, represented by the graph, and graph edges correspond to interactions between entities in different sets. See Figure 4.4 for an example k-partite graph constructed from the CDR data analysed in this chapter.

After the data is transformed as a graph, we utilise similarity or distance measures between graphs, such as the ones presented in Section 2.3.4, which are built based on parameters such as time or users. Based on these measures, we find the events or users that deviate from the normal behaviour, as presented in Figure 4.1. We discuss social graphs and k-partite graphs in more detail in the following sections, after providing some information on the dataset used to illustrate the graph-based approaches.

4.2. The IEEE VAST'08 CDR dataset

The IEEE Symposium on Visual Analytics Science and Technology (VAST) annually holds a data analytics challenge, and releases a number of datasets to be used by the competitors. For the IEEE VAST 2008 challenge, four heterogeneous synthetic datasets were provided; we use the *cellphone calls* dataset from VAST'08 to illustrate our proposed

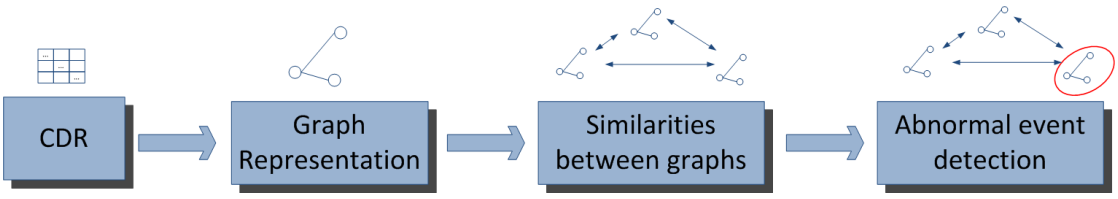


Figure 4.1.: The architecture of the graph-based abnormal event detection scheme.

Event no.	From	To	Date-Time	Duration (sec.)	Cell id
1	349	23	20060601 0008	1634	1
2	379	364	20060601 0011	640	24
3	392	27	20060601 0012	1182	29
4	17	339	20060601 0014	578	23
5	272	251	20060601 0015	887	29
6	282	13	20060601 0015	1015	19
7	28	5	20060601 0045	1175	22
8	209	71	20060601 0054	1120	28
9	80	120	20060601 0058	588	11

Table 4.1.: Sample events from the IEEE VAST'08 dataset. Each row is a network event, i.e. a mobile voice call, and the columns are the event attributes.

graph-based anomaly detection and correlation methods. Table 4.1 shows a sample from the VAST'08 dataset. This dataset contains the following attributes for each event, i.e. each call record:

- *From* is the unique identifier for the subscriber who initiated the voice call,
- *To* is the unique identifier for the subscriber who received the call,
- *Datetime* is the timestamp of the event, i.e. when the call was initiated,
- *Duration* is the length of the call in seconds, and
- *Cell id* is the unique identifier of the cell in which the caller was camping when the call was initiated.

4.3. Social Graphs

We consider network events occurring in the mobile network, and model this activity as a directional graph that contains all subscribers and network events. An *event* in this context is a single "billable" occurrence of network service usage as captured by components in the mobile core network for charging and billing purposes. For instance, when a mobile user makes a voice call, an event is generated which contains information about that phone call, such as the call duration, the date and time when the call was made, the number that was called, etc. Such an event is called a *Call Detail Record (CDR)*, and they are suitable for the detection and correlation of suspicious service usage patterns in order to uncover attacks that are reflected in the billing records. The most common attributes in CDRs are:

- Originating subscriber, i.e. caller,
- Terminating subscriber, i.e. callee,
- Event timestamp,
- Service usage duration, and
- Service type (voice call, SMS, data, etc.).

Let us define the social graph that captures the network activity as the *directed* graph $G = (V, E, L, F)$, which is sampled over time using a constant period dt. Every time instance t_i, where $dt = t_{i+1} - t_i$, results in a new graph $G_i = (V_i, E_i, L_i, F_i)$, where $V_i = \{v_0, v_1, \ldots, v_n\}$ is the set of nodes representing the subscribers at that time instance, and $E_i = \{e_0, e_1, \ldots, e_k\}$ are the directed interactions between the subscribers, e.g. voice calls and SMS. Furthermore, G_i contains a set of labels $L_i = \{l_0, l_1, \ldots, l_z\}$, where each label l_j, $j \in [1, z]$ corresponds to a network service (e.g. voice call, SMS). Finally, we define the mapping function $F_i : E_i \to L_i$, which assigns a label to each edge.

Below, we discuss how social graphs can be used to analyse network activity in order to uncover anomalous behaviour such as SMS spam.

4.3.1. Analysis of SMS spam

Mobile malware is clearly on the rise [33, 55, 94], and one of the most common mobile malware is SMS spammers, which operate as follows. Initially, the user receives an SMS with a malicious URL, or clicks on a malicious advertisement that links to a malicious URL. As a result of visiting the malicious website, the mobile is infected with malware,

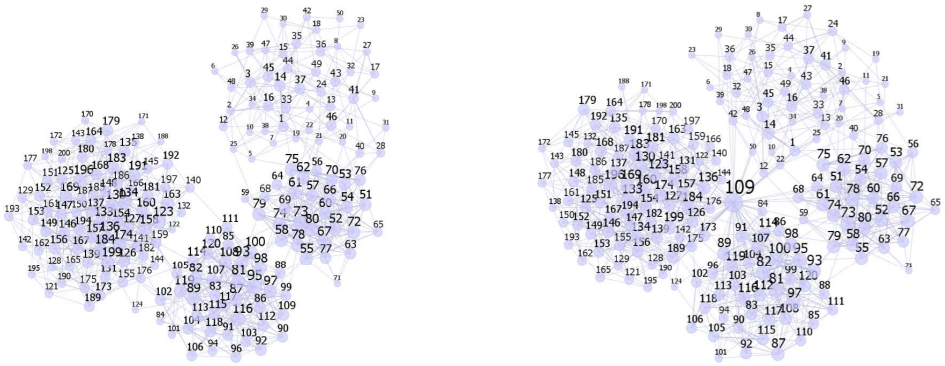
which often masquerades as a benign application. The malware usually turns the infected device into a botclient, which receives command and control (C2) messages from one or more C2 servers, allowing the botmaster control over the behaviour of the infected devices, i.e. the mobile botnet. The botclient normally receives a list of mobile numbers and spam messages [99], upon which it sends the spam SMS messages from the list to the given mobile numbers. This procedure is repeated periodically, possibly with different lists of numbers and messages depending on the configuration of the botnet and the instructions of the botmaster.

Social graphs are useful for the detection of SMS spam activity in the network, and also for the identification of the infected users, enabling the service provider to inform the infected users and to mitigate the effects of the SMS spam, for example by dropping the SMS messages identified as spam at the SMS-C server. In order to demonstrate this procedure, two synthetic datasets were generated: (i) normal SMS traffic, shown in Figure 4.3(a), and (ii) spam SMS traffic sent from one compromised device, shown in Figure 4.3(b). For the synthetic generation of CDR records, the social characteristics of user behaviour were taken into account. Specifically, the social graph created from the CDR records has a structure in which individuals within communities tend to be linked via strong ties as represented by the large number of communication events between them, whereas communities tend to be connected to other communities via weak ties [85]. This behavior is captured in Figure 4.3(a), where four distinct communities with a total size of 200 users were synthetically generated. The intra-community communication rate is high (3–6 SMS per user), and it is uniformly distributed within the community, while the number of inter-community communications is small (0.02–0.06 SMS per user) [82]. For the generation of the spam SMS dataset, a compromised device was added to the network, which sent SMS to a list of 50 random numbers [53,61] not necessarily belonging to the same community. This behaviour is captured in Figure 4.3(b), which shows that user 109 sends spam SMS to a specific set of users that belong to different communities. By comparing the graphs for expected or normal behaviour and for what is currently observed in the network, anomalies such as SMS spam can be detected, as illustrated by this example.

4.4. K-partite Graphs

In this section, we discuss how k-partite graphs can be used to represent network activity and for network analysis. Unlike the social graph based method presented above, the k-partite graph captures all attributes of network activity, including cell ids and date information. Thus, the k-partite graph model takes into account more information than

Figure 4.2.: The modus-operandi of the SMS spam malware: 1) The user receives an SMS with a malicious URL. 2) He visits the URL and gets infected with malware. 3) The infected device receives instructions from the C2 server. 4) The infected device sends the spam SMS messages to mobile numbers. 5) The infected device retrieves a new list of numbers or waits for further instructions.



(a) Normal SMS traffic. The nodes of the graph represent users, while the edges represent SMS exchanged between users. We can identify four distinct communities based on social interactions.

(b) Spam SMS traffic. User 109 sends spam SMS to multiple users belonging to different communities.

Figure 4.3.: Social graphs capturing SMS-based activity in the mobile network. This example is based on synthetically generated data.

the social graph, and can be utilised for detection of a wider set of abnormal events, including botnets and changes in user behaviour.

In k-partite graphs, vertices are divided into k disjoint groups $\{V_1, \ldots, , V_k\}$, such that no edge connects the vertices in the *same* group. More formally, a k-partite graph G is defined as: $G = (V, E)$, where $V = V_1 \cup V_2 \cup \ldots V_k$, $V_j = \{n_i \mid 1 \leq i \leq N_j\}$, $\forall\, j \in [1, k]$, N_j is the number of vertices in the j-th vertex group V_j, and $E \subset \bigcup_{j=2}^{k} \{V_1 \times V_j\}$. The CDR data is mapped to a k-partite graph in the following way: nodes in V_1 correspond to unique CDR records, while nodes in $V_{j>1}$ correspond to attribute values for each record. Therefore, a k-partite graph shows the connections of the CDR records to attribute values of $k-1$ different record attributes.

Figure 4.4 shows a very simple k-partite graph built using the CDR records given in Table 4.2. Since the example CDR set has 4 attributes, the k-partite graph has 4 disjoint groups, represented with different colors in the figure. The records are represented by the two "event id" vertices colored in white, and the data is captured in the form of edges connecting each record to its attribute values.

Event id	From	To	Date
0	1	2	2006-10-01
1	1	3	2006-10-02

Table 4.2.: Sample CDR records used to illustrate how k-partite graphs are constructed from CDR records. Note that the number of attributes and records in the dataset have been significantly reduced for this example. We have added the "event id" attribute to the dataset as a unique identifier for each record.

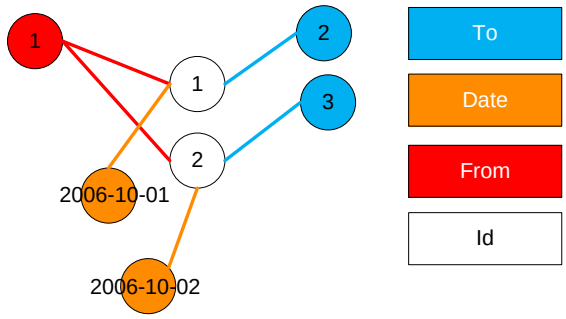


Figure 4.4.: The k-partite graph representation of the CDR dataset presented in Table 4.2. The "event id" nodes uniquely identify each CDR record, and are connected to the attribute values. Different attributes are shown with different colors.

4.5. Graph-based Abnormal Event Detection

This section presents the procedure of applying the graph representations of the raw data for abnormal event detection. The main procedure consists of the following steps:

- Build a series of graphs based on specific parameters, such as time stamp, or signalling message.
- Use graph matching to find the distances between every pair of graphs.
- Detect abnormal graphs, which deviate from the normal behaviour. This abnormal event detection procedure is based on multi-dimensional scaling (MDS).

Note that we illustrate the procedure of "constructing a sequence of graphs capturing network activity at different instances in time, applying a graph matching method to the graphs, and performing anomaly detection based on the similarity score of the graphs" using k-partite graphs only. However, the procedure is the same for social graphs, which are useful to capture anomalies in the social structure of the mobile users.

4.5.1. Building a Graph Sequence for Abnormal Event Detection

A series of graphs is created in order to represent changes based on specific parameters. The assumption is that abnormal events will cause large changes in the graph weights and/or structure, which can be detected automatically utilising the graph matching techniques discussed in Section 2.3.4.

There are many ways to build the series of graphs based on the task at hand. For example, in the case that the task is detection of anomalous periods of activity, the parameter by which the sequence of graphs is created is time, i.e. each graph represents the activity (signalling or CDR) over a different time point. A specific list of parameters that are utilised for building of the graph sequence is presented below:

- Time: Detection of abnormal time periods
- User: Detection of abnormal users, whose behaviour deviates from the rest
- Time/User: Detection of time periods during which a specific user changed behaviour.
- Signalling Message: Detection of abnormal signalling messages, whose behaviour deviates from the rest.

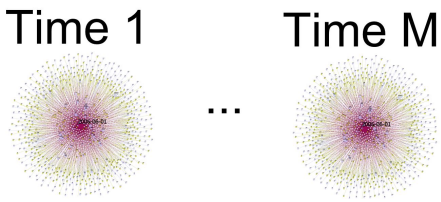


Figure 4.5.: Creation of a sequence of k-partite graphs utilising the time parameter. Each graph represents the k-partite representation of the CDR activity over a specific time instance.

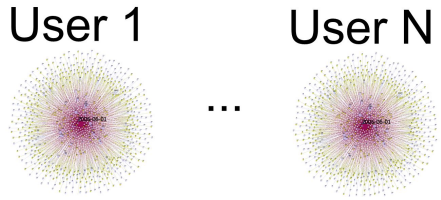


Figure 4.6.: Creation of a sequence of k-partite graphs utilising the user parameter. Each graph represents the k-partite representation of the CDR activity of a specific user.

It should be noted that this list is not exhaustive, but is found to be the most useful in detecting anomalies, and is the one implemented in the abnormal event detection prototype.

Examples of building the sequence of graphs are depicted in Figures 4.5 and 4.6. Figure 4.5 shows an example of graph sequence built taking into account the time parameter, i.e. each graph in the sequence represents the CDR activity of a specific time instance, while Figure 4.5 illustrates an example of graph sequence built taking into account the user parameter, i.e. each graph represents the CDR activity of a specific user.

4.5.2. Graph Matching

The use of graph matching is the first step towards abnormal event detection. The main goal is to utilise the generated graph sequence so as to find specific graphs which are very different from the rest, and thus deviate from the normal behaviour. The assumption is that abnormal events, such as DDoS attack or spam SMS campaigns, create graphs in which the structure and/or the weights of its elements are distinctly different from the

rest of the graphs, and thus can be detected through graph matching methodologies.

Section 2.3.4 presented a review of the graph matching techniques that have been proposed in the literature. Each of the proposed methods takes as input two graphs $G_1, G_2 \in \mathbb{G}$, and returns their structural similarity or dissimilarity (distance), i.e. $d : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{R}^+$, where $\mathbb{G}$ is the domain of graphs, and $\mathbb{R}^+$ is the space of positive real numbers. Following this definition, and using the generated graph sequence $G_1, \ldots, G_N$, the distance matrix $\mathbf{D} = [d_{ij}]$ is built, where each element d_{ij} represents the distance of the graphs G_i, G_j, i.e. $d_{ij} = d(G_i, G_j)$.

For the exact calculation of the distance between two graphs, Graph Edit Distance (GED) methodologies are utilised (cf. Section 2.3.4), due to their simplicity, speed, and accuracy in finding anomalies. Specifically two GED distance metrics are used. The first, defined in Eq. 2.15, takes into account only the structural characteristics of the graphs, namely the number of nodes and edges of graph G_i and the number of the common nodes and edges in the two graphs. Thus the distance function contains the number of graph operations that happens through the transition $G_i \rightarrow G_j$, i.e. the aggregation of all the graph's node and edge deletions and insertions. The main disadvantage of this GED metric is that it does not take into account the weights of the input graphs. Thus, it will give the same result in graphs which have the same edge connections, but different weights on them. In such cases, it is more appropriate to use a distance formula that embeds the difference of the edge weights. One such metric which takes into account the Euclidian distances of the weights of the adjacency matrices is given by [64]:

$$d_{ij} = \sqrt{\sum_{r=1}^{k} \sum_{c=r+1}^{k} \left(\mathbf{H}_i\left(r,c\right) - \mathbf{H}_j\left(r,c\right)\right)^2} \tag{4.1}$$

Two additional similarity metrics from the literature have also been considered. These metrics have been proposed for the comparison of two sets, but can also be applied to graphs. Specifically, they are applied in order to find similarities between users and time periods. The first is the Jaccard index [62] which is defined as follows:

$$s_{JI}\left(G_k, G_z\right) = \frac{|V_k \cap V_z|}{|V_k \cup V_z|} \tag{4.2}$$

This metric captures the degree in which the two sets of nodes have common entries, or in other words the two graphs have common behaviour.

Finally, the last similarity metric is the Sorensen-Dice index [107], which is defined as follows:

$$s_{SD}\left(G_k, G_z\right) = \frac{2\,|V_k \cap V_z|}{|V_k| + |V_z|} \tag{4.3}$$

4.5.3. Abnormal event detection using k-partite graphs

As described earlier, the CDR dataset comes from the IEEE VAST 2008 challenge. In this challenge, a group of users changed their mobile devices in the last three days of the monitoring period. The goal of the challenge was to detect this behaviour through visual analytics techniques. The procedure for detecting anomalies utilising graph representations can be summarised in the following: (i) create a sequence of graphs $G_1, \cdots, G_N$ based on a specific parameter, (ii) calculate the pairwise distances in order to construct the distance matrix $\mathbf{D} = [d_{ij}]$, and (iii) use visualisation methods to identify anomalous graphs in the sequence. However, the last step requires the initial graphs to be transformed into a low dimensional space, using any of the dimensionality reduction techniques described in Section 2.2.4. The results presented in the sequel are obtained by applying the MDS method (cf. Section 2.2.4) which produces N 2D coordinates whose distances are proportional to the distances between the graphs.

The first objective is to detect abnormal time periods, which means that the sequence of graphs has to be created taking into account the time parameter. Thus, each k-partite graph in the sequence represents the CDR activity of one hour, to a total of ten days. The result is depicted in Figure 4.7. Two clusters have emerged, one representing the CDR activity in the first seven days, and one in the last three days. This means that the CDR activity changed in the last three days, a fact which needs further investigation.

In addition to the hour parameter, the day parameter has also been utilised to create the sequence of graphs. Thus, each k-partite graph in the sequence represents the CDR activity of one day, to a total of ten days. Two GED based distance metrics are used, which are described in Eqs. (2.15) and (4.1). The result of applying these two distance metrics and the MDS method is depicted in Figure 4.8. In both cases, the differentiation in the last three days is apparent. Indeed in the last three days, a certain set of users changed phone numbers and continued to use the network services for communication. This behaviour is suspicious and needs further investigation, which could be facilitated through the visualisation methods developed in T5.2. The use of the day granularity helps in the analysis of past behaviour, root cause analysis, and identification of malware infection vectors.

The second objective is the detection of anomalous users, i.e. users that have different behaviour from the rest. To this end, the sequence of graphs is built based on the user parameter. Thus, each k-partite graph in the sequence represents the CDR activity of a specific user. As in the previous case, the two GED metrics (2.15) and (4.1) are used. MDS is then applied to the distance matrix derived from these metrics, in order to project the data onto the 2D plane, and create user friendly visualisation of the behaviour of the users. As shown in Figure 4.9, anomalous users stand out from the crowd of normal

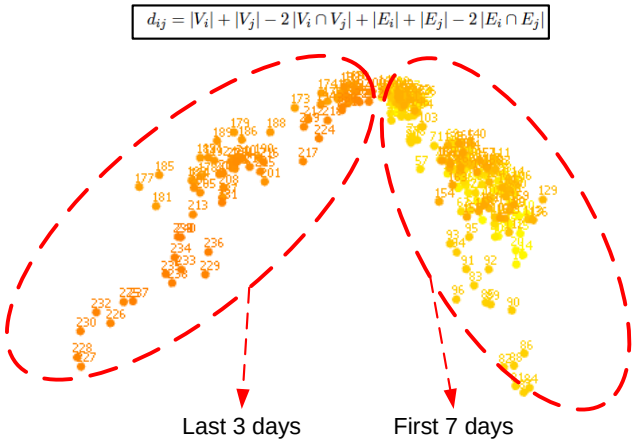


Figure 4.7.: Application of the abnormal event detection technique which utilises MDS and dissimilarity-based graph matching. The sequence of graphs is created using the hour parameter, i.e. each graph represents one hour of CDR activity. Two clusters have emerged, one representing the CDR activity in the first seven days, and one in the last three days.

users, and are easily detected using the proposed approach. Specifically, the following users are found to be anomalous in Figure 4.9: 1,2,3,5,306, and 309. According to the solution of the VAST challenge, these users are indeed included in the set of anomalous users. This demonstrates the efficiency of the proposed approach in finding anomalous dates and users.

In addition to MDS, similarity matrices have also been used to illustrate similarities between distinct users and time periods. The results of the application of the Jaccard index and SorensenDice coefficient similarity metrics, described in Section 4.5.2, are depicted in Figures 4.10 and 4.11. Specifically, Figure 4.10 shows the similarities between multiple graphs, each representing the CDR activity of one day in the CDR dataset. Filtering has been applied in order to focus on the most important events. It is clear in both metrics that two distinct clusters of activity are observed, one in the first seven days, and one in the last three days. This type of behaviour was also apparent in the MDS representation depicted in Figure 4.8, and is indeed part of the solution of the challenge.

Figure 4.11 illustrates the result of the application of the similarity metrics to the most active users during the ten day period. The aim is the identification of users

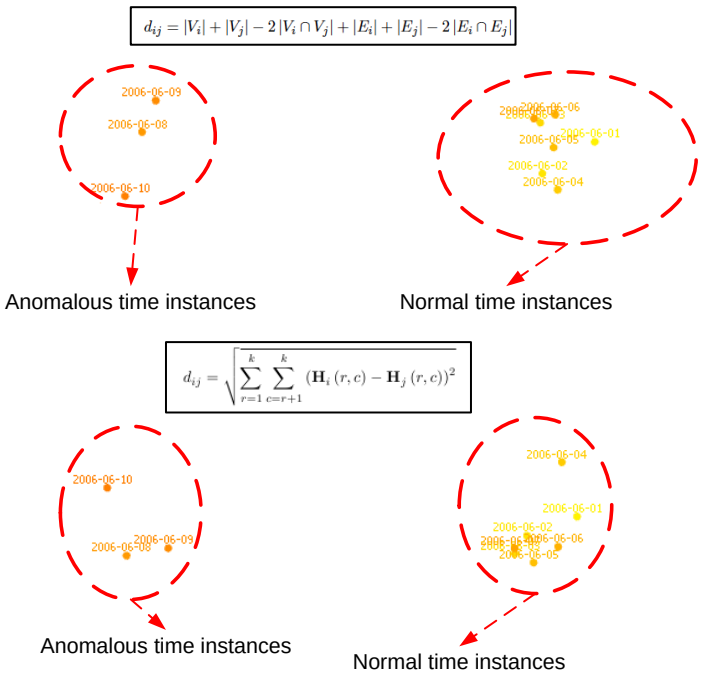


Figure 4.8.: Application of the MDS abnormal event detection method and the GED in Eqs. (2.15) and (4.1). The sequence of graphs is created using the day parameter, i.e. each graph represents one day of CDR activity.

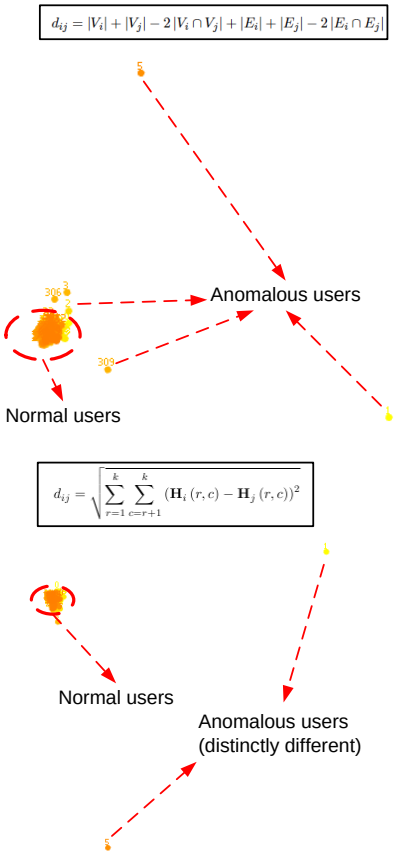
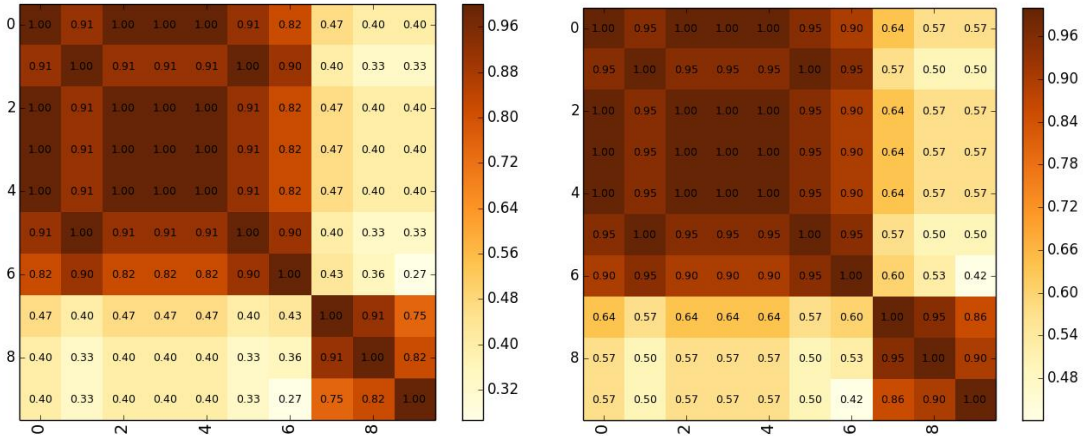
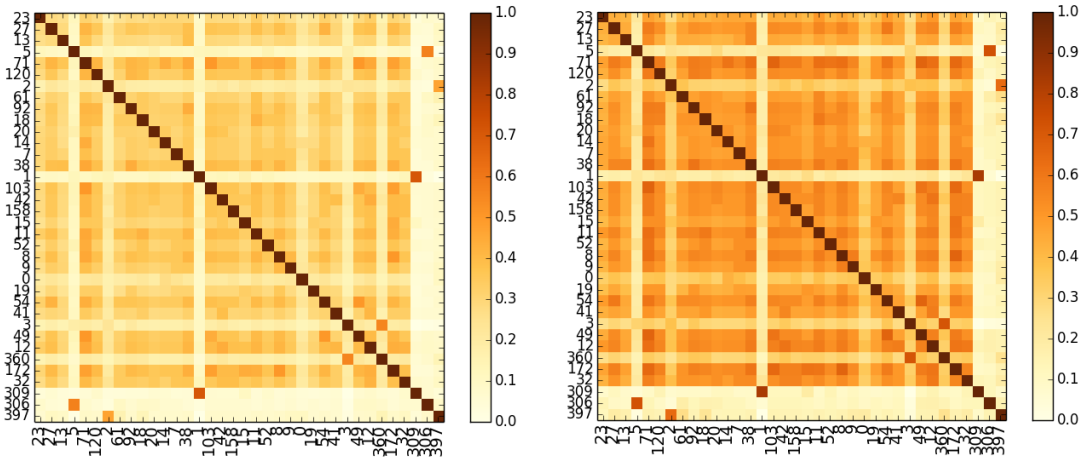


Figure 4.9.: Application of the MDS abnormal event detection method and the GED in Eqs. (2.15) and (4.1). The sequence of graphs is created utilising the user parameter, i.e. each graph represents the CDR activity of one user. The anomalous users are users: 1,2,3,5,306, and 309. These users are indeed responsible for the anomalies observed in the VAST 2008 challenge.



(a) Jaccard index (b) Sorensen-Dice coefficient

Figure 4.10.: The matrices between the ten days of CDR activity. Red color represents high and yellow low similarity between the days of the corresponding row and column. Two clusters are clearly visible, one regarding the first seven days and one for the last three.



(a) Jaccard index (b) Sorensen-Dice coefficient

Figure 4.11.: The similarity matrices between the most active users for the days of CDR activity. Red color represents high and yellow low similarity between the users of the corresponding row and column. Users (1,309), (2,397), (5,306),(3,360) are very similar, while they have no similarities with other users.

with common behaviour, which leads to the detection of interesting patterns for further analysis. In this specific case, users (1,309), (2,397), and (5,306) are very similar. Indeed according to the solution of the VAST 2008 challenge, these users are the ones responsible for the change in the CDR activity of the last three days. Specifically, these pairs of Ids belong to users that were active in distinct periods, i.e in the first seven and the last three days, while they also have high similarity, since they communicate with the same set of users. This indicates that these pairs of mobile devices Ids were actually used by the same users. Also, these users show another interesting similar pattern, namely each user does not have high similarity with any other users, but one. This is evident in the similarity matrices of Figure 4.11 by the white lines that run through it. Observing these white lines, we can identify another pair of users (3,360) showing high similarity. These users are indeed contributing to the anomalies hidden in the VAST CDR dataset.

4.6. Summary and Future Work

This chapter presented a graph correlation approach utilised in NEMESYS as a pre-processing step towards anomaly detection, which takes place in WP4. Towards this end, billing related records (CDR) were transformed into sequences of graph representations, where each graph in the sequence represents either (i) the CDR activity in a specific time instance, or (ii) the CDR activity of one user for the total time period. The goal in the first case is the detection of time instances that are distinctly different from the rest, while the in the second case is the detection of users that have different behaviour from the rest. For the comparison between the graphs in the generated sequence, graph matching techniques were applied. Then dimensionally reduction through MDS was used to transform the calculated differences into a user-friendly 2D visual representation, allowing to detect anomalies as outliers in the 2D space. This correlation approach was applied on the VAST2008 challenge regarding CDR analysis, and was found to be very efficient in identifying distinctly different users and time instances.

Future work includes the application of additional graph matching and dimensionality reduction techniques, so as to provide feature representations in which the graphs are distinctly separated, hence increasing the accuracy of the anomaly detection algorithms. We will also investigate how to identify the minimum number of features that most accurately describe the correlation results, without much loss of information, which may require reducing the dimensionality of data to more than 2. Further directions include the application of clustering methods in the features extracted from the graph correlation analysis, allowing to separate the space into regions of distinct behaviour. This separation reduces the size of the dataset, and is an efficient method for noise

reduction for the anomaly detection algorithms. Finally, graph sequences will be created for each user, in order to identify the time periods in which each user changed behaviour. This is an important step that must be applied before the application of root cause analysis and attack attribution algorithms.

5. A Model-based Approach to Anomaly Detection

5.1. Introduction

Research in communication systems has a solid background of modelling methodologies such as stochastic processes, queueing systems, graph models, etc. These methods are routinely and successfully utilised to describe communication systems and to analyse, optimise and improve their performance, but they are rarely used for security. The typical approach for anomaly detection in networks is data driven, whereby detection is performed by collecting and analysing large volumes of data, and neglecting the underlying communication system.

In this chapter we develop a quantitative model-based framework for correlating mobile user activities with their impact on different network components so as to perform anomaly detection more effectively and reduce false alarms. The approach incorporates modelling of the mobile network at different levels of abstraction to properly represent the components and the processes that are prone to anomalous behaviour. The natural choice for this approach is multi-class queueing models [35, 39]. Such models require average service times and task sequences to be known, and can provide estimates of both averages and variances of the times that it would take to undertake signalling or call processing functions, as well as of access times to web sites and other services, in the presence of a large population of mobile users in the network. Thus it is more suited for network threats that map into *congestion* in either the signaling or data planes. The aim is to allow anomaly detection algorithms to obtain quick estimates of the impact of a suspected set of users, so that abnormal but non performance impacting behaviours are not incorrectly flagged as malicious. Specifically, for a subset of users which are identified by the anomaly detection algorithm, if the estimated average quantities for a population scaled up to the current observed numbers in the network, for the same user set, deviates significantly from the current measured values in the network, then one can infer some level of anomaly. The estimates for the scaled up population can be calculated from the queueing models in a very fast and straightforward manner. Finally, the modelling approach allows to predict the future effect of a suspected set of users on the network (e.g. during rush hour), so that appropriate mitigation steps could be taken in advance.

5.1.1. Motivation

The model-based approach involves representing how the communication system functions at the level of each mobile connection and is motivated by several factors. *First*, the number of mobile users that we need to monitor and deal with in real time is very large. Thus a clear and understandable uniform approach is needed to deal with each individual mobile call, emphasising the similarities and common parameters. Anomalies can then be detected via deviations from normal parameters or behaviours. *Second*, the computational tools that are being developed in WP4 for anomaly detection and mitigation need to be based on sound principles; mathematical models allow us to evaluate and validate such algorithms based on clear underlying model assumptions, even though the use of these models in various practical situations will include conditions when some of the model assumptions are not satisfied. Thus mathematical models will need to be tested and validated through simulation and practical measurements. *Third*, due to the large size of the systems we need to deal with, the mathematical models will have to be decomposable, both in terms of obtaining analytical and numerical solutions, e.g. in terms of product forms [36–38, 40], and in terms of distributed processing for reasons of scalability [3]. Again, the mathematical and decomposable structure also provides a handle for decomposing and distributing the computational tasks.

The focus of model construction is on identifying and modelling the individual steps that a mobile user makes regarding:

- Call establishment, including the connection to base stations, access points, and call management,
- Monitoring and billing, and the interactions between the mobile operator's resources and the network for monitoring and billing,
- Accesses that the call may make to sensitive resources such as web sites for privileged information interrogation,
- Call processing or service steps that may require that the mobile user identity itself is sent to the network or external resources, or provide other sensitive information (e.g. personal addresses) at certain operational steps,
- The access to web sites that are used for purchasing and billing.

Indeed, in order to develop detection capabilities of a practical value it is vital to formulate a unified analytical framework which explicitly describes the main *internal resources* of the network architecture, including both the internal aspects regarding base station,

access points and call management and billing, and the *sensitive external resources* that the mobile user may access during its call. Since our approach will have to be effective in situations where hundreds of thousands of mobile users may be active in a given network simultaneously, we need to address both:

- The case where only a small percentage of mobile users come under attack at a given time, but these attacks are nevertheless of high value to the attacker so that we must be able to detect relatively rare events in a very large ensemble
- Situations where attacks affect the signalling system and are significantly disturbing a large fraction of the ongoing mobile connections

In all cases one may need to deal with real-time detection, mitigation and possibly attack elimination, as well as data collection for deferred ulterior analysis. Our approach will be integrated into the learning-based detection algorithms developed in WP4, and will also be linked to the visualisation tools.

5.2. The G-Network Model

Consider a mobile network providing a set of services I such as calls, data, sms, etc. to the users. When a user initiates an access to a service $i \in I$, a session starts. Generally, the session is comprised of different phases including initiation, service provisioning and termination. We denote by J the set of session phases. Moreover, during a session the user utilises network resources such as channels, controllers, timers, etc., and we denote this set of resources by N. Therefore at any time t, an active session is characterised by a triple $(i, j, n) \in (I, J, N)$, which we will refer to as the state of the session. The state of the network at time t is given by $x(t) = (x_1(t), ..., x_N(t))$, where $x_n(t)$ is the state of node n describing the sessions currently being handled at the node.

We model the system as a queueing network in which the servers (and buffers) represent the resources of the cellular network. An example of a network comprising 10,000 user equipments (UEs), 7 base stations and a radio network controller (RNC) is presented in Figure 5.1. The model includes both user and control plane traffic, allowing us to capture in great detail the dynamics of the mobile network. It is assumed that all the resources have exponential service time distributions, and we denote by $r(i, j, n)$ the service rate of session of type i and phase j at node n. Sessions of type i and phase j arrive at node n with rate $\Lambda_{i,j,n}$ following an independent Poisson process. Additionally, we represent control-plane traffic generated by different network components by L types of signals which are responsible for establishment, termination as well as various state transitions during a session's lifetime. The control signals of type l arriving at node n

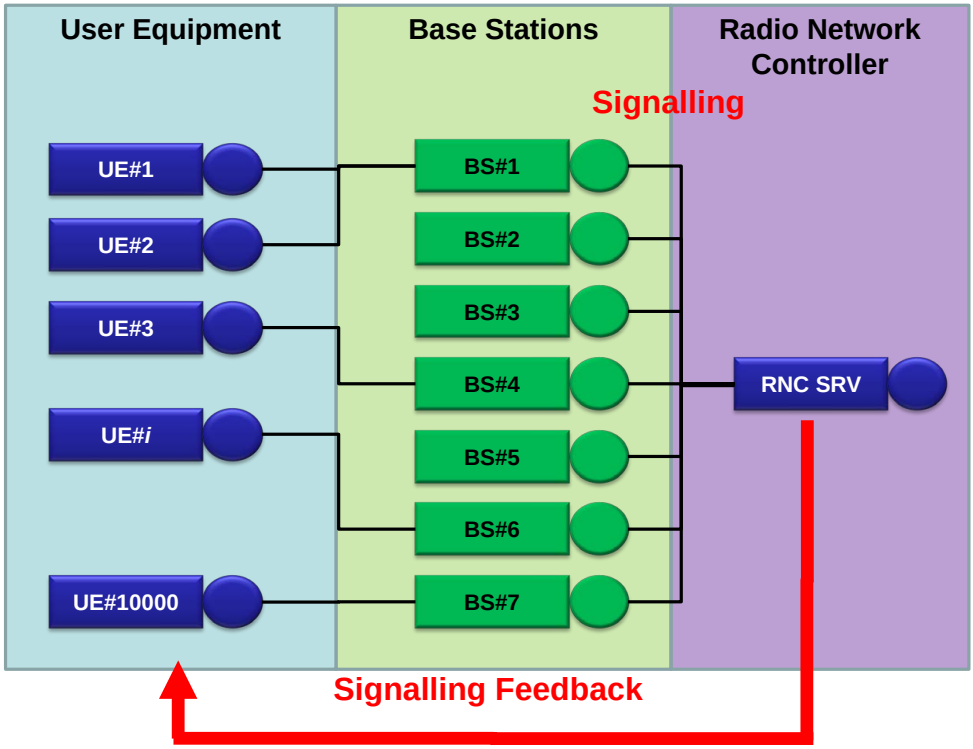


Figure 5.1.: Queueing model of a network containing 10,000 mobile users, 7 base stations and an RNC.

form a Poisson process with rate $\lambda_{l,n}$. When the service of a session of type i and phase j is completed at node n, it transits to node m and its phase changes to k with probability $P^{+}[i,j,n][i,k,m]$, or it becomes a control signal of class l and transits to node m with probability $P^{-}[i,j,n][l,m]$. It may also depart from the network with probability $d[i,j,n]$. The network state change may also be triggered by control signals generated by other network elements. If upon arrival, a control signal finds an empty queue it simply disappears; otherwise it triggers a state transition. A control signal of class l triggers a transition of the session of type i and phase j at node n with probability $K_{n,l,i,j}$. $Q_{n,m,k,l}$

denotes the probability of a customer of class k after being removed by a control signal from node n to move to node m as customer of class l. Obviously the probabilities must sum up to 1, that is:

$$\sum_{j=1}^{N}\sum_{l=1}^{R} P^{+}_{i,j,k,l} + \sum_{j=1}^{N}\sum_{m=1}^{S} P^{-}_{i,j,k,m} + d_{i,k} = 1, \tag{5.1}$$

The theory of G-Networks [39] can then be applied to analyse the system described above. In G-Networks, the users' sessions running in the system in their corresponding phases are represented by multi-class customers, where the set of the classes is given by the cartesian product $I \times J$, with $R = |I \times J|$ being the number of customer classes. For First In First Out (FIFO) service mode, if the following conditions hold:

$$\begin{aligned} r(i,j,n) &= r(a,b,n), \quad \forall i,j,a,b \\ K_{n,l,i,j} &= K_{n,w,a,b}, \ \forall i,j,a,b,l,w \end{aligned}$$

then the probability distribution possesses the product-form property and is given by:

$$P(x) = G\prod_{i=1}^{N}\prod_{k=1}^{R} q_{i,k}^{x_{i,k}} \tag{5.2}$$

where G is a normalisation factor such that $\Sigma_x P(x) = 1$ and the quantities $q_{i,k} > 0, i \in [1,...,N],\ k \in [1,...,R]$ are the solution for the system of non-linear equations given in (5.3)-(5.5):

$$q_{i,k} = \frac{\Lambda_{i,k} + \Lambda^{+}_{i,k}}{r_{i,k} + \sum_{i=1}^{S} K_{i,m,k}(\lambda_{i,m} + \lambda^{-}_{i,m})} \tag{5.3}$$

with:

$$\begin{aligned} \Lambda^{+}_{i,k} =& \sum_{j=1}^{N}\sum_{l=1}^{R} P^{+}_{j,i,l,k} r_{j,l} q_{j,l} + \sum_{j=1}^{N}\sum_{l=1}^{R}\sum_{h=1}^{N}\sum_{m=1}^{S}\sum_{s=1}^{R} r_{j,l} q_{j,l} P^{-}_{j,h,l,m} K_{h,m,s} q_{h,s} Q_{h,i,s,k} \\ &+ \sum_{j=1}^{N}\sum_{m=1}^{S}\sum_{s=1}^{R} \lambda_{j,m} K_{j,m,s} q_{j,s} Q_{j,i,s,k} \end{aligned} \tag{5.4}$$

$$\lambda^{-}_{i,m} = \sum_{j=1}^{N}\sum_{l=1}^{R} P^{-}_{j,i,l,k} r_{j,l} q_{j,l} \tag{5.5}$$

The dynamics of the system are then characterised by the steady-state probability distribution given in (5.2). In general, comparison of the first moments of the performance measure of interest may not reveal the full picture of what is going on in the network, especially when some local distortions or anomalies occur in network operations. Therefore, we are interested in a metric that naturally combines all the available information contained in the probability distribution.

5.3. Information Divergence in G-Networks

The developed model can be used to evaluate how the observed network's behaviour diverges from a reference (typical or normal) behaviour. This divergence, when quantified properly, will posses the desirable properties of an anomaly measure, namely:

- Aggregating various communication and system aspects into one number.
- Scalable with the network size.
- Applicable to arbitrarily selected network components.

As described in Chapter 2, a common metric for the distance between two probability distributions u and v, representing respectively the reference and observed behaviours, is information or KL divergence $d_{KL}(u, v)$ given in Eq. (2.8). When the probability distributions u and v follow the form in (5.2) we get:

$$u(x) = u(x_1, ..., x_N) = \prod_{j=1}^{N} (1 - q_{u,j}) q_{u,j}^{x_j} \tag{5.6}$$

and

$$v(x) = v(x_1, ..., x_N) = \prod_{j=1}^{N} (1 - q_{v,j}) q_{v,j}^{x_j} \tag{5.7}$$

Now substituting (5.6) and (5.7) into the expression for d_{KL} (2.8) we get a closed-form expression for the information divergence in G-Networks:

$$d_{KL}(u, v) = \sum_{j=1}^{N} \left[\ln \left(\frac{1 - q_{u,j}}{1 - q_{v,j}} \right) + \frac{q_{u,k}}{1 - q_{u,k}} \ln \left(\frac{q_{u,j}}{q_{v,j}} \right) \right] \tag{5.8}$$

The derivations leading to the result above are presented in Appendix A. Note that $d_{KL}(u, v)$ depends only on the loads $q_{u,j}$ at the various network resources, which are much easier to measure than the patterns of control signals and sessions of individual users.

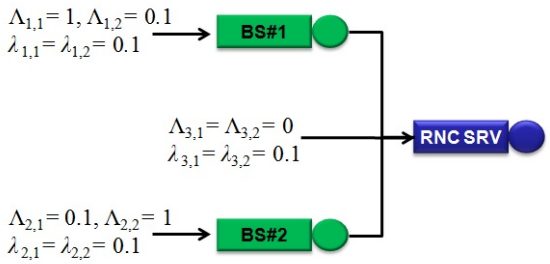


Figure 5.2.: Illustration of the G-network model for a subnetwork consisting of 2 base stations and 1 RNC.

5.3.1. Signalling Attack Example

We demonstrate the above modelling framework and the usage of the d_{KL} divergence for correlation of mobile users behaviour and the resulting impact on the network through the following example. Suppose that the wireless network presented in Figure 5.1 is partially observed, so that the monitored part consists of two base stations and 1 RNC as shown in Figure 5.2. The traffic consists of two types of services, voice calls and web browsing, and there are two types of signalling traffic controlling the corresponding two services. This subnetwork forms a G-Network consisting of $N = 3$ nodes, $R = 2$ customer classes, and $S = 2$ classes of control signals. Let the model parameters have the following values:

$$\Lambda_{[N\times R]} = \begin{pmatrix} 1 & 0.1 \\ 0.1 & 1 \\ 0 & 0 \end{pmatrix}, \ \lambda_{[N\times R]} = \begin{pmatrix} 1 & 0.1 \\ 0.1 & 1 \\ 0 & 0 \end{pmatrix}, \ r_{[N\times R]} = \begin{pmatrix} 2\,2 \\ 2\,2 \\ 2\,2 \end{pmatrix}, \ K = [0.5]_{N\times S\times R}$$

$$P^+ = [0]_{N\times N\times R\times R} \qquad P^- = [0]_{N\times N\times R\times R}$$
$$P^+[i,i,j,j] = 0.1 \quad i,j = \{1,2\} \qquad P^-[1,3,i,j] = 0.1 \quad i,j = \{1,2\}$$
$$P^+[i,3,j,j] = 0.4 \quad i,j = \{1,2\} \qquad P^-[1,1,i,j] = 0.1 \quad i,j = \{1,2\}$$
$$Q = [0]_{N\times N\times R\times R}$$
$$Q[2,3,2,2] = Q[3,2,2,2] = 0.5$$
$$Q[1,3,1,1] = Q[3,1,1,1] = 0.5$$

The numerical solution of the non-linear equations (5.3)-(5.5) for the values of q and

$E[x]$ yields:

$$q_{[N\times R]} = \begin{pmatrix} 0.5091 & 0.0497 \\ 0.05261 & 0.5389 \\ 0.2275 & 0.2251 \end{pmatrix}, \quad E[x]_{[N\times R]} = \begin{pmatrix} 1.0369 & 0.0523 \\ 0.0556 & 1.1685 \\ 0.2946 & 0.2904 \end{pmatrix}$$

Let λ_{attack} be the vector of signaling attack rates, $0 \leq \alpha \leq 1$ be the attack intensity, so that the effective arrival rate of control signals from both normal and malicious behaviour is $\lambda_{eff} = (1-\alpha)\lambda + \alpha\lambda_{attack}$. We will examine the sensitivity of the d_{KL} divergence on the attack intensity α for three different attack rate vectors:

$$\lambda_{attack_1} = \begin{pmatrix} 0.2 & 0.1 \\ 0.1 & 0.1 \\ 0.1 & 0.1 \end{pmatrix}, \quad \lambda_{attack_2} = \begin{pmatrix} 0.1 & 0.1 \\ 0.1 & 0.2 \\ 0.1 & 0.1 \end{pmatrix}, \quad \lambda_{attack_3} = \begin{pmatrix} 0.1 & 0.1 \\ 0.1 & 0.1 \\ 0.2 & 0.1 \end{pmatrix}$$

The results are presented in Figure 5.3, where each plot shows the divergence d_{KL} caused by the respective attack rate vector, at each node as well as the entire network. Due to the symmetry of the traffic patterns, which results from the specific rates of incoming traffic and the routing probabilities in the example, the first and the second attack rate vectors, λ_{attack_1} and λ_{attack_2}, affect the network and each node approximately in the same manner. However, when the attack rates are represented by λ_{attack_3}, where number 3 is the most affected, we see that the divergence is significant both at this specific node as well as the entire network, whereas it is less pronounced in nodes 1 and 2. This observation suggests that our method could be useful for capturing the correlation between the behaviour of users and the impact on the system not only at the network level, but also at the nodes level, where divergence at a node increases in proportion to how the node is impacted by the attack. This could help in the identification of the root causes of anomalies.

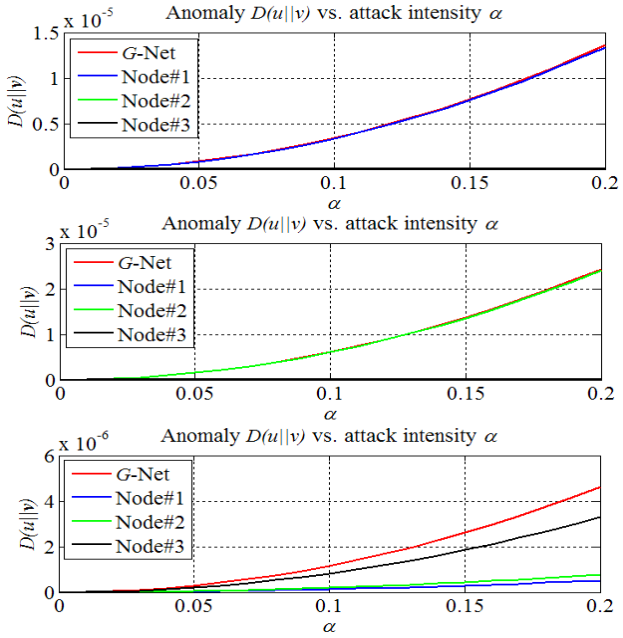


Figure 5.3.: KL divergence in G-networks vs attack intensity α for the attack rate vectors λ_{attack_1} (top), λ_{attack_2} (middle) and λ_{attack_3} (bottom).

6. Conclusions

This deliverable presented a number of frameworks for representing and analysing mobile network traffic, focusing on signaling and billing data available to the operators. Several techniques have been applied to datasets obtained from operational 3G/4G mobile networks, in order to reduce the number of monitored traffic variables, allow efficient processing of the data, and enhance the performance of the anomaly detection algorithms.

First, we analysed features extracted from time series signaling data related to the HLR and MSC components, in order to understand their reference behaviour and identify the information they provide about the network. The features were extracted using time series decomposition, entropy analysis and dimensionality reduction. We also examined the correlations between the extracted features, and identified some underlying patterns in the signaling traffic under both normal conditions and in the presence of anomalous traffic instances. The efficiency of the approach in identifying relevant features for anomaly detection was demonstrated for both datasets.

Then, we presented a graph correlation approach that can be used as a pre-processing step for anomaly detection. In this method, billing or CDR records were transformed into sequences of graph representations, where each graph in a sequence represents either a single CDR activity in a specific time instance, or a CDR activity of one user for the total time period, allowing the detection of anomalous time instances and users, respectively. Graph matching techniques were subsequently applied to measure the differences between graphs which in turn were visualised on the 2D space using MDS in order to facilitate the detection of outliers. The approach was applied on the VAST2008 challenge regarding CDR analysis, and was found to be very efficient in identifying distinctly different users and time instances.

Finally, we developed a model-based approach using multi-class queueing models, which allows to conduct quick what-if analysis to determine whether an observed behaviour is normal or malicious in order to perform signaling-based anomaly detection. Specifically, the model enables anomaly detection algorithms to obtain quick estimates of the impact of a suspected set of users, so that abnormal but non performance impacting behaviours are not incorrectly flagged as malicious. It also allows to predict the future effect of a suspected set of users on the network (e.g. during rush hour), so that appropriate mitigation steps could be taken in advance.

The final abnormal event detection module, which integrates the methods and features identified in this study and the detection algorithms developed in WP4, will be presented in the second version of this deliverable.

A. Derivation of KL Divergence in G-networks

The derivation of (5.8) consists in substituting the probability distribution of the reference behaviour (5.6) and the observed one (5.7) into the definition of KL divergence in (2.8):

$$
\begin{aligned}
d_{KL}(u,v) &= \sum_x u(x) \ln \frac{u(x)}{v(x)} \\
&= \sum_{x_1,...,x_N} u(x_1,...,x_N) \ln \frac{u(x_1,...,x_N)}{v(x_1,...,x_N)} \\
&= \sum_{x_1,...,x_N} u(x_1,...,x_N) \ln \frac{\prod_{j=1}^N (1-q_{u,j}) q_{u,j}^{x_j}}{\prod_{j=1}^N (1-q_{v,j}) q_{v,j}^{x_j}} \\
&= \sum_{x_1,...,x_N} u(x_1,...,x_N) \ln \left[\prod_{j=1}^N \left(\frac{1-q_{u,j}}{1-q_{v,j}} \right) \prod_{j=1}^N \left(\frac{q_{u,j}}{q_{v,j}} \right)^{x_j} \right] \\
&= \sum_{x_1,...,x_N} u(x_1,...,x_N) \left[\ln \prod_{j=1}^N \left(\frac{1-q_{u,j}}{1-q_{v,j}} \right) + \ln \prod_{j=1}^N \left(\frac{q_{u,j}}{q_{v,j}} \right)^{x_j} \right] \\
&= \left[\ln \prod_{j=1}^N \frac{1-q_{u,j}}{1-q_{v,j}} \right] \underbrace{\sum_{x_1,...,x_N} u(x_1,...,x_N)}_{=1} \\
&\quad + \sum_{(x_1,...,x_N)} u(x_1,...,x_N) \ln \left(\prod_{j=1}^N \left(\frac{q_{u,j}}{q_{v,j}} \right)^{x_j} \right) \\
&= \sum_{j=1}^N \ln \left(\frac{1-q_{u,j}}{1-q_{v,j}} \right) + \sum_{x_1,...,x_N} u(x_1,...,x_N) \ln \prod_{j=1}^N \left(\frac{q_{u,j}}{q_{v,j}} \right)^{x_j} \\
&= \sum_{j=1}^N \ln \left(\frac{1-q_{u,j}}{1-q_{v,j}} \right) + \sum_{x_1,...,x_N} u(x_1,...,x_N) \sum_{j=1}^N x_j \ln \left(\frac{q_{u,j}}{q_{v,j}} \right)
\end{aligned}
$$

$$= \sum_{j=1}^{N} \ln\left(\frac{1-q_{u,j}}{1-q_{v,j}}\right) + \sum_{j=1}^{N} \ln\left(\frac{q_{u,j}}{q_{v,j}}\right) \underbrace{\sum_{x_1,\ldots,x_N} u(x_1,\ldots,x_N)x_j}_{=E_u[x_j]}$$

$$= \sum_{j=1}^{N} \left[\ln\left(\frac{1-q_{u,j}}{1-q_{v,j}}\right) + E_u[x_j] \ln\left(\frac{q_{u,j}}{q_{v,j}}\right)\right] \tag{A.1}$$

In the following, we obtain the expression for $E_u[x_k]$ based on (5.6):

$$\begin{aligned}
E_u[x_k] &= \sum_{x_1,\ldots,x_k,\ldots,x_N} x_k \; u(x_1,\ldots,x_k,\ldots,x_N) \\
&= \sum_{x_1,\ldots,x_k,\ldots,x_N} x_k \prod_{j=1}^{N} (1-q_{u,j}) q_{u,j}^{x_j} \\
&= \prod_{j\neq k}^{N} \left(\sum_{x_j=0}^{\infty} (1-q_{u,j}) q_{u,j}^{x_j}\right) \sum_{x_k=0}^{\infty} x_k (1-q_{u,k}) q_{u,k}^{x_k} \\
&= \prod_{j\neq k}^{N} \left((1-q_{u,j}) \sum_{x_j=0}^{\infty} q_{u,j}^{x_j}\right) (1-q_{u,k}) \sum_{x_k=0}^{\infty} x_k q_{u,k}^{x_k} \\
&= \prod_{j\neq k}^{N} \underbrace{\frac{(1-q_{u,j})}{(1-q_{u,j})}}_{=1} \left[(1-q_{u,k}) \frac{q_{u,k}}{(1-q_{u,k})^2}\right] \\
&= \frac{q_{u,k}}{1-q_{u,k}}
\end{aligned} \tag{A.2}$$

Substituting (A.2) into (A.1) results in (5.8).

Bibliography

[1] IEEE VAST Challenge. http://www.cs.umd.edu/hcil/VASTchallenge08/, 2008.

[2] 3GPP. Study on core network overload (CNO) solutions. TS 23.843, 3rd Generation Partnership Project (3GPP), 2013.

[3] J. Aguilar and E. Gelenbe. Task assignment and transaction clustering heuristics for distributed systems. *Information Sciences*, 97(1-2):199–219, 1997.

[4] U. Aickelin, D. Dasgupta, and F. Gu. Artificial immune systems. In E. K. Burke and G. Kendall, editors, *Search Methodologies*, pages 187–211. Springer US, 2014.

[5] N. Al-Rousan, S. Haeri, and L. Trajkovic. Feature selection for classification of bgp anomalies using bayesian models. In *Proc. International Conference on Machine Learning and Cybernetics (ICMLC'12)*, volume 1, pages 140–147, Jul 2012.

[6] T. A. Almeida, J. M. G. Hidalgo, and A. Yamakami. Contributions to the study of SMS spam filtering: New collection and results. In *Proc. 11th ACM Symposium on Document Engineering (DocEng'11)*, pages 259–262, Mountain View, California, USA, 2011.

[7] R. Ambauen, S. Fischer, and H. Bunke. Graph edit distance with node splitting and merging, and its application to diatom identification. In *Graph Based Representations in Pattern Recognition*, pages 95–106. Springer, 2003.

[8] C. Amrutkar, M. Hiltunen, T. Jim, K. Joshi, O. Spatscheck, P. Traynor, and S. Venkataraman. Why is my smartphone slow? on the fly diagnosis of underperformance on the mobile internet. In *Proc. 43rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN'13)*, pages 1–8, Budapest, Hungary, Jun 2013. IEEE Computer Society.

[9] A. B. Ashfaq, S. Rizvi, M. Javed, S. A. Khayam, M. Q. Ali, and E. Al-Shaer. Information theoretic feature space slicing for statistical anomaly detection. *Journal of Network and Computer Applications*, 41:473 – 487, May 2014.

[10] B. Auffarth, M. López, and J. Cerquides. Comparison of redundancy and relevance measures for feature selection in tissue classification of CT images. In P. Perner, editor, *Advances in Data Mining. Applications and Theoretical Aspects*, volume 6171 of *LNCS*, pages 248–262. Springer Berlin Heidelberg, 2010.

[11] A. Aziz, A. Azar, M. Salama, A. Hassanien, and S.-O. Hanafy. Genetic algorithm with different feature selection techniques for anomaly detectors generation. In *Proc. Federated Conference on Computer Science and Information Systems (FedCSIS'13)*, pages 769–774, krakow, Poland, Sep 2013.

[12] M. Baltatu et al. State-of-the-art for security threats and attacks against mobile devices & analysis of current practices. NEMESYS Deliverable D1.1, Apr 2013.

[13] A. Bar, A. Paciello, and P. Romirer-Maierhofer. Trapping botnets by DNS failure graphs: Validation, extension and application to a 3g network. In *Proc. 5th IEEE International Traffic Monitoring and Analysis Workshop (TMA'13)*, pages 393–398, Turin, Italy, Apr 2013.

[14] P. Barford, J. Kline, D. Plonka, and A. Ron. A signal analysis of network traffic anomalies. In *Proc. 2nd ACM SIGCOMM Workshop on Internet Measurment (IMW '02)*, pages 71–82, Marseille, France, Nov 2002.

[15] M. Becher, F. C. Freiling, J. Hoffmann, T. Holz, S. Uellenbeck, and C. Wolf. Mobile security catching up? revealing the nuts and bolts of the security of mobile devices. In *Proc. IEEE Symposium on Security and Privacy (SP '11)*, pages 96–111, Oakland, CA, May 2011. IEEE Computer Society.

[16] I. Borg and P. J. F. Groenen. *Modern multidimensional scaling: Theory and applications*. Springer Series in Statistics, 2nd edition, 2005.

[17] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander. LOF: Identifying density-based local outliers. *ACM SIGMOD Rec.*, 29(2):93–104, May 2000.

[18] H. Bunke and G. Allermann. Inexact graph matching for structural pattern recognition. *Pattern Recognition Letters*, 1(4):245–253, 1983.

[19] H. Bunke, P. J. Dickinson, M. Kraetzl, and W. D. Wallis. *A graph-theoretic approach to enterprise network dynamics*, volume 24 of *Progress in Computer Science and Applied Logic (PCS)*. Birkhäuser Boston, 2007.

[20] M. Burgess, H. Haugerud, S. Straumsnes, and T. Reitan. Measuring system normality. *ACM Trans. Comput. Syst.*, 20(2):125–160, May 2002.

[21] P. Chaovalit, A. Gangopadhyay, G. Karabatis, and Z. Chen. Discrete wavelet transform-based time series analysis and mining. *ACM Comput. Surv.*, 43(2):6:1–6:37, Feb 2011.

[22] C. Chatfield and A. Collins. Principal component analysis. In *Introduction to Multivariate Analysis*, pages 57–81. Springer US, 1980.

[23] R. Chavarriaga, H. Sagha, and J. del Millán. Ensemble creation and reconfiguration for activity recognition: An information theoretic approach. In *Proc. IEEE International Conference on Systems, Man, and Cybernetics (SMC'11)*, pages 2761–2766, Anchorage, AK, Oct 2011.

[24] R. B. Cleveland, W. S. Cleveland, J. E. McRae, and I. Terpenning. STL: A seasonal-trend decomposition procedure based on loess. *Journal of Official Statistics*, 6(1):3–73, 1990.

[25] A. Coluccia, A. D'Alconzo, and F. Ricciato. Distribution-based anomaly detection in network traffic. In *Data Traffic Monitoring and Analysis*, pages 202–216. Springer, 2013.

[26] A. D'Alconzo, A. Coluccia, F. Ricciato, and P. Romirer-Maierhofer. A distribution-based approach to anomaly detection and application to 3G mobile traffic. In *Proc. 28th IEEE Conference on Global Telecommunications (GLOBECOM'09)*, pages 2888–2895, Honolulu, Hawaii, 30 Nov – 4 Dec 2009.

[27] A. D'Alconzo, A. Coluccia, and P. Romirer-Maierhofer. Distribution-based anomaly detection in 3G mobile networks: From theory to practice. *Int. J. Netw. Manag.*, 20(5):245–269, Sept. 2010.

[28] G. Del Vescovo and A. Rizzi. Automatic classification of graphs by symbolic histograms. In *Proc. IEEE International Conference on Granular Computing (GRC'07)*, pages 410–416, Fremont, CA, Nov 2007.

[29] G. Del Vescovo and A. Rizzi. Online handwriting recognition by the symbolic histograms approach. In *Proc. IEEE International Conference on Granular Computing (GRC'07)*, pages 686–690, Fremont, CA, Nov 2007.

[30] Developing Solutions. The basic architecture of a 3G/4G network core. `http://www.developingsolutions.com/products/`, 2014.

[31] L. Dolberg, J. Francois, and T. Engel. Multi-dimensional aggregation for DNS monitoring. In *Proc. IEEE 38th Conference on Local Computer Networks (LCN'13)*, pages 390–398, Oct 2013.

[32] S. M. Emran and N. Ye. Robustness of canberra metric in computer intrusion detection. In *Proc. IEEE Workshop on Information Assurance and Security*, West Point, NY, Jun 2001.

[33] A. P. Felt, M. Finifter, E. Chin, S. Hanna, and D. Wagner. A survey of mobile malware in the wild. In *Proc. 1st ACM W'shop on Security and Privacy in Smartphones and Mobile Devices (SPSM'11)*, pages 3–14, Chicago, IL, 2011.

[34] T. Gärtner, J. W. Lloyd, and P. A. Flach. Kernels for structured data. In S. Matwin and C. Sammut, editors, *Inductive Logic Programming*, volume 2583 of *LNCS*, pages 66–83. Springer Berlin Heidelberg, 2003.

[35] E. Gelenbe. Probabilistic models of computer systems. *Acta Inform.*, 12(4):285–303, 1979.

[36] E. Gelenbe. Product-form queueing networks with negative and positive customers. *J. Appl. Probab.*, 28(3):656–663, 1991.

[37] E. Gelenbe. G-networks with signals and batch removal. *Probability in the Engineering and Informational Sciences*, 7(3):335–342, 7 1993.

[38] E. Gelenbe. G-networks with triggered customer movement. *J. Appl. Probab.*, 30(3):742–748, 1993.

[39] E. Gelenbe and A. Labed. G-networks with multiple classes of signals and positive customers. *Eur. J. Oper. Res.*, 108(2):293–305, 1998.

[40] E. Gelenbe and R. R. Muntz. Probabilistic models of computer systems: Part i (exact results). *Acta Inform.*, 7(1):35–60, 1976.

[41] N. Gobbo, A. Merlo, and M. Migliardi. A denial of service attack to GSM networks via attach procedure. In A. Cuzzocrea, C. Kittl, D. Simos, E. Weippl, and L. Xu, editors, *Security Engineering and Intelligence Informatics*, volume 8128 of *LNCS*, pages 361–376. Springer, 2013.

[42] M. A. Hall. *Correlation-based feature selection for machine learning.* PhD thesis, The University of Waikato, 1999.

[43] M. Hamdi, N. Grira, and N. Boudriga. Detecting distributed computer network attacks: A multi-dimensional wavelet approach. In *Proc. 12th IEEE International Conference on Electronics, Circuits and Systems (ICECS'05)*, pages 1–5, Dec 2005.

[44] J. Hu and S. Guo. Segment-based anomaly detection with approximated sample covariance matrix in wireless sensor networks. *IEEE Transactions on Parallel and Distributed Systems*, 99(PrePrints):1, 2014.

[45] B. J. Jain and K. Obermayer. Structure spaces. *Journal of Machine Learning Research*, 10:2667–2714, 2009.

[46] D. Jiang, J. Liu, Z. Xu, and W. Qin. Network traffic anomaly detection based on sliding window. In *Proc. International Conference on Electrical and Control Engineering (ICECE'11)*, pages 4830–4833, Sept 2011.

[47] N. Jiang, Y. Jin, A. Skudlark, and Z.-L. Zhang. Understanding SMS spam in a large cellular network: Characteristics, strategies and defenses. In *Research in Attacks, Intrusions, and Defenses*, pages 328–347. Springer, 2013.

[48] I. Jolliffe. *Principal component analysis*. Wiley Online Library, 2005.

[49] G. Jurman, S. Riccadonna, R. Visintainer, and C. Furlanello. Canberra distance on ranked lists. In *Proc. NIPS'09 Advances in Ranking Workshop*, pages 22–27, Whistler, Canada, Dec 2009.

[50] H. Kashima, K. Tsuda, and A. Inokuchi. Marginalized kernels between labeled graphs. In T. Fawcett and N. Mishra, editors, *Proc. 20th International Conference on Machine Learning (ICML'03)*, pages 321–328, Washington, DC, Aug 2003.

[51] S. Kaski and J. Peltonen. Dimensionality reduction for data visualization [applications corner]. *IEEE Signal Processing Magazine*, 28(2):100–104, 2011.

[52] S. A. Khayam and H. Radha. Linear-complexity models for wireless MAC-to-MAC channels. *Wireless Networks*, 11(5):543–555, 2005.

[53] J. Kirk. Android botnet abuses people's phones for SMS spam. PCWorld Australia, Dec 2012.

[54] A. Lakhina, K. Papagiannaki, M. Crovella, C. Diot, E. D. Kolaczyk, and N. Taft. Structural analysis of network traffic flows. *SIGMETRICS Perform. Eval. Rev.*, 32(1):61–72, Jun 2004.

[55] N. Leavitt. Mobile security: finally a serious problem? *IEEE Computer*, 44(6):11–14, 2011.

[56] P. P. Lee, T. Bu, and T. Woo. On the detection of signaling DoS attacks on 3G/WiMax wireless networks. *Computer Networks*, 53(15):2601–2616, Oct 2009.

[57] W. Lee and D. Xiang. Information-theoretic measures for anomaly detection. In *Proc. IEEE Symposium on Security and Privacy (SP'01)*, pages 130–143, Oakland, CA, May 2001. IEEE Computer Society.

[58] Y.-J. Lee, Y.-R. Yeh, and Y.-C. F. Wang. Anomaly detection via online oversampling principal component analysis. *IEEE Transactions on Knowledge and Data Engineering*, 25(7):1460–1470, July 2013.

[59] V. I. Levenshtein. Binary codes capable of correcting deletions, insertions and reversals. In *Soviet physics doklady*, volume 10, page 707, 1966.

[60] C. Lever, M. Antonakakis, B. Reaves, P. Traynor, and W. Lee. The core of the matter: Analyzing malicious traffic in cellular carriers. In *Proc. Network and Distributed System Security Symposium (NDSS'13)*, pages 1–16, San Diego, CA, Feb 2013. Internet Society.

[61] J. Leyden. Android trojan taints US mobes, spews 500,000 texts a day. The register, Dec 2012.

[62] L. Leydesdorff. On the normalization and visualization of author co-citation data: Salton's cosine versus the jaccard index. *Journal of the American Society for Information Science and Technology*, 59(1):77–85, 2008.

[63] H. Liu, J. Sun, L. Liu, and H. Zhang. Feature selection with dynamic mutual information. *Pattern Recognition*, 42(7):1330–1339, 2009.

[64] L. Livi and A. Rizzi. The graph matching problem. *Pattern Analysis and Applications*, 16(3):253–283, 2013.

[65] G. Lyberopoulos et al. Use case analysis and user scenarios - first version. NEMESYS Deliverable D1.2.1, Oct 2013.

[66] G. Maier, A. Feldmann, V. Paxson, and M. Allman. On dominant characteristics of residential broadband internet traffic. In *Proc. 9th ACM SIGCOMM Conference on Internet Measurement Conference (IMC'09)*, pages 90–102, Chicago, Illinois, USA, 2009.

[67] D. Maslennikov and Y. Namestnikov. Kaspersky security bulletin 2012: The overall statistics for 2012. Kaspersky Lab, Dec 2012.

[68] K. mei Zheng, X. Qian, and N. An. Supervised non-linear dimensionality reduction techniques for classification in intrusion detection. In *Proc. International Conference on Artificial Intelligence and Computational Intelligence (AICI'10)*, volume 1, pages 438–442, Oct 2010.

[69] S. Mika, B. Schölkopf, A. J. Smola, K.-R. Müller, M. Scholz, and G. Rätsch. Kernel PCA and de-noising in feature spaces. In *Proc. 1998 Conference on Advances in Neural Information Processing Systems II*, pages 536–542. MIT Press, 1999.

[70] K. Muller, S. Mika, G. Ratsch, K. Tsuda, and B. Scholkopf. An introduction to kernel-based learning algorithms. *IEEE Transactions on Neural Networks*, 12(2):181–201, 2001.

[71] C. Mulliner, N. Golde, and J.-P. Seifert. SMS of death: From analyzing to attacking mobile phones on a large scale. In *Proc. 20th USENIX Conference on Security (SEC'11)*, pages 24–24, San Francisco, CA, Aug 2011.

[72] C. Mulliner, S. Liebergeld, M. Lange, and J.-P. Seifert. Taming Mr Hayes: Mitigating signaling based attacks on smartphones. In *Proc. 42nd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN '12)*, pages 1–12, Boston, MA, 2012. IEEE Computer Society.

[73] C. Mulliner and J.-P. Seifert. Rise of the iBots: Owning a telco network. In *Proc. 5th International Conference on Malicious and Unwanted Software (MALWARE'10)*, pages 71–80, Nancy, Lorraine, Oct 2010.

[74] M. Neuhaus and H. Bunke. A probabilistic approach to learning costs for graph edit distance. In *Proc. 17th International Conference on Pattern Recognition (ICPR'04)*, volume 3, pages 389–393. IEEE, 2004.

[75] M. Neuhaus and H. Bunke. Self-organizing maps for learning the edit costs in graph matching. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 35(3):503–514, 2005.

[76] M. Neuhaus and H. Bunke. A convolution edit kernel for error-tolerant graph matching. In *Proc. 18th International Conference on Pattern Recognition (ICPR'06)*, pages 220–223, Hong Kong, Aug 2006. IEEE.

[77] M. Neuhaus and H. Bunke. A random walk kernel derived from graph edit distance. In D.-Y. Yeung, J. T. Kwok, A. Fred, F. Roli, and D. Ridder, editors, *Structural, Syntactic, and Statistical Pattern Recognition*, volume 4109 of *LNCS*, pages 191–199. Springer Berlin Heidelberg, 2006.

[78] M. Neuhaus and H. Bunke. *Bridging the gap between graph edit distance and kernel machines*. World Scientific Publishing, 2007.

[79] M. Neuhaus and H. Bunke. A quadratic programming approach to the graph edit distance problem. In *Graph-Based Representations in Pattern Recognition*, pages 92–102. Springer, 2007.

[80] R. Newson. Parameters behind "nonparametric" statistics: Kendall's tau, somers' D and median differences. *Stata Journal*, 2(1):45–64, 2002.

[81] S. Novakov, C.-H. Lung, I. Lambadaris, and N. Seddigh. Studies in applying PCA and wavelet algorithms for network traffic anomaly detection. In *Proc. IEEE 14th International Conference on High Performance Switching and Routing (HPSR'13)*, pages 185–190, Jul 2013.

[82] G. Palla, A.-L. Barabási, and T. Vicsek. Quantifying social group evolution. *Nature*, 446(7136):664–667, 2007.

[83] C. Pascoal, M. Rosario de Oliveira, R. Valadas, P. Filzmoser, P. Salvador, and A. Pacheco. Robust feature selection and robust PCA for internet traffic anomaly detection. In *Proc. IEEE INFOCOM 2012*, pages 1755–1763, Mar 2012.

[84] H. Peng, F. Long, and C. Ding. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8):1226–1238, 2005.

[85] M. A. Porter, J.-P. Onnela, and P. J. Mucha. Communities in networks. *Notices of the AMS*, 56(9):1082–1097, 2009.

[86] G. Reddig. OTT service blackouts trigger signaling overload in mobile networks, Sep 2013.

[87] Rethink Wireless. DoCoMo demands Google's help with signalling storm, Jan 2012.

[88] F. Ricciato, A. Coluccia, A. D'alconzo, D. Veitch, P. Borgnat, and P. Abry. On the role of flows and sessions in internet traffic modeling: an explorative toy-model. In *Proc. IEEE GLOBECOM 2009*, pages 1–8, Honolulu, HI, 30 Nov– 4 Dec 2009.

[89] F. Ricciato, P. Svoboda, E. Hasenleithner, and W. Fleischer. On the impact of unwanted traffic onto a 3G network. In *Proc. 2nd Int. W'shop Security, Privacy and Trust in Pervasive and Ubiquitous Computing (SecPerU'06)*, pages 49–56, Lyon, France, Jun 2006.

[90] K. Riesen and H. Bunke. Approximate graph edit distance computation by means of bipartite graph matching. *Image and Vision Computing*, 27(7):950–959, 2009.

[91] K. Riesen and H. Bunke. *Graph classification and clustering based on vector space embedding*. World Scientific Publishing Co., Inc., River Edge, NJ, USA, 2010.

[92] V. Roth and V. Steinhage. Nonlinear discriminant analysis using kernel functions. In S. A. Solla, T. K. Leen, and K. Müller, editors, *Advances in neural information processing systems 12 (NIPS'99)*, 1999.

[93] H. Sagha, H. Bayati, J. d. R. Millán, and R. Chavarriaga. On-line anomaly detection and resilience in classifier ensembles. *Pattern Recognition Letters*, 34(15):1916–1927, 2013.

[94] A.-D. Schmidt, H.-G. Schmidt, L. Batyuk, J. H. Clausen, S. A. Camtepe, S. Albayrak, and C. Yildizli. Smartphone malware evolution revisited: Android next target? In *Proc. 4th International Conference on Malicious and Unwanted Software (MALWARE'09)*, pages 1–7, Montreal, QC, Oct 2009. IEEE.

[95] V. A. Siris and F. Papagalou. Application of anomaly detection algorithms for detecting SYN flooding attacks. *Computer Communications*, 29(9):1433–1442, 2006.

[96] S. Song, L. Ling, and C. N. Manikopoulo. Flow-based statistical aggregation schemes for network anomaly detection. In *Proc. IEEE International Conference on Networking, Sensing and Control (ICNSC'06)*, pages 786–791, 2006.

[97] N. Suri, M. Murty, and G. Athithan. Unsupervised feature selection for outlier detection in categorical data using mutual information. In *Proc. 12th International Conference on Hybrid Intelligent Systems (HIS'12)*, pages 253–258, Dec 2012.

[98] N. Suzuki, K. Hirasawa, K. Tanaka, Y. Kobayashi, Y. Sato, and Y. Fujino. Learning motion patterns and anomaly detection by human trajectory analysis. In *Proc. IEEE International Conference on Systems, Man and Cybernetics (ISIC'07)*, pages 498–503, Montreal, Que, Oct 2007.

[99] Symantec Security Response. Pikspam: An SMS spam botnet, 2012.

[100] P. Szilagyi and S. Novaczki. An automatic detection and diagnosis framework for mobile communication systems. *IEEE Transactions on Network and Service Management*, 9(2):184–197, Jun 2012.

[101] A. G. Tartakovsky, A. S. Polunchenko, and G. Sokolov. Efficient computer network anomaly detection by changepoint detection methods. *IEEE Journal of Selected Topics in Signal Processing*, 7(1):4–11, 2013.

[102] J. B. Tenenbaum, V. De Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

[103] S. W. Thomas Gaertner, Peter Flach. On graph kernels: Hardness results and efficient alternatives. In *Learning Theory and Kernel Machines*, volume 2777 of *LNCS*, pages 129–143. Springer Berlin Heidelberg, 2003.

[104] P. Traynor, W. Enck, P. McDaniel, and T. La Porta. Mitigating attacks on open functionality in SMS-capable cellular networks. *IEEE/ACM Transactions on Networking*, 17(1):40–53, Feb 2009.

[105] P. Traynor, M. Lin, M. Ongtang, V. Rao, T. Jaeger, P. McDaniel, and T. L. Porta. On cellular botnets: Measuring the impact of malicious devices on a cellular network core. In *Proc. 16th ACM Conf. on Computer and Communications Security (CCS'09)*, pages 223–234, Chicago, Illinois, USA, Nov 2009.

[106] J. Venna and S. Kaski. Comparison of visualization methods for an atlas of gene expression data sets. *Information Visualization*, 6(2):139–154, 2007.

[107] F. Vilisics and E. Hornung. Urban areas as hot-spots for introduced and shelters for native isopod species. *Urban ecosystems*, 12(3):333–345, 2009.

[108] A. Wade. Ranking data, 2010.

[109] M. Wählisch, A. Vorbach, C. Keil, J. Schönfelder, T. C. Schmidt, and J. H. Schiller. Design, implementation, and operation of a mobile honeypot. *CoRR*, abs/1301.7257:1–6, Jan 2013.

[110] T. Xia, G. Qu, S. Hariri, and M. Yousif. An efficient network intrusion detection method based on information theory and genetic algorithm. In *Proc. 24th IEEE International Performance, Computing, and Communications Conference (IPCCC'05)*, pages 11–17, Apr 2005.

[111] M. Xie, S. Han, and B. Tian. Highly efficient distance-based anomaly detection through univariate with pca in wireless sensor networks. In *Proc. IEEE 10th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom'11)*, pages 564–571, Nov 2011.

[112] Y. Xie, H.-A. Kim, D. R. O'Hallaron, M. K. Reiter, and H. Zhang. Seurat: A pointillist approach to anomaly detection. In E. Jonsson, A. Valdes, and M. Almgren, editors, *Recent Advances in Intrusion Detection*, volume 3224 of *LNCS*, pages 238–257. Springer Berlin Heidelberg, 2004.

[113] Q. Yang and F. Li. Support vector machine for intrusion detection based on LSI feature selection. In *Proc. 6th World Congress on Intelligent Control and Automation (WCICA'06)*, volume 1, pages 4113–4117, Dalian, 2006.

[114] L. Yu and H. Liu. Feature selection for high-dimensional data: A fast correlation-based filter solution. In *Proc. the2nth International Conference on Machine Learning (ICML'03)*, pages 856–863, Washington, DC, Aug 2003.

[115] L. Yu and H. Liu. Efficient feature selection via analysis of relevance and redundancy. *J. Mach. Learn. Res.*, 5:1205–1224, Dec 2004.

[116] B. Zhang, J. Yang, J. Wu, D. Qin, and L. Gao. MCST: Anomaly detection using feature stability for packet-level traffic. In *Proc. 13th Asia-Pacific Network Operations and Management Symposium (APNOMS'11)*, pages 1–8, Sep 2011.

[117] D. Zhang, Z.-H. Zhou, and S. Chen. Non-negative matrix factorization on kernels. In Q. Yang and G. Webb, editors, *PRICAI 2006: Trends in Artificial Intelligence*, volume 4099 of *LNCS*, pages 404–412. Springer Berlin Heidelberg, 2006.

[118] J. Zhang, F.-C. Tsui, M. M. Wagner, and W. R. Hogan. Detection of outbreaks from time series data using wavelet transform. In *Proc. AMIA Annual Symposium*, volume 2003, pages 748–752, Washington, DC, Nov 2003.

[119] G. Zhao, J. Yang, G. Hura, L. Ni, and S.-H. Huang. Correlating TCP/IP interactive sessions with correlation coefficient to detect stepping-stone intrusion. In *Proc. International Conference on Advanced Information Networking and Applications (AINA'09)*, pages 546–551, May 2009.

[120] Y. Zhou and X. Jiang. Dissecting Android malware: Characterization and evolution. In *Proc. IEEE Symp. on Security and Privacy (SP'12)*, pages 95–109, May 2012.