Imperial College London

Novel Graph Analytics for Enhancing Data Insight

Mr. Stavros Papadopoulos

Supervisor: Prof. Erol Gelenbe

November 2015

Submitted in part fulfillment of the requirements for the degree of Doctor of Philosophy in Electrical Engineering and Electronic Engineering of Imperial College London and the Diploma of Imperial College London

The author confirms that the work described in this thesis is his own, and all other works are appropriately referenced.

"The copyright of this thesis rests with the author and is made available under a Creative Commons Attribution Non-Commercial No Derivatives licence. Researchers are free to copy, distribute or transmit the thesis on the condition that they attribute it, that they do not use it for commercial purposes and that they do not alter, transform or build upon it. For any reuse or redistribution, researchers must make clear to others the licence terms of this work"

Abstract

Graph analytics is a fast growing and significant field in the visualization and data mining community, which is applied on numerous high-impact applications such as, network security, finance, and health care, providing users with adequate knowledge across various patterns within a given system. Although a series of methods have been developed in the past years for the analysis of unstructured collections of multi-dimensional points, graph analytics has only recently been explored. Despite the significant progress that has been achieved recently, there are still many open issues in the area, concerning not only the performance of the graph mining algorithms, but also producing effective graph visualizations in order to enhance human perception.

The current thesis deals with the investigation of novel methods for graph analytics, in order to enhance data insight. Towards this direction, the current thesis proposes two methods so as to perform graph mining and visualization.

Based on previous works related to graph mining, the current thesis suggests a set of novel graph features that are particularly efficient in identifying the behavioral patterns of the nodes on the graph. The specific features proposed, are able to capture the interaction of the neighborhoods with other nodes on the graph. Moreover, unlike previous approaches, the graph features introduced herein, include information from multiple node neighborhood sizes, thus capture long-range correlations between the nodes, and are able to depict the behavioral aspects of each node with high accuracy. Experimental evaluation on multiple datasets, shows that the use of the proposed graph features for the graph mining procedure, provides better results than the use of other state-of-the-art graph features.

Thereafter, the focus is laid on the improvement of graph visualization methods towards enhanced human insight. In order to achieve this, the current thesis uses non-linear deformations so as to reduce visual clutter. Non-linear deformations have been previously used to magnify significant/cluttered regions in data or images for reducing clutter and enhancing the perception of patterns. Extending previous approaches, this work introduces a hierarchical approach for non-linear deformation that aims to reduce visual clutter by magnifying significant regions, and leading to en-

hanced visualizations of one/two/three-dimensional datasets. In this context, an energy function is utilized, which aims to determine the optimal deformation for every local region in the data, taking the information from multiple single-layer significance maps into consideration. The problem is subsequently transformed into an optimization problem for the minimization of the energy function under specific spatial constraints. Extended experimental evaluation provides evidence that the proposed hierarchical approach for the generation of the significance map surpasses current methods, and manages to effectively identify significant regions and deliver better results.

The thesis is concluded with a discussion outlining the major achievements of the current work, as well as some possible drawbacks and other open issues of the proposed approaches that could be addressed in future works.

Acknowledgements[1]

First, I would like to thank my principal supervisor, Prof. Erol Gelenbe, for the opportunity for this research, as well as for his guidance and advice in completing the thesis. I would also like to especially thank my collaborative supervisor at the Information Technologies Institute (ITI), Dr. Dimitrios Tzovaras, without whom this thesis would not be possible, for his supervision and support throughout the whole period of my research. Furthermore, I wish to thank Dr. Anastasios Drosou (ITI) for his invaluable assistance in every aspect, with his knowledge and experience.

[1] A major part of this Thesis was suported through project FP7-ICT-317888-NEMESYS funded by the 7th framework program, coordinated by Prof. Erol Gelenbe. The opinions expressed in this paper are those of the authors and do not necessarily reflect the views of the European Commission.

To my beautiful soon to be wife, without whose help and support this thesis would not be completed.

Contents

1 Introduction **18**
1.1 Motivation 19
1.1.1 Motivation for graph mining 19
1.1.2 Motivation for graph visualization 20
1.1.3 Problem Formulation 21
1.2 Originality Achievements of the Thesis 23
1.3 Thesis outline 26

2 Literature survey **27**
2.1 Graph features used in graph mining 27
2.2 Graph clustering methods 31
2.3 Visual clutter reduction 34
2.3.1 Visual variables 34
2.3.2 Use of non-liner deformations for clutter reduction . . 35
2.4 Critical summary 38
2.4.1 Graph mining 38
2.4.2 Graph visualization 39

3 Novel Graph Features for behavioral analysis of the graph nodes **41**
3.1 Graph structure 42
3.2 Graph neighborhoods 43
3.3 Extraction of graph features 45
3.3.1 Volume 47
3.3.2 Edge Entropy 47
3.3.3 Graph Entropy 48
3.3.4 Edge Weight Ratio 48

3.3.5 Average Outward/Inward Edge Weight 49
3.4 Complexity analysis 50
3.5 Security analysis 51

4 A Hierarchical Magnification Approach for enhancing the Insight in Graph Visualizations 54
4.1 Generation of the hierarchical significance map 56
4.2 Multi-resolution grid 59
4.3 Multi-dimensional grid deformation 60
4.4 Different choices for the generation of the significance map . . 65

5 Experimental evaluation 66
5.1 Datasets 66
5.1.1 Datasets used for the evaluation of the graph features 66
5.1.2 Datasets used for the evaluation of the hierarchical clutter reduction approach 71
5.2 Experimental evaluation of the graph features 74
5.2.1 Anomaly detection methods selected for comparison . 74
5.2.2 Evaluation metrics 77
5.2.3 Scenario 1: SMS flood 77
5.2.4 Scenario 2: Spam SMS 81
5.2.5 Scenario 3: RRC-based attacks 84
5.2.6 Scenario 4: Malware infection 88
5.3 Experimental evaluation of the hierarchical magnification approach 95
5.3.1 Evaluation Metrics 95
5.3.2 BGPGraph Visualization magnification 96
5.3.3 Graph Visualization magnification 99
5.3.4 Scatterplot Visualization magnification 101

6 Application of the Magnification Approach for enhancing non-graph visualizations 105
6.1 Datasets 105
6.1.1 Word frequencies dataset 105
6.1.2 US Census unemployment dataset 106
6.1.3 Image 106

6.2 Experimental evaluation of the hierarchical magnification approach . . . 106
6.2.1 Evaluation Metrics . . . 106
6.2.2 Word Clouds . . . 107
6.2.3 Choropleth maps . . . 110
6.2.4 Image resizing . . . 113
6.2.5 User evaluation results . . . 113
6.2.6 Time performance . . . 115

7 Conclusions 117
7.1 Summary of the Thesis . . . 117
7.2 Critical discussion . . . 119
7.3 Future work . . . 120

A Alternative method for clutter reduction in one-dimensional datasets 136
A.1 Visual mapping function . . . 137
A.2 Clutter reduction for one-dimensional mapping . . . 138
A.2.1 Optimization Procedure . . . 140

B BGPGraph: Hierarchical Visualization of BGP Routing Changes 147
B.1 Introduction . . . 148
B.2 Graph Creation . . . 148
B.3 Clustering of the AS-Graph . . . 151
B.4 Visualizing AS relationships using edge direction . . . 155
B.5 Interaction with the Graph Hierarchy . . . 158
B.6 Quantification of the information . . . 159
B.6.1 Entropy of the input signal . . . 159
B.6.2 Entropy of the output signal . . . 161
B.7 Applications of the information quantification . . . 162
B.7.1 Minimization of information loss induced by the clustering algorithm . . . 162
B.7.2 Using Entropy to detect events . . . 164

List of Figures

3.1 A schematic representation of the procedure followed for the generation of the graph features 42
3.2 The graph structure of the communication events in an example dataset 43
3.3 An example of k-neighborhood graph 44
3.4 Graph traversal 45
3.5 Examples of graph entropy. 49
3.6 Examples of classifier regions for a two dimensional space, i.e. utilizing two features. Red regions correspond to anomalous feature vector and green regions correspond to normal feature vectors. 52

4.1 Method overview 55
4.2 Generation of the hierarchical significance map. The significance map is crated for each layer of the grid hierarchy. Afterwards, each layer is normalized, and all the layers are superpositioned for the generation of the final hierarchical significance map. 56
4.3 Hyperrectangle intersection in multiple dimensions, in which hyperrectangles f_j^l and f_i^{l+1} intersect. In the specific case that $n_l = 2^l$ partitions per dimension, if two hyperrectangles intersect, it follows that one contains the other. 58
4.4 An example of comparison between Gaussian blurring and hierarchical significance map. 59

4.5 (a) The hypercubes of the grid in *layer M* of the hierarchy. Hypercubes f_i^{mul} and f_j^{mul} are highlighted in red and green colors. (b) The hypercubes of the multi-resolution grid. The sets of hypercubes $P(f_i^{mul})$ and $P(f_j^{mul})$ are highlighted using red and green colors. 61

5.1 The number of infected mobile devices at each time instance of the simulation. 71
5.2 Examples of decision tree generation. 76
5.3 Evaluation of the graph features on the SMS flood dataset. . 79
5.4 The histograms of the two features with the highest importance according to the Gini index for the SMS flood dataset. The corresponding features are the volume (k=1) and the Graph entropy (k=1). 79
5.5 Evaluation of the graph features on the SMS flood dataset. . 80
5.6 Evaluation of the graph features on the Spam SMS dataset . 82
5.7 The histograms of the two features with the highest importance according to the Gini index for the Spam SMS dataset. The corresponding features are the Average outward weight (k=0) and the Edge weight ratio(k=0). 83
5.8 Evaluation of the graph features on the Spam SMS dataset. . 84
5.9 Evaluation of the graph features on the RRC-based attacks dataset. 86
5.10 The histograms of the two features with the highest importance according to the Gini index for the RRC-based attacks dataset. The corresponding features are the Average outward weight (k=0) and the Average inward weight (k=1). 87
5.11 Evaluation of the graph features on the RRC-based attacks dataset. 87
5.12 Evaluation of the graph features on the malware infection dataset. 90

5.13 The histograms of the two features with the highest importance according to the Gini index for the malware infection dataset for the task of identifying anomalous devices in the current and the next two time periods, given the current state of the network. The corresponding features for the current time period are the Average outward weight (k=0) and the Edge entropy (k=1), while the corresponding features for the next two time periods are the Average outward weight (k=1) and the Edge entropy (k=2). 91
5.14 Evaluation of the graph features on the malware infection dataset. 92
5.15 Evaluation of the graph features on the malware infection dataset 93
5.16 Evaluation of the graph features for anomaly detection in the current and future time periods. 94
5.17 Evaluation of all the anomaly detection in the current and future time periods with respect to the TPR scores (a) and FPR scores (b). 94
5.18 The Entropy plot of level-11 of the clustering hierarchy, for the period 23-Dec-2004 to 24-Dec-2004. The time step is set to $\Delta T = 30$ *sec*. It is apparent that the information drastically increases around 9:24 GMT, 24-Dec-2004. 97
5.19 Abstract view of level 11 visualizing the event that took place on Dec 24, 2004 concerning AS-9121. (a) The green vertex gains a lot of IPs while the neighboring ASes lose IPs. Furthermore, the green path takes over the traffic from many other neighboring paths (red paths). (b) It is apparent that AS-9121 is the one that hijacks a big portion of the Internet traffic. 98
5.20 Application of the magnification approach on the *BGPGraph* [1] visualization of the BGP hijacking event, which took place on 24-Dec-2004. AS-9121 due to a router misconfiguration announced a large number of prefixes that did not belong to it. 99
5.21 Results of the application of the Magnification approach on a graph-based visualization with magnification factor equal to 3. 100

5.22 (a) SP metric for Figure 5.21. (b) PP metric Figure 5.21. (c) PP metric for Figure 5.23. 101
5.23 Results of the application of the Magnification approach on a scatterplot visualization with magnification factor equal to 3. 102
5.24 Application of the Magnification approach on a 3D scatterplot visualization using the forest fires dataset [2]. 103
5.25 Application of the generalized scatterplots [3]. (a) Input data. (b) Result of the generalized scatterplot. The red ellipsoids highlight two regions, the first is empty and not well filled after the deformation, and the second compresses the clusters and distorts the local patterns. 104

6.1 Results of the application of the Magnification approach on the word cloud visualization with magnification factor equal to 4. 107
6.2 The evaluation results using the SP (a) and JSD (b) and Kendall tau (c) metrics applied on Figures 6.1(g), 6.1(h), and 6.1(i) respectively. 109
6.3 Results of the application of the Magnification approach on a choropleth map, visualizing the unemployment data for each US area in 2011. 110
6.4 This figure shows the application of the JSD metric, in order to measure the distance of the color distribution created from the deformation methods in Figure 6.3, from the optimum uniform distribution (according to [4]). 111
6.5 Application on image resizing. 112
6.6 Running times for regular and multi-resolution grids for all the datasets in this paper measured in milliseconds. 116

A.1 The mapping function $F : \Re^n \rightarrow V^m$ that is used by every visualization approach. 137
A.2 A histogram illustrating the distribution of the BGP IP changes caused by the BGP announcements. 141
A.3 The value of the objective function of information loss for various output bin partitions. 142
A.4 Comparison of the optimization methods for various number of bins N, and for various input distributions. 143

A.5 The relationship of the number of iterations of the Downhill Simplex method and the number of bins N. 144
A.6 The relationship of the number of bins and the information loss for four input distributions. 145

B.1 Computation of the edges and vertices weights. At time T_i, AS-1 owns $m_i^1 = 150$ IPs, AS-2 owns $m_i^3 = 100$ IPs and the edge between them is used by $m_i^2 = 150$ IPs. After some time $\Delta T = T_k - T_i$, these values change and become $m_k^1 = 200$, $m_k^2 = 200$ and $m_k^3 = 90$. This difference constitutes the weights of the edges and the vertices: $weight_1 = \Delta m^1 = 50$, $weight_2 = \Delta m^2 = 50$ and $weight_3 = \Delta m^3 = -10$. The magnitude of the corresponding weight is mapped to the radius of the vertex or the width of edge. Green color represents positive and red color negative weight values. 151
B.2 Example announcements. (a) This is an empty graph. All the ASes own 0 IPs and the edges are used by 0 IPs. (b) Two announcements are made. The weights of the edges that are shown are the square roots of the actual weights $\sqrt{Weight}$. (c) AS-4 hijacks part of AS-3 prefix. This is shown by positive weight (green color) in the path <3549 5 1 2 4> and negative weight (red color) in the path <3549 1 2 3>. 152
B.3 The mapping of the weights to the Graph Hierarchy. The numbers that are shown represent weights and the blue ellipsoids represent clusters. The mapping is calculated using equations (B.11) and (B.12). 154
B.4 Cluster Hierarchy of the AS-Graph. As it is shown the number of vertices is reduced as the level increases, but the topology of the original graph (a), is preserved. The monitoring point is the orange node. 156
B.5 Visualizing the AS relationships using directed and undirected edges. Directed edges illustrate Provider-to-Costumer relationships, and undirected edges illustrate Peer-to-Peer and Sibling-to-Sibling relationships. 157
B.6 Interaction methods with the Graph hierarchy. The vertices inside of the blue ellipsoids represent the set of disjoint pairs S_{dis}^0. The orange color represents the monitoring point. . . . 158

B.7 The mapping function $F : \Re^n \rightarrow V^m$ from the edge weight to the edge width. Green color represents positive weight values and red negative. 161
B.8 The relationship between the entropy loss rate $En^{in}_{(loss-rate)}(J^l)$ for various M values and the levels of the hierarchy. 164
B.9 The Entropy plot concept. The length of each bar represents the magnitude of the edges or vertices entropy, at the predefined time window. 166

List of Tables

3.1 An overview of the proposed graph features. 46

5.1 Anomaly detection results on the SMS flood dataset. 10% of the dataset was utilized for training. 78
5.2 Anomaly detection results on the spam SMS dataset. 10% of the dataset was utilized for training. 81
5.3 Anomaly detection results on the RRC-based attacks dataset. 10% of the dataset was utilized for training. 85
5.4 Anomaly detection results on the malware infection dataset. . 88
5.5 Anomaly detection results on the malware infection dataset. . 89

6.1 The results of the user evaluation studies on 22 individuals. The participants had to select between the proposed hierarchical approach and other state-of-the-art methods 114

B.1 The edges with the biggest entropy and their corresponding rate of the total edge entropy 163

Abbreviation List

AIG	American International Group
AS	Autonomous System
ASN	Autonomous System Number
BGP	Border Gateway Protocol
CDR	Call Detail Record
COI	Communities Of Interest
DC	Drought Code
DCH	Dedicated Channel
DDoS	Distributed Denial of Service
DoS	Denial of Service
DMC	Duff Moisture Code
FFMC	Fine Fuel Moisture Code
FP	False Positive
FPR	False Positive Rate
GED	Graph Edit Distance
GMT	Greenwich Mean Time
HLR	Home Location Register
HVS	Human Vision System
IEEE	Institute of Electrical and Electronics Engineers
IP	Internet Protocol
ISI	Initial Spread Index
JSD	Jensen-Shannon Divergence
KLD	Kullback-Leibler Divergence
kNN	k-Nearest Neighbor
MELO	Multiple Eigenvector Linear Ordering

MFPT	Mean First Passage Time
PCA	Principal Component Analysis
PhD	Doctor of Philosophy
PP	Pattern Preservation
RF	Random Forest
RRC	Radio Resource Control
SMS	Short Message Service
SP	Size perception
SVM	Support Vector Machine
TP	True Positive
TPR	True Positive Rate
US	United States

Chapter 1

Introduction

Contents

1.1 Motivation 19

1.2 Originality Achievements of the Thesis 23

1.3 Thesis outline 26

Graph analytics has been a focal point under the research scope of information technologies, as the need of its applications becomes more evident in the continuous development of cutting-edge technologies including computer, social, opinion, auction, financial and web-page networks [5].

Specifically, graph analytics is the study and analysis of data that can be transformed into a graph representation consisting of nodes and links (edges). Data used in a wide range of applications can be intuitively formulated into a graph, providing a holistic view of the correlations that an entity participates in. For example, computer networks can be represented as graphs where the routers/computers are the nodes and the physical connections between them are the edges. Another example is mobile networks, in which the mobile devices are the nodes and edges represent communications between them. In many applications, including both visualization and analytics fields, graphs are seemingly ubiquitous.

Graph analytics is a multi-disciplinary field, consisting of diverse areas such as: information visualization, human-computer interaction, graph drawing, and data mining. The first three areas concern the methods used for graph visualization and interactive exploration. The fourth area is related to graph mining, which champions knowledge discovery through algorithmic computation.

In the case that the task of graph analytics is known beforehand, then graph mining is the appropriate approach to follow. Graph mining utilizes the vast computational resources that exist today, in order to analyze large graphs fast and efficiently. Specific tasks for which graph mining has been used, include the detection of anomalous nodes or subgraphs, community detection, and object recognition.

On the other hand, if the analysis task is not known beforehand, then the use of graph visualization is more appropriate. Graph visualization includes the human cognition in the analysis loop, and enables the interactive exploration of graph related datasets. It is used in many cases where the human may simply examine the data to learn more about it, gain insight about it, or make new discoveries.

Although significant progress has been achieved in the field of graph analytics, there are still many open issues, concerning the performance of the graph mining algorithms and delivery of effective graph visualizations, that need to be addressed.

1.1 Motivation

This section presents the challenges and open issues of the graph analytics field, with respect to both graph mining and graph visualization.

1.1.1 Motivation for graph mining

Graph mining utilizes algorithmic computation to identify structural features and their interrelationships in graphs. There are a number of diverse tasks in graph mining, such as [6] [5]: Detection of abnormal subgraphs/edges/nodes, community detection (graph clustering/partitioning) [7] [8], social network analysis, and image segmentation (through graph partitioning) [9]. Irrespectively of a specific task, the graph mining algorithms are confronted with similar challenges.

One of the major challenges in this field is the definition of the descriptive graph features that are able to efficiently capture the behavioral patterns of each node or subgraph. Different features can severely affect the accuracy of the graph mining algorithms, since each feature has a specific capacity in capturing different patterns in the graph. In this respect, this thesis focuses on the definition of novel node features in graph-based datasets, which are

able to efficiently identify the behavioral patterns of each node, and thus, result in improved accuracy of the graph mining algorithms in a variety of applications.

The previously proposed graph features were generated either using the entire graph as input, or using egonets[1] [5]. Dynamic neighborhood sizes have not been yet considered in the literature. This thesis proposes the use of dynamic neighborhood sizes, in order to capture the structure of the data from different behavioral perspectives and detect correlations and relationships more efficiently when compared to static neighborhood sizes. Extended experimental evaluation on multiple datasets provides evidence that the use of dynamic neighborhood sizes can significantly increase the accuracy of the graph mining algorithms.

It is also important for the graph mining algorithm to be fast, and able to be applied on large graphs. This means that the graph features must be able to be computed fast. In this respect the graph features proposed in this thesis can be applied on large graphs, since they are efficient to compute.

1.1.2 Motivation for graph visualization

Visual clutter [10] is a particularly significant issue in the graph visualization field, since the graphs have an inherently small visualization capacity. The reason for this is that they are comprised of a small number of available edge widths and node sizes, in order to visually represent the edge and node attributes. Visual clutter can be also caused when large graphs are visualized on small display devices, which reduce the visualization space and its information capacity. According to Rosenholtz et al. [11] clutter is defined as the state in which excess items, or their representation or organization, lead to a degradation of performance at some task. As a result users can be misled into deriving wrong conclusions, while also the decision confidence on erroneous decisions is increased [12].

Non-lineal deformations based on content aware significance maps, have been proposed in the literature for the reduction of visual clutter (as presented in Chapter 2). These non-lineal deformations are used in order to magnify highly cluttered regions of the data, and enhance their perception by the human analyst.

[1]The egonet of a node includes the node itself, its neighbors, and all the interactions between these nodes

For non-lineal deformation applications, the most efficient method proposed in the literature is the utilization of a regular grid [13] [14]. While the regular grids, which have the same resolution in all the regions of the input data, work well, they waste computational resources due to the large number of optimization variables. The reason for this is that the different regions of the input data are not equally important. This suggests that the utilization of a multi-resolution grid, that encodes significant regions with high and non-significant regions with low resolution, could have the potential to be more efficient. While multi-resolution grids have been utilized in the context of volume rendering [15], their application on non-linear deformation has not been thoroughly investigated in the literature. Towards this end, this thesis utilizes a multi-resolution grid for non-linear deformation, in order to research their effect in the grid deformation procedure and evaluate the final results, when compared to other state-of-the-art methods.

The previously proposed approaches for non-lineal deformations utilize significance maps that were created by taking into account, only one grid resolution. The use, however, of information from only one grid resolution, makes the identification of significant regions less accurate, since high resolutions lose significant regions of large size, while low resolutions cannot encode significant regions of small size. Inspired by the way human visual perception works [16], this thesis proposes the use of a hierarchical significance map, in order to address the aforementioned issue. This approach takes into account information from multiple layers of different grid resolutions, and is able to efficiently identify both small and large sizes of significant regions.

1.1.3 Problem Formulation

As the amount, types and sources of data continue to expand, the demand for better analysis and interpretation of the input information is highly accelerated. Existing graph analytics approaches used in a variety of tasks can be discriminated based on their application field and the expected goal. In well defined tasks, the use of graph mining approaches has been the primary method used for analysis. For example, in tasks of clustering and anomaly detection, graph mining is able to deliver accurate results. On the contrary, when the task is not well defined, or the goal is simply exploration of the dataset or presentation of the results, then graph visualization is the leading method utilized. In each case, there is a number of issues that should be

considered when analyzing a graph.

Graph mining is primarily concerned with the detection of different patterns in the graph. To achieve this goal, one of the fundamental and most frequently used approach is the extraction of graph features. Different features are able to identify and capture different graph patterns, and thus, each feature is appropriate for a particular task. In order to alleviate this issue, multiple features are used in combination, in order to capture the multifaceted characteristics that exist in the graph. Based on the above, novel features need to be defined in order to identify different aspects of the data, and finally, lead to algorithms with high accuracy in specific tasks.

The recent increase in information technologies has led to the collection of large volumes of data. This fact creates the need for fast graph mining algorithms that are able to deal with large graphs and quickly identify patterns. In order to achieve this goal, the new graph features should be able to be computed fast and efficiently, so as to render them able for use in large graphs.

Graph visualization is a major part of graph analytics and is mainly related to the presentation of the graph datasets to the human analyst for knowledge discovery and exploratory purposes. In this respect, its primary concern is the mapping of the input data to the visual attributes of the visualization, so as to enhance the human insight. To fulfill this outcome, the graph visualization should have low visual clutter, in order to aid for the better comprehension of the graph by the human analyst. Lowering the size of visual clutter leads to reduction of information loss and the subsequent enhancement of data patterns that exist in the graph.

Visual clutter in graph visualizations is primarily caused by two sources. The first source is the positioning of the graph on the visual display. Different positioning of the nodes/edges can highlight different patterns in the graph and thus, lead to different analysis results. The positioning of the graph nodes/edges is also affected when resizing the graph visualization on small displays. Multiple metrics have been suggested for measuring the efficiency of different graph positioning methods of the same graph. These measures are called aesthetic measures [17], and try to capture how pleasing and comprehensive a graph positioning is. In other words, the generation of aesthetically appealing graphs is more than creating beautiful graphs, but is associated with readability and understanding. Examples of frequently used

aesthetic measures include, but are not limited to: minimize edge crossings [18], keep edge lengths uniform [19], and maximize the minimum edge crossing angles [20]. Recent methods have also been proposed in order to cluster the edges of the graph that have similar directions, so as to reduce the total edge congestion. This method of clustering the edges is called Edge bundling [21]. The second source of visual clutter in graph visualizations is the mapping of graph attributes to the visual attributes of the visualization. The main issue during this mapping procedure is that the graph visualization has an inherently small visual capacity, due to the small number of available colors and edge/node sizes that can be successfully discriminated by the human analyst. Based on the above, novel graph visualization techniques should be developed to address the issue of visual clutter in graph visualizations, and subsequently lead to more efficient analysis methods.

The objective of the current thesis is the to propose novel graph analytics methods for enhancing data insight and deliver deeper understanding of the graph-structured datasets. In this respect, it deals both with graph mining and graph visualization.

The expected outcome of the current thesis will be the definition and thorough study of novel graph features, which will be able to efficiently identify graph patterns, and lead to increasing the accuracy of graph mining algorithms. The proposed graph features should be experimentally evaluated on a variety of graph datasets, and compared with other state-of-the-art features.

Additionally to the above, the current thesis will also investigate different methods for clutter reduction in graph visualizations. In particular, this thesis will focus on the development of a novel method for clutter reduction. The proposed approach will be attempted to be as fast as possible, so as to be used in real time analysis. Extended experimental evaluation will also be conducted, in order to compare the proposed approach with existing clutter reduction approaches, and provide evidence regarding the superiority of the suggested solution.

1.2 Originality Achievements of the Thesis

The current thesis deals with the development of novel methods for graph analytics. The main contribution offered to the State-of-the-Art technologies

by the current thesis can be summed up in the following bullets:

- Novel graph features are proposed for graph mining, namely: Volume, Edge entropy, Graph Entropy, Edge weight ration, and Average Outward/Inward edge weight. These features capture different behavioral aspects of each node, and when used in combination, provide a complete view of the node behavior.

- Utilization of dynamic graph neighborhoods for defining the graph features (and not only egonets as performed in previous approaches). The use of dynamic graph neighborhoods enables the identification of large-scale correlations and relationships between the different nodes of the graph, which has as a result the increase in the accuracy of the graph mining algorithms.

- The novel features suggested in this thesis were experimentally verified to have a higher precision than other state-of-the-art features in a variety of graph datasets. The task considered for the experiments is the identification and prediction of anomalous users in a mobile network environment. Multiple scenarios are considered, namely: Detection of SMS spamers, Detection of users that attack the availability of the network, and Prediction of future malware infections for each user.

- The current thesis deals with clutter reduction in graph visualizations using non-linear deformations. The proposed approach utilizes a hierarchical significance map that takes the information from multiple grid layers at different grid resolutions into consideration. This has as a result the more efficient identification of significant regions with different sizes, since small resolutions capture large regions and high resolutions capture small regions.

- Definition and application of a multi-resolution grid that allows less data distortion on the significant regions. This is achieved by allowing regions of low significance to have a smaller number of cells, while regions of high significance have a large number of cells that enable for more continuous distortion. The proposed multi-resolution grid also provides faster optimization by reducing the number of problem variables.

- The proposed clutter reduction approach has been applied on graph visualizations to enhance and simplify the identification of patterns in the graph. The proposed approach has also been applied for clutter reduction on additional non-graph visualizations, such as wordclouds, choropleth maps, and on the field of image resizing. The results of these experiments provide evidence that using the proposed hierarchical non-linear deformation, results in better visualizations with less clutter, when compared to other state-of-the-art approaches.

The following publications have been produced during the study for the completion of the current thesis.

Journals

1. S. Papadopoulos, K. Moustakas, A. Drosou, D. Tzovaras, *"BGPGraph: Detecting and Visualizing Internet Routing Anomalies*, IET Information Security, 2015.

2. S. Papadopoulos, A. Drosou, D. Tzovaras, *"A Novel Graph-based Descriptor for the Detection of Billing-related Anomalies in Cellular Mobile Networks"*, IEEE Transactions on Mobile Computing, accepted for publication, 2016.

3. S. Papadopoulos, A. Drosou, D. Tzovaras, *"A Hierarchical Magnification Approach for enhancing the Insight in Image and Volume Visualizations"*, IEEE Transactions on Multimedia, under review.

Conferences

1. S. Papadopoulos, K. Moustakas, D. Tzovaras, *"Hierarchical Visualization of BGP Routing Changes Using Entropy Measures"*, 8th International Symposium on Visual Computing, July 16-18, 2012.

2. S. Papadopoulos, K. Moustakas, D. Tzovaras, *"BGPViewer: Using Graph representations to explore BGP routing changes"*, 18th International Conference on Digital Signal Processing (DSP), 1-3 July 2013.

1.3 Thesis outline

The rest of the thesis is organized as follows: Chapter 2 presents a review, with respect to the graph features and visual clutter reduction proposed in the literature. This section also presents the methods in which the proposed approaches were utilized for graph-based analysis and for enhancing the analytical potential of information visualization approaches.

Chapter 3 presents the novel graph features proposed in this thesis. These features take information from different neighborhood sizes into account, and thus, are able to efficiently characterize the behavior of each node in the graph.

Chapter 4 presents the proposed approach for clutter reduction in graph visualizations. This is achieved by using significance maps and non-linear grid deformation. In contrast to previous approaches, a hierarchical approach is utilized on saliency and entropy significance maps, in order to generate visual descriptors that are able to identify significant regions in the visualization with varying sizes. These hierarchical maps are afterwards used for non-linear magnification of the identified significant regions, so as they are easier to perceive.

Chapter 5 presents the experimental evaluation of the proposed approaches. The graph descriptors are applied on multiple graph structured datasets for the purpose of identifying nodes that are anomalous, and behave different from the normal nodes. The analysis is performed with respect to the accuracy of the results, and the effect of the different neighborhood sizes on the results. This chapter also presents applications of the proposed visual clutter reduction approach of graph visualizations.

Chapter 6 presents additional applications of the proposed visual clutter reduction approach on non-graph visualizations. Multiple visualization methods are considered, while the proposed approach is also applied for content aware image resizing.

Finally, Chapter 7 presents a summary of the thesis and future work directions.

Chapter 2

Literature survey

Contents

2.1 Graph features used in graph mining 27
2.2 Graph clustering methods 31
2.3 Visual clutter reduction 34
2.4 Critical summary 38

This Section presents the related work with regards to the research areas investigated in this PhD. It is focused on the presentation of features and their uses in graph mining, and in the illustration of the current methods for clutter reduction in information visualization.

2.1 Graph features used in graph mining

The graph features can be partitioned into two main groups: features extracted from the entire graph, and features extracted from local neighborhoods. The section that follows presents the features proposed in the literature for graph analysis and the way they are used. One of the most common goals in the graph analysis techniques is the detection of abnormal nodes or abnormal graphs. Anomalies are considered to be graph objects (nodes/edges/substructures) that are significantly different from the majority of the reference objects in the graph [5].

The node-level features characterize the behavior of single nodes with respect to their interactions with their neighbors. They are usually extracted considering the egonet of each node. The egonet of a node includes the node

itself, its direct neighbors which are connected with the initial node with an edge, and all the connections among these nodes. The most simple feature that can be extracted from an egonet is the in/out degree of the node. The paragraphs that follow provide a review of the features proposed in the literature for the analysis of graph-structured data.

A number of publications focuses on the generation of features that capture the centrality of each node. Central nodes are nodes that lay a large number of paths connecting two random nodes. Removing central nodes from the graph reduces node-to-node connection to a large degree. One of the earliest works in centrality features is the work of Freeman et al. [22]. The authors utilize the degree in which a node lays in the shortest path between the other nodes in the graph, in order to determine its centrality. Bonacich et al. [23] present two features, the eigenvectors and the alpha-centrality. Both features are used in order to measure the centrality of each node. The authors demonstrate the results of their features on a small dataset representing marriages between families in order to identify the most central family. Noh et al. [24] utilize random walks on the graph in order to compute its centrality. The authors propose the use of the mean first passage time (MFPT)[1] between two nodes, in order to determine the relative speed in which a node can receive and spread information over the network, using the random process.

Other node-level features have been also utilized in the literature in order to capture different aspects of the graph data. For example, Watts et al. [25] utilize a local clustering coefficient feature in order to characterize the degree in which the nodes belong to clusters in a small-world network structure[2]. The local clustering coefficient is defined as the ratio between the nodes of the egonet, with respect to the maximum number of nodes it could have. Kang et al. [26] propose the use of node radius as a feature. The node radius is the distance between the corresponding node and a reachable node that is the farthest away from it. The authors use plots of this feature and the number of nodes in the graph that have a specific radius value, in order to visually identify central nodes and outliers.

The proposed node-level features can be analyzed in various ways, in-

[1]MFTP is the expected time required to reach a target node from a source node, using a random walk procedure.

[2]Small-world networks are graphs that have a number of densely connected nodes/ clusters. These clusters are loosely connected with each other with small number of edges

cluding standard data mining procedures such as classification and outlier detection. Akoglu et al. [27] propose Oddball, a method for graph analysis based on node-level features. They utilize four features: 1) Number of neighbors in the egonet, 2) Number of edges in the egonet, 3) Total weight summation of the egonet, and 4) Principal eigenvalue of the weighted adjacency matrix of the egonet. The authors analyze these features in pairs, utilizing their pairwise correlations. These correlations are captured in the form of power laws. The points that have a large distance from the prediction of the power law correlation are considered anomalous. Henderson et al. [28] propose Refex, a method that is able to recursively combine multiple node-level features in other to generate more complex features that capture behavioral information. The authors utilize summation and mean functions on the node-level features as combination operations. At each iteration, the generated features are pruned, based on their pair-wise correlations. The process is repeated until no new features can be added. Kang et al. [29] propose Net-ray, that utilizes the features instead of the actual graph data for visualizing very large datasets. The authors use node-level features such as in/out degree, number of triangles in the egonet. Scatter and distribution plots are used to visualize correlations between pairs of features and distributions of single features. Kang et al. also propose an outlier detection method based on k-means clustering on the points of the plots.

Another category of graph features are the graph-level features. Graph-level features characterize the entire graph. They are extracted by taking into account the entire graph and not small node neighborhoods like node-level features. Simple graph-level distributors already exist in the graph theory domain, such as radius, and diameter. Radius of the graph is the minimum eccentricity[3] of any vertex. Diameter is the maximum eccentricity of any vertex in the graph. In addition to these simple features, multiple more complex features have been proposed in the literature.

One of the first graph-level features proposed in the literature is the global clustering coefficient, proposed by Luce et al. [30]. The global clustering coefficient captures the degree in which there are clusters in the graph, and its structure resembles a small world network. The authors define this feature by utilizing triplets of nodes. A triplet is a set of three nodes, that

[3]The eccentricity of a vertex is the largest geodesic distance between this vertex and all the other vertices in the graph

are connected by either two or three undirected edges. The global clustering coefficient is defined as the ratio of triangles (multiplied by three) with respect to the number of triplets in the graph. Opsah et al. [31] extended the definition of global clustering coefficient to include weighted networks. The authors utilize aggregation functions on the weights on the triangles and triplets in order to represent each one of them as a single value. The aggregation fucntions proposed are arithmetic/geometric mean, maximum, and minimum. The global clustering coefficient is defined as the ratio of the summation of the values for the triangles and the triplets respectively. Korner et al. [32] propose graph entropy, which measures the structural information content of the graph, based on the distribution of the edge connections. The graph entropy is introduced as a problem of determining the best possible encoding of the information, emitted by a source in which pairs of symbols may be indistinguishable. The symbols represent distinct entities in a set of objects which, in the case of graph entropy, are the vertices. Two symbols are distinguishable if they are connected through an edge, and indistinguishable otherwise.

The generation of graph-level features can also be accomplished by combining the node-level features. Possible combinations include arithmetic / geometric mean, variance, maximum, and minimum, mode (e.g. average node degree and principal eigenvalue [5]). It is also possible to directly utilize the distributions of the node-level features. For example Kang et al. [33] utilize the distribution of the node degree in order to study how a specific graph evolves over time.

The global features have the capacity to capture patterns with respect to the global graph structure, while the local features have the capacity to capture patterns with respect to the egonet around a node. There might be cases, however, in which a pattern is more efficiently captured using intermediate neighborhood sizes for feature extraction (i.e. larger from egonets but smaller from the entire graph). In order to overcome this issue, Cortes et al. [34] proposed the use of Communities Of Interest (COI). The COIs are subgraphs centered around a specific node, which include the central node and all its k-hop neighbors and edges. Cortes et al. utilized COIs with $k = 1$ (i.e. egonets) and $k = 2$ for the identification of new fraudulent accounts. The authors utilized two methods to achieve the detection of fraudulent accounts: 1) They concluded that the probability of an account

being fraudulent is an increasing function of the number of fraudulent accounts that exist in the $k = 2$ COI (i.e. Guilt by Association), and 2) They proposed a metric that captures the pairwise distances between different $k = 1$ and $k = 2$ COIs, and used it to find COIs that are very similar to known fraudulent COIs. In this respect, the concept of COIs has also been utilized in further network analysis studies and applications, such as email prioritization [35] by Johansen et al. and malware mitigation in corporate networks [36] by McDaniel et al.

Inspired from the aforementioned works, the current research work acts complementarily to the COI approach, and in particular, it extends the COI definition to also include all the inter-COI communications. Additionally, unlike the COI and other graph-based approaches [34] [35] [36] [27] [28] [29] that utilize static neighborhood sizes, the approach proposed in this research work utilizes dynamic neighborhood sizes for different values of k (i.e. $k = 0, 1, , N$), and incorporates them in the analysis procedure.

2.2 Graph clustering methods

Graph features have also been used for measuring the similarity between the nodes of the graph, and for generating clusters based on these similarities. More formally, graph clustering is the process of grouping the vertices of the graph into distinct groups (i.e. clusters), taking into consideration the edge structure of the graph in such a way that within each cluster there are many edges, while between the clusters there are relatively few edges [37]. In some of the clustering literature (the most common of which refer to social network analysis), a cluster in a graph is also called a community [8] [9]. Clustering on graphs has many applications, such as gene expression analysis [38], natural language processing [39], galaxy formation [40], and image segmentation [7]. Surveys regarding the recent graph clustering algorithms can be found on [37] and [41].

One of the most common ways to perform clustering on a graph is to utilize spectral graph theory. This class of methods is known as spectral clustering methods. The main motivation behind these methods is that when a graph is formed as a collection of k-disjoint cliques, then its normalized Laplacian will be a blockdiagonal matrix that has eigenvalue zero with multiplicity k. In addition, the corresponding eigenvectors will serve

as indicators regarding the cluster membership of each vertex in the graph, i.e. the elements of the clique have similar values with each other (of the same magnitude), and a different value (of larger magnitude) than all the other vertices. Even thought, the addition of edges between the cliques and the removal of some edges within the cliques would cause some divinations, some of the underlying structure can still be evident in the eigenvectors of the Laplacian [37].

Spectral clustering algorithms are related to graph-cut problems. The reason for this relationship is that they use one or more eigenvectors from Laplacian matrices of a graph that must be clustered, which are solutions of a relaxation of some graph-cut problems [41]. The graph-cut problems try to identify the edges to be eliminated from the graph in order to produce k connected subgraphs. The ultimate goal is the generation of well separated clusters, corresponding to the k connected subgraphs. Different criteria have been suggested in the literature for the identification of the aforementioned set of edges, the most common of which are the [41]: minimum cut, minimum ratio cut, minimum normalized cut, and modularity maximization.

The spectral clustering algorithms proposed in the literature can be classified to two main categories: recursive two-way clustering algorithms, and direct k-way clustering algorithms. The recursive two-way clustering algorithms find the Fiedler eigenvector (second smallest eigenvalue of the Laplacian) and perform two-way partitions recursively until a k-way partition is found. The k-way spectral algorithms find the first $d \geq k$ eigenvectors and try to directly cluster the data in k groups, using heuristics.

Hagen and Kahng [42] were the first to propose a recursive two-way spectral clustering approach for the k-way ratio cut problem. They propose an algorithm based on a study about the one-dimensional quadratic placement problem [43] and on the linear ordering of the Fiedler eigenvector. The authors were also the first that established the connection between the eigenvectors of the Laplacian matrix and the solution of the relaxed two-way ratio cut problem. Later, Shi and Malik [44] [45] proposed a recursive two-way nominalized cut algorithm for the k-way nominalized cut problem, based on the generalized eigenvalue problem. The authors also proved that the minimization of the two-way nominalized cut problem can be relaxed to the Rayleigh quotient. With respect to modularity maximization, Newman [46] proposed a method that groups vertices with a higher-than-average density

of edges connecting them. The problem is formulated as a maximization of the benefit function known as "modularity", which is afterwards formulated in terms of the eigenspectrum of a matrix (modularity matrix) and solved using eigen-decomposition and recursive two-way partitioning.

All the aforementioned methods for spectral graph clustering rely on a recursive two-way partitioning in order to identify a k-way partition. The specific clutster selected for further partitioning is selected based on various heuristics. However, if the graph has more than two clusters, there might be cases in which these methods produce unsatisfactory and unstable results [41]. For this reason, many authors proposed direct k-way spectral clustering methods to solve these problems.

Alpert et al. [47] were the first to propose the utilization of all the eigenvectors of the Laplacian for the k-way partitioning problem. Specifically, they utilized a greedy algorithm based on linear ordering of the eigenvectors, named Multiple Eigenvector Linear Orderings (MELO). The authors showed that this approach outperforms previous two-way partitioning schemes both in terms of speed and cluster quality. Bach and Jordan [48] proposed a new cost function in order to evaluate how close the eigenstructure of a laplacian matrix is to a given partition, and tried to minimize it in order to identify the clusters. Extending previous approaches to directed weighted graphs, Capoccia et al. [49] proposed a spectral clustering based on the correlations between the eigenvectors of the Laplacian to determine the cluster structure.

In addition to using the spectrum of the graph for clustering, other methods have been proposed for the production of graph-related features and the clustering of the vertices based on these features. Newman and Girvan [8] [9] proposed the use of the edge betweenness feature for graph clustering. According to the authors, edge betweenness is the number of shortest paths connecting any pair of vertices that pass through the edge. The proposed algorithm assumes the edges with high betweenness to be links between clusters, and partitions the network into clusters by removing a number of edges with high betweenness values. Each edge removal is followed by a recalculation of the betweenness values, since the shortest paths might have been altered. Fortunato et al. [50] propose a hierarchical clustering approach that is closely related to the betweenness method proposed by Newman and Girvan. The difference is that instead of the betweenness values, they use information centrality, which is defined for each edge as the relative decrease

in the value of the average efficiency [51] of the graph that is caused by removing the corresponding edge. According to Latora and Marchiori [51], the efficiency of a pair of edges is defined as the inverse of their distance in the graph, while the average efficiency of the graph is defined as the average of the efficiencies for all ordered pairs of vertices.

2.3 Visual clutter reduction

This section presents the methods proposed in the literature for visual clutter reduction. These methods change the mapping of the input data to the different visual variables, in order to enhance patterns and reduce information loss. Under this consideration, this section starts with a presentation of the visual variables, and continues with the presentation of the different clutter reduction techniques suggested in the literature.

2.3.1 Visual variables

Every visualization system utilizes a visual mapping function. The role of the mapping function is to map the input dataset to the different visual variables used by each visualization approach. There are many visual variables [52], each one representing a different channel of the human vision system.

Bertin [53] presents a set of seven visual variables, which are: Position, Size, Shape, Value-lightness, Color, Orientation, and Texture. Furthermore, Bertin presents how the different visual variables can be combined so as to define appropriate data marks, describing appropriate units of information. Some basic marks are 1) the Points, which are dimensionless locations on the plane, that might also have size, shape or color, and 2) the Lines, which represent information with a certain length, but no area and therefore no width. Other data marks are the Areas, Surfaces, and Volumes.

Mackinlay [54] extended the work of Bertin by ranking the different visual variables according to the task at hand and the type of information they can convey better. For example, he suggests that in order to represent Quantitative information, Position and Length are the most descriptive visual variables, while in the case of Nominal (categorical) data, Position and Color-hue are the best choice. Furthermore, Mackinlay also includes Connectivity as an important visual variable.

Recently, Ware [52] describes many aspects of the visual variables, the

way they interact, and their perception by the human vision system. In addition to the previous work, he describes many additional visual variables, such as Motion and Blinking, each of which is described by many characteristics, such as direction, speed, and frequency. Furthermore, Ware also presented Opacity, Shadow, Blur, and Depth as additional visual variables.

In the context of the present work, the clutter reduction procedure is demonstrated with regards to three visual variables: Position, Size, and Color. The proposed framework can be, however, applied in any other visual variable, as long as it has a specific number of quantization levels, to which the input information is mapped. It should also be emphasized that the human perceptual system has different decoders for different visual variables. Thus each visual variable should have a specific mapping function that takes into account its perceptual characteristics.

2.3.2 Use of non-liner deformations for clutter reduction

Over the last decade, many researchers have focused their efforts on reducing visual clutter [10] [55] and capturing important properties of the data for extracting meaningful information [55]. One of the most popular methods for clutter reduction (review available at [10]) is spatial deformation, where the positions of the visual objects (e.g. a node of a graph, a word in context preserving word clouds, or an object in an image) on the display changes to accommodate the enhanced visualization of data patterns. The spatial transformation is a method used to define the visual mapping function. The mapping function is where the transformation of the visual variables takes place, not only concerning the positions of the visual objects, but other visual variables as well, such as size and color. The mapping of the data to the visual variables is very important, since it constitutes the analytical power of every visualization system by changing the perceived patterns. The most commonly utilized mapping functions for generating visualizations are the linear and logarithmic mappings, which do not take into account the content of the visualizations. A few researchers have focused on this important aspect of the visualization.

Regarding spatial deformation of the position/size of the visual objects, Keim et al. [3] present the generalized scatterplots, a method to deform the positions of the points on the scatterplots and reduce visual clutter. The positions are deformed based on the underlining distribution of the data.

The authors also make use of point displacement around the initial point of reference, in order to further reduce visual clutter in high density areas. In the field of flow visualization, Tao et al. [56] suggest the utilization of a 3D deformation method, based on an entropy significance map. The proposed approach magnifies regions with high entropy in the flow visualizations, and reduces visual clutter. The previous methods were only adjusted for specific visualizations, and they did not take into account the human perception for clutter reduction. Based on these limitations, Wu et al. [14] present a general visualization resizing framework in order to scale every visualization approach on small displays. The authors utilize a clutter and a degree of interest map in order to generate a 2-dimensional significance map. The positions of the data objects (i.e. scatterplot point positions, graph layout, and word positions on a context preserving word cloud) are distorted in order to allow for the visualization of significant visualization areas on small screen sizes. Similarly, Wang et al. [57] propose a method for Focus+Context visualization of volumetric data based on 3D grid deformation. The authors utilize a significance map in order to magnify important regions at the expense of reducing the size of less significant regions. The limitations of the aforementioned approaches is that they utilize significance maps created by only considering one grid layer with a specific resolution, and thus, they lose important information. The introduction of the hierarchical significance map in this work, is able to more efficiently identify significant regions, and provides improved results.

Focus+Context techniques, such as fish-eye views [58], have been used for the visualization of data on mobile devices [59]. Sarkar and Brown [60] propose the use of fish-eye distortions for the visual exploration of graphs. Extending previous methods, which only relied on one type of deformation, Keahey and Robertson [61] present general non-linear magnification deformations, which are used for combining multiple deformations and smoothly interpolating between magnified and normal views. Carpendale et al. [62] present further improvements on fish-eye distortion methods, which make high "magnification in context" more feasible, by enabling lenses to be used on top of other lenses, and thus, effectively multiplying their magnification power. The limitation of the Focus+Context techniques is that the focus points must be selected by the user manually, depending on the task at hand. To address this issue, Feng et al. [63] propose a Focus+Context tech-

nique for the visualization of dynamic graphs. The significance of each vertex and edge in the graph can be either manually defined by the analyst, or automatically determined, based on graph related properties. The proposed approach utilizes deformations of a triangular mesh in order to achieve the desired Focus+Context result based on the significance values of the graph regions. The work presented in this thesis utilizes an automatically generated significance map that enables the detection of focus points for the magnification procedure.

Cluttering is not limited to information visualization, since it is a generic visualization issue. It is also frequently met in resizing images, in which significant objects are deformed and their perception becomes difficult. For this reason non-uniform resizing techniques have been proposed in the literature. These resizing methods can be classified into two categories [64]: discrete and continuous. Discrete approaches, such as seam carving [65], remove pixels from the image in order to preserve the most significant content, based on the significance of each pixel. Continuous methods [13] have the advantage of not removing image content. They process the image as a continuous domain and perform a continuous geometric deformation to change their size. The deformation process is guided by a significance map, which identifies which objects should be less deformed. Similarly to the aforementioned spatial deformation methods, the utilized significance maps in the case of image resizing are generated, taking into account only one grid layer resolution, and thus, lose important information.

The methods mentioned until now, deal only with the spatial visual variables, which refer to the size/position/shape of the visual objects. Very few approaches have been proposed for non-spatial visual variables, such as the color. For example, Thompson et al. [66] propose a method for color mapping that takes into account the input data distribution. The authors produce palettes utilizing perceptual color distance, in order to emphasize prominent values in the data. They also introduce two different color mappings, one for the visual detection of large differences in data values, and one for the visual detection of similar values, so that small differences can be detected. Eisemann et al. [67] propose the use of a simple projection technique based on angular interpolation, in order to distort the dataset before mapping its values to colors. The user of the visualization can select the desired distortion factor. Maciejewski et al. [68] utilize Box-Cox transformations

for transforming the distribution data as close to normal as possible, before applying a color scheme for choropleth maps. Unlike previous methods, the approach proposed in this paper is applied on both spatial and non-spatial visual variables.

Towards the direction of reducing visual clutter, this thesis presents an extension of the previous works on deformation using Focus+Context techniques. This extension refers to the introduction of a hierarchical approach for the generation of the significance map, which takes into account information from multiple layers of different resolutions, and thus, enhances the detection of significant patterns or objects. The proposed approach results in less distortion of significant objects, when compared to previous methods, leading to better results. Additionally, unlike previous approaches, the magnification method is applied on both spatial and non-spatial visual variables.

2.4 Critical summary

This section explains the main issues found in the literature and the main contributions of this thesis.

2.4.1 Graph mining

One of the main issues in graph mining is the formulation of descriptive graph features that would be able to efficiently discriminate between the different behavioral patterns of each node or subgraph in the graph. Since different features are able to capture different patterns, this selection is very important, because it can severely affect the accuracy of the graph mining algorithms. In this respect, this thesis focuses on the definition of novel node features in graph-based datasets, which are able to efficiently identify the behavioral patterns of each node, and thus, result in improved accuracy of the graph mining algorithms in a variety of applications.

Previous literature work on the definition of the graph features relied on using the entire graph as input, or using egonets [5]. Dynamic neighborhood sizes have not yet been considered in the literature. The use of dynamic neighborhood sizes captures information from different perspectives, and thus, is more informative for the graph mining task. Towards this end, this thesis proposes the use of dynamic neighborhood sizes, in order to capture the structure of the data from different behavioral perspectives

and detect correlations and relationships more efficiently when compared to static neighborhood sizes. The experimental results provide evidence that the use of dynamic neighborhood sizes can increase the accuracy of the graph mining algorithms in a variety of different datasets.

Finally, the feature extraction procedure must be fast, in order to enable the graph mining algorithms to be applied on large graphs. Under this consideration, the graph features proposed in this thesis can be applied on large graphs, since they are efficient to compute.

2.4.2 Graph visualization

One of the main challenges in the graph visualization field is how to deal with visual clutter [10]. Visual clutter is very common on graph visualization since the graphs have an inherently small visualization capacity, due to the fact that the graphs are comprised of a small number of available edge widths and node sizes that are used to visualize the edge and node attributes. Another reason that can cause visual cluttering is when large graphs are visualized on small display devices.

Previous literature work proposed the use of non-linear deformations based on content aware significance maps, in order to reduce the amount of visual clutter. These non-linear deformations are used in order to magnify highly cluttered regions of the data, and enhance their perception by the human analyst. One of the most common methods for non-linear deformation of the data is to utilize a regular grid [13] [14]. Although regular grids, work well, they waste computational resources due to the large number of optimization variables, since not all the different regions of the input data are equally important. This indicates that it is possible to utilize a multi-resolution grid, which provides more quantization levels on regions with high significance, and less quantization levels on less significant regions. This would have as a result the reduction of the optimization parameters while still maintaining high performance. While multi-resolution grids have been utilized in the context of volume rendering [15], their application on non-linear deformation has not been thoroughly investigated in the literature. Under this consideration, this thesis utilizes a multi-resolution grid for non-linear deformation and compares it to other state-of-the-art methods, so as to research their effect in the grid deformation procedure and evaluate the final results.

Finally, the previously proposed approaches for non-linear deformations utilize significance maps that were created by taking into account only one grid resolution. This fact, however, makes the identification of significant regions less accurate, since high resolutions lose significant regions of large size, while low resolutions cannot encode significant regions of small size. Inspired by the hierarchical way human visual perception works [16], this thesis utilizes a hierarchical significance map in order to address this issue. The proposed approach is able to efficiently identify both small and large sizes of significant regions, since it takes into account information from multiple layers with different grid resolutions.

Chapter 3

Novel Graph Features for behavioral analysis of the graph nodes

Contents

3.1 Graph structure 42
3.2 Graph neighborhoods 43
3.3 Extraction of graph features 45
3.4 Complexity analysis 50
3.5 Security analysis 51

This section presents the graph features proposed in this thesis. A schematic representation of the procedure followed for the generation of the graph features is illustrated in Figure 3.1. Firstly, the raw data are transformed into graph-structured data (Section 3.1). Afterwards multiple neighborhoods with different sizes are extracted from each node of the graph (Section 3.2). The proposed graph features are extracted for each of these neighborhoods (Section 3.3). These features are used by the analysis methods in order to derive the analysis results. Finally, the results are presented to the analyst.

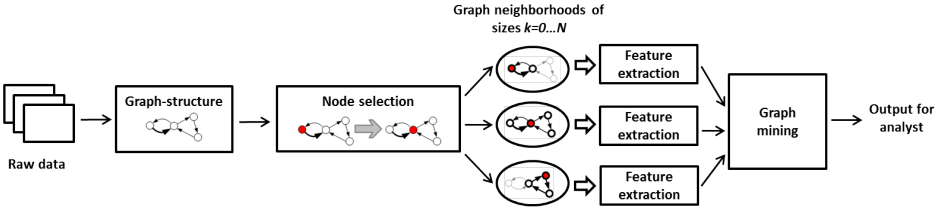


Figure 3.1: A schematic representation of the procedure followed for the generation of the graph features. Firstly, the raw data are transformed into a graph. Multiple graph neighborhoods of different sizes k are extracted for each node in the input graph. Multiple features are afterwards extracted from each graph neighborhood, which are fed to the specific analysis techniques. Finally, the results are presented to the analyst.

3.1 Graph structure

A graph is a graphical representation of input information that captures its main structural and/or relational characteristics. A graph $G(V, E)$ comprises of a set of vertices $V = \{v_1, v_2, ..., v_{|V|}\}$ and a set of edges $E = \{e_1, e_2, ..., e_{|E|}\}$, where $E \subset V \times V$. A graph in this case is a weighted and directed graph in which $W : E \to \mathbb{R}$ is a function that takes as input a specific edge and returns its corresponding edge weight.

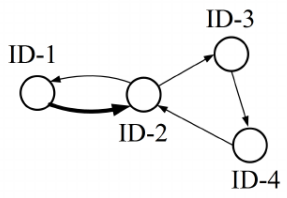
For example, the nodes of the graph are the source and destination in a mobile network, e.g. user IDs for calls/SMS and IP servers for cellular Internet traffic. The edges connect specific sources to their destinations and are directed, while the weights of the edges represent specific attributes of the Call/SMS/Internet records, e.g. the number of calls/SMSes, or the size of the Internet packets. The exact weights of the edges are selected based on the task at hand, and the scenario under investigation. In this respect, the graph provides a holistic view of the communication activity in the mobile network, and affords a first step towards identifying anomalous behaviors.

Figure 3.2(a) shows an example of records representing the origin and destination of each communication event for four entities: ID-1, ID-2, ID-3, and ID-4. In this example, ID-1 communicated two times with ID-2, and ID-2 one time with ID-3. The graph for this example is created by associating each user with a vertex, and by connecting users that are involved in at least one communication event. The edge weights represent the number of communication events between entities. The direction of the edges encodes

the direction of the communication. The resulting graph of the example in Figure 3.2(a) is illustrated in Figure 3.2(b), where the widths of the edges reflect their weights: the edge connecting ID-1 to ID-2 has weight 2, and all other edges have weight 1.

Record ID	Origin	Destination
1	ID-1	ID-2
2	ID-1	ID-2
3	ID-2	ID-1
4	ID-2	ID-3
5	ID-3	ID-4
6	ID-4	ID-2

(a)



(b)

Figure 3.2: (a) Example of record data representing the origin and destination of each communication event. (b) The graph structure of the communication events in the example data of (a). The weights of the edges represent the number of communication events between the corresponding entities (e.g. ID-1 communicated two times with ID-2).

3.2 Graph neighborhoods

The graph structure provides a method to represent the node activities in a network. Yet in order to identify specific patterns in the graph (e.g anomalous behaviors), multiple smaller graphs are extracted from the initial graph, each representing the neighborhoods of a vertex. The resulting graphs are subsequently used for feature extraction and their subsequent analysis.

Let $N_k(v_i)$ denote the set of k-neighbors of vertex $v_i \in V$. This set is comprised of all the nodes that have graph geodesic distance smaller than or equal to k, where the geodesic distance $GD(v_i, v_j)$ between two vertices v_i and v_j is the length of the shortest path connecting them. Hence, the k-neighbors of vertex $v_i \in V$ are defined as:

$$N_k(v_i) = \{v_j \in V | \forall j \; such \; that \; GD(v_i, v_j) \leq k\} \tag{3.1}$$

Graph neighborhood extraction is the process of creating a new graph for each vertex v_i, denoted as $G_i(V_i, E_i)$, where $V_i = N(v_i) \bigcup v_i$, $E_i = \{e_j = \{v_k, v_h\} \| \forall e_j \in E,\ and\ v_k, v_h \in V_i\}$, and E is the set of edges of the initial graph G. Figure 3.3(a) shows an example of k-neighborhood graph for different values of k, where the graph is created for the central large vertex; the larger the value of k the larger the resulting subgraph G_i.

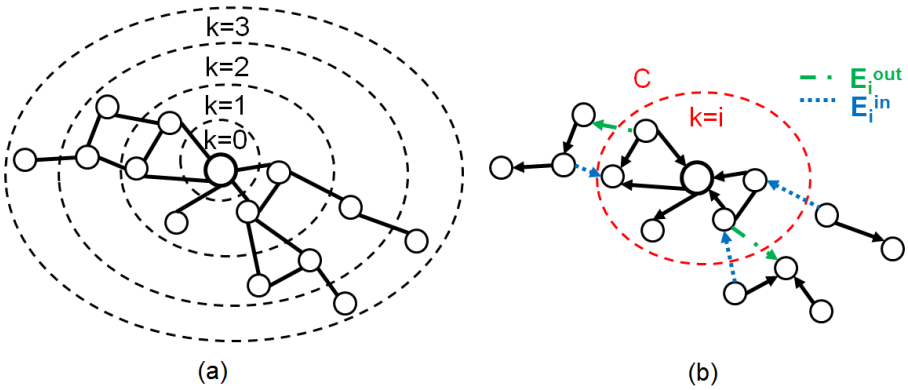


Figure 3.3: (a) Example of k-neighborhood graph for the central large vertex under different values of k. (b) Example of inward and outward directed edges for the k-neighborhood graph, where $k = i$, which comprised of all the edges that intersect the dashed red ellipsoid C.

Since the graph is directed, we denote by e_j^s the source vertex of edge $e_j \in E_i$, and by e_j^d its destination vertex. For each k-neighbors of vertex $v_i \in V$, the set of outward directed edges represent all the edges that have their source in the set $N_k(v_i)$ and their destination outside of this set. More precisely, the set of outgoing edges for $N_k(v_i)$ is given by:

$$E_i^{out} = \{e_j \in E_i | \forall e_j^s \in N_k(v_i) \text{ and } e_j^d \notin N_k(v_i)\} \tag{3.2}$$

The set of ingoing edges E_i^{in} for the set $N_k(v_i)$ is defined in a similar manner. Figure 3.3(b) shows the set of inward and outward directed edges for each k-neighborhood of the central vertex in Figure 3.3(a). It can be observed that all the edges that intersect the dashed red ellipsoid C belong to either the set of ingoing edges (edges that are directed towards the centre) or the set of outgoing edges (edges that are directed away from the centre).

Figure 3.4 illustrates the graph neighborhood extraction process, showing the graphs that are created in each step of the iteration using k-neighborhoods

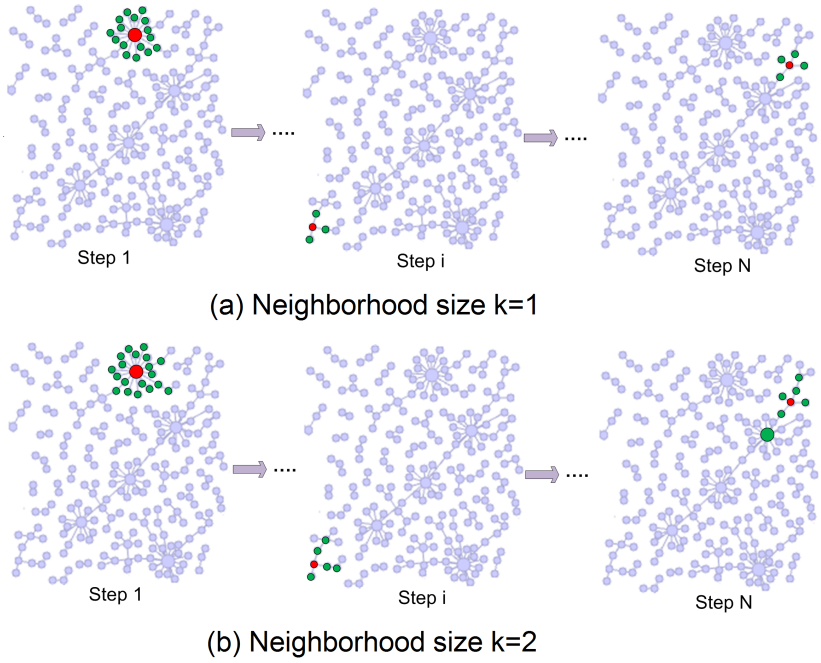


Figure 3.4: An example of graph neighborhoods with $k = 1$ in (a) and $k = 2$ in (b). The red vertex represents the active vertex in each step, and green vertices are its k-neighbors in the graph.

for $k = 1$ and $k = 2$. All the vertices of the graph are traversed, and for the specific vertex under consideration in the current iteration (called active vertex), all its k-neighborhoods are utilized to create the corresponding neighborhood graph. The red node in Figure 3.4 represents the active vertex in each step and the green vertices are its k-neighborhoods in the graph. The end result is the creation of a set of neighborhood graphs, one for each vertex: $\{G_1, G_2, .., G_{|V|}\}$, where $|V|$ is the number of vertices of the entire graph G.

3.3 Extraction of graph features

This section describes how the aforementioned graph neighborhoods can be used to extract relevant features that are able to characterize the network activity of each entity (e.g. user or server). Many features have been proposed in literature for egonets (an overview is provided in [5]), including but not limited to [27] [28]: number of neighbors, node in/out degree, number

of triangles, total weight, principal eigenvalue, and number of within-egonet edges. These features take into account only the egonet of each node. This fact limits their descriptive capacity. In addition, these features do not consider the interactions of the node neighborhoods with the rest of the graph nodes. The features proposed in this thesis are extracted from multiple neighborhood sizes, and take into consideration multiple interaction patterns. The justification for choosing these features is also provided in the paragraphs that follow. Comparison with other feature schemes is provided in Chapter 5.

Table 3.1: An overview of the proposed graph features.

#	**Feature**	**Short Description**	**Definition Range**
1	Volume	The number of edges of the graph	$k \geq 1$
2	Edge Entropy	The entropy of the weights of the edges in the graph	$k \geq 1$
3	Graph Entropy	The graph entropy, which captures the structural characteristics of the graph	$k \geq 1$
4	Edge Weight Ratio	The ratio of the sum of weights of outward to inward edges.	$k \geq 0$
5	Average Outward/Inward Edge Weight	The average value of the weights of the Outward/Inward Edges.	$k \geq 0$

Table 3.1 provides an overview of the proposed graph-based features, while more details for each feature are provided in the paragraphs that follow. These features are extracted from multiple neighborhood sizes $k \in \{0, ..., N\}$. N represents the maximum neighborhood size to be taken into account, and

is equal to the graph diameter[1]. The definition range for the values of neighborhood sizes k for each feature is also provided in Table 3.1.

3.3.1 Volume

The volume feature f_{vol} captures the size of the graph, and consequently the degree of network activity, which is important for identifying anomalies related to large network activities. The volume feature is defined as follows:

$$f_{vol}^{G_i} = \sum_{e_j \in E_i} g\left(W(e_j)\right), \tag{3.3}$$

where:

$$g(x) = \begin{cases} 1, & \text{for } |x| \neq 0 \\ 0, & \text{for } |x| = 0 \end{cases}$$

and G_i denotes the neighborhood graph of vertex V_i, E_i its set of edges, and $W(e_j)$ the weight of edge $e_j \in E_j$.

3.3.2 Edge Entropy

Shannon's entropy takes as input a set of symbols and their underlying distribution, and returns their average amount of information; the input symbols represent distinct entities in a set of objects, which in this case are the edge weights. Hence the edge entropy f_{ee} captures the amount of information in the edge weights as characterized by their underling distribution. Large difference in the entropy value between distinct graphs indicates that they have large differences in their weight distributions. The edge entropy of a graph G_i is defined as:

$$f_{ee}^{G_i} = -\sum_{j=1}^{Y^i} \frac{y_j^i}{y_{total}^i} \log\left(\frac{y_j^i}{y_{total}^i}\right) \tag{3.4}$$

where Y^i is the number of different edge weight instances of graph, G_i, y_j^i is the number of occurrences of the j-th weight, and $y_{total}^i = \sum_{j=1}^{Y^i} y_j^i$ is the total number of weight occurrences. It should be noted that this feature characterizes two possible types of network behavior: change in the volume of the edge weights, and change in their underling distribution. Large network

[1]The graph diameter is the longest shortest path (i.e. the longest graph geodesic distance) between any two graph vertices of a graph

disturbances not only increase the volume of the edge weights, but also have a direct effect on their distribution. The entropy feature is introduced to capture this effect.

3.3.3 Graph Entropy

Graph entropy f_{ge} measures the structural information content of the graph based on the distribution of the edge connections. The graph entropy is introduced in [32] as a problem of determining the best possible encoding of the information emitted by a source in which pairs of symbols may be indistinguishable. As noted earlier, the symbols represent distinct entities in a set of objects which, in the case of graph entropy, are the vertices. Two symbols are distinguishable if they are connected through an edge, and indistinguishable otherwise. Examples of the entropy value of known graphs (illustrated in Figure 3.5) include:

- A complete graph with n vertices has graph entropy $log_2(n)$ bits, since all the vertices are distinguishable. In this case, f_{ge} is equivalent to Shannon's entropy.
- A complete bi-partite graph has entropy 1 bit, since it can be partitioned into two sets of indistinguishable vertices.
- A graph with no edges has entropy 0 bits, since all the vertices are indistinguishable.

More formally, Korner's entropy is defined as [69]:

$$f_{ge}^{G_i} = \min_{X,Y} I(X \wedge Y) \tag{3.5}$$

where $I(X \wedge Y)$ is the mutual information of the variables X and Y that have the following properties: the variable X is uniformly distributed and taking its values on the vertices of G_i, while Y on the stable sets of G_i, and their joint distribution is such that $X \in Y$ with probability 1. A subset Y of the vertices V_i of an undirected graph $G_i = (V_i, E_i)$ is a stable set if no edge in the graph has both endpoints in Y.

3.3.4 Edge Weight Ratio

The edge weight ratio feature f_{wr} captures the total difference in the values of the weights between the edges of a graph G_i directed outward from v_i

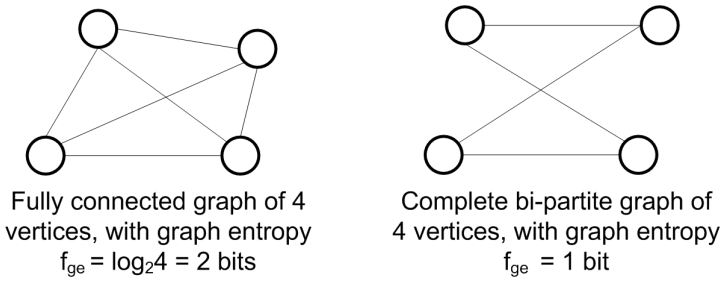


Figure 3.5: Examples of graph entropy.

and inward to v_i. Thus, this feature captures anomalies regarding large imbalances between the communication directions. For example, the number of SMSes sent by spammers is usually much larger than what they receive, while a normal user is likely to exhibit comparable number of messages in both directions. We define the edge weight ratio feature of graph G_i as:

$$f_{wr}^{G_i} = \frac{\sum\limits_{e_j \in E_i^{out}} W(e_j)}{\sum\limits_{e_j \in E_i^{in}} W(e_j)} \tag{3.6}$$

where E_i^{out} and E_i^{in} are the set of outward and inward directed edges for the k-neighbors of graph G_i, as described in (3.2).

3.3.5 Average Outward/Inward Edge Weight

The average outward edge weight f_{avout} represents the ratio of the volume of traffic generated by a node to the number of its destinations:

$$f_{avout}^{G_i} = \frac{\sum_{e_j \in E_i^{out}} W(e_j)}{|E_i^{out}|} \tag{3.7}$$

This feature can be useful to detect, for example, spammers or port scanning worms that typically have high communication activity directed to a large number of destinations, resulting in an average outward edge weight close to 1. On the other hand, normal users are likely to communicate with a small set of destinations, leading to $f_{avout} >> 1$ for high activity users. Similarly, the average inward edge weight f_{avin} can be defined as in (3.7) but instead using E_i^{in}; this feature will exhibit very small values (close to 0) for the malicious behaviors mentioned above, and moderate to high values for normal behavior, which may further increase when a user receives anomalous traffic.

3.4 Complexity analysis

The storage requirements of the proposed anomaly detection approach are comprised of the features extracted for each node of the graph. There are 5 features extracted from different neighborhood sizes. Experiments on multiple cases showed that only features collected from neighborhood sizes $k < 3$ are informative. Thus, the storage complexity of the proposed approach is $O(20|V|) = O(|V|)$, which is linear with respect to the number of nodes $|V|$ of the entire graph. For the computation of the computational complexity, the proposed anomaly detection approach is separated into three steps, namely: 1) Extraction of graph neighborhoods, 2) Feature extraction, and 3) Classification. The complexity of extracting graph neighborhoods of size k, utilizing a breadth-first-search approach starting from one node is $O(e^k)$, where $e = |E|/|V|$ is the average number of edges for each node, while $|E|$ and $|V|$ are the number of edges and nodes of the entire graph. Performing this computation for each node, the complexity of extracting graph neighborhoods of size k becomes $O(|V|e^k)$. The complexity of the feature extraction step (not taking into account the graph entropy feature) is related with the computation of multiplications and additions on the edge weights of each neighborhood, which is also related to the number of edges of each neighborhood. These computations are computed for each node, and thus, the complexity of the feature extraction step, without the graph entropy feature, is $O(|V|e^k)$. In this complexity, however, we also need to add the extraction of the graph entropy feature. For the calculation of the graph entropy, we first need to calculate the stable sets of the graph, which is in general an NP-hard problem [70]. This means that for large graphs the calculation of the graph entropy is very difficult. For this reason, the graph entropy feature is calculated only for small values of (i.e. $k \leq 2$). For very large graphs, this computation can be further reduced to $k = 1$ (the experimental results show that the graph entropy feature is most accurate for $k = 1$ in the cases under consideration). For $k = 1$, and given the fact that all nodes in the graph neighborhood are connected with the central node, the calculation of the stable sets has complexity $O(|V|e)$, since the problem is solved by simple traversal of the nodes in the graph neighborhood. Thus, the total complexity of the feature extraction step is $O(|V|e^k)$. Finally, the complexity of using a trained random forest classifier is $O(1)$ for one node and, thus, $O(|V|)$ for the

entire graph. Thus, the total complexity of the proposed anomaly detection approach is $O(|V|(|E|/|V|)^k)$.

Based on the total computational complexity $O(|V|(|E|/|V|)^k)$, the storage complexity $O(|V|)$, and the nature of the problem, , there are two main approaches used in order to apply the proposed approach on a very large graph. The first approach is to use of multi-threading and parallel programing. Specifically, the graph is separated into smaller subgraphs, comprised of neighbors around central nodes, which are independently used for feature extraction and classification. The central nodes are chosen so that each subgraph has roughly equal size, and thus, the computational burden is distributed equally to the different processes. Similarly to other big data software that is designed for large, distributed networks, such processes could be carried out by utilizing cloud-based infrastructures. The second approach is to reduce the complexity of the problem. There are two main methods that can reduce the complexity of the algorithm: 1) Reduce the size of the graph, 2) Extract less computationally demanding features. With respect to reducing the graph size, filtering can be applied in order to remove edges with low weight, which contribute less to the patterns under consideration, as done in [34]. Alternately, less computationally demanding features can be extracted depending on the task at hand. For example, for the detection of SMS spammers, the weights of the outward edges carry a lot of significant information, and thus, for such an application, only features related to this pattern can be extracted at the first iteration. More features can be extracted in later iterations of the algorithm for a more in depth analysis, after the total number of nodes has been reduced in previous iterations using information from the reduced set of features. Additionally, the complexity of the feature extraction step can be also reduced by limiting the number of available neighborhood sizes k, for example to $k \in [0, c]$, where c is a small constant.

3.5 Security analysis

One might argue that in the case that the proposed anomaly detection features were publicly known, the attackers might be able to employ a defense strategy to avoid detection. This section presents an analysis of when and how this might be feasible.

In order to be able to bypass detection, the attacker must have knowledge of not only the features, but also the regions of the space defined by these features that are considered to be anomalous by the classification algorithm. The shape of these regions depends on the specific classifier that is used. In the case of the Random Forest (RF) classifier used in the context of this thesis, the classification regions are hyper-rectangles. More formally, each region is defined as a set of vectors in the n-dimensional space as follows:

$$R_j \subset \mathbb{R}^n, \ where \bigcup_j R_j = \mathbb{R}^n \tag{3.8}$$

where n is the number of features. Afterwards, for binary classification the regions are separated into two sets, the set of normal regions $N = \{R_j\}, \forall j$ such that R_j corresponds to normal feature vectors and $A = \{R_i\}, \forall i$ such that R_i corresponds to anomalous feature vectors. Figure 3.6 shows two examples of region separation, corresponding to two classifiers. In this example two features are utilized in order to define a two dimensional space. Red regions correspond to anomalous feature vector and green regions correspond to normal feature vectors.

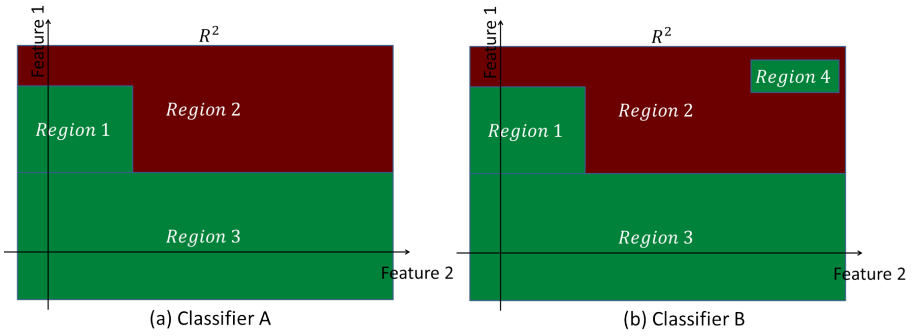


Figure 3.6: Examples of classifier regions for a two dimensional space, i.e. utilizing two features. Red regions correspond to anomalous feature vector and green regions correspond to normal feature vectors.

Knowledge of the aforementioned classification regions and the feature space could potentially allow an attacker to bypass detection. For example in the case of the classifier depicted in Figure 3.6(a), the attacker must operate in the regions *Region 1* and *Region 3*, avoiding *Region 2*. Thus, in order to increase the value of the Feature 1 as much as possible, the attacker must

operate inside *Region 1*. This means that the utilization of this different behavior by the attacker in order to bypass detection, acts in itself as a mitigation procedure, since it reduces the effects of the attack to the owners of the network devices and/or the network itself.

In the case of Figure 3.6(b), however, the attacker could further increase the value of the Feature 1 by trying to operate inside *Region 4*, which is a safe island inside the anomalous region *Region 2*. Knowledge of such safe islands could potentially allow the attacker to act more freely and affect the network. The existence of such safe islands depends on the parameters of the classifier and the dataset used during its training phase. This means that the designer of the security system, and in the general case the designer of every security system, must adjust the parameters and the training dataset of the algorithm in order to try to eliminate such safe islands, and subsequently limit the available feature vectors that an attacker can use. It should be highly underlined, however, that this issue does not have to do with the proposed features, but has to do with the classifier utilized for detection, and is inherent in any security system that uses features for detection purposes.

Chapter 4

A Hierarchical Magnification Approach for enhancing the Insight in Graph Visualizations

Contents

4.1 Generation of the hierarchical significance map . 56
4.2 Multi-resolution grid 59
4.3 Multi-dimensional grid deformation 60
4.4 Different choices for the generation of the significance map 65

This section presents the proposed clutter reduction approach in multiple dimensions. Two significance maps are utilized, namely, saliency and entropy. These significance maps are enhanced using a hierarchical approach, in order to generate new maps, that are better able to characterize the significant regions of the visualization. The reason for this enhancement is that the hierarchical approach takes into account information form multiple significance layers with different resolutions and combines them. Thus, the hierarchical significance map is able to identify small significant regions visible from a high resolution significance map, and large significant regions visible from the low resolution map.

The hierarchical significance map is used to guide the non-linear de-

formation procedure. The result is the magnification of regions with high significance values, at the expense of reducing the size of less significant regions. As a result, significant regions are enhanced, and are easier perceived by the analyst.

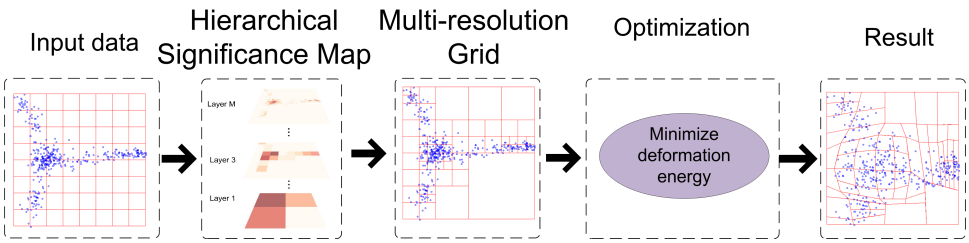


Figure 4.1: Method overview: The first step is the generation of hierarchical significance map. Afterwards the multi-resolution grid is generated and the optimization is applied onto this grid based on the hierarchical significance map in order to identify its optimal deformation. The result is a deformed visualization space which enhances cluttered areas of the display.

Figure 4.1 presents an overview of the proposed visualization non-linear deformation approach. Taking into account the input dataset, the first step is the generation of the hierarchical significance map, which defines regions of the input space that are important and need to be magnified. Independent of the type of the input data that can be either N-dimensional points, word clouds, or images, multiple grid resolutions are defined, while a significance map is generated for each such resolution. The significance maps are afterwards combined for the generation of the hierarchical significance map, which takes into account both small and large significant regions, which contain small and large data patterns.

The second step is the generation of the multi-resolution grid, which assigns a larger number of hyperrectangles in the regions of the input space with high significance. The hyperrectangles are N-dimensional representations of rectangles. Each hyperrectangle represents a cell in the grid. In the case of 2 dimensions each cell is a rectangle, and in 3 dimensions each cell is a rectangular cuboid. The advantage of the multi-resolution grid, when compared to the uniform grid, is that it more effectively preserves local patterns in the deformed space. Furthermore, the optimization space (number of variables) of the multi-resolution grid is much smaller when compared to the uniform grid, and thus, the optimization procedure is much faster.

Afterwards, the grid deformation energy is defined based on the significance of each hyperrectangle. This energy term allows significant hyperrectangles to be more magnified, in order to enhance the patterns that exist in them. The optimization procedure minimizes the grid deformation energy and enhances significant regions of the input data/image spaces. The core idea of the optimization approach is to assign to each hyperrectangles a size that corresponds to its underling significance, while keeping their distortion to a minimum.

4.1 Generation of the hierarchical significance map

The hierarchical significance map takes into account multiple grid resolutions, and is able to efficiently identify significant regions. This hierarchical procedure has been used before in the literature for the generation of saliency maps on images [71] [16] and meshes [72]. The advantage of this hierarchical definition over the single-layer approach, is that it captures both large and small patterns [73], since low resolution grids might lose small patterns but can capture large patterns, while the opposite stands true for high resolution grids.

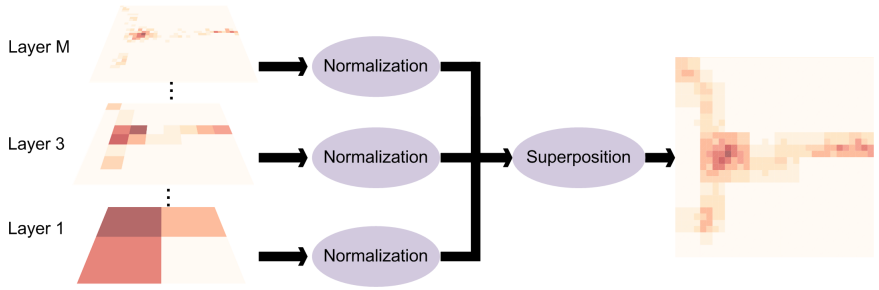


Figure 4.2: Generation of the hierarchical significance map. The significance map is crated for each layer of the grid hierarchy. Afterwards, each layer is normalized, and all the layers are superpositioned for the generation of the final hierarchical significance map.

The procedure of generating the hierarchical significance map is illustrated in Figure 4.2. A hierarchy of layers is generated, each one covering the input space with different grid resolutions. The significance map is calculated and normalized for each layer separately. Afterwards, the significance maps of each individual layer are superpositioned for the generation of the

hierarchical significance map which will be used for the deformation procedure.

For the generation of the hierarchical significance map, the grid of each layer of the hierarchy must be defined. Each multi-dimensional grid $G = (V, E, F)$ is comprised of a set of vertices V, a set of edges E and a set of hyperrectangles F, where $V = [\boldsymbol{v_0^T}, \boldsymbol{v_1^T}, ..., \boldsymbol{v_{end}^T}]$ and $v_i \in \mathbb{R}^N$ is the vertex position in the N-dimensional space. The vertices and edges partition the input space into a multi-dimensional grid comprised of hyperrectangles of the same size. The reason why the hyperrectangles have the same size is that in each layer, the grid is uniform, i.e. each dimension is equally partitioned. An example of equal hyperrectangle size is provided in Figure 6.1(a).

Each hyperrectangle $f_i \in F$ is comprised of a set of edges $E(f_i) \subset E$ and vertices $V(f_i) \subset V$. The total number of hyperrectangles is $\prod_{i=1}^{N} n_i$, where n_i is the number of partitions per dimension. Each partition in a dimension is a line segment, equal in size with all the other partitions in the same dimension.

Let $G_l = (V_l, E_l, F_l)$ denote the grid in *layer* l with n_l partitions per dimension. Each grid G_l has $n_l = 2^l$ partitions per dimension, e.g. the grid of *layer* 1, G_1 has $n_1 = 2$ partitions per dimension. The value of l is within $l \in [1, .., M]$, where M is the maximum level allowed, and depends on the maximum resolution allowed by the multi-resolution grid defined in Section 4.2.

The first step towards the generation of the hierarchical significance map is the definition of a single-layer high resolution significance map in the input data space. There are multiple choices for the selection of the appropriate single-layer significance map. In the context of this work, the experimental results have been created and compared using both entropy [74] and saliency maps [14] [13]. Afterwards, the calculation of the significance value for each hyperrectangle is defined as the average significance value of the region covered by it, while the significance map of layer l is defined as S_l. The significance value of each hyperrectangle f_i^l is denoted as $s_i^l \in S_l$. The second step is the normalization of significance maps of each individual layer, for the subsequent superposition of the maps. The operator $\aleph(.)$ defined in [16] is utilized so as to normalize each significance map,.

The same normalization operator $\aleph(.)$ has also been used in other research works (e.g. [72] and [75]), and has as an effect the promotion of sig-

nificance maps, which are comprised of a small number of high significance values, while efficiently suppressing significance maps with a large number of similar values.

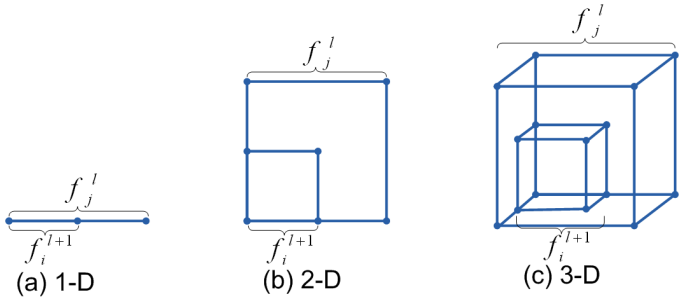


Figure 4.3: Hyperrectangle intersection in multiple dimensions, in which hyperrectangles f_j^l and f_i^{l+1} intersect. In the specific case that $n_l = 2^l$ partitions per dimension, if two hyperrectangles intersect, it follows that one contains the other.

The final hierarchical significance map is created for the resolution of the last layer of the grid hierarchy G_M. To achieve this, let us denote as $Q\left(f_i^l\right)$, where l is the level of the hierarchy and $f_i^l \in F_l$ is a specific hyperrectangle belonging to layer l, as the set of hyperrectangles from all the layers of the hierarchy, so that they intersect with the N-dimensional hyperrectangle f_i^l. In the specific case of this paper that $n_l = 2^l$ partitions per dimension are considered, if two hyperrectangles intersect, it means that one contains the other (as shown in Figure 4.3). The same stands true for every $n_l = r^l$ for $r \in \mathbb{N}_{>1}$. In the same notion $Q_s\left(f_i^l\right)$ is the set of significance values of all the hyperrectangles belonging to $Q\left(f_i^l\right)$. The superpositioning of the different maps is performed on *layer* M and the final hierarchical significance map, denoted as S^{hier}, is calculated for each hyperrectangle in *layer* M. Let s_i^h be the final hierarchical significance value of hyperrectangle $f_i^M \in F_M$. The calculation of s_i^h is described in the equation that follows: $s_i^{hier} = \sum_{s_j^l \in Q_s\left(f_i^M\right)} \aleph\left(s_j^l\right)$.

Finally, the hierarchical significance map is defined as $S^{hier} = \bigcup s_i^{hier}$.

Compared to the single layer significance map, the hierarchal significance map is smoother. Although similar smoothing results can be obtained by applying a simple blurring function, such as Gaussian, on the single layer map, the hierarchical map is able to better identify both small and large

regions of high significance. On the other hand, the application of a blurring function enhances the large regions of significant cells in the grid, at the expense of losing the significance of smaller regions that contain equally significant cells in the grid. An example of comparison between Gaussian blurring and hierarchical significance map is illustrated in Figure 4.4.

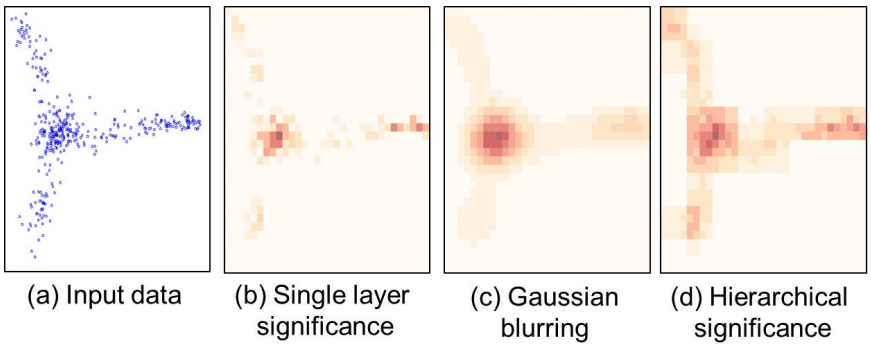


Figure 4.4: An example of comparison between Gaussian blurring and hierarchical significance map. (a) Input dataset. (b) Single layer entropy significance map. (c) Gaussian blurring of single layer map. (d) Hierarchical entropy significance map. Small regions of significant grid cells are lost by applying the Gaussian blurring. The Hierarchical significance map is able to identify all the significant regions, while also being smoother than the single layer map.

4.2 Multi-resolution grid

The multi-resolution grid is a non-uniform grid on the N-dimensional space, in which the distribution of hyperrectangles is not uniform, but instead distributes more hyperrectangles in regions with high significance. The advantage of the multi-resolution grid is that regions with high significance are less distorted when magnified, since they have higher resolution. In addition, the optimization space of the vertex positions of the grid is much smaller when compared to the full resolution grid, and thus, has better speed performances as well as produces better results.

The algorithm for the creation of the multi-resolution grid is similar to the one used for the creation of a quad tree. The algorithm takes as input a specific user-defined significance *threshold*. In the context of this work this

threshold is set to a value that results in the reduction of the number of optimization variables by 20% to 50%, depending on the application. This algorithm has as a result that the significance value of each hyperrectangle is the largest possible value, which is smaller than the provided *threshold* (i.e. the infimum). The multi-resolution grid is denoted as $G_{mul}(V_{mul}, E_{mul}, F_{mul})$, where F_{mul} are hyperrectangles of the multi-resolution grid, V_{mul} the vertices and E_{mul} the edges.

The significance value s_i^{mul} of each hyperrectangle $f_i^{mul} \in F_{mul}$ in the multi-resolution grid is calculated from the hierarchical significance map S^{hier} defined in Section 4.1. For this calculation, the values of hierarchical significance map S^{hier} have to be mapped to the hyperrectangles in F_{mul}. As explained in Section 4.1, the hierarchical significance map S^{hier} is defined for each hyperrectangle of the grid G_M in the maximum layer of the grid hierarchy *layer* M.

Let $P(f_i^{mul})$ denote the set of all hyperrectangles from grid G_M, which intersect with $f_i^{mul} \in F_{mul}$. Additionally, let $P_s(f_i^{mul})$ denote the set of significance values of all the hyperrectangles belonging to $P(f_i^{mul})$. Figure 4.5(b) presents the hypercubes of the multi-resolution grid. The sets of hypercubes $P(f_i^{mul})$ and $P(f_j^{mul})$ are highlighted in this figure using red and green colors respectively. The procedure of calculation of the significance value of each hyperrectangle $f_i^{mul} \in F_{mul}$ in the multi-resolution grid G_{mul} is defined as follows:

$$s_i^{mul} = \sum_{\forall s_j^{hier} \in P_s\left(f_i^{mul}\right)} s_j^{hier} \tag{4.1}$$

The significance map of the multi-resolution grid G_{mul} is defined as $S^{mul} = \bigcup s_i^{mul}$.

4.3 Multi-dimensional grid deformation

This section follows the work presented in [13] and [14] regarding the grid deformation procedure and for quantifying the optimum distortion of each hyperrectangle. The deformation procedure presented here has been firstly proposed by [13] for image resizing. The same procedure has afterwards been utilized by [14] for resizing data visualizations. The deformation procedure is briefly presented here. More details can be found in [13] and [14].

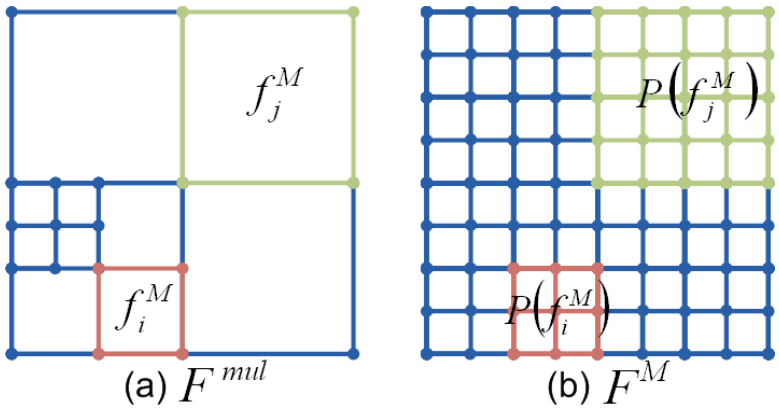


Figure 4.5: (a) The hypercubes of the grid in *layer* M of the hierarchy. Hypercubes f_i^{mul} and f_j^{mul} are highlighted in red and green colors. (b) The hypercubes of the multi-resolution grid. The sets of hypercubes $P(f_i^{mul})$ and $P(f_j^{mul})$ are highlighted using red and green colors.

The purpose of grid deformation is to enlarge significant regions of the input space, and visualize any patterns that were previously hidden due to high cluttering. The deformation method takes as input a grid $G = (V, E, F)$ and its significance map S, where V is the matrix of vertices, E matrix of edges, and F the matrix of hyperrectangles. These matrices have specific relationships between them, i.e. $F = QV$ and $E = HV$, where Q and H are also matrices which depend on the grid. The result of deformation is a new grid $G' = \left(V', E', F'\right)$, created by changing the positions of the vertices in V, according to the given significance map.

According to [13] and [14], the ideal deformation of each $\boldsymbol{v_i} \in V$ into a new position should be defined as $\boldsymbol{v_i'} = c * \boldsymbol{v_i}$, where c is the scale factor. This equation refers to the uniform scaling case, and requires an extension of the area covered by the data in the input space. In the case that the area covered by the data in the input space is considered static, this extension is not possible. One of the alternatives proposed in the literature (see Section 2) is to introduce a non-linear deformation of each hyperrectangle, so that more space is given to more significant hyperrectangles [13] [14].

The main objective of the utilized grid deformation method is to minimize the total deformation energy, defined as the distance from the uniformly scaled position. Let $f_i \in F$ be a hyperrectangle defined as $f_i^T = q_i V$, where

q_i is a $2^N \times |V|$ matrix (N is the dimension) and its element in the r_{th} row and c_{th} column is defined as:

$$q_{i,rc} = \left\{ \begin{array}{cc} 1 & , if\, v_c \in f_i,\, and\, f_{i,r} = v_c \\ 0 & , else \end{array} \right\} \tag{4.2}$$

where f_i, r is the r_{th} vertex in f_i. Given $Q = \left[q_0^T, q_1^T, ..., q_{|F|}^T\right]^T$, the matrix of hyperrectangles is defined as $F = QV$.

The uniformly scaled hyperrectangle is defined as $f_i^{'} = c_i f_i$, where c_i is the desired scale matrix for the $i_t h$ hyperrectangle. Let $F^T = [f_1, f_2, .., f_{|F|}]$ be a matrix of all the quads, and W_F be a $|F| \times |F|$ matrix of the significance of each quad, while its element in the r_{th} row and c_{th} column is defined as:

$$W_{F,rc} = \left\{ \begin{array}{cc} \sqrt{s_r} & , if\, r = c \\ 0 & , else \end{array} \right\} \tag{4.3}$$

where s_r is the significance of hyperrectangle f_r. Furthermore, let C be the desired scale matrix C with:

$$C_{rc} = \left\{ \begin{array}{cc} c_r & , if\, r = c \\ 0 & , else \end{array} \right\} \tag{4.4}$$

The total hyperrectangle deformation energy is defined as follows:

$$\left\| W_F F^{'} - W_F C F \right\|^2 \tag{4.5}$$

or alternatively:

$$\left\| W_F Q V^{'} - W_F C Q V \right\|^2 \tag{4.6}$$

The minimization of the total hyperrectangle deformation energy allows for the hyperrectangles with large significance to have a smaller distance from their uniformly scaled version (which refers to the lack of distortion for all the objects), and thus, under the constraint of static space, significant hyperrectangles are enlarged more than the less significant ones.

An additional energy term that is used for the deformation procedure is the edge bending, as proposed in [14] and [13]. This term captures the case in which significant data patterns occupy multiple connected hyperrectangles, and tries to prevent their distortion [13] [14]. Given an edge e_k, its uniformly scaled version is defined as $e_k^{'} = l_k e_k$, where l_k is a 2×2 scale matrix. Let

$E^T = [e_1, e_2, .., e_{|E|}]$ be a matrix of all the edges, and W_E be a $|E| \times |E|$ matrix of the significance of each edge, while its element in the r_{th} row and c_{th} column is defined as:

$$W_{E,rc} = \left\{ \begin{array}{cc} \sqrt{s_r} & , if\ r = c \\ 0 & , else \end{array} \right\} \tag{4.7}$$

where s_r is the average significance factor for all the hyperrectangles in which edge e_r belongs. Furthermore, let L be the desired scale matrix L with:

$$L_{rc} = \left\{ \begin{array}{cc} l_r & , if\ r = c \\ 0 & , else \end{array} \right\} \tag{4.8}$$

The total edge bending energy is defined as follows:

$$\left\| W_E E^{'} - W_E L E \right\|^2 \tag{4.9}$$

or alternatively:

$$\left\| W_E H V^{'} - W_E L H V \right\|^2 \tag{4.10}$$

where H is a matrix such that $E = HV$. The edge bending energy term scales the edge lengths and tries to retain the edge orientations, after the multi-dimensional grid deformation.

Finally, the optimum deformed grid is defined as a solution to the following minimization problem:

$$\arg\min_{V'} \left\| W_F (QV^{'} - CQV) \right\|^2 + \left\| W_E (HV^{'} - LHV) \right\|^2 \tag{4.11}$$

$$subject\ to:$$

$$\left\{ \begin{array}{cc} v^{'}_{i,d} = \min[d] & , if\ v^{'}_{i,d}\ is\ on\ the \\ & lower\ boundary \\ & of\ dimention\ d \\ v^{'}_{i,d} = \max[d] & , if\ v^{'}_{i,d}\ is\ on\ the \\ & upper\ boundary \\ & of\ dimention\ d \\ \min[d] \leq v^{'}_{i,d} \leq \max[d] & , else \end{array} \right\}$$

where $v_{i,d}^{'}$ is the coordinate of vertex $v_i \in V$ in the d_{th} dimension, and $\min[d]$ and $\max[d]$ are the lower and upper boundaries of the d_{th} dimension as defined by the input dataset. After the minimization of the total deformation energy D defined in equation 4.11, the new grid $G^{'} = \left(V^{'}, E, F\right)$ is found, where $V^{'} = \bigcup_i v_i^{'}$. The dataset is deformed to this new grid $G^{'}$, taking into account the initial grid G and utilizing linear interpolation in the multi-dimensional space.

The scale factors of each hyperrectangle are also smoothed, as suggested in [14] and [13], in order to allow for neighboring hyperrectangles to have similar deformations, and thus, lead to perceptually consistent results [14] [13]. The smoothing procedure is defined as an optimization problem, by minimizing the following energy term:

$$\sum_{f_i \in F} \sum_{f_j \in N(f_i)} \frac{1}{2}(s_i - s_j)(c_i^{'} - c_j^{'})^2 + \sum_{f_i \in F} s_i (c_i^{'} - c_j)^2 \tag{4.12}$$

where $c_i^{'}$ is the new scale factor of hyperrectangle f_i, and $N(f_i)$ is the set of neighbors of hyperrectangle f_i. In the case of a regular grid, the set of neighbors of each hyperrectangle has the same size. On the contrary, in the case of multi-resolution grid, the set of neighbors of a hyperrectangle might have different size for different hyperrectangles. This however, is not an issue, since the set of neighbors is well defined in all cases.

The optimization problem presented in this section is a quadratic problem, solved using an iterative procedure [13]. It should be noted that the optimization procedure presented in this section takes as input any kind of grid, either regular or multi-resolution as defined in Section 4.2. In the case of the multi-resolution grid however, the experimental results reveal that the hyper-rectangles in the grid can be deformed into triangles. This might create an issue since the deformed grid, guides the scale-and-stretch of the final visualization results. In order to alleviate this issue, and also make sure that the quality of the visualization is high, the proposed approach utilizes a deformation threshold for each cell of the grid, in order to stop them from deforming too much.

An alternative method for non-linear deformation in one-dimensional datasets is presented in Appendix A. This method was firstly presented in [1] that was published in the context of this thesis. The non-liner deformation presented in Appendix A is a similar method for clutter reduction

to the hierarchical method presented in this section, but instead of using significance maps to guide the deformation procedure, it measures the exact information loss caused by the mapping procedure and tries to minimize it. It is possible to achieve similar results using both methods. In order to achieve these similar results, the distribution of the output signal is measured. The distribution of the data is measured using histograms, i.e. using a standard binning procedure and measuring how many data points fall into each bin. The goal is to make this distribution as uniform as possible, so as to have large entropy. The difference however is that the hierarchical approach presented in this section is more generic than the one-dimensional case presented in Appendix A, since it can be applied in any number of dimensions, taking as input any type of significance map. In addition, the proposed approach results in a quadratic optimization problem that can be efficiency solved. More details can be found in Appendix A.

4.4 Different choices for the generation of the significance map

For image resizing, multiple significance maps have been proposed including saliency maps [16] [76] and edge detection techniques [13]. These models take into account the way the human eye works in order to identify important regions in the image that contain visual objects, and thus, should be less distorted. These methods are efficient for 2-dimensional visualizations, but are not scalable to multiple dimensions. To overcome this issue, the use of an additional significance map is proposed in this section, which identifies highly cluttered regions, and can be applied to any number of dimensions. The corresponding significance map represents the entropy of the data instances (e.g. [56]). Specifically, the significance value of each hyperrectangle in the entropy significance map, is defined as the shannon's entropy [74] of all the data instances that belong to it. All the aforementioned significance maps are applied in multiple visualizations, and the results are compared with respect to multiple evaluation metrics.

Chapter 5

Experimental evaluation

Contents

5.1 Datasets 66
5.2 Experimental evaluation of the graph features . . 74
5.3 Experimental evaluation of the hierarchical magnification approach 95

This section presents the results of the experimental evaluation with respect to the graph analytic techniques presented in this thesis. Specifically, Section 5.1 presents the datasets utilized for the evaluation procedure. Section 5.2 presents the evaluation results of the proposed graph features in graph mining applications. Section 5.3 presents the evaluation results of the proposed hierarchical approach for clutter reduction in graph visualizations.

5.1 Datasets

5.1.1 Datasets used for the evaluation of the graph features

This section presents the datasets used for the evaluation of the graph features. All these datasets have been generated in the context of the research project NEMESYS [77] [78], which is founded by European Commission. The main goal of the NEMESYS project was the development of novel methods for the detection and mitigation of anomalies in mobile cellular networks [79] [80].

These datasets correspond to billing-related information describing cases of Distributed Denial of Service (DoS) attacks against the core network.

In general these types of attacks can be detected from either the signaling (control) or the billing (data) planes. The signaling plane in the mobile network is comprised of all the signals that control or are needed for the network services (e.g. Call Forwarding enable/disable or Call handover). Previous research [81] [82] has demonstrated the effect of several attack scenarios against the core network using the signaling plane. One such example is demonstrated by Traynor et al. [81], who proposed an attack that overloads the Home Location Register (HLR) with call Forwarding enable/disable signals. On the other hand, the billing plane is comprised of actual data sent/received by the mobile devices, including Call Detail Records (CDR)[1], and Internet traffic. For example, a large volume of SMSes (Short Message Service) can degrade the availability of the cellular phone network [84] [85] and deny voice service to cities in the size of Washington D.C. and Manhattan [85]. It should be underlined that this thesis targets the detection of anomalies using information from the billing plane.

This thesis focuses on the analysis of billing information. The reason for focusing only on billing information has to do with the creation of the communication graph. This graph represents the users' communication events (Calls/SMSes/Internet), which are captured from the billing information. On the other hand, the signaling data have as a recipient the mobile network, and in this respect the destinations of the signals are static. Although the signaling information could be used for the creation of the communication graph, the drawback is that the structure of this graph is static, since it does not depend on the communication patterns of the users. This means that the use of a graph for the analysis of signaling anomalies does not offer any additional advantages, when compared to simple approaches that take into account only the volume or frequency of the sent signals.

SMS flood

A simulation dataset is used to demonstrate the efficiency of the proposed anomaly detection approach in identifying unknown attackers, and to compare the results with other state-of-the-art anomaly detection methods.

GEDIS Studio [86] has been utilized for the creation of the dataset, which is an online tool for generating CDR data. The generation of the data is

[1]CDRs include attributes such as [83]: source, destination, time, duration, call type (e.g. voice, SMS, MMS etc), any fault condition encountered etc.

controlled by user-profile files, which describe patterns of usage for a specific user. A usage pattern defines the number of messages to be generated in the simulation period, the type of messages and the correspondent(s). A user-profile can contain several usage patterns, defining various types of calls originating from or terminating to a mobile device.

The CDR data correspond to the SMS flood scenario described in [84]. A number of 4800 users have been simulated. According to [84], the normal SMS traffic rate for one network sector is 0.7 messages/sec. For 4800 users, this corresponds to 12.6 messages/day/user. The users were divided into two usage groups, one with a large rate of 17.79 messages/day/user (2000 users) and one with a small rate of 8.89 messages/day/user (1800 users), with a total average of the original 12.6 messages/day/user. A period of 7 days has been simulated. A 10% ratio of the first group of users and a 5% ratio of the second group of users constitute the mobile devices, which have been infected by the SMS flooding malware during this period. In the last hour of the 7th day, the infected devices cause an SMS flood, by increasing their rate of SMS traffic by 8 times their normal rates, according to the setting of [84].

The above information and-user groups have been described with proper usage profile files in the GEDIS Studio tool. The generated CDR data for the SMS traffic consist of the following fields:

- *Origin*: The Identifier of the phone sending the SMS.
- *Destination*: The Identifier of the phone receiving the SMS.
- *Datetime*: Date and time that the SMS was sent/received.

In this case the origin and destination of the communication defined in Figure 3.2(a) are the mobile phones, while the communication events are SMSes.

Spam SMS

This scenario simulates an SMS spam campaign. For the simulation of this scenario, the OMNeT++ simulation framework [87] was used (similarly to the simulation presented in [88] and [89]). There are in total 10,000 mobile devices which are divided into three distinct groups according to behavior: (i) 4,000 users exhibit low levels of SMS activity, (ii) 3,000 users exhibit

medium levels of SMS activity, and (iii) 4,000 users exhibit high levels of SMS activity. In the simulations, 102 mobile devices, uniformly distributed across the three distinct groups, are assumed to be infected with spam malware. The malware does not have a diurnal cycle, and it regularly sends a configurable number of spam SMS messages, with each message being sent to a destination chosen randomly from the mobile's contact list. Normal mobile users have a very small probability of responding to a spam SMS. This dataset was simulated for a period of 7 days. The goal in this experiment is to identify the infected mobile devices.

The generated CDR data for the spam SMS dataset consist of the following fields:

- *Origin*: The Identifier of the phone sending the SMS.
- *Destination*: The Identifier of the phone receiving the SMS.
- *Datetime*: Date and time that the SMS was sent/received.

In this case the origin and destination of the communication defined in Figure 3.2(a) are the mobile phones, while the communication events are SMSes.

RRC-based attacks

This section presents the application of the graph features to the dataset for Radio Resource Control (RRC)-based attacks described in detail in [88], [89], and [90]. For the simulation of this scenario, the OMNeT++ simulation framework [87] was used. There are two classes of anomalous users in this dataset: (i) malicious mobile devices that send Internet packets whenever a demotion from the DCH (Dedicated Channel) state is assumed to have occurred, and (ii) misbehaving mobile devices that send periodic packets whenever the user is inactive. There are in total 200 users, of which 100 are anomalous, 50 in the first class and 50 in the second one. The goal of this experiment is to evaluate the performance of the proposed algorithm in identifying the anomalous users. There are in total 30 normal Internet servers that communicate with all the users, and 6 malicious servers that communicate only with the anomalous users. The total simulation period is 5 hours.

For each user, multiple attributes are collected concerning Internet traffic. Specifically, this dataset is comprised of the following billing-related data:

- *Origin*: The Identifier of the phone/server sending the Internet packet.
- *Destination*: The Identifier of the phone/server receiving the Internet packet.
- *Size*: The size in bytes of the Internet packet sent/received.
- *Datetime*: Date and time that the Internet packet was sent/received.

In this case the origin and destination of the communication defined in Figure 3.2(a) are the mobile phones and the Internet Servers, while the communication events are Internet packets.

Malware infection

This scenario is similar to the spam SMS scenario. The infected mobile devices send a large number of spam SMSes towards different destinations. The difference in this case is that the simulation included the malware propagation over time, and the infection of new mobile devices gradually increases. There are 2000 users in total, that communicate by means of SMS. The infected mobile devices send randomly a small number of SMSes that contain a link to a malicious website, which if the recipient of the SMS visits, is infected by the corresponding malware. This propagation procedure is simulated by utilizing an infection probability. Specifically, if an infected mobile device sends an SMS to a normal device, there is a small probability that it will become infected. The graph of connections between the users has diameter equal to 11. The simulation starts with one infected mobile device.

Figure 5.1 shows the number of infected mobile devices at each time instance of the simulation. Starting from one infected device at time period 0, the number of infected mobile devices increases gradually, and results in all the users being infected at time period 100.

In this case the origin and destination of the communication defined in Figure 3.2(a) are the mobile phones, while the communication events are SMSes.

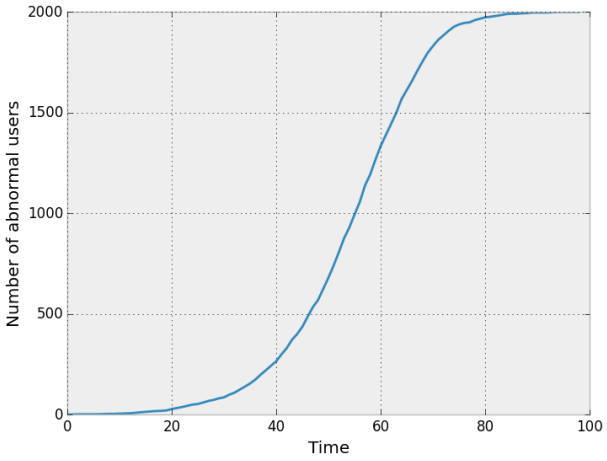


Figure 5.1: The number of infected mobile devices at each time instance of the simulation. The horizontal axis represents the index of the time instance of the simulation, and the vertical the number of abnormal users in each time instance. The simulation starts with 1 infected user, and ends with all the users being infected at time instance 100.

Similarly to the previous scenarios, the goal in this scenario is the identification of infected mobile devices at each time period. But unlike previous scenarios, a second goal is also set in this malware infection scenario. Specifically, the second goal is the identification of the mobile devices that will be infected in the near future (i.e. 1 to 10 time periods in the future).

5.1.2 Datasets used for the evaluation of the hierarchical clutter reduction approach

BGP routing changes

The goal of this dataset is to perform detention of Border Gateway Protocol (BGP)-related anomalies in the network, as well as identify the Autonomous Systems (AS) that are responsible for the anomalies.

BGP is the de facto protocol used in the Internet today for the exchange of routing information. The Internet consists of a large number of networks called AS. Within the Internet, an AS is a collection of connected Internet Protocol (IP) routing prefixes under the control of one or more network operators that present a common, clearly defined routing policy to the Internet.

A unique number (ASN) is allocated to each AS for use in BGP routing. AS numbers are important because the ASN uniquely identifies each network on the Internet.

The basic components of BGP are the announcements. A BGP announcement contains information about the destination prefix and the AS path followed to reach that prefix. An announcement has the form {<prefix> : <AS-path>}. For example the following announcement: {<91.194.12.0/21> : <AS-3549 AS-1 AS-2 AS-3>}, consists of the Prefix: <91.194.12.0/21> which describes 2048 IP addresses [2] and the AS path: <AS-3549 AS-1 AS-2 AS-3>. The last AS in the path (AS-3) is the one that produced the announcement and is the owner of these 2048 IP addresses. Furthermore, the path describes the ASes that one should traverse through to send data to AS-3. The BGP announcement data used here, were collected from the RIPE repository [91].

For the sake of demonstration, a graph based tool that was developed in [1] will be utilized, which from now on, will be called *BGPGraph*. *BGPGraph* was introduced for the purpose of visualizing BGP (Border Gateway Protocol) [92] routing changes. *BGPGraph* (presented in Appendix B) uses a graph metaphor in order to represent the topology of the Internet on the Autonomous System (AS) level. In other words, the vertices of this graph represent ASes and the edges physical connections between two ASes.

The basic methodology used by the *BGPGraph* is to define a time window and calculate the amount of traffic change that was caused by the incoming BGP announcements, measured in number of IP addresses. Afterwards, this information is mapped onto the widths of the edges and the radii of the vertices. This means that there are two mapping functions: $F : \mathbb{N}^1 \rightarrow V^1$, one for the edge widths and one for the vertex radii. The visualization space is represented with V, and corresponds to the set of distinct edge widths and vertex radii on the graph visualization, to which the input information is mapped. More details about this mapping procedure can be found on Appendix B, while equations (B.5) and (B.4) define the set V. It should be noted that the input space is in this case $\mathbb{N}^1$, since the input information represents number of IP addresses, which are integers. Finally, a hierarchical

[2] Each prefix, e.g. <91.194.12.0/21> consists of two numbers, namely, the IP address, i.e. 91.194.12.0, and the number of bits that are static beginning from the left of the IP address when written in binary format. For IP version 4, there are 32 bits that specify the IP address. Thus, the prefix <91.194.12.0/21> describes $2^{32-21} = 2^{11} = 2048$ IPs

clustering method is introduced in [1], that produces a hierarchy of coarse-to-fine graphs, so as to reduce the size of the visualized graphs.

This dataset is used in order to demonstrate the effects of the one-dimensional discriminators on a graph-based visualization of the input data.

Collaboration network

The goal of this dataset is to perform visualization resizing to fit smaller/ different-aspect-ratio screens, identical to the goal proposed by Wu et al. [14], with which the method proposed in this thesis is compared and evaluated.

This dataset is used by a graph-based visualization. It represents a collaboration network [93] from the e-print arXiv, and covers scientific collaborations between authors for all the papers submitted to General Relativity and Quantum Cosmology category. Each node represents an author, while undirected edges connect authors that have co-authored a paper. The data covers papers in the period from January 1993 to April 2003. The graph is comprised of 5242 nodes and 14496 edges.

Forest Fires dataset

The goal of this dataset is to perform visualization resizing to fit smaller/ different-aspect-ratio screens, identical to the goal proposed by Wu et al. [14], with which the method proposed in this thesis is compared and evaluated.

This dataset is used by a 3-dimensional scatterplot visualization. Although strictly speaking the scatterplot is not a graph, for a visualization perceptive it can be considered to be a graph without edges. The Forest Fires dataset [2], contains meteorological data from burned forests in the north-east region of Portugal. It contains multiple attributes such as: spatial coordinates of the burned forests, date of the fire, wind speed, relative humidity, temperature, Fine Fuel Moisture Code (FFMC), Duff Moisture Code (DMC), Drought Code (DC), and Initial Spread Index (ISI).

INSPIRE dataset

The goal of this dataset is to perform visualization resizing to fit smaller/ different-aspect-ratio screens, identical to the goal proposed by Wu et al. [14], with which the method proposed in this thesis is compared and evaluated.

This dataset is used by a scatterplot visualization. The INSPIRE [94] dataset reveals the topic distribution of IEEE Vis, IEEE InfoVis and, IEEE VAST proceeding papers published from 2006 to 2008. For the generation of this dataset, Wong et al. [94] collected the articles published in the corresponding IEEE conferences from 2006 to 2008. Afterwards, the authors examined the words that appear in each document in order to determine its thematic content, as well as its semantic similarity with the rest of the documents. The document data were afterwards projected onto a 2-dimensional space in order to visualize their relative thematic similarities.

5.2 Experimental evaluation of the graph features

This section presents the results of the application of the proposed graph-based features in multiple simulation datasets. Four scenarios are considered for the validation procedure, 1) SMS flood, 2) Spam SMS, and 3) RRC-based attacks, 4) Malware infection. These scenarios are common anomalies in the mobile network that were described in recent papers [84] [95] [96] [88], and are used as representative examples. This section compares the detection rates of the proposed approach with existing approaches for anomaly detection.

5.2.1 Anomaly detection methods selected for comparison

The methods selected for comparison and evaluation, belong to two groups: 1) Anomaly detection methods in mobile networks utilizing billing data, and 2) Anomaly detection methods using graph representations and local feature extraction. With respect to the first group, the following methods were selected: Lim et al. [84] and Yan et al. [95] which present methods for the detection of SMS flood events, and Xu et al. [96] which present a method for detection of SMS spammers. All these methods are supervised anomaly detection methods. On the other hand, with respect to the second group, the following methods were selected: Akoglu et al. [27] (Oddball), Henderson et al. [28] (Refex), and Kang et al. [29] (net-ray), which present generic methods for anomaly detection using graph-based data. The Oddball and net-ray are unsupervised anomaly detection methods, and the Refex is a supervised anomaly detection method. In addition to the default methods presented in these graph-based anomaly detection papers belonging to the second group, the extracted features were also utilized for classification based

on the supervised RF classifier, for fair comparison. Thus, two approaches are evaluated in each case, the default method presented in each paper, and a method based on an RF classifier.

Supervised classification has been shown to outperform unsupervised classification in cases of known types of classes, e.g. known attacks in networks [97] or identifying cell types [98]. However, the accuracy of supervised classification algorithms deteriorates significantly in the presence of unknown attacks [97]. In this research work, a specific indicative set of scenarios is presented and analyzed using a supervised classification approach, so as to evaluate the efficiency of the proposed features in detecting known attacks. The evaluation of the proposed features in unknown attacks is out of the scope of this thesis.

There are a number of methods that may be used for supervised classification. The RF clarifier, however, has been shown to outperform other supervised learning algorithms [99, 100] in a variety of datasets. In addition, this method does not suffer from over-fitting problems, is less sensitive to outlier data, and calculates automatically the importance of each feature in the classification task [101]. One of the most common methods for calculating the importance of each feature is the Gini index [102] [103], which is also the one utilized in this paper. Hence we use the RF algorithm along with the features that we have described in order to perform anomaly detection.

The Gini index [102] [103] of a data subset T_{sub} according to feature f_i is defined as follows:

$$Gini(f_i, T_{sub}) = 1 - \sum_{j=1}^{c} (p_j)^2 \tag{5.1}$$

where c is the number of different classes that the input data will be classified to, and p_j is the percentage of points belonging to the j_{th} class in the T_{sub}. The Gini index is minimum (with value zero) only when points from one class belong to T_{sub}. For the selection of the next feature f_k to be used for splitting a node of the decision tree, the following equation is utilized:

$$f_k = \arg \min_{f_k, T^q_{sub}} \left(\sum_{q=1}^{t} \left(\frac{|T^q_{sub}|}{|T|} Gini(f_k, T^q_{sub}) \right) \right) \tag{5.2}$$

where $|T|$ is the size of the dataset and $|T^q_{sub}|$ is the size of the q_{th} partition. The number of partitions on each iteration of the algorithm is t. In other

words, the partition feature f_k at each iteration of the algorithm is chosen (over all the possible f_k in the dataset) so that the sum of the Gini indexes over the different partitioning schemes T_{sub}^q (over the same feature f_k) is minimized, and thus, the classes are better separated according to f_k. In the case that the value of equation (5.2) is zero, then the separation is perfect and only points from one class belong to each T_{sub}^q.

An alternative metric to Gini Index that is commonly used for finding the best feature to partition the dataset at each iteration of the algorithm is the entropy [102]:

$$Entropy(f_i, T_{sub}) = \sum_{j=1}^{c} (p_j log\,(p_j)) \tag{5.3}$$

where c is the number of different classes that the input data will be classified to, and p_j is the percentage of points belonging to the j_{th} class in the T_{sub}. Similarly to the Gini index, the entropy is minimum (with value zero) only when points from one class belong to T_{sub}. The equation for finding the best partitioning feature according to the entropy is similar to equation (5.2).

The RF is a collection of multiple decision trees, trained on random samples extracted from the initial dataset. The final classification result is calculated using majority voting. Figure 5.2 shows an example of decision tree generation for the classification of two colored circles: red and green. The background color represents the classification result at the specific iteration, as calculated from the largest class belonging to the corresponding partition. The total space is iteratively partitioned into smaller regions along each feature, in order to create the space partitioning that best represents the data.

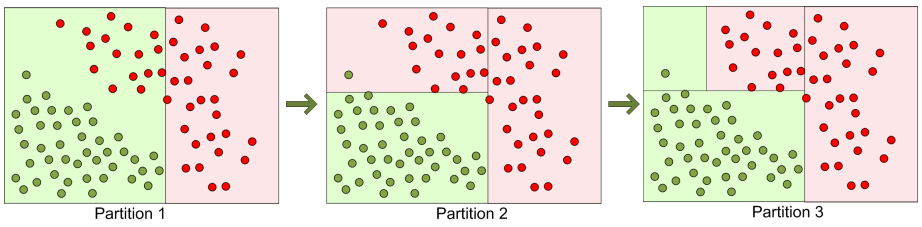


Figure 5.2: Examples of decision tree generation for the classification of two colored circles (red and green). The total space is iteratively partitioned into smaller regions along each feature, so as to create a partitioning that best represents the data.

5.2.2 Evaluation metrics

The result of all the anomaly detection methods is a binary classification. The evaluation of the efficiency of such results takes place by comparing them with the ground truth as follows::

- True Positive Rate (TPR) or recall: $\frac{TP}{P}$
- False Positive Rate (FPR): $\frac{FP}{N}$

where:

- True Positive (TP): number of anomalies correctly identified
- False Positive (FP): number of normal events incorrectly identified as anomalies
- Positive (P): number of anomalies
- Negative (N): number of normal events

5.2.3 Scenario 1: SMS flood

Since the important factor in this dataset is SMS activity, the graph is created to represent this behavior. Specifically, the vertices of the graph represent the mobile devices, and the edges represent SMS activity between two devices. The weight of an edge represents the number of messages exchanged between the corresponding devices. Thus, each neighborhood graph represents the SMS activity of each user with respect to the selected destinations and the number of messages directed towards them. The graph represents the activity of 1 user for 1 day.

For the training of the RF classifier, 10% of the dataset was utilized, uniformly distributed across the two distinct groups of behavior, while the rest of the users were used for testing the detection accuracy of the algorithm.

Table 5.1 presents the results of the application of the graph features on the SMS flood dataset by utilizing all the possible neighborhood sizes $k \in 0..N$ for feature extraction. The same table also presents the results of the anomaly detection methods proposed in [84] and [95], for identifying the mobile subscribers involved in the SMS flood attacks, and the methods proposed in [27], [28], and [29] for the detection of anomalies in graph-based

Table 5.1: Anomaly detection results on the SMS flood dataset. 10% of the dataset was utilized for training.

Method	**TPR**	**FPR**
Proposed features	**99.40%**	**0%**
OddBall [27]	31.58%	2.74%
OddBall [27] with RF	99.12%	0.07%
Refex [28]	97.66%	0.14%
Refex [28] with RF	97.66%	0.21%
Net-Ray [29]	40.06%	0.93 %
Net-Ray [29] with RF	99.12%	**0%**
Method in [84]	93.2%	1.4%
Method in [95]	96.5%	2.1%

representations of data. In the case of graph-based anomaly detection methods two cases were examined: 1) the standard method presented in the respective papers, and 2) the features presented in the respective papers, but by utilizing an RF classifier. In all of the cases except for the Refex [28], the RF classifier provided better results than the default method proposed in each paper. The reason for this result, is that the OddBall [27] and Net-Ray [29] methods are unsupervised methods, which are at a disadvantage on a supervised classification task, as previously mentioned in Section 5.2.1. On the other hand, the Refex method utilizes a log-Forest supervised classifier, which has similar performance to the RF classifier. The methods proposed in [84] and [95] have good performance, since they were specifically developed for the detection of SMS flood attacks in the mobile network. Their performance, however, is lower when compared to the used of graph-related features followed by supervised classification. This means that the use of the graph structure for the generation of the features conveys more information than the simple use of the history of each user. Finally, Table 5.1 shows that the best results, with the highest TPR and the lowest FPR, are obtained by utilizing the features proposed in this thesis.

Figure 5.3 shows the evaluation of the graph features by taking different

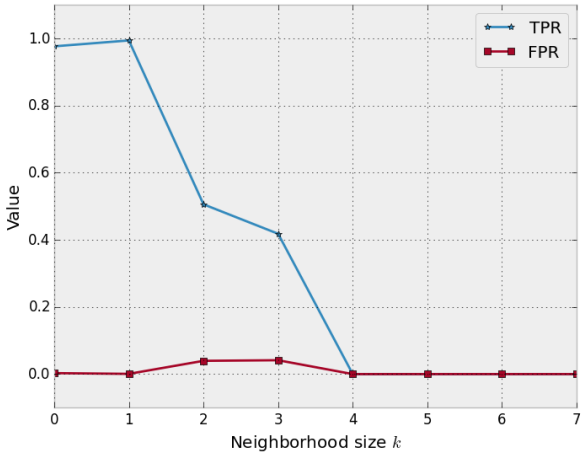


Figure 5.3: Evaluation of the graph features on the SMS flood dataset, by taking into account only the features extracted from different values of neighborhood sizes k, and not all together (as done in the default method).

values of neighborhood sizes separately, and not all together (as done in the default method). The anomalous users are best found by taking into account only their SMS activity ($k = 0$), and the SMS activity of their neighbors ($k = 1$). Further increase in the value of k deteriorates the classification efficiency. The reason for this is that a neighbor of an anomalous user might or might not be anomalous himself, and thus, the addition of extra neighbors adds noise. The graph diameter in this case is 7.

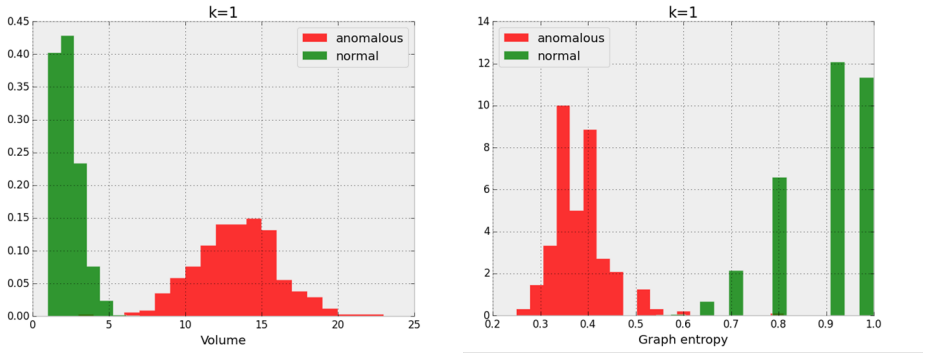
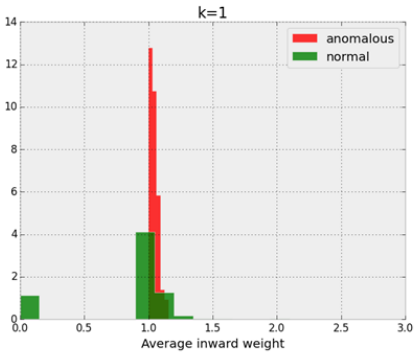


Figure 5.4: The histograms of the two features with the highest importance according to the Gini index for the SMS flood dataset. The corresponding features are the volume (k=1) and the Graph entropy (k=1).

Figure 5.4 shows the histograms of the best features, which have the highest importance according to the Gini index. The best features are the volume and the graph entropy for $k = 1$, since the large number of SMSes sent by abnormal users on different destinations has a direct effect on these two features. The reason for this is that the specific type of increased SMS behavior, changes the topology and the weight of the egonet, i.e. 1) increases the weights of the edges and 2) adds extra edges in star-shape around the source node that lower the graph entropy.

Figure 5.5(a) illustrates the histogram of a feature with average highest importance according to the Gini index, namely the Average inward weight (k=1). Due to the overlapping of the classes, this feature can not discriminate between normal and abnormal users very efficiently. Figure 5.5(b) shows the ordering of the feature importances based on the Gini index, and the TPR and FPR of each single feature when used alone for anomaly detection (i.e. not used in combination with any other feature). The first two features are able to capture the larger amount of anomalous related patterns with high accuracy, while the accuracy drops significantly for the rest of the features. The Average inward weight (k=1) feature shown in Figure 5.5(a) has TPR 15.21% and FPR 0%.



(a)

Order	Feature	TPR	FPR
1	Graph Entropy (k=1)	99.12%	0.07%
2	Volume (k=1)	98.24%	0.07%
3	Edge Entropy (k=1)	46.19%	0.00%
4	Average Inward Edge Weight (k=0)	30.12%	0.00%
5	Average Inward Edge Weight (k=1)	15.21%	0.00%

(b)

Figure 5.5: Evaluation of the graph features on the SMS flood dataset. (a) The histogram of a feature with average gini value, namely the Average inward weight (k=1). (b) The ordering of the feature importances based on the Gini index, and the TPR/FPR of each single feature when used alone for anomaly detection (i.e. not used in combination with any other feature).

5.2.4 Scenario 2: Spam SMS

Similarly to the SMS flood case, the significant factor in the spam SMS dataset is the exchange of SMSes. Thus, the graph is created to represent this activity. Specifically, each vertex of the graph represents a mobile subscriber, and each edge the direction of an SMS between two of them. The weight of each edge represents the number of SMSes sent from the source to the destination vertex. The graph represents the activity of each user for all 7 days.

Table 5.2: Anomaly detection results on the spam SMS dataset. 10% of the dataset was utilized for training.

Method	**TPR**	**FPR**
Proposed features	**98.05%**	0.01%
OddBall [27]	33.01%	0.11%
OddBall [27] with RF	87.37%	0.01%
Refex [28]	8.73%	**0%**
Refex [28] with RF	7.76%	0.03%
Net-Ray [29]	33.01%	8.86 %
Net-Ray [29] with RF	38.83%	0.07%
Method in [96] (PCA)	87.2%	0.03%
Method in [96] (all features)	79.4%	0.10%

For the training of the RF classifier, 10% of the dataset was used, uniformly distributed across the three distinct groups of spam behavior, while the rest of the users were used for testing the detection accuracy of the algorithms.

Table 5.2 presents the results of the application of the proposed features (utilizing all the possible neighborhood sizes for feature extraction) on the task of detecting SMS spammers in a mobile network. The same table also presents the results of the graph-based methods [27] [28] [29] on the same task, as well as the results of the method proposed by Xu et al. [96], which is a method specifically designed for the detection of spammers in mobile

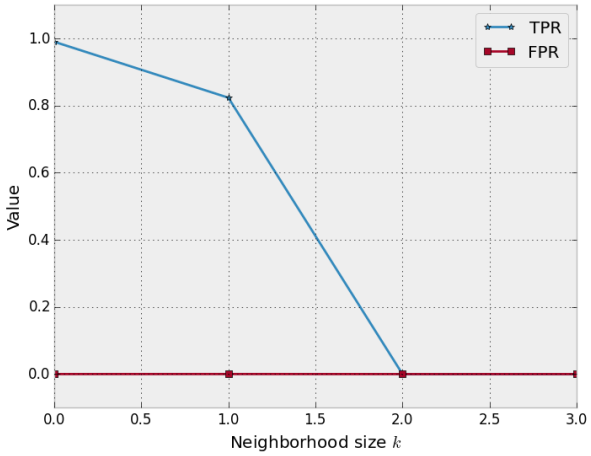


Figure 5.6: Evaluation of the graph features on the Spam SMS dataset, by taking into account only the features extracted from different values of neighborhood sizes k, and not all together (as done in the default method).

networks. Similarly to the previous experiments, the RF classifier outperforms the default method proposed in OddBall [27] and Net-Ray [29], since these two methods are unsupervised. However, amongst the graph-based anomaly detection methods, the OddBall with RF provides the best results. The reason for this is that in the spam SMS dataset, the anomalous users have increased activity towards the mobile phones in their contact list. This means that specific features are needed in order to capture this behavior, which does not include structural changes in the graph, but just changes in the weights of the edges. The features are not provided by either Net-Ray or Refex. Specifically, the Refex method utilizes a recursive method for feature generation, which also prunes correlated features in each step. The pruning method is a fast binning based method, which might not always be accurate. This means that the required features for detection were either not generated at all, or pruned by mistake in a step of the algorithm. This is evident in both the default Refex method, which utilizes supervised log-forest classifiers, and the use of RF. On the other hand, the OddBall method has a specific feature, namely, the sum of the edge weights in the egonet, that is capable of capturing the aforementioned spam SMS behavior. Finally, the method proposed by Xu et al. [96] is able to outperform most of the graph-based approaches (expect for the OddBall with RF), since it was specifically

designed for the detection of spammers. Xu et al. proposed a number of features that were used for supervised classification with a Support Vector Machine (SVM). Principal Component Analysis (PCA) is also utilized for feature reduction, so as to increase the accuracy of the algorithm. Finally, the best results are obtained by utilizing the graph features proposed in this thesis with the RF classifier, with the highest TPR and very low FPR.

Figure 5.6 shows the evaluation results of the graph features by taking into account the features extracted from different values of neighborhood sizes k separately. In this case, the graph diameter is 3, and thus this is the largest k value beyond which the TPR and FPR become zero. The best results are obtained with the proposed graph features, using neighborhood size $k = 0$. The reason for this is that since a spam message concerns two users, a normal user whose graph geodesic distance to an anomalous user is 2, is not necessarily anomalous, and thus, the addition of extra neighbors adds noise.

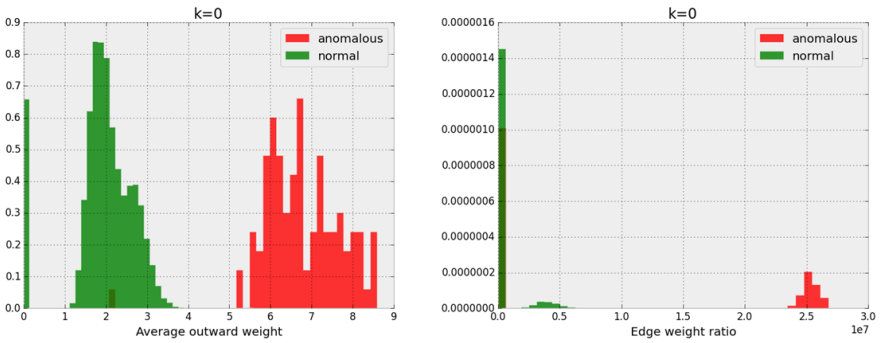
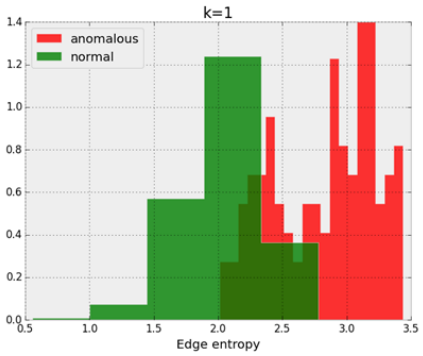


Figure 5.7: The histograms of the two features with the highest importance according to the Gini index for the Spam SMS dataset. The corresponding features are the Average outward weight (k=0) and the Edge weight ratio(k=0).

Figure 5.7 shows the histograms of the two features with the highest importance according to the Gini index. The best features are the average outward weight and the edge weight ratio for $k = 0$, since the spammers send more SMSes than they receive, and thus, this behavior directly affects these two features. Large outward SMSes increase the value of the average outward weight feature, and also make the edge weight ratio (i.e. the number of SMSes sent with respect to the number of SMSes received) larger for a



(a)

Order	Feature	TPR	FPR
1	Average Outward Edge Weight (k=0)	98.03%	0.03%
2	Edge Weight Ratio (k=0)	94.12%	0.03%
3	Edge Entropy (k=1)	63.72%	0.23%
4	Edge Weight Ratio (k=1)	9.80%	0.89%
5	Graph Entropy (k=1)	0.00%	0.00%

(b)

Figure 5.8: Evaluation of the graph features on the Spam SMS dataset. (a) The histogram of a feature with average importance according to the Gini index, namely the Edge entropy (k=1). (b) The ordering of the feature importances based on the Gini index, and the TPR/FPR of each single feature when used alone for anomaly detection (i.e. not used in combination with any other feature).

large number of spammers. As shown in this figure, and also in Figure 5.6, the best features are obtained from the graph neighborhood with $k = 0$, since it captures the direct communications of the spammers.

Figure 5.8(a) illustrates the histogram of a feature with average importance according to the Gini index, namely the Edge entropy (k=1). As shown by the histogram, half of both the normal and anomalous classes overlap, and thus, the use of only this feature for anomaly detection would not provide good results. Figure 5.5 (b) shows the ordering of the feature importances based on the Gini index, as well as the TPR and FPR of each single feature when used for anomaly detection. The first two features have very high accuracy, while the rest have significantly lower accuracy. The Edge entropy (k=1) feature shown in Figure 5.8(a) has TPR 63.72% and FPR 0.23%.

5.2.5 Scenario 3: RRC-based attacks

For the creation of the graph, the Internet traffic is utilized. Specifically, the vertices of the graph represent either mobile devices in the cellular network or Internet servers. The edges represent Internet communications between

cellular mobile devices and Internet servers, while their weight represents the size of the packets exchanged in the communication events. The graph was created for the entire period of 15 days.

Table 5.3: Anomaly detection results on the RRC-based attacks dataset. 10% of the dataset was utilized for training.

Method	**TPR**	**FPR**
Proposed Features	**99.00%**	**0.74%**
OddBall [27]	0%	16.29%
OddBall [27] with RF	93.06%	4.44%
Refex [28]	96.04%	16.29%
Refex [28] with RF	98.02%	5.18%
Net-Ray [29]	0.99%	20.74%
Net-Ray [29] with RF	95.05%	**0.74%**

Table 5.3 shows the results of the application of the different anomaly detection methods on the task of identifying infected mobile devices which are involved in an RRC-based attack. Similarly to the results of the previous experiments, the presence of ground truth makes this problem more suitable for use with supervised classification. This is also shown in the other experimental results where the supervised methods outperform the unsupervised ones. This dataset, in contrast to the rest of the datasets in this thesis is particularly challenging for unsupervised classification methods, since both Oddball and Net-Ray are not able to detect the anomalous users, while they also give a large number of false positives. The best results are obtained by utilizing the features proposed in this thesis.

Figure 5.9 illustrates the evaluation of the graph features by taking into account only the features extracted from specific values of neighborhood sizes k. In this case, neighborhood sizes $k = 0$ and $k = 1$ provide the best results, while the graph diameter is 2.

Figure 5.10 shows the histograms of the two features with the highest Gini index values. The best features are the average outward weight for $k = 0$ and the average inward weight for $k = 1$. The average outward weight

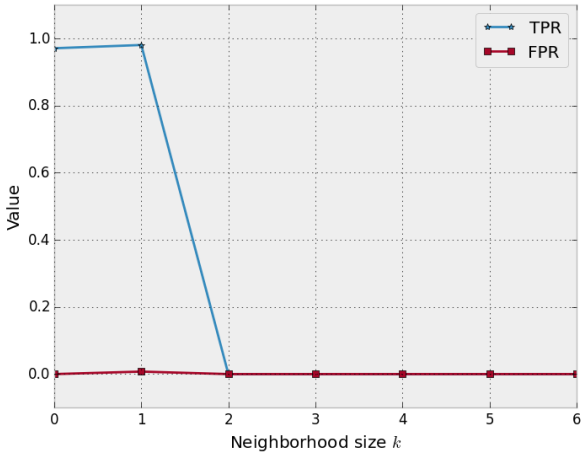


Figure 5.9: Evaluation of the graph features on the RRC-based attacks dataset, by taking into account only the features extracted from different values of neighborhood sizes k, and not all together (as done in the default method).

feature for $k = 0$ captures the fact that the abnormal users communicate with a specific set of anomalous Internet servers and exchange a large volume of traffic. The average inward weight feature for $k = 1$ includes the abnormal servers in the neighborhood of the abnormal users. Thus, this feature captures the fact that the a number of abnormal users sent large traffic volumes towards these servers, i.e. these neighborhoods have a large average inward weight.

Figure 5.11(a) illustrates the histogram of the feature with the fourth highest importance according to the Gini index, namely, the Edge weight ratio (k=1). In this case, the separation between normal and abnormal users using only the corresponding feature is more difficult than the two best features shown in Figure 5.10, since there is some overlapping of both classes in the histogram. However, as shown in Figure 5.5(b), the TPR of this feature is 97.32% and the FPR is 0.86%, which means that it is able to identify a large number of anomalous users with very low false positives. This feature captures the increased traffic activity towards the k=1 neighborhood, since in the case of anomalous users, this neighborhood contains a malicious server that is the communication destination of multiple other malicious users. Figure 5.5(b) shows the ordering of the features according to their

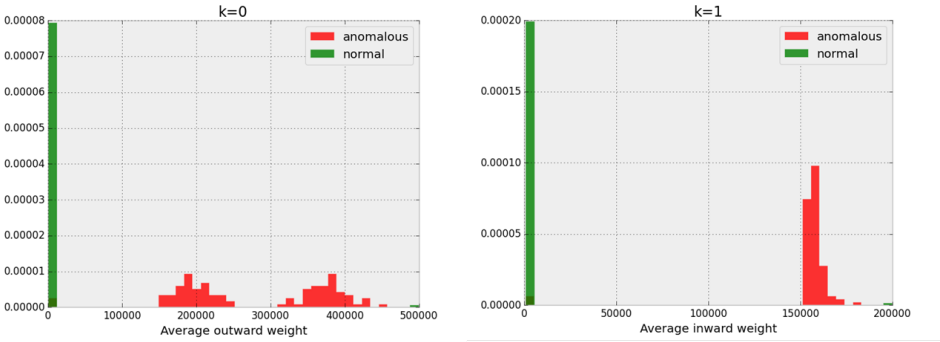
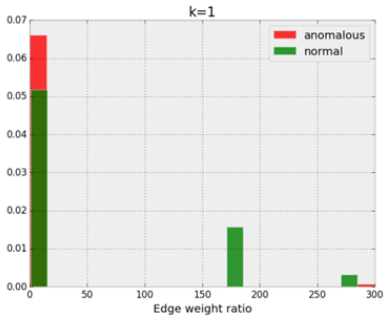


Figure 5.10: The histograms of the two features with the highest importance according to the Gini index for the RRC-based attacks dataset. The corresponding features are the Average outward weight (k=0) and the Average inward weight (k=1).

importance based on the Gini index. The same subfigure also shows the results of the application of each individual feature for the task of anomaly detection. The first four features in this case have all high performance, i.e. above 97% TPR and below 1% FPR.



(a)

Order	Feature	TPR	FPR
1	Average Outward Edge Weight (k=0)	99.00%	0.73%
2	Average Inward Edge Weight (k=1)	98.63%	0.73%
3	Edge Weight Ratio (k=0)	98.63%	0.86%
4	Edge Weight Ratio (k=1)	97.32%	0.86%
5	Volume (k=1)	93.33%	5.14%

(b)

Figure 5.11: Evaluation of the graph features on the RRC-based attacks dataset. (a) The histogram of a feature with average importance according to the Gini index, namely the Edge weight ratio (k=1). (b) The ordering of the feature importances based on the Gini index, and the TPR/FPR of each single feature when used alone for anomaly detection (i.e. not used in combination with any other feature).

5.2.6 Scenario 4: Malware infection

For the creation of the graph, each mobile device is represented by a node, and each SMS communication between two nodes is represented by an edge. The direction of the edge shows the direction of communication, and the weight of the edge the number of SMSes sent through it. One graph is created for each time period.

Table 5.4: Anomaly detection results on the malware infection dataset, for the goal of detecting the infected mobile devices at each time instance. 10% of the dataset was utilized for training.

Method	**TPR**	**FPR**
Proposed features	**99.82%**	**0.01%**
OddBall [27]	48.63%	1.17%
OddBall [27] with RF	99.67%	0.10%
Refex [28]	97.21%	0.86%
Refex [28] with RF	98.76%	0.61%
Net-Ray [29]	4.12%	4.77%
Net-Ray [29] with RF	69.15%	14.08%

Table 5.4 shows the results of the different graph-based anomaly detection methods for the task of detecting the mobile devices that are infected in the current time period, given the current state of the network. As shown in all the other cases, the supervised methods outperform the unsupervised. The worst results regarding the supervised methods are obtained from the Net-Ray method, since all the graph-based features extracted from this method take into account only the structure of the graph, and not the weight of the edges. In this dataset, the spam SMS behavior of each infected mobile device is similar to the one presented in Section 5.2.4, where the infected devices sent spam SMS to the numbers found in the contact list of the phone. In contrast, however, to Section 5.2.4, the Refex method was able, using the recursive feature generation and pruning steps, to efficiently generate features that capture this type of behavior, and thus, has increased accuracy. As shown in this table, the best results are obtained by

utilizing the graph-based features proposed in this thesis, which were able to efficiently discriminate between normal and abnormal behavior.

Table 5.5: Anomaly detection results on the malware infection dataset, for the goal of detecting the infected mobile devices at the next 2 time periods, given the current state of the network (i.e. using the features of the current time period). 10% of the dataset was utilized for training.

Method	**TPR**	**FPR**
Proposed features	**89.69%**	**0.23%**
OddBall [27]	38.21%	1.22%
OddBall [27] with RF	88.19%	2.61%
Refex [28]	84.42%	1.41%
Refex [28] with RF	89.40%	1.71%
Net-Ray [29]	3.57%	4.82%
Net-Ray [29] with RF	65.26%	18.24%

Table 5.5 shows the results of the different anomaly detection methods for the task of detecting the mobile devices that will be infected in the next 2 time periods given the current state of the network, i.e. is a prediction task. The results are similar to the results presented in Table 5.4, but the random nature of the infection in the future reduces the accuracy of the algorithms. Similarly to Table 5.4, this table also shows that the best results in this prediction task are provided by the graph-based features proposed in this thesis.

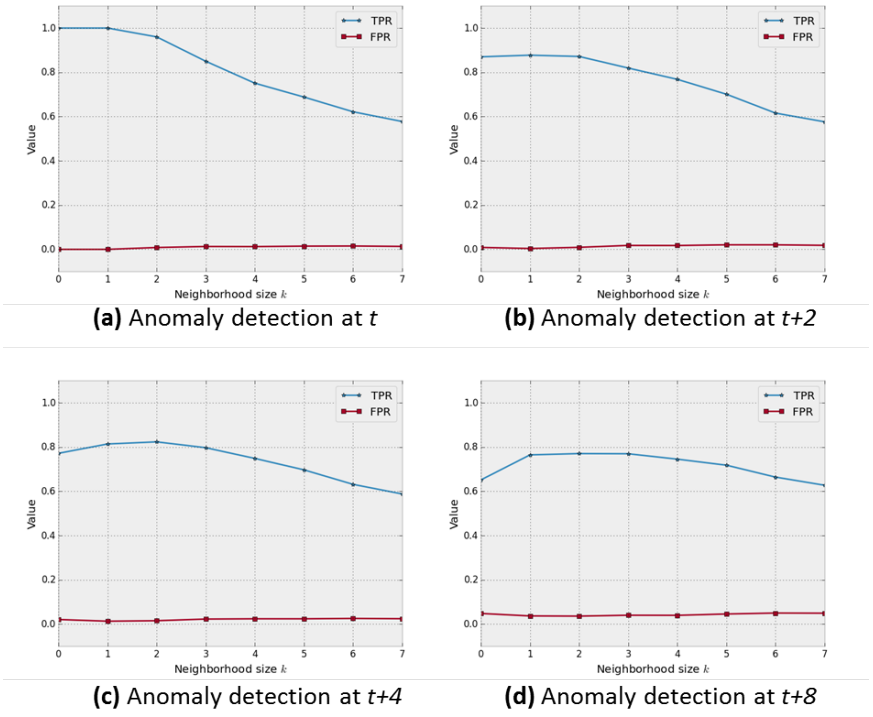


Figure 5.12: Evaluation of the graph features on the malware infection dataset, by taking into account only the features extracted from different values of neighborhood sizes k, and not all together (as done in the default method).

Figure 5.12 illustrates the evaluation of the graph features by taking into account only the features extracted from specific values of neighborhood sizes k. Four different cases are shown in this figure. Anomaly detection at each current time instance t, and anomaly detection at the next 2, 4, and 8 time instances $t+2$, $t+4$, and $t+8$. In the first case, the best neighborhood is $k = 0$, while predicting further in the future cases the best neighborhood size k raise to $k = 1$ for $t + 2$, and $k = 2$ for $t + 4$ and $t + 8$.

Figure 5.13 shows the two features with the highest Gini index for the malware infection dataset. Two cases are shown, one for anomaly detection at the current time instance t and one for anomaly detection at the next 2 time instances $t + 2$. In both cases, the same features provide the best separation between normal and abnormal mobile devices. The only difference is the neighborhood size for which they were extracted. Specifically, for the

time period t the features are extracted from neighborhood sizes $k = 0$ and $k = 1$, while for the time period $t + 2$ the best features are extracted from neighborhood sizes $k = 1$ and $k = 2$.

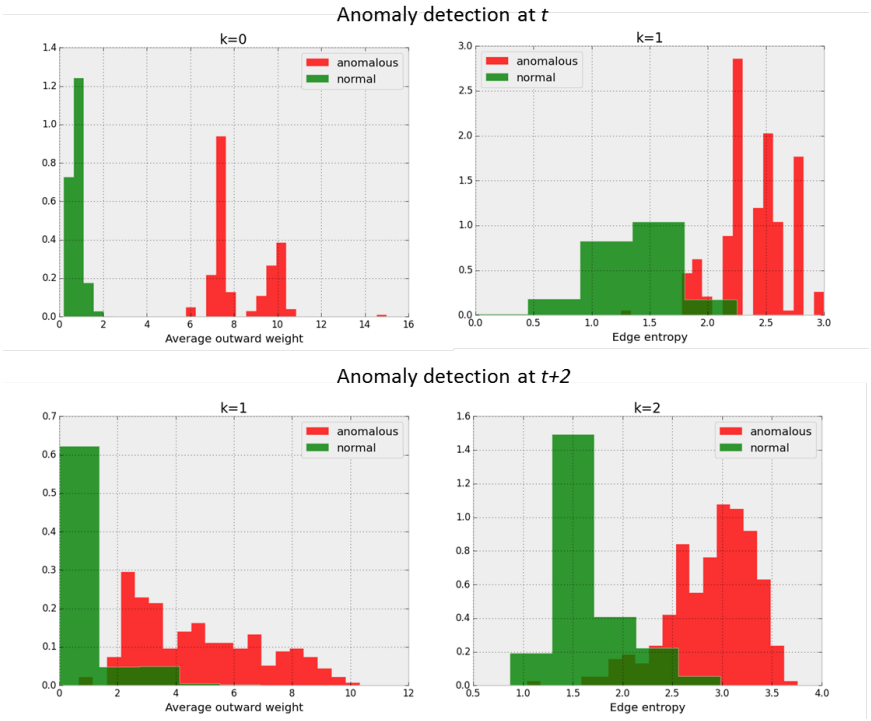
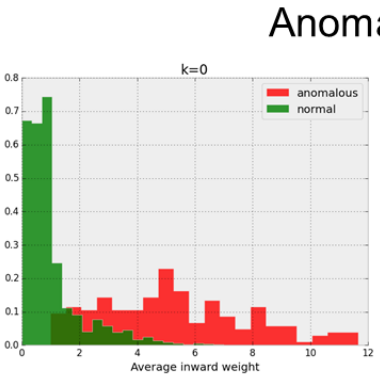


Figure 5.13: The histograms of the two features with the highest importance according to the Gini index for the malware infection dataset for the task of identifying anomalous devices in the current and the next two time periods, given the current state of the network. The corresponding features for the current time period are the Average outward weight (k=0) and the Edge entropy (k=1), while the corresponding features for the next two time periods are the Average outward weight (k=1) and the Edge entropy (k=2).

Figures 5.14(a) and 5.15(a) illustrate the histogram of a feature with average importance according to the Gini index. In both cases, the separation between normal and abnormal users is difficult using only the corresponding feature for anomaly detection, since there is a large overlapping between the classes of normal and anomalous users. Specifically, Figure 5.14(a) shows the Average inward edge weight (k=0) feature for the task of detecting anomalous users at the current time period, given the current state of the network. Since the anomalous users have increased outwards activity, this feature can-

not capture this pattern. In addition, Figure 5.15(a) shows the Edge weight ratio (k=1) feature for the task of predicting the anomaly status of the users at the next two time periods, given the current state of the network. The Edge weight ratio (k=1) would have large values in the case that the anomalous users are only the k=1 neighborhood (i.e. large probability of infection), but it would have values around 1 in the case the anomalous users are found in both the k=1 and k=2 neighborhoods, i.e. large inwards and large outwards activity, despite the fact that the probability of infection is still high. For this reason the Edge weight ratio (k=1) feature has low accuracy.



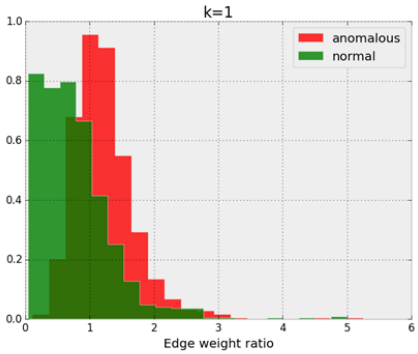
Order	Feature	TPR	FPR
1	Average Outward Edge Weight (k=0)	99.22%	0.02%
2	Edge Entropy (k=1)	87.50%	0.51%
3	Edge Entropy (k=2)	76.78%	3.37%
4	Average Outward Edge Weight (k=1)	74.55%	3.49%
5	Average Inward Edge Weight (k=0)	68.75%	3.49%

Figure 5.14: Evaluation of the graph features on the malware infection dataset. (a) The histogram of a feature with average importance according to the Gini index, namely the Average inward weight (k=0). (b) The ordering of the feature importances based on the Gini index, and the TPR/FPR of each single feature when used alone for anomaly detection (i.e. not used in combination with any other feature).

Figures 5.14(b) and 5.15(b) show the ordering of the feature importances based on the Gini index. The same figures also show the TPR and FPR that result from using only one feature for anomaly detection, in an attempt to quantify the feature importances. Figure 5.14(b) corresponds to anomaly detection at the current time period, given the current state of the network, while Figure 5.15(b) corresponds to anomaly detection/prediction at the next two time periods, given the current state of the network. Figure 5.14(b) shows that most of the anomalous patterns are captured only by the first

feature, while the accuracy drops for the rest of the features. The first feature captures the pattern that the anomalous users sent a large number of spam SMSes. Figure 5.15(b) shows that most of the anomalous-related patterns are captured by the first two features, with TPR over 80% and FPR below 4%. The accuracy drops significantly for the rest of the features.

Anomaly detection at t+2



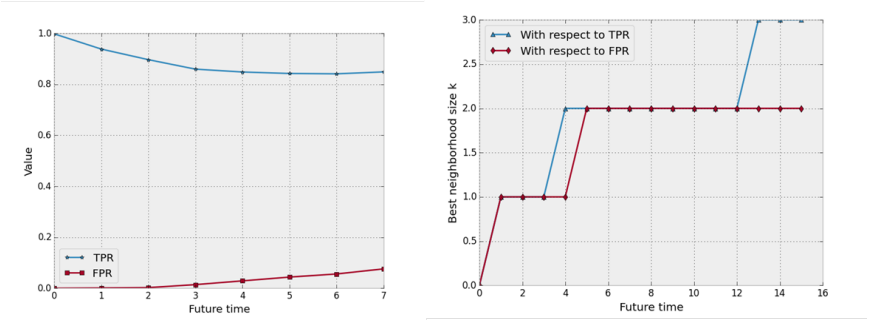
(a)

Order	Feature	TPR	FPR
1	Average Outward Edge Weight (k=1)	82.41%	0.63%
2	Edge Entropy (k=2)	81.31%	3.99%
3	Edge Entropy (k=1)	67.76%	0.23%
4	Edge Weight Ratio (k=1)	31.86%	6.43%
5	Edge Entropy (k=3)	22.77%	6.66%

(b)

Figure 5.15: Evaluation of the graph features on the malware infection dataset. (a) The histogram of a feature with average importance according to the Gini index, namely the Edge weight ratio (k=1). (b) The ordering of the feature importances based on the Gini index, and the TPR/FPR of each single feature when used alone for anomaly detection (i.e. not used in combination with any other feature).

Figure 5.16 shows the evaluation results of the graph features for anomaly detection in the current and future time periods. Specifically, Figure 5.16 (a) shows the TPR and FPR scores with respect to the future time periods that were used for anomaly detection. For the current time period (i.e. $t = 0$) that does not include future predictions, the TPR score is close to 100% and the FPR score close to 0%. For future predictions, the TPR score decreases and the FPR score increases. The further away the prediction is in the future, the worst the scores. The reason for these results is the probabilistic malware infection behavior that makes the accurate prediction difficult. Figure 5.16 (b) shows the neighborhood sizes k that have either the highest TPR or the lowest FPR with respect to the future time periods that

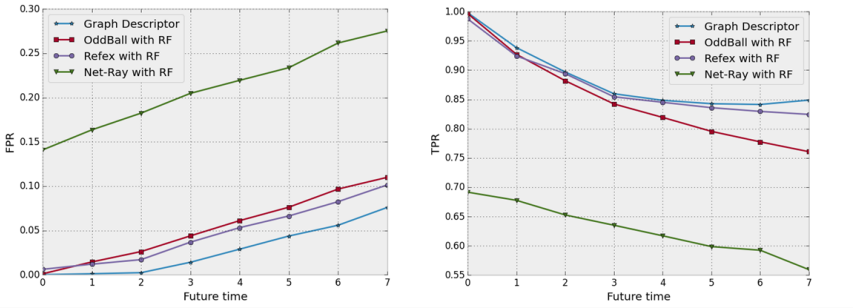


(a) Evaluation of future predictions

(b) Best neighborhood sizes for future predictions

Figure 5.16: Evaluation of the graph features for anomaly detection in the current and future time periods. (a) The TPR and FPR scores with respect to the number of future time periods that were used for anomaly detection. (b) The best neighborhood size k for either the TPR of the FPR scores with respect to the number of future time periods that were used for anomaly detection.

were used for anomaly detection. For the current time period (i.e. $t = 0$), the best neighborhood size is $k = 0$. Predictions further into the future cause the best neighborhood sizes to increase to $k = 1$, $k = 2$, and $k = 3$.



(a) Evaluation of future predictions with respect to FPR

(b) Evaluation of future predictions with respect to TPR

Figure 5.17: Evaluation of all the anomaly detection in the current and future time periods with respect to the TPR scores (a) and FPR scores (b).

Figure 5.17 shows the evaluation of all the anomaly detection methods (utilizing the RF classifier) in the current and future time periods with re-

spect to the TPR scores (Figure 5.17(a)) and FPR scores (Figure 5.17(b)). In all the cases, the best method is to use the proposed graph features, and the second best method is the Refex [28] with RF.

5.3 Experimental evaluation of the hierarchical magnification approach

5.3.1 Evaluation Metrics

This section presents the evaluation metrics used for evaluating the efficiency of the proposed hierarchical resizing/magnification approach, and comparing it with other state-of-the-art methods. It should be highly underlined that due to the fact that there are no quantitative metrics in the literature for evaluating the visualization resizing/magnification results, these metrics have been proposed in the context of this thesis, as an attempt to more formally and quantitatively evaluate the proposed approach.

Pattern Preservation

The *Pattern Preservation* (PP) evaluation metric captures the degree in which the local regions in the magnified space preserve the patterns that existed in the input space. This metric is based on graph matching on the kNN (k-Nearest Neighbors) graphs between the two spaces, the input and magnified (example of this metric is presented in [104]). Graph representations of the input data, and subsequent graph matching, have been used in the literature for the comparison of 3D objects and point clouds (e.g. [105] [106] [107]), as well as for speaker identity recognition (e.g. [104]). A large change in the number of neighbors for each data instance represents a large change of the initial pattern due to the magnification procedure. The graph matching distance is found using the graph edit distance [108] between the labeled kNN graphs of the input and magnified spaces. More formally, the Pattern Preservation metric is defined as follows:

$$PP = GED\left(G_{kNN}(P,k), G_{kNN}(P',k)\right) \tag{5.4}$$

where PP is the pattern preservation metric, GED(.) is a function that takes as input two graphs, and returns their Graph Edit Distance (GED) [108], and $G_{kNN}(P,k)$ is the kNN graph created taking as input a set of points

$P = \{p_1, p_2, .., p_n\}$ for $p_i \in \mathbb{R}^2$ or $p_i \in \mathbb{R}^3$, and the value of k. The set of points P is the set of points of the visualization on the screen before the application of the magnification procedure (i.e. input visualization), while P' is the set of points of the visualization on the screen after the application of the magnification procedure (i.e. output/magnified visualization). Thus, the PP metric captures how the local neighborhood of each point has changed due to the application of the non-linear magnification approach.

Size Perception

The *Size Perception* (SP) evaluation metric is based on the visual perception of the different object sizes. This metric captures the degree in which the new object sizes in the magnified space, are perceived as the correct sizes, which is important in the analytical task. According to Stevens Power Law [109] the perception of the intensity of a sensory stimuli is a power function of the actual intensity of the input stimuli as follows:

$$\psi(I) = kI^a \tag{5.5}$$

where I is the intensity of the input stimuli, $\psi(I)$ is the subjective magnitude of the sensation, a is an exponent that depends on the sensory type of the stimulation, and k a constant. In case of size perception, according to [110], $k = 0.98$ and $a = 0.87$. The proposed evaluation metric measures the absolute difference of the actual input size, from the size perceived by the visualization:

$$SP = \sum_i |\psi(I_i) - L_i| \tag{5.6}$$

where $\psi(I_i)$ is the perceived size of the i_{th} visual object, I_i is the size of the visual object after the application of the magnification procedure, and L_i is the initial size of the i_{th} visual object as specified from the input data, and before the application of the magnification approach. Thus, this metric measures how different the perceived final size of the magnified visualization is, when compared to the one that was originally supposed to be perceived. Smaller differences correspond to better results.

5.3.2 BGPGraph Visualization magnification

This section presents the application of the proposed approach on the BGP-Graph visualization approach for clutter reduction. BGPGraph was devel-

oped in the context of this thesis, and aims at the analysis of BGP routing changes and anomalies. BGPGraph is presented in Appendix B.

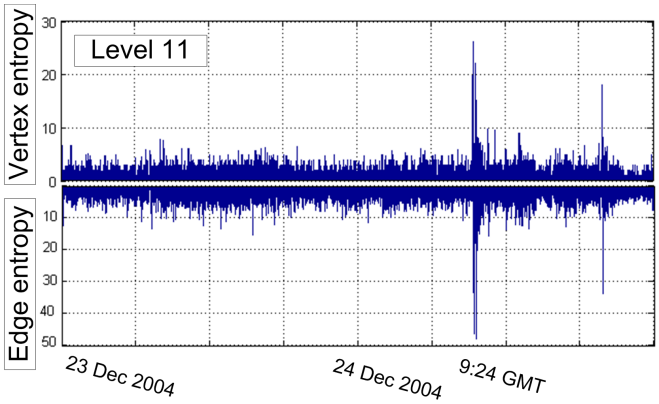
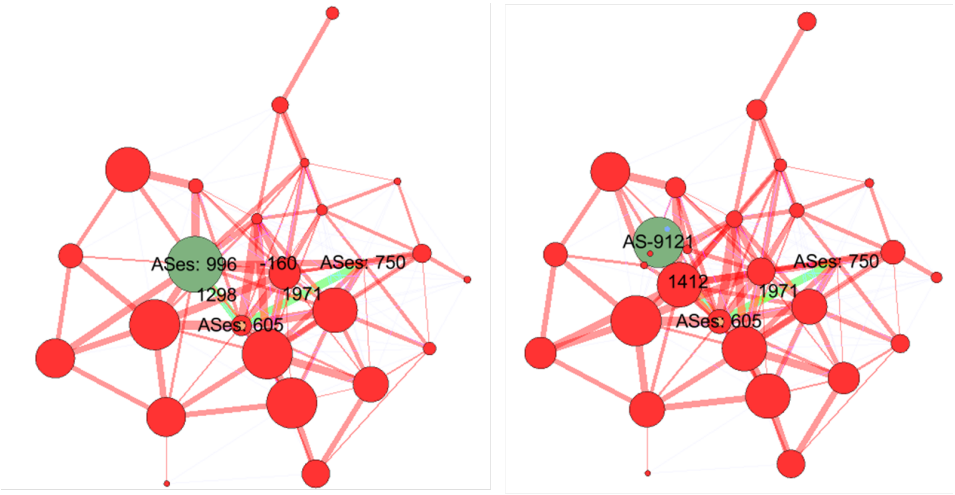


Figure 5.18: The Entropy plot of level-11 of the clustering hierarchy, for the period 23-Dec-2004 to 24-Dec-2004. The time step is set to $\Delta T = 30$ *sec*. It is apparent that the information drastically increases around 9:24 GMT, 24-Dec-2004.

Figure 5.18 depicts the entropy plot of level-11 of the clustering hierarchy, spanning a two-days period, from 23-Dec-2004 to 24-Dec-2004. Observing this overview plot, a pattern emerges. At 9.24 GMT the AS-Graph suddenly increased its input information content by a factor of five. This is an indicator that something might be happening. As a result the user selects a small time window around 9.24 GMT and creates the AS-Graph visualizing the level-11 of the clustering hierarchy, which is depicted in Figure 5.19.

Observing Figure 5.19(a), it is apparent that one cluster of ASes gains weight (gains IPs), which is represented with green color, while all the neighboring clusters lose weight (lose IPs), a fact which is shown using red color. This behavior is repeated in the case of the edges, in which all the edges, except for one path of edges, lose weight.

After "semantic zooming" to the cluster that is gaining weight (Figure 5.19(b)) it is clear that AS-9121 is responsible for this event, since it gains weight (announces a lot of IPs) and all it's neighbor vertices lose weight (lose IPs). It is apparent that the event concerns a clear case of prefix hijack. This observation is verified by the NANOG discussion forums. It's confirmed that AS-9121 originated almost all the prefixes in the Internet, thus making a route through AS-9121 more preferable than some of the



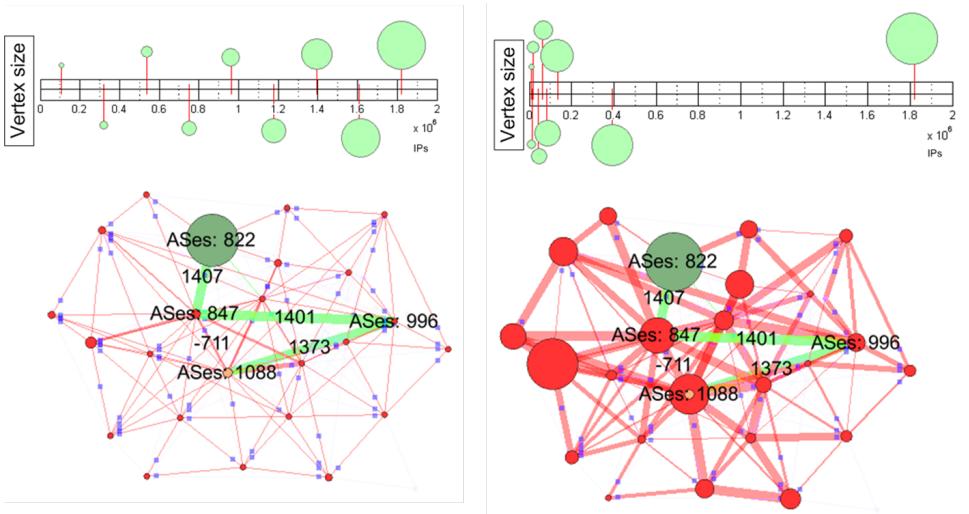
(a) Abstract view of level 11 of the clustering hierarchy

(b) Abstract view of level 11 after zooming to the green vertex

Figure 5.19: Abstract view of level 11 visualizing the event that took place on Dec 24, 2004 concerning AS-9121. (a) The green vertex gains a lot of IPs while the neighboring ASes lose IPs. Furthermore, the green path takes over the traffic from many other neighboring paths (red paths). (b) It is apparent that AS-9121 is the one that hijacks a big portion of the Internet traffic.

longer but genuine routes. Although, this event was caused by an error and not a malicious act, the consequences had a global impact: for many hours several Internet users were unable to reach a large number of Websites.

Figure 5.20 illustrates the result of the magnification procedure on the *BGPGraph* visualization approach. It is apparent from this Figure that the linear mapping (Figure 5.20(a)) induces large information loss with regards to the propagation of the event to the ASes that are depicted. On the other hand the magnified view is more informative regarding this fact. The vertex mapping labels on the upper part of Figure 5.20 illustrate the relationship between the size of the vertices and the actual weight value they represent. It is apparent from these labels that since the majority of the vertex weights have low value, the magnified view uses more size discrimination levels for the lower part of the distribution.



(a) Linear mapping (b) Magnification factor = 4.5

Figure 5.20: Application of the magnification approach on the *BGPGraph* [1] visualization of the BGP hijacking event, which took place on 24-Dec-2004. AS-9121 due to a router misconfiguration announced a large number of prefixes that did not belong to it.

5.3.3 Graph Visualization magnification

This section demonstrates the application of the Magnification approach in a graph based visualization. In case of large graphs, patterns in the data are hidden by the high cluttering that is generated. Additionally, in the case of graph visualization in small displays, the cluttering becomes worse, and thus, the exploration of the graph structure is more difficult.

For the magnification of the graph visualization, each node is attached to a cell in the 2-dimensional cell, and its new position is found by interpolating on the four vertices of the cell. The node size is changed based on how the size of the associated grid cell changes. The initial node size encodes its degree on the graph. The initial positioning of the graph was calculated using a force directed layout algorithm [111].

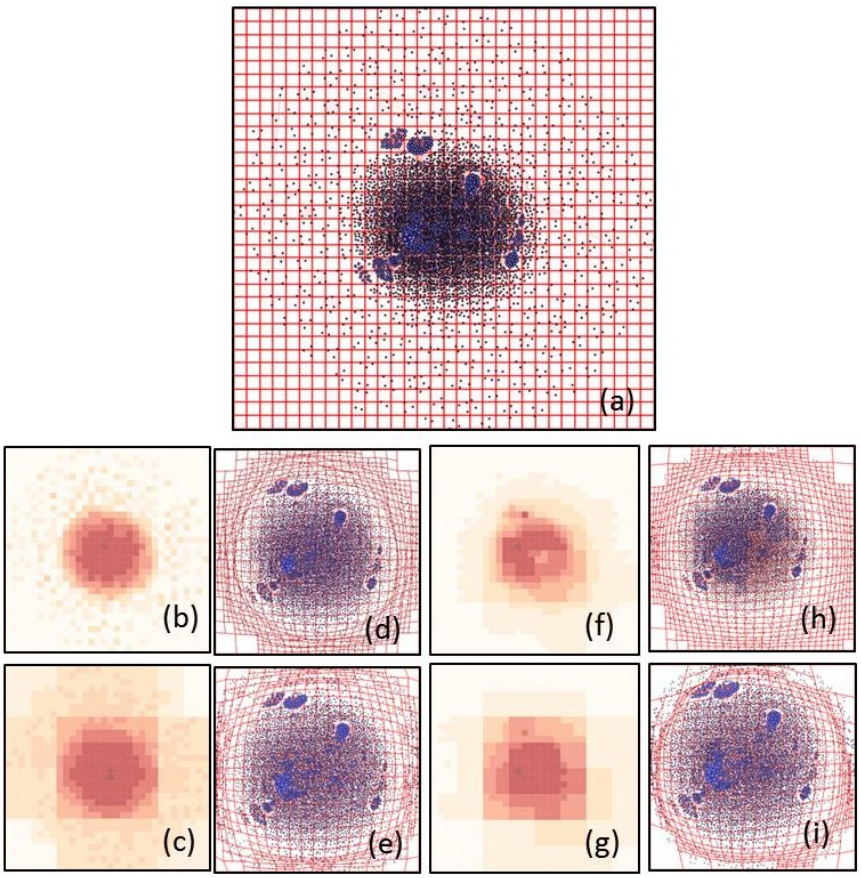


Figure 5.21: Results of the application of the Magnification approach on a graph-based visualization with magnification factor equal to 3. (a) Input visualization. (b),(c) One layer and hierarchical significance maps generated using the entropy of each hyperrectangle. (f),(g) One layer and hierarchical significance maps generated from the saliency based significance map proposed in [14]. (d),(e),(h),(i) Magnification results using the significance maps shown on the left of each subfigure.

A real graph dataset is used for the experimental demonstration of the proposed approach. This dataset represents a collaboration network [93] from the e-print arXiv.

The results of the application of the Magnification approach on the graph visualization are presented in Figure 5.21. Figure 5.21(a) presents the input graph, and Figures 5.21(b) and 5.21(c) the one layer and hierarchical entropy maps. As shown in these figures, the hierarchical entropy map is smoother around significant regions, and thus, preserves better the local

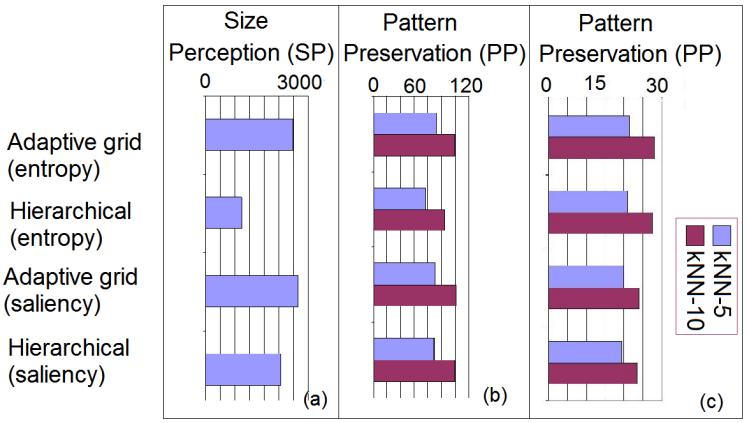


Figure 5.22: (a) SP metric for Figure 5.21. (b) PP metric Figure 5.21. (c) PP metric for Figure 5.23.

patterns and introduces less distortion. Figure 5.21(d) presents the magnification procedure using the adaptive grid proposed in [14] and the one layer significance map, and Figure 5.21(e) presents the magnification procedure using the multi-resolution grid and the hierarchical significance map. Similarly, Figures 5.21(f), 5.21(g), 5.21(h), 5.21(i) represent the same result, but instead using the saliency map proposed in [14]. The proposed approach magnifies the clusters in the datasets more that the single layer significance map. Thus, there is better utilization of the empty space.

Figure 5.22(a) and (b) presents the values of the SP and PP evaluation metrics on the results shown in Figure 5.21, using adaptive grid [14] and hierarchical grid, and utilizing entropy and saliency as the significance maps. According to both metrics, the best results are obtained from the hierarchical grid approach proposed in this work, using the entropy-based significance map.

5.3.4 Scatterplot Visualization magnification

This section presents the application of the Magnification approach on scatterplot visualizations. Although, strictly speaking the scatterplot is not a graph, for a visualization perceptive it can be considered to be a graph without edges. High cluttering occurs when visualizing scatterplots in small displays, or when the input data distribution is very skewed on a specific region. The magnification approach automatically detects the highly cluttered

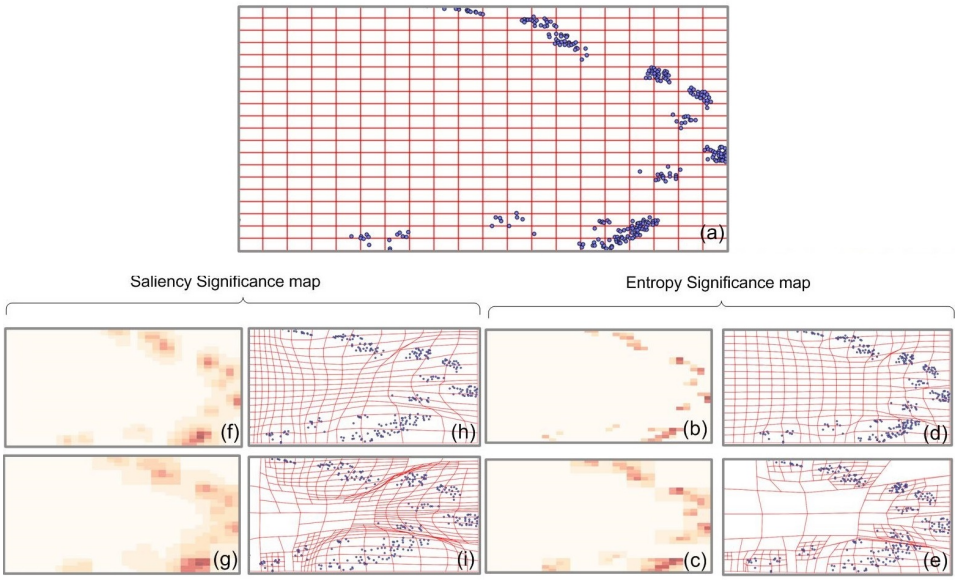


Figure 5.23: Results of the application of the Magnification approach on a scatterplot visualization with magnification factor equal to 3. (a) Input visualization. (b)-(c) One layer and hierarchical significance maps generated using the entropy of each hyperrectangle. (d)-(e) Results of the significance maps shown in (b) and (c) using the method proposed in [14] and in this paper respectively. (f)-(g) One layer and hierarchical significance maps generated from the saliency based significance map proposed in [14]. (h)-(i) Results of the significance maps shown in (f) and (g) using the method proposed in [14] and in this paper respectively.

regions and magnifies them in order to reveal any hidden patterns.

Each point in the scatterplot is attached to a specific hyperrectangle of the grid. After the magnification procedure, the new point positions are found by means of interpolation on the new hyperrectangle positions. The size of the points does not change.

Figure 5.23 presents the results of the application of the magnification approach on the scatterplot visualization of the aforementioned INSPIRE dataset. Two types of significance maps were utilized, the entropy and saliency [14] maps, using both one layer and hierarchical significance definition. The hierarchical significance map takes better advantage of the empty space, and enlarges more the significant areas, with minimum deformation of the local pattern distribution.

As shown in Figure 5.23, large magnification of significant regions can result in distinct clusters being positioned close to each other in the deformed space. This result is unavoidable. But according to the PP evaluation metric and the user evaluation results, this is not enough to destroy the original patterns found in the input dataset.

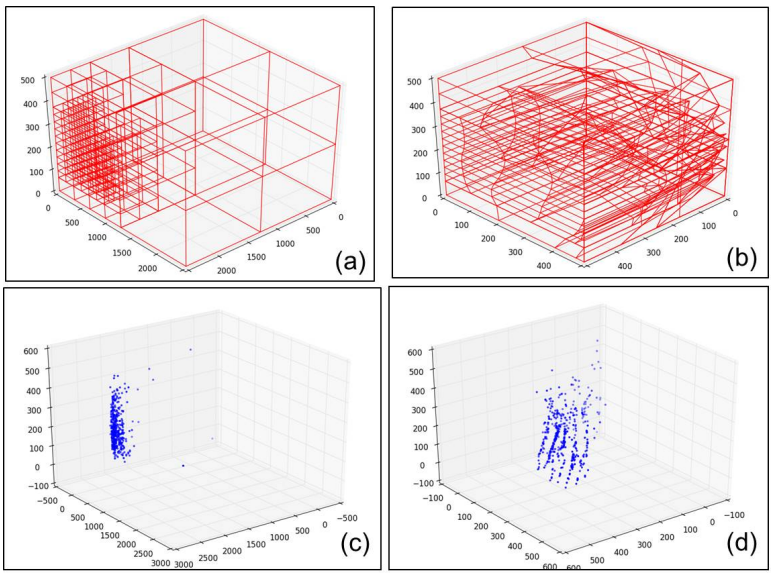


Figure 5.24: Application of the Magnification approach on a 3D scatterplot visualization using the forest fires dataset [2].

The values of the PP evaluation metric applied on Figure 5.23 are shown in Figure 5.22(c). Since the size of the points in the scatterplot remains the same after the magnification procedure, only the PP metric is used. This figure reveals that the best results are obtained from the hierarchical grid approach proposed in this work, using the saliency-based significance map proposed in [14].

Figure 5.24 presents an example of application of the Magnification approach on a 3D scatterplot visualization using the forest fires dataset. Specifically, the Fine Fuel Moisture Code (FFMC), Drought Code (DC), and the Initial Spread Index (ISI), are used for the generation of the scatterplot. The hierarchical grids for the initial and magnified scatterplots are shown in Figures 5.24(a) and 5.24(b), using entropy significance map, while the corresponding magnification results are shown in Figures 5.24(c) and 5.24(d). The correlations between the visualized variables, which are hidden in the

initial view due to high cluttering, are visualized better in the magnified result.

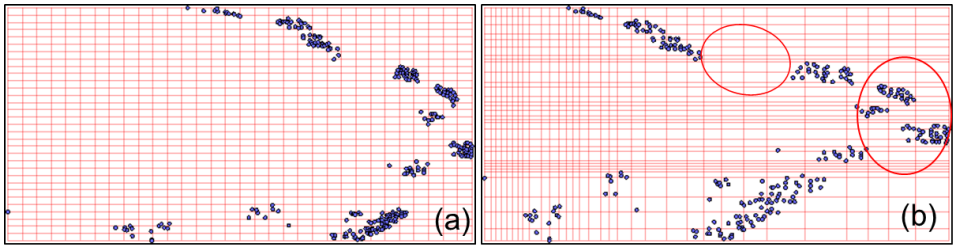


Figure 5.25: Application of the generalized scatterplots [3]. (a) Input data. (b) Result of the generalized scatterplot. The red ellipsoids highlight two regions, the first is empty and not well filled after the deformation, and the second compresses the clusters and distorts the local patterns.

Figure 5.25 shows the application of generalized scatterplots [3] in the data from INSPIRE [94]. In contrast to the proposed approach, the generalized scatterplots are not able to fill the empty space well, and distort more the local patterns. The red ellipsoids highlight two regions, the first is empty and not well filled after the deformation, and the second compresses the clusters and distorts the local patterns.

It should be noted that in both cases of 2-dimensional and 3-dimensional scatterplot magnification, the deformation procedure destroys the chart scales. Due to the non-uniform deformation of each hyperrectangles, the deformation of the data with respect to each axis is also not uniform. This is a limitation of the proposed approach, in the sense that in order to enhance the patterns found in the data, it loses the axis scales. This limitation can also be found in [14].

Chapter 6

Application of the Magnification Approach for enhancing non-graph visualizations

Contents

6.1 Datasets 105

6.2 Experimental evaluation of the hierarchical magnification approach 106

This section presents the application of the Magnification approach to the following non-graph visualization approaches: 1) Word clouds, 2) Choropleth maps, and 3) Images.

6.1 Datasets

6.1.1 Word frequencies dataset

The goal of this dataset is to perform visualization resizing to fit smaller/ different-aspect-ratio screens, identical to the goal proposed by Wu et al. [14], with which the method proposed in this thesis is compared and evaluated.

This dataset is used by the word cloud visualization. It is comprised of 13,828 news articles spanning one year (from 2008 to 2009) that were related to American International Group (AIG). For the generation of the

word cloud, the frequency of appearance of each word on the news articles is computed. The size of the words on the Word cloud is used to visually represent the frequency of each word. Only the most frequent words are kept and visualized, while the rest are discarded.

6.1.2 US Census unemployment dataset

The goal of this dataset is to perform visualization resizing to fit smaller/ different-aspect-ratio screens, identical to the goal proposed by Wu et al. [14], with which the method proposed in this thesis is compared and evaluated.

This dataset is used by the choropleth visualization approach. For this visualization the unemployment of 2011 from the US Census [112] dataset is used. This dataset is composed of the number of citizens that were unemployed on 2011 for all the US counties. The number of unemployed citizens is mapped onto the color of each county on the choropleth visualization.

6.1.3 Image

The goal of this dataset is to perform visualization resizing to fit smaller/ different-aspect-ratio screens, identical to the goal proposed by Wu et al. [14], with which the method proposed in this thesis is compared and evaluated. This image was retrieved from [13].

6.2 Experimental evaluation of the hierarchical magnification approach

6.2.1 Evaluation Metrics

Jensen−Shannon divergence

The Jensen−Shannon Divergence (JSD) [113] is used in probability theory in order to measure the distance between two one-dimensional distributions. It is based on the Kullback−Leibler divergence [113] , and is a symmetric metric. Given two distributions P and R, and denoting as $KLD(P||R)$ the Kullback−Leibler divergence between them, the JSD metric is defined as follows:

$$JSD\left(P||Q\right) = \frac{1}{2}KLD\left(P||M\right) + \frac{1}{2}KLD\left(Q||M\right) \tag{6.1}$$

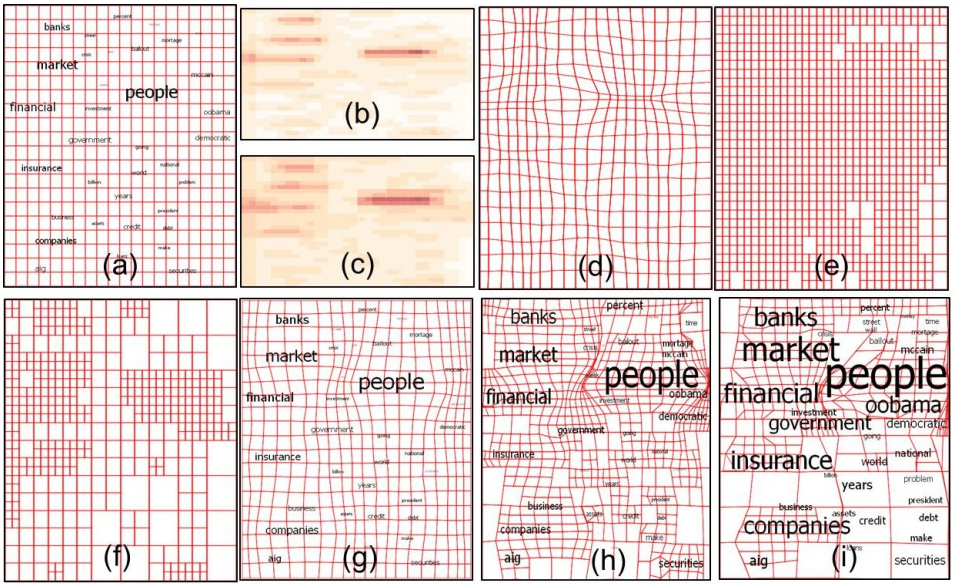


Figure 6.1: Results of the application of the Magnification approach on the word cloud visualization with magnification factor equal to 4. (a) Input visualization (b)-(c) One layer and hierarchical significance maps generated from the saliency based significance map proposed in [14]. (d)-(e) Significance aware grid and adaptive grid proposed in [14]. (f) Proposed multi-resolution grid. (g)-(h) Results obtained from the one layer significance map (in (b)) applied on the grids in subfigures (d) and (e). (i) Result obtained from the hierarchical significance map (in (c)) applied on the grid in subfigure (f). The proposed approach in (i), takes better advantage of the empty space, and enlarges more the words of the visualization.

where $M = \frac{P+Q}{2}$.

When using the logarithm of base 2 for the calculation of the Kullback−Leibler divergence, the JSD metric is bounded in $[0, 1]$, with 0 representing two identical distributions, and 1 representing two completely different distributions.

6.2.2 Word Clouds

In this experiment, the efficiency of the proposed multi-resolution magnification approach is demonstrated on the word clouds visualization. This is especially important in the case of small displays, so as to preserve the

relative positioning of the words during resizing, and enable the fast identification of the words. The preservation of the relative word positioning is important for context-preserving word clouds [114], in which the relative positions of the words on the cloud encode sematic relationships between the words.

The resizing of the word cloud in smaller screens can cause cluttering. The most common reason for this type of cluttering is that some words can become very small to read/understand, which unavoidably leads to information loss. High cluttering is also caused by not preserving the relative sizes of the words in the deformed space. In order to solve this issue, we utilize non-linear deformation that increases the size of the words, while also preserving the relative sizes of the words. The increase in the size of the words is performed in such a way (through the use of the significance map and the deformation procedure), so as to preserve the relative sizes of the words as much as possible, and thus, have a minimum change in the relationship between different words. Thus, the size of the words in the magnified space is (almost) proportional to the size of the words in the input space.

In this context, each word is considered as a visual item, which is attached on all the grid cells that contain it. After the grid deformation, the position of the words is adjusted by means of interpolation, while the final word resizing is found from the associated cell deformations. In addition, in order to avoid word overlapping, a collision avoidance algorithm was implemented, which can change the positions of the words in case of collision.

In this experiment, a context-preserving word cloud by a force-directed algorithm [114] was used on a real dataset with 13,828 news articles spanning one year (from 2008 to 2009) that were related to American International Group (AIG). The context-preserving word cloud results in positioning semantically-similar words close to each other.

Figure 6.1 presents the results of the application of the proposed Magnification approach on the aforementioned word cloud visualization. Figure 6.1(a) shows the input data, while Figures 6.1(b) and 6.1(c) illustrate the one layer and hierarchical significance maps generated using as base the significance map proposed in [14]. As shown in these figures, the hierarchical significance map is smoother around significant regions. Three types of grids are shown in Figures 6.1(d)-(f), the first two proposed in [14] and the last is the multi-resolution grid proposed in this work. The corresponding results

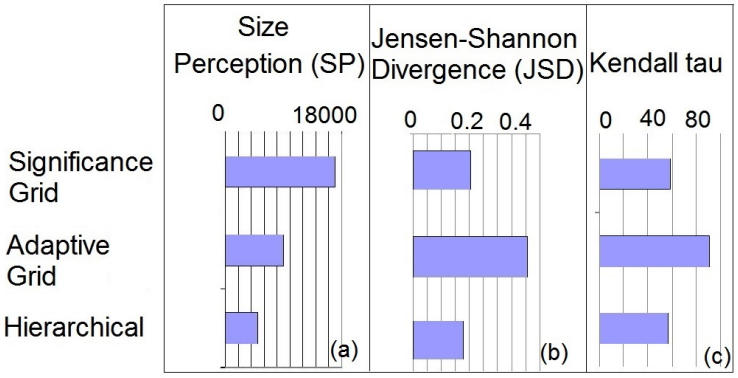


Figure 6.2: The evaluation results using the SP (a) and JSD (b) and Kendall tau (c) metrics applied on Figures 6.1(g), 6.1(h), and 6.1(i) respectively. Smaller values represent better results. The hierarchical approach provides the best results in all the metrics.

of each of these grids are shown in Figures 6.1(g)-(i), in which the first two utilize the one layer significance map, while the last utilizes the hierarchical significance map. Figure 6.1(i) utilizes better the empty space for the magnification of the words, and produces better results.

Figure 6.2 presents the evaluation results applied on Figure 6.1, using the SP and JSD evaluation metrics presented in Section 6.2.1, as well as the Kendal tau distance metric presented in the next paragraphs. The SP metric shows that the multi-resolution grid has the smallest distance between the perception of the sizes of the words, and their intended size, and thus, it is more perceptually consistent. The JSD metric is used to compare the normalized distribution of the sizes of the words in the input word cloud with respect to the word sizes in the deformed spaces. The proposed approach provides the better results.

In order to measure how the relative sizes of the words change after the application of each deformation method, we utilize the Kendall tau distance [115]. This is a distance metric between two ordered lists. In this case the first ordered list is the list of sizes of the input words, and the second is the list of words in the magnified result. Figure 6.2(c) shows the results of this metric. Smaller values represent a greater preservation of the relative ordering of the sizes in the words. The smallest Kendall tau distance is obtained by the proposed hierarchical approach.

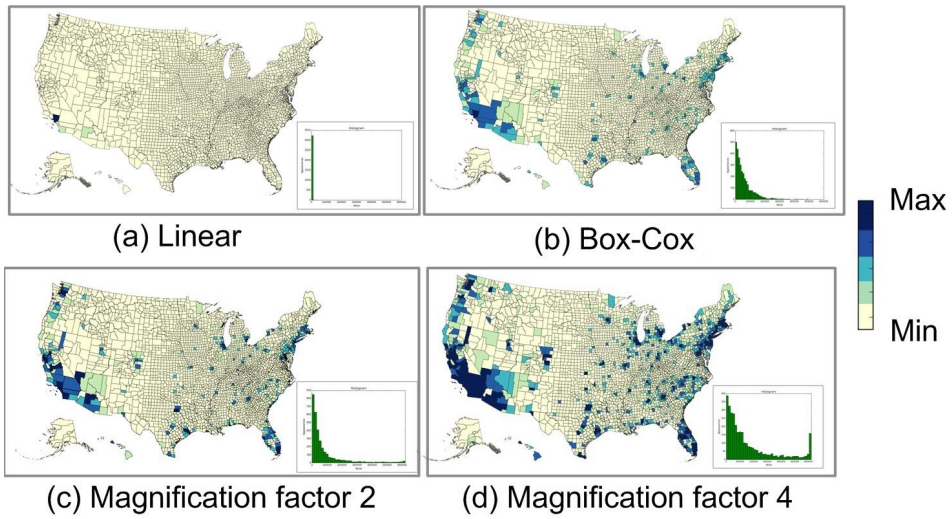


Figure 6.3: Results of the application of the Magnification approach on a choropleth map, visualizing the unemployment data for each US area in 2011. The lower right part of each subfigure shows the corresponding distribution. (a) Lineal mapping. (b) Box-Cox mapping [68]. (c) Magnification factor equal to 2. (d) Magnification factor equal to 4. Although Subfigure (d) loses some information regarding the high values of the dataset, it provides more color resolution to the lower ranges of the data (where the majority of the data values lies), and thus, small differences between the different states are better visualized and perceived.

6.2.3 Choropleth maps

This section presents the application of the Magnification approach on choropleth maps, in which color is utilized to encode quantitative information. In this visualization, each area of the map has a specific attribute value which is mapped onto its color. The magnification approach is applied on the 1-dimensional input dataset, and afterwards, the magnified space is mapped onto the color values. For this visualization the unemployment of 2011 from the US Census [112] dataset is used.

Figure 6.3 presents the results of the Magnification approach on the choropleth map. Figure 6.3(a) presents the case of linear mapping, Figure 6.3(b) the Box-Cox transformation proposed by [68], while Figures 6.3(c) and 6.3(d) presents the results with magnification factor manually selected to be equal to 2 and 4 respectively. The right lower part of the figures shows

the histogram of the distribution of the data in each case. In the case of linear mapping shown in Figure 6.3(a), due to the fact that the distribution is very skewed towards zero, most of the information is cluttered and is not visualized. On the other hand, the visualization on Figures 6.3(c)-(d) is more uniformly distributed, and the choropleth map shows more information regarding the status of the civilian labor force to the different areas of the US. It should be noted that the results and distribution of the data for the [68] mapping resemble the results of the magnification approach for magnification factor equal to 4. The visualization in Figures 6.3(d) provides more information regarding the distribution of the data in the lower ranges of its values, and the small differences between the different states are more evident.

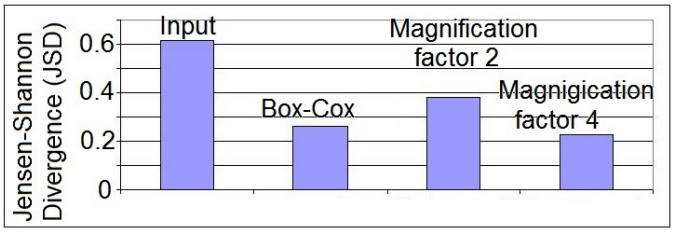


Figure 6.4: This figure shows the application of the JSD metric, in order to measure the distance of the color distribution created from the deformation methods in Figure 6.3, from the optimum uniform distribution (according to [4]). Using magnification factor equal to 4, the results are better than Box-Cox mapping.

For the evaluation of the deformation results in Figure 6.3, the work presented by Brewer and Pickle [4] is used. Specifically, Brewer and Pickle contacted multiple user studies in order to evaluate the effectiveness of the different color mapping techniques in choropleth maps, and investigate how they affected user performance in various tasks. They found that the best mapping method is the quantile mapping, which assigns the same number of data objects to each of the available colors. This is in fact a uniform distribution in the color plane. Taking this fact into account, the use of the JSD metric is used to measure the distance of the distribution of the colors created by the deformation procedure, from the optimum uniform distribution.

The results of the JSD metric are presented in Figure 6.4, which show

that increasing the magnification factor results in more uniform distributions, and thus, better results. Using magnification factor equal to 4, the results are better than Box-Cox mapping.

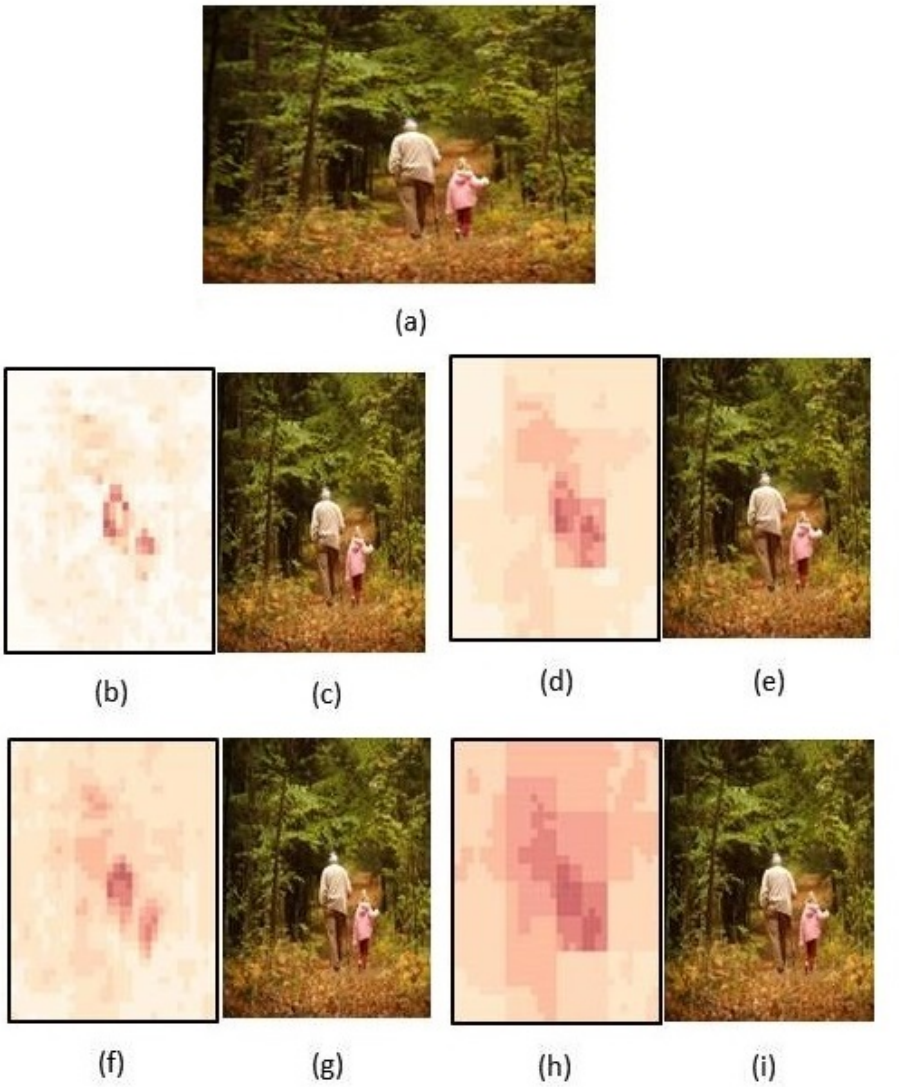


Figure 6.5: Application on image resizing. (a) Input image. (b),(f) Significance maps proposed in [13] and in [16]. (d),(h) Hierarchical significance maps using the Significance maps proposed in [13] and in [16] for each level of the hierarchy. (c),(e),(g),(i) The results using the significance maps on the left of each figure.

It should be noted that the JSD value is used in order to select the most

appropriate scale factor. Specifically, given a set of possible magnification factors the JSD value is computed for each one of them. Afterwards, the magnification factor that has the best JSD value is selected as the most appropriate. In the case of Figure 6.3, the magnification factor equal to 4 was the best.

6.2.4 Image resizing

The hierarchical magnification approach proposed in the context of this work is a general approach, which takes as input an image or a dataset, and a significance map, and enhances significant regions of the image/data. Under this consideration, this section demonstrates the efficiency of the proposed approach in resizing images. Visualization of images with different aspect ratio can result in distortions of significant objects, which make their recognition more difficult, resulting in low quality images. The resizing or magnification approach is required to identify significant objects that should be magnified in the new display.

In this context, the magnification approach is compared with the image resizing method proposed in [13], utilizing the significance maps proposed in [13] and [16]. Figure 6.5 presents the results of the application of the magnification approach on one image. Figure 6.5(a) presents the input image. Figures 6.5(b) and 6.5(f) present the significance maps proposed in [13] and in [16], while Figures 6.5(d) and 6.5(h) utilize the same maps as a base (i.e. [13] and [16]), in order to generate the hierarchical significance maps. Figures 6.5(c), 6.5(e), 6.5(g), and 6.5(i) present the results using the significance maps on the left of each figure. Compared to the results from the single layer maps, the hierarchical map enhances more the significant objects, which are closer to their initial size. Specifically, the old man and the young girl are magnified more using the hierarchical significance map, and thus, are less distorted.

6.2.5 User evaluation results

This section presents the results of the user evaluation studies. The participants were presented with a variety of visualization methods that were magnified utilizing either the proposed magnification approach, or other state-of-the-art methods. The applications considered resizing/magnification of:

image, graph, scatterplot, word cloud and color (in choropleth maps). In the case of the image resizing, the method proposed by Wang et al. [13] has been used for comparison with the proposed approach. In the case of graph, scatterplot, and word cloud magnification the approach proposed by Wu et al. [14] was used for comparison. Finally, in the case of the choropleth map, the Box-Cox mapping [68] has been used for comparison with the proposed approach. For the evaluation studies, the same datasets presented in this paper were utilized.

Three images/visualizations were given for each application, the input image, and the two result images (produced by applying the proposed approach and the other state-of-the-art method respectively). The participants were asked to select one of the result images. The following questions were asked to the participants depending on the application:

- In the image application: Which result represents better the input image?
- In the graph, scatterplot, and word cloud application: Which result provides better understanding/insight about the input image?
- In the choropleth map application: Which result represents more information about the input data?

Table 6.1: The results of the user evaluation studies on 22 individuals. The participants had to select between the proposed hierarchical approach and other state-of-the-art methods

	Hierarchical	**Other**	**p-value (chi-square)**
Image	86.37%	13.63%	0.0006
Graph	68.18%	31.82%	0.088
Scatterplot	63.64%	36.36%	0.2008
Choropleth	95.45%	4.55%	0.00002
Wordcloud	81.82%	18.18%	0.0028
Overall	79.09%	20.91%	1.046e-09

The results of the user evaluation studies are illustrated in Table 6.1. As shown in this figure, the participants selected the results of hierarchical approach more often than the competing method. The difference is larger in the image, choropleth map, and word cloud visualizations. Smaller differences can be found in the graph and scatterplot applications. General discussion with the users after the evaluation revealed two distinct groups. The first group preferred displays with dense clusters, that had a large distance between them, and this is the reason they selected the less magnified result (i.e. the competing method in [14], since the magnification using the hierarchical method was generally larger). On the other hand, the second group found that the more magnified view provided better insight regarding the fine details within a cluster. Table 6.1 also presents the results of hypothesis testing, i.e. chi-square goodness of fit, in order to identify the difference of the observed distributions with respect to random guess. In all the cases except for the scatterplot and graph visualizations, the results are indeed statistically significant. Overall, 79.09% of the participants selected the hierarchical approach with p-value below 0.005.

6.2.6 Time performance

This section presents a comparison of running times, between regular and multi-resolution grids. The multi-resolution grid reduces the optimization space by assigning less hyperrectangles to less significant regions. Thus, compared to the regular grid, the total number of hyperrectangles in a multi-resolution grid is less, without affecting the quality of the results. The comparison takes place for a 32×32 2-dimensional regular grid, and a multi-resolution 2-dimensional grid with maximum resolution equal to 32×32. In all cases, the number of hyperrectangles was reduced from 1024 (for regular) to around 200-800 (for multi-resolution). The experiments were conducted on an Intel core 2 CPU 2,4 GHz, 3 GB RAM, 32-bit windows-XP desktop PC. Although, not being a direct scope of the current work, we have also implemented the algorithm in GPU-compatible version, and the advancement in execution time has shown an acceleration of roughly 8x.

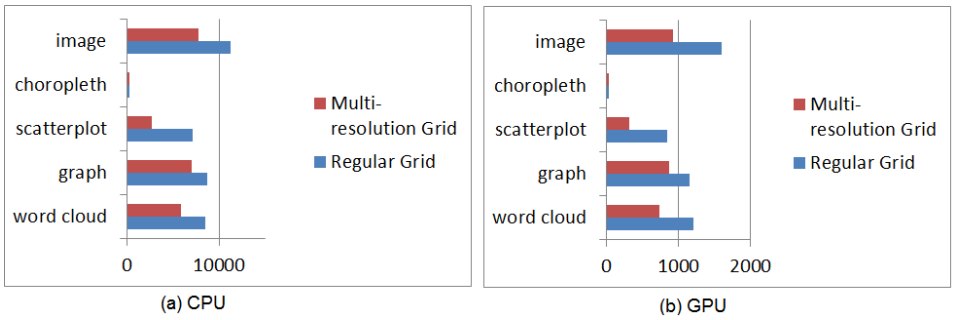


Figure 6.6: Running times for regular and multi-resolution grids for all the datasets in this paper measured in milliseconds.

The running times of the regular and multi-resolution grids for all the datasets presented in the previous sections are presented in Figure 6.6. As shown in Figure 6.6, the multi-resolution grid runs 1.5 times faster than the regular grid for all datasets. This Figure shows the total time until convergence. The average time it takes to run each iteration of the algorithm is around 600 ms for regular and 400 ms for multi-resolution on a single CPU.

Chapter 7

Conclusions

Contents

7.1 Summary of the Thesis 117

7.2 Critical discussion 119

7.3 Future work 120

The present thesis dealt with the development and investigation of novel graph analytics techniques for enhancing the data analysis procedure and provide improved data insight. The research focused on two fields of graph analytics, namely, graph mining and graph visualization. In the next few sections, a summary of the current thesis follows in Section 7.1, along with a critical discussion in Section 7.2, and a few ideas for future work in Section 7.3.

7.1 Summary of the Thesis

The thesis started with a thorough analysis and reference to the state-of-the-art approaches in the graph analytics field, attempting, this way, to clarify the limitations encountered in the previously proposed approaches and the need for novel analysis methods. After presenting and briefly analyzing the most significant works regarding existing graph analytics techniques, the thesis presents the proposed extensions of current state-of-the-art and the motivation towards the introduction of a new graph analytics techniques.

In particular, this thesis presented novel graph mining and visualization techniques expected to deliver deeper understanding of the behavioral pattern formulations of the nodes inside a graph. In the context of graph

mining, novel graph features were proposed, in order to efficiently capture different patterns inside a graph. The suggested method includes information from dynamic neighborhood sizes, and thus, is rendered better in the identification of correlations between the different nodes, when compared to methods that utilize static neighborhood sizes.

The efficiency of the proposed graph features has been evaluated on different graph datasets. In particular, four different datasets have been utilized, for the data mining task of identifying anomalous nodes in the graph. The proposed features were also compared with alternative state-of-the-art features and methods for anomaly detection in the same tasks. All experimental results exhibited very promising recognition rates, achieving high true positive and low false positive rates that surpass the alternative methods. Furthermore, analysis with different neighborhood sizes revealed that depending on the scenario and the detection goal, different neighborhood sizes provide different accuracies.

Additionally to the above, the current thesis also dealt with enhancing the graph visualizations used in the context of graph analytics. Towards this goal, this thesis introduced a hierarchical magnification approach that enables the interactive reduction of visual clutter, and the magnification of significant regions of the data in multiple dimensions. Significant regions are uniformly magnified with minimum distortion, while the distortion is distributed to the less significant areas of the display. The proposed hierarchical significance map combines the information of the significance maps over multiple grid scales, based on each hyperrectangle's intersections with other hyperrectangles, and the significance of each grid using a normalization operator. The proposed approach identifies both small and large objects/patterns, and scales them with minimum distortion in the magnified space. In addition, the introduction of the multi-resolution grid reduces the number of variables in the optimization space, and thus, decreases the execution time of the algorithm. The efficiency of the proposed approach was demonstrated on multiple visualization approaches, and the results were found to be superior to previous methods using multiple evaluation metrics and user studies.

7.2 Critical discussion

The list below summarizes the most significant achievements of the current thesis, while a more detailed discussion follows thereafter:

- Introduction of novel graph features for graph mining.
- Generation of graph features using dynamic graph neighborhoods
- Experimentally verified the higher accuracy of the analysis methods when using the proposed features, compared to existing state-of-the-art features
- Introduction of a hierarchical magnification approach for clutter reduction in graph visualizations.
- Utilization of a multi-resolution grid for non-linear magnification that makes the optimization procedure faster.
- Experimentally verified the superiority of the hierarchical magnification approach when compared to other magnification approaches for clutter reduction in visualizations.

The advantage of the proposed graph features is that they were able to efficiently identify different behavioral patterns in a variety of applications, and also surpassed existing graph features and methods in achieving higher accuracy. Despite this fact, however, the proposed graph features have not been proven to be more accurate in any graph-related task. This means that a different set of features could be possibly considered, in order to be able to cover a wider range of applications. The ideal goal, however, would be the automatic generation of the most relevant features with respect to the specific application field.

The introduction of different neighborhood sizes is one of the main contributions of the current thesis. Analysis with different neighborhood sizes revealed that depending on the scenario and the detection goal, different neighborhood sizes provide different accuracies. When compared to the use of static neighborhood sizes, the dynamic neighborhoods added more descriptive power to the features, and further improved data insight. The utilization of the random forest classifier was able to efficiently identify the most relevant neighborhoods for each feature, and provided high accuracy

results. The drawback, however of the proposed dynamic neighborhood sizes is that not all neighborhoods are used by the analysis procedure. In tasks different from anomaly detection, this could possibly affect the accuracy of the graph mining task. Specific feature selection techniques should be developed and employed so as to select the most appropriate neighborhood sizes in each application.

The proposed hierarchical magnification approach has been found to reduce clutter more efficiently than existing methods, in a variety of graph and non-graph visualizations. The hierarchical significance map is able to efficiently identify both small and large size significant regions, and magnify them better compared to using single significance maps. An important issue that has only been less extensively covered in this thesis, is the selection of the magnification factor. In some cases, such as in the choropleth maps, the magnification factor was automatically detected amongst a list of alternatives, trying to maximize the Jensen−Shannon Divergence (JSD) metric. In cases of image/visualization resizing, the magnification factor is set by utilizing the intrinsic input and output dimensions. Instead, if the analyst was simply interested in magnification for clutter reduction, the magnification factor was manually selected. This indicates the need for an automatic identification of the most appropriate magnification factor that would provide the best trade-off between the magnification level and the distortion error.

7.3 Future work

A possible plan for future work based on the outcomes and the achievements of the current thesis regards the investigation of the items presented in the following list:

- Development of an automatic method for the generation of the most relevant graph features depending on the given application.
- Examination and proposal of alternative methods for the detection of the most appropriate graph neighborhood sizes.
- Development of a method for the automatic identification of the most appropriate magnification factor for visualization resizing.
- Research of additional irregular grid formations and their effect on the deformation of visualization.

Specifically, an important part of future work includes the definition of automatic methods for the generation of the most relevant graph features depending on the given application. The features should be dynamically generated and not comprised of a static list. This approach could extent the graph features for application in a large range of applications, maintaining high accuracy across them. An example of this automatic feature definition can be found in [116], where the authors utilize the classification results in order to improve the utilized features.

Another important research direction includes the definition of novel feature selection techniques and the investigation of existing, for the task of defining the most appropriate graph neighborhood sizes. This could further increase the accuracy of the proposed features, and render them applicable to a wider range of graph-related applications.

Furthermore, an issue that could also be of scientific interest includes the automatic definition of the magnification factor utilized in the proposed hierarchical magnification approach. This would remove the burden from the analyst to manually select the magnification factor, and provide him/her with the best result in order to start the analytical exploration.

Last but not least, another future research direction is the investigation of alternative grid formations and their effect on the deformation of visualization. In the current thesis, the multi-resolution grid has been utilized, and found to provide good and fast results in many applications. There might exist, however, alternative grid formations, e.g. triangular grids, that might provide results that are more efficiently magnified and have less distortion.

Bibliography

[1] Stavros Papadopoulos, Konstantinos Moustakas, and Dimitrios Tzovaras. Hierarchical Visualization of BGP Routing Changes Using Entropy Measures. In George Bebis, Richard Boyle, Bahram Parvin, Darko Koracin, Charless Fowlkes, Sen Wang, Min-Hyung Choi, Stephan Mantler, Jürgen Schulze, Daniel Acevedo, Klaus Mueller, and Michael Papka, editors, *Advances in Visual Computing*, volume 7432 of *Lecture Notes in Computer Science*, pages 696–705. Springer Berlin Heidelberg, 2012.

[2] Paulo Cortez and Aníbal de Jesus Raimundo Morais. A data mining approach to predict forest fires using meteorological data. 2007.

[3] Daniel A Keim, Ming C Hao, Umeshwar Dayal, Halldor Janetzko, and Peter Bak. Generalized scatter plots. *Information Visualization*, 9(4):301–311, 2010.

[4] Cynthia A Brewer and Linda Pickle. Evaluation of methods for classifying epidemiological data on choropleth maps in series. *Annals of the Association of American Geographers*, 92(4):662–681, 2002.

[5] Leman Akoglu, Hanghang Tong, and Danai Koutra. Graph based anomaly detection and description: a survey. *Data Mining and Knowledge Discovery*, pages 1–63, 2014.

[6] Deepayan Chakrabarti and Christos Faloutsos. Graph mining: Laws, generators, and algorithms. *ACM Computing Surveys (CSUR)*, 38(1):2, 2006.

[7] Zhenyu Wu and Richard Leahy. An optimal graph theoretic approach to data clustering: Theory and its application to image segmentation.

Pattern Analysis and Machine Intelligence, IEEE Transactions on, 15(11):1101–1113, 1993.

[8] Mark E J Newman and Michelle Girvan. Mixing patterns and community structure in networks. In *Statistical mechanics of complex networks*, pages 66–87. Springer, 2003.

[9] Michelle Girvan and Mark E J Newman. Community structure in social and biological networks. *Proceedings of the national academy of sciences*, 99(12):7821–7826, 2002.

[10] Geoffrey Ellis and Alan Dix. A taxonomy of clutter reduction for information visualisation. *Visualization and Computer Graphics, IEEE Transactions on*, 13(6):1216–1223, 2007.

[11] Ruth Rosenholtz, Yuanzhen Li, Jonathan Mansfield, and Zhenlan Jin. Feature congestion: a measure of display clutter. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 761–770. ACM, 2005.

[12] Stefano Baldassi, Nicola Megna, and David C Burr. Visual clutter causes high-magnitude errors. *PLoS biology*, 4(3):e56, 2006.

[13] Yu-Shuen Wang, Chiew-Lan Tai, Olga Sorkine, and Tong-Yee Lee. Optimized scale-and-stretch for image resizing. In *ACM Transactions on Graphics (TOG)*, volume 27, page 118. ACM, 2008.

[14] Yingcai Wu, Xiaotong Liu, Shixia Liu, and Kwan-Liu Ma. ViSizer: a visualization resizing framework. *Visualization and Computer Graphics, IEEE Transactions on*, 19(2):278–290, 2013.

[15] Chaoli Wang, Jinzhu Gao, Liya Li, and Han-Wei Shen. A multiresolution volume rendering framework for large-scale time-varying data visualization. In *Volume Graphics, 2005. Fourth International Workshop on*, pages 11–223. IEEE, 2005.

[16] Laurent Itti, Christof Koch, and Ernst Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on pattern analysis and machine intelligence*, 20(11):1254–1259, 1998.

[17] Chris Bennett, Jody Ryall, Leo Spalteholz, and Amy Gooch. The Aesthetics of Graph Visualization. *Computational Aesthetics*, 2007:57–64, 2007.

[18] Martyn Taylor and Peter Rodgers. Applying graphical design techniques to graph visualisation. In *Information Visualisation, 2005. Proceedings. Ninth International Conference on*, pages 651–656. IEEE, 2005.

[19] Therese Biedl, Joe Marks, Kathy Ryall, and Sue Whitesides. Graph multidrawing: Finding nice drawings without defining nice. In *Graph Drawing*, pages 347–355. Springer, 1998.

[20] H C Purchase. Metrics for graph drawing aesthetics. *Journal of Visual Languages & Computing*, 13(5):501–516, 2002.

[21] Danny Holten and Jarke J Van Wijk. Force-Directed Edge Bundling for Graph Visualization. In *Computer Graphics Forum*, volume 28, pages 983–990. Wiley Online Library, 2009.

[22] Linton C Freeman. A set of measures of centrality based on betweenness. *Sociometry*, pages 35–41, 1977.

[23] Phillip Bonacich and Paulette Lloyd. Eigenvector-like measures of centrality for asymmetric relations. *Social Networks*, 23(3):191–201, 2001.

[24] Jae Dong Noh and Heiko Rieger. Random walks on complex networks. *Physical review letters*, 92(11):118701, 2004.

[25] Duncan J Watts and Steven H Strogatz. Collective dynamics of small-worldnetworks. *nature*, 393(6684):440–442, 1998.

[26] U Kang, Charalampos E Tsourakakis, Ana Paula Appel, Christos Faloutsos, and Jure Leskovec. Hadi: Mining radii of large graphs. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 5(2):8, 2011.

[27] Leman Akoglu, Mary McGlohon, and Christos Faloutsos. Oddball: Spotting anomalies in weighted graphs. In *Advances in Knowledge Discovery and Data Mining*, pages 410–421. Springer, 2010.

[28] Keith Henderson, Brian Gallagher, Lei Li, Leman Akoglu, Tina Eliassi-Rad, Hanghang Tong, and Christos Faloutsos. It's who you know: graph mining using recursive structural features. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 663–671. ACM, 2011.

[29] U Kang, Jay-Yoon Lee, Danai Koutra, and Christos Faloutsos. Net-ray: Visualizing and mining billion-scale graphs. In *Advances in Knowledge Discovery and Data Mining*, pages 348–361. Springer, 2014.

[30] R Duncan Luce and Albert D Perry. A method of matrix analysis of group structure. *Psychometrika*, 14(2):95–116, 1949.

[31] Tore Opsahl and Pietro Panzarasa. Clustering in weighted networks. *Social networks*, 31(2):155–163, 2009.

[32] J. Körner. Coding of an information source having ambiguous alphabet and the entropy of graphs. *Transactions of the Sixth Prague Conference on Information Theory*, pages 411–425, 1973.

[33] U Kang, Mary McGlohon, Leman Akoglu, and Christos Faloutsos. Patterns on the connected components of terabyte-scale graphs. In *Data Mining (ICDM), 2010 IEEE 10th International Conference on*, pages 875–880. IEEE, 2010.

[34] Corinna Cortes, Daryl Pregibon, and Chris Volinsky. Communities of interest. *Intelligent Data Analysis (IDA)*, 6(3):211–219, 2002.

[35] Lisa Johansen, Michael Rowell, Kevin R B Butler, and Patrick Drew McDaniel. Email Communities of Interest. In *In Proceedings of the Conference on Email and Anti-Spam (CEAS)*, 2007.

[36] Patrick Drew McDaniel, Subhabrata Sen, Oliver Spatscheck, Jacobus E van der Merwe, William Aiello, and Charles R Kalmanek. Enterprise Security: A Community of Interest Based Approach. In *Proceedings of Network and Distributed Systems Security 2006 (NDSS)*, 2006.

[37] Satu Elisa Schaeffer. Graph clustering. *Computer Science Review*, 1(1):27–64, 2007.

[38] Curtis Huttenhower, Avi I Flamholz, Jessica N Landis, Sauhard Sahi, Chad L Myers, Kellen L Olszewski, Matthew A Hibbs, Nathan O Siemers, Olga G Troyanskaya, and Hilary A Coller. Nearest Neighbor Networks: clustering expression data based on gene neighborhoods. *Bmc Bioinformatics*, 8(1):1, 2007.

[39] Akira Ushioda and J Kawasaki. Hierarchical clustering of words and application to NLP tasks. In *Proceedings of the Fourth Workshop on Very Large Corpora*, pages 28–41, 1996.

[40] Simon D M White and Carlos S Frenk. Galaxy formation through hierarchical clustering. *The Astrophysical Journal*, 379:52–79, 1991.

[41] Maria C V Nascimento and Andre CPLF De Carvalho. Spectral methods for graph clustering–a survey. *European Journal of Operational Research*, 211(2):221–231, 2011.

[42] Lars Hagen and Andrew B Kahng. New spectral methods for ratio cut partitioning and clustering. *Computer-aided design of integrated circuits and systems, ieee transactions on*, 11(9):1074–1085, 1992.

[43] Kenneth M Hall. An r-dimensional quadratic placement algorithm. *Management science*, 17(3):219–229, 1970.

[44] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(8):888–905, 2000.

[45] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. In *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, pages 731–737. IEEE, 1997.

[46] Mark E J Newman. Finding community structure in networks using the eigenvectors of matrices. *Physical review E*, 74(3):36104, 2006.

[47] Charles J Alpert, Andrew B Kahng, and So-Zen Yao. Spectral partitioning with multiple eigenvectors. *Discrete Applied Mathematics*, 90(1):3–26, 1999.

[48] FRBMI Jordan and F Bach. Learning spectral clustering. *Adv. Neural Inf. Process. Syst*, 16:305–312, 2004.

[49] Andrea Capocci, Vito D P Servedio, Guido Caldarelli, and Francesca Colaiori. Detecting communities in large networks. *Physica A: Statistical Mechanics and its Applications*, 352(2):669–676, 2005.

[50] Santo Fortunato, Vito Latora, and Massimo Marchiori. Method to find community structures based on information centrality. *Physical review E*, 70(5):56104, 2004.

[51] Vito Latora and Massimo Marchiori. Efficient behavior of small-world networks. *Physical review letters*, 87(19):198701, 2001.

[52] Colin Ware. *Information Visualization: Perception for Design*, volume 22 of *The Morgan Kaufmann series in interactive technologies / ed. Stuart Card. - San Francisco : Morgan Kaufmann.* Morgan Kaufmann, 2004.

[53] Jacques Bertin. *Semiology of Graphics*, volume 94. University of Wisconsin Press, 1983.

[54] Jock Mackinlay. Automating the design of graphical presentations of relational information. *ACM Transactions on Graphics*, 5(2):110–141, 1986.

[55] Enrico Bertini. Quality Metrics in High-Dimensional Data Visualization: An Overview and Systematization. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2203–2212, 2011.

[56] Jun Tao, Chaoli Wang, Ching-Kuang Shene, and Seung Hyun Kim. A Deformation Framework for Focus+Context Flow Visualization. *Visualization and Computer Graphics, IEEE Transactions on*, 20(1):42–55, jan 2014.

[57] Yu-Shuen Wang, Chaoli Wang, Tong-Yee Lee, and Kwan-Liu Ma. Feature-preserving volume data reduction and focus+ context visualization. *Visualization and Computer Graphics, IEEE Transactions on*, 17(2):171–181, 2011.

[58] George W Furnas. *Generalized fisheye views*, volume 17. ACM, 1986.

[59] Bernd Karstens, René Rosenbaum, and Heidrun Schumann. Information presentation on mobile handhelds. In *Proceedings of IRMA*, 2003.

[60] Manojit Sarkar and Marc H Brown. Graphical fisheye views of graphs. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 83–91. ACM, 1992.

[61] T Alan Keahey and Edward L Robertson. Techniques for non-linear magnification transformations. In *Information Visualization, IEEE Symposium on*, page 38. IEEE Computer Society, 1996.

[62] Sheelagh Carpendale, John Ligh, and Eric Pattison. Achieving higher magnification in context. In *Proceedings of the 17th annual ACM symposium on User interface software and technology*, pages 71–80. ACM, 2004.

[63] Kun-Chuan Feng, Chaoli Wang, Han-Wei Shen, and Tong-Yee Lee. Coherent time-varying graph drawing with multifocus+ context interaction. *Visualization and Computer Graphics, IEEE Transactions on*, 18(8):1330–1342, 2012.

[64] Ariel Shamir and Olga Sorkine. Visual media retargeting. In *ACM SIGGRAPH ASIA 2009 Courses*, page 11. ACM, 2009.

[65] Michael Rubinstein, Ariel Shamir, and Shai Avidan. Multi-operator media retargeting. In *ACM Transactions on Graphics (TOG)*, volume 28, page 23. ACM, 2009.

[66] David Thompson, Janine Bennett, C Seshadhri, and Ali Pinar. A provably-robust sampling method for generating colormaps of large data. In *Large-Scale Data Analysis and Visualization (LDAV), 2013 IEEE Symposium on*, pages 77–84. IEEE, 2013.

[67] Martin Eisemann, Georgia Albuquerque, and Marcus Magnor. Data driven color mapping. *Proceedings EuroVA*, 2011.

[68] Ross Maciejewski, Avin Pattath, Sungahn Ko, Ryan Hafen, William S Cleveland, and David S Ebert. Automated box-cox transformations for improved visual encoding. *Visualization and Computer Graphics, IEEE Transactions on*, 19(1):130–140, 2013.

[69] Gábor Simonyi. Perfect graphs and graph entropy. An updated survey. *Perfect graphs*, pages 293–328, 2001.

[70] Nicolas Bourgeois, Bruno Escoffier, Vangelis Th Paschos, and Johan M M van Rooij. A bottom-up method and fast algorithms for max independent set. In *Algorithm Theory-SWAT 2010*, pages 62–73. Springer, 2010.

[71] Stas Goferman, Lihi Zelnik-Manor, and Ayellet Tal. Context-aware saliency detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(10):1915–1926, 2012.

[72] Shixiang Jia, Caiming Zhang, Xuemei Li, and Yuanfeng Zhou. Mesh resizing based on hierarchical saliency detection. *Graphical Models*, 2014.

[73] Qiong Yan, Li Xu, Jianping Shi, and Jiaya Jia. Hierarchical saliency detection. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 1155–1162. IEEE, 2013.

[74] Thomas M Cover and Joy A Thomas. *Elements of information theory*. John Wiley & Sons, 2012.

[75] Chang Ha Lee, Amitabh Varshney, and David W Jacobs. Mesh saliency. In *ACM Transactions on Graphics (TOG)*, volume 24, pages 659–666. ACM, 2005.

[76] Ruth Rosenholtz. Search asymmetries? What search asymmetries? *Perception & Psychophysics*, 63(3):476–489, 2001.

[77] NEMESYS project. http://www.nemesys-project.eu/nemesys/, 2015.

[78] Erol Gelenbe, Gökçe Görbil, Dimitrios Tzovaras, Steffen Liebergeld, David Garcia, Madalina Baltatu, and George Lyberopoulos. NEMESYS: Enhanced network security for seamless service provisioning in the smart mobile ecosystem. In *Information Sciences and Systems 2013*, pages 369–378. Springer, 2013.

[79] Omer H Abdelrahman, Erol Gelenbe, Gökçe Görbil, and Boris Oklander. Mobile network anomaly detection and mitigation: The NEMESYS approach. In *Information Sciences and Systems 2013*, pages 429–438. Springer, 2013.

[80] Erol Gelenbe, Gokce Gorbil, Dimitrios Tzovaras, Steffen Liebergeld, David Garcia, Madalina Baltatu, and George Lyberopoulos. Security for smart mobile networks: The NEMESYS approach. In *Privacy and Security in Mobile Systems (PRISMS), 2013 International Conference on*, pages 1–8. IEEE, 2013.

[81] Patrick Traynor, Michael Lin, Machigar Ongtang, Vikhyath Rao, Trent Jaeger, Patrick McDaniel, and Thomas La Porta. On cellular botnets: measuring the impact of malicious devices on a cellular network core. In *Proceedings of the 16th ACM conference on Computer and communications security*, pages 223–234. ACM, 2009.

[82] Nicola Gobbo, Alessio Merlo, and Mauro Migliardi. A Denial of Service Attack to GSM Networks via Attach Procedure. In *Security Engineering and Intelligence Informatics*, pages 361–376. Springer, 2013.

[83] Julie K Petersen. *The telecommunications illustrated dictionary*. CRC Press, 2002.

[84] Eun Kyoung Kim, Patrick McDaniel, and Thomas La Porta. A detection mechanism for SMS flooding attacks in cellular networks. In *Security and Privacy in Communication Networks*, pages 76–93. Springer, 2013.

[85] William Enck, Patrick Traynor, Patrick McDaniel, and Thomas La Porta. Exploiting open functionality in {SMS}-capable cellular networks. In *Proc. 12th ACM Conf. Computer and Communications security (CCS'05)*, pages 393–404, Alexandria, VA, 2005.

[86] GenieLog. GEDIS Studio online, 2014.

[87] Omnet. Omnet++, http://www.omnetpp.org/, 2015.

[88] G Gorbil, O H Abdelrahman, M Pavloski, and E Gelenbe. Modeling and Analysis of RRC-Based Signalling Storms in 3G Networks. *Emerging Topics in Computing, IEEE Transactions on*, 4(1):113 – 127, 2016.

[89] O H Abdelrahman and E Gelenbe. Signalling storms in 3G mobile networks. In *Communications (ICC), 2014 IEEE International Conference on*, pages 1017–1022, 2014.

[90] Mihajlo Pavloski and Erol Gelenbe. Signaling Attacks in Mobile Telephony. In *SECRYPT*, pages 206–212, 2014.

[91] RIPE Network Coordination Centre (available at http://www.ripe.net). Routing Information Service project (RIS).

[92] Y Rekhter, T Li, and S Hares. A Border Gateway Protocol 4 (BGP-4). *Request for Comment (RFC): 1771*, 1995.

[93] Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. Graph evolution: Densification and shrinking diameters. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 1(1):2, 2007.

[94] Pak Chung Wong and Jim Thomas. Visual analytics: building a vibrant and resilient national science. *Information Visualization*, 8(4):302–308, 2009.

[95] Guanhua Yan, Stephan Eidenbenz, and Emanuele Galli. Sms-watchdog: Profiling social behaviors of sms users for anomaly detection. In *Recent Advances in Intrusion Detection*, pages 202–223. Springer, 2009.

[96] Qian Xu, Evan Wei Xiang, Qiang Yang, Jiachun Du, and Jieping Zhong. Sms spam detection using noncontent features. *IEEE Intelligent Systems*, 27(6):44–51, 2012.

[97] Pavel Laskov, Patrick Düssel, Christin Schäfer, and Konrad Rieck. Learning intrusion detection: supervised or unsupervised? In *Image Analysis and Processing–ICIAP 2005*, pages 50–57. Springer, 2005.

[98] Luis Guerra, Laura M McGarry, Víctor Robles, Concha Bielza, Pedro Larranaga, and Rafael Yuste. Comparison between supervised and unsupervised classifications of neuronal cell types: a case study. *Developmental neurobiology*, 71(1):71–82, 2011.

[99] Rich Caruana, Nikos Karampatziakis, and Ainur Yessenalina. An empirical evaluation of supervised learning in high dimensions. In *Proceedings of the 25th international conference on Machine learning*, pages 96–103. ACM, 2008.

[100] Rich Caruana and Alexandru Niculescu-Mizil. An empirical comparison of supervised learning algorithms. In *Proceedings of the 23rd international conference on Machine learning*, pages 161–168. ACM, 2006.

[101] Jehad Ali, Rehanullah Khan, Nasir Ahmad, and Imran Maqsood. Random forests and decision trees. *IJCSI International Journal of Computer Science Issues*, 9(5), 2012.

[102] Lior Rokach and Oded Maimon. *Data mining with decision trees: theory and applications*. World scientific, 2014.

[103] Saul B Gelfand, C S Ravishankar, and Edward I Delp. An iterative growing and pruning algorithm for classification tree design. In *Systems, Man and Cybernetics, 1989. Conference Proceedings., IEEE International Conference on*, pages 818–823. IEEE, 1989.

[104] Ville Hautamäki, Tomi Kinnunen, and Pasi Fränti. Text-independent speaker recognition using graph matching. *Pattern Recognition Letters*, 29(9):1427–1432, 2008.

[105] Masaki Hilaga, Yoshihisa Shinagawa, Taku Kohmura, and Tosiyasu L Kunii. Topology matching for fully automatic similarity estimation of 3D shapes. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 203–212. ACM, 2001.

[106] Tony Tung and Francis Schmitt. The Augmented Multiresolution Reeb Graph Approach for Content-based Retrieval of 3d Shapes. *International Journal of Shape Modeling*, 11(1):91–120, 2005.

[107] Athanasios Mademlis, Petros Daras, Apostolos Axenopoulos, Dimitrios Tzovaras, and Michael G Strintzis. Combining topological and geometrical features for global and partial 3-D shape retrieval. *Multimedia, IEEE Transactions on*, 10(5):819–831, 2008.

[108] Xinbo Gao, Bing Xiao, Dacheng Tao, and Xuelong Li. A survey of graph edit distance. *Pattern Analysis and applications*, 13(1):113–129, 2010.

[109] Jozef J Zwislocki. Stevens' Power Law. *Sensory Neuroscience: Four Laws of Psychophysics*, pages 1–80, 2009.

[110] Cynthia A Brewer. Cartography: thematic map design. *Cartographic Perspectives*, (17):26–27, 1994.

[111] Thomas M J Fruchterman and Edward M Reingold. Graph drawing by force-directed placement. *Software: Practice and experience*, 21(11):1129–1164, 1991.

[112] Census. US Census Bureau (available at http://www.census.gov/), 2014.

[113] Dominik Maria Endres and Johannes E Schindelin. A new metric for probability distributions. *Information Theory, IEEE Transactions on*, 49(7):1858–1860, 2003.

[114] Weiwei Cui, Yingcai Wu, Shixia Liu, Furu Wei, Michelle X Zhou, and Huamin Qu. Context preserving dynamic word cloud visualization. In *Pacific Visualization Symposium (PacificVis), 2010 IEEE*, pages 121–128. IEEE, 2010.

[115] Ronald Fagin, Ravi Kumar, and D Sivakumar. Comparing top k lists. *SIAM Journal on Discrete Mathematics*, 17(1):134–160, 2003.

[116] Serkan Kiranyaz, Turker Ince, and Moncef Gabbouj. *Multidimensional particle swarm optimization for machine learning and pattern recognition*. Springer, 2014.

[117] Min Chen Min Chen and H Jaenicke. An information-theoretic framework for visualization. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1206–1215, 2010.

[118] Gerhard Venter. Review of optimization techniques. *Encyclopedia of aerospace engineering*, 2010.

[119] Eric Ziegel, William Press, Brian Flannery, Saul Teukolsky, and William Vetterling. *Numerical Recipes in C: The Art of Scientific Computing*, volume 29. Cambridge University Press, nov 1987.

[120] Tamara G Kolda, Robert Michael Lewis, and Virginia Torczon. Optimization by direct search: New perspectives on some classical and modern methods. *SIAM review*, 45(3):385–482, 2003.

[121] Vincent Granville, Mirko Krivánek, and J-P Rasson. Simulated annealing: A proof of convergence. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 16(6):652–656, 1994.

[122] Sanja Singer and Saša Singer. Complexity analysis of Nelder-Mead search iterations. In *Proceedings of the 1. Conference on Applied Mathematics and Computation, Dubrovnik, Croatia*, pages 185–196. PMF–Matemati{č}ki odjel, Zagreb, 1999.

[123] Stavros Papadopoulos, Konstantinos Moustakas, and Dimitrios Tzovaras. BGPViewer: Using Graph representations to explore BGP routing changes. In *Proceedings of the 18th International Conference on Digital Signal Processing (DSP)*, 2013.

[124] Mohit Lad, Dan Massey, and Lixia Zhang. Visualizing Internet routing changes. *IEEE transactions on visualization and computer graphics*, 12(6):1450–60, 2006.

[125] T. Wong, V. Jacobson, and C. Alaettinoglu. Internet Routing Anomaly Detection and Visualization. *2005 International Conference on Dependable Systems and Networks (DSN'05)*, pages 172–181, 2005.

[126] Michalis Faloutsos, Petros Faloutsos, and Christos Faloutsos. On power-law relationships of the internet topology. In *ACM SIGCOMM Computer Communication Review*, volume 29, pages 251–262. ACM, 1999.

[127] Jeroen B J Smeets and Eli Brenner. Grasping Weber's law., 2008.

[128] Hitesh Ballani, Paul Francis, and Xinyang Zhang. A study of prefix hijacking and interception in the internet. *ACM SIGCOMM Computer Communication Review*, 37(4):265, 2007.

[129] Emden R Gansner, Yehuda Koren, and Stephen C North. Topological fisheye views for visualizing large graphs. *IEEE transactions on visualization and computer graphics*, 11(4):457–68, 2005.

[130] R B Urquhart. Algorithms for computation of relative neighborhood graph. *Electronics Letters*, 16:556–557, 1980.

[131] M De Berg, O Cheong, M Van Kreveld, and M Overmars. *Computational Geometry: Algorithms and Applications*, volume 17. Springer, 2008.

[132] J W Jaromczyk and M Kowaluk. A note on relative neighborhood graphs. *Proceedings of the third annual symposium on Computational geometry SCG 87*, (V):233–241, 1987.

[133] Lixin Gao Lixin Gao. On inferring autonomous system relationships in the Internet. *IEEE/ACM Transactions on Networking*, 9(6):733–745, 2001.

[134] Xenofontas Dimitropoulos, Dmitri Krioukov, Marina Fomenkov, Bradley Huffaker, Young Hyun, George Riley, and Others. AS relationships: Inference and validation. *ACM SIGCOMM Computer Communication Review*, 37(1):29–40, 2007.

Appendix A

Alternative method for clutter reduction in one-dimensional datasets

Contents

A.1 Visual mapping function 137

A.2 Clutter reduction for one-dimensional mapping . 138

This section presents an alternative method for clutter reduction in one-dimensional visualizations. This method is similar to the approach proposed in Section 4, since they both reduce information loss and increase data insight. The method presented in this section, however, differs in the sense that it measures directly the information loss caused by the mapping procedure, and tries to minimize it. The two approaches can have similar results by adding a specific step on the hierarchical magnification approach. Specifically, the JSD metric is used in order to measure the difference between the distribution of the deformed one-dimensional space, and the uniform distribution. Since the uniform distribution has the maximum entropy, the smaller difference results in the smaller information loss. Thus, utilizing the JSD metric, it is possible to find the magnification factor that results in the lowest information loss amongst a set of alternative values.

This section starts with the presentation of the visual mapping function. Afterwards, it presents the research work performed with respect to one-dimensional mapping. In this case, the goal is to reduce the information loss

caused by the mapping procedure, so as to visualize as much information as possible and reduce visual clutter.

A.1 Visual mapping function

Generally, the mapping procedure in the visualization is performed by the mapping function: $F : \Re^n \rightarrow V^m$, where $\Re^n$ is the input information in the space of real numbers and has n features, and V^m is the visualization space (visual variables) that has m features. The first step towards the optimization of the visual mapping is the formal definition of the mapping function F, which is depicted in Figure A.1. From now on, the input dataset is considered the input signal, while the visualization result is considered the output signal.

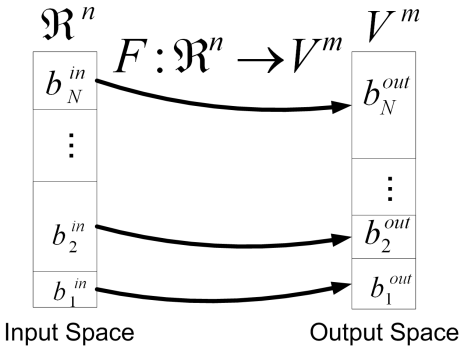


Figure A.1: The mapping function $F : \Re^n \rightarrow V^m$ that is used by every visualization approach. The input $\Re^n$ and the output V^m spaces, are separated into N bins and sections. Afterwards, each input bin b_i^{in} is mapped onto the corresponding output section b_i^{out}.

The mapping function F is formulated in such a way that it describes the majority of the visualization approaches. To begin with, the input space $\Re^n$ is partitioned into a set $B^{in} = \{b_i^{in} \mid i \in [1, N]\}$ of N bins, such that:

$$\bigcup b_i^{in} \equiv \Re^n \; and \; b_i^{in} \cap b_k^{in} = \emptyset \; \forall \; i \neq j, \; i, j \in [1, N] \tag{A.1}$$

where the superscript in represents the input space and the subscript i the index of the corresponding bin. In addition, the output space V^m is also separated into a set $B^{out} = \{b_i^{out} \mid i \in [1, N]\}$ of N sections, such that:

$$\bigcup b_i^{out} \equiv V^m \; and \; b_i^{out} \cap b_k^{out} = \emptyset \; \forall \; i \neq j, \; i,j \in [1,N] \qquad (A.2)$$

where the superscript *out* represents the output space.

The purpose of the mapping function F is to map each input bin b_i^{in} to the corresponding output section b_i^{out} for all $i \in [1,N]$. The number of bins N is a user defined variable that affects the degree of non-linearity of the mapping, as well as the amount of the information mapped onto the visual variables. The mapping procedure followed by the mapping function F is depicted in Figure A.1. It should be noted that using the proposed mapping function and changing the bin and section separations and the number of bins N, many well known mappings can be described, such as linear or logarithmic mapping. For instance, if the input and output signals are separated into two sets of equal sized bins and sections, the mapping procedure that is followed is linear.

The human perception system has different decoders for different visual channels. As a result the mapping of each input bin b_i^{in} to the corresponding output section b_i^{out}, must take into account the perceptual characteristics of the corresponding visual variable, and change accordingly.

A.2 Clutter reduction for one-dimensional mapping

In order to formulate the optimum mapping function $F : \Re^1 \rightarrow V^1$, Information Theory is used. More specifically, entropy measures are used to measure the information content of both the input information and the visualization result. The information content of the visualization result depends on two parameters: the mapping function F and the display size [117]. For a constant display size, the optimum mapping function F is considered the one that given a specific space partitioning into N bins, maximizes the information content of the visualization, and thus minimizes its information loss. It should be noted that due to the limitation of the space partitioning, the maximization of the output entropy does not necessarily mean that the entropy has its global maximum value. The value of the maximum output entropy depends on the number of bins N, in which case the higher the value of N the more close is the maximum entropy to its global maximum.

In order to measure the amount of information content, Shannon's entropy is used:

$$H(X) = -\sum_{i=1}^{t} p_i \log(p_i) \tag{A.3}$$

where $X = \{x_i \mid i \in [0,t]\}$ is a random variable and p_i is the probability of the random variable X to take the value x_i. In the case that the random variable X represents the values of the input dataset $X \equiv r^1 \in \Re^1$, then Shannon's entropy measures the information content of the input signal, while in the case that the random variable X represents visual variables $X \equiv v^1 \in V^1$, then the information content of the output signal is measured instead. It should be emphasized that $H\left(v^1\right)$ should not be confused with the visualization capacity, and is just a measure of the information currently being visualized.

Using Shannon's entropy, the information loss that is caused by the visual mapping procedure is defined as:

$$H^{loss}\left(v^1\right) = \frac{H\left(r^1\right) - H\left(v^1\right)}{H\left(r^1\right)} \tag{A.4}$$

where $r^1 \in \Re^1$ represents the input signal and $v^1 \in V^1$ the output signal. Finally, as it was explained earlier, the optimum mapping function is considered the one that given a specific space partitioning into N bins, minimizes the information loss or in other words (since $H\left(\Re^1\right) = const$) maximizes the information content of the visualization. As a result, the optimization procedure is the process of defining the appropriate bin and section sizes of the input and the output signals given a static value of N, so as to satisfy the following equations:

$$\left[B^{in}, B^{out}\right] = \arg \min_{[B^{in},B^{out}]} \left(H^{loss}\left(v^1\right)\right) \tag{A.5}$$

or:

$$\left[B^{in}, B^{out}\right] = \arg \max_{[B^{in},B^{out}]} \left(H\left(v^1\right)\right) \tag{A.6}$$

As it was earlier explained, the value of the maximum visualization entropy depends on the number of bins N selected by the user. The higher the value of N, the higher the output entropy, up to the point that it reaches its global maximum. This effect of increasing the entropy, has as a result

the reduction of cluttering, as well as better space utilization by the visualization. On the other hand, higher entropy also introduces a non-linear space distortion which makes it difficult for the user to decode the visualized information. As a compromise between these opposite goals, the user has the choice to interactively change the number of bins N according to the displayed visualization result and the detected patterns.

The reason behind the increase of the visual information, lays in the fact that large information loss could cause the analyst to misunderstand the magnitude of the patterns that are visible, or not be able to detect any patterns at all. On the other hand by reducing the information loss, more information is presented to the analyst, enabling him/her to take more informed decisions, or just focus on interesting parts of the visual display (by means of filtering or zooming).

A.2.1 Optimization Procedure

This sections discusses issues related to the optimization procedure. Specifically, it addresses issues related to the selection of the appropriate optimization methodology to minimize the information loss defined in equation (A.5), the complexity of the optimization procedure, and the selection of the appropriate number of bins N.

Optimization Methods Comparison

Equation (A.5) defines an optimization problem, and the selection of the optimization procedure is very important in order to acquire solutions close to the global maximum of the objective function. There have been proposed a great variety of optimization techniques in the literature [118], based on different approaches. For experimentation though, four of them have been selected and compared in the context of this research, namely: Gradient Descent [119], Downhill Simplex [120], Compass Search [120], and Simulated Annealing [121].

Gradient Descent calculates the direction of the gradient in each iteration and progresses with small steps towards it. The Downhill Simplex method, also known as NelderMead method, makes use of simplex changes in the N-dimensional space of optimization, and simple rules to change the worst nodes of the simplex towards better ones. Compass Search estimates

the value of the objective function in the neighborhood of the current point, by using a set of predefined search directions and search steps, and progresses towards the best point. If no best point is found, the search step decreases. Finally, the Simulated Annealing searches randomly around the current point for better solutions and progresses towards them iteratively. The advantage of this method is that it allows some probability of progression towards a worse point, so as to avoid getting stuck in local minima. This probability is decreasing as the number of iterations increases.

These four optimization methods were compared with respect to their performance in finding a good solution of the objective function. Multiple input distributions were selected and the results of each of the optimization methods were collected and compared. The input distributions selected are: Normal, Exponential, Mixture of 20 Normal Distributions, and the BGP IP change distribution. The last distribution is the input distribution for the BGPGraph approach described in Appendix B, and as shown in figure A.2 follows a distribution that resembles the Cauchy distribution.

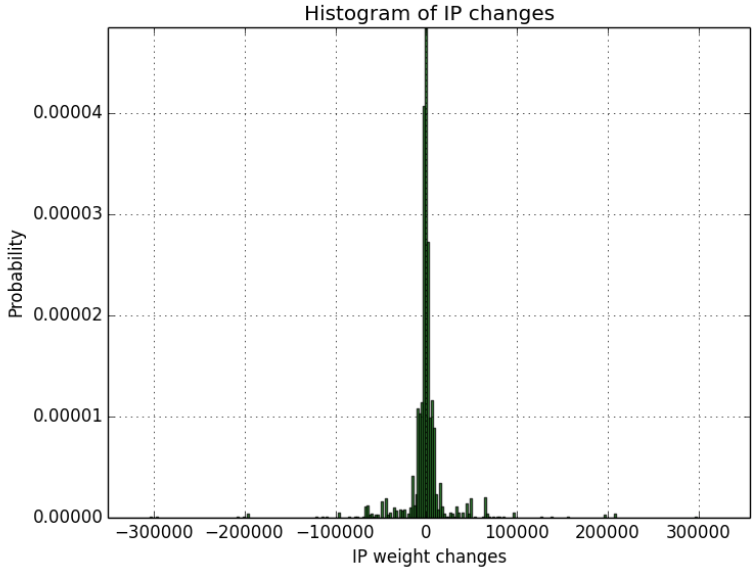


Figure A.2: A histogram illustrating the distribution of the BGP IP changes caused by the BGP announcements, and utilized by the proposed BGPGraph approach (Appendix B) for visualizing BGP routing changes.

In the case of two bins ($N = 2$), the optimization space has been ex-

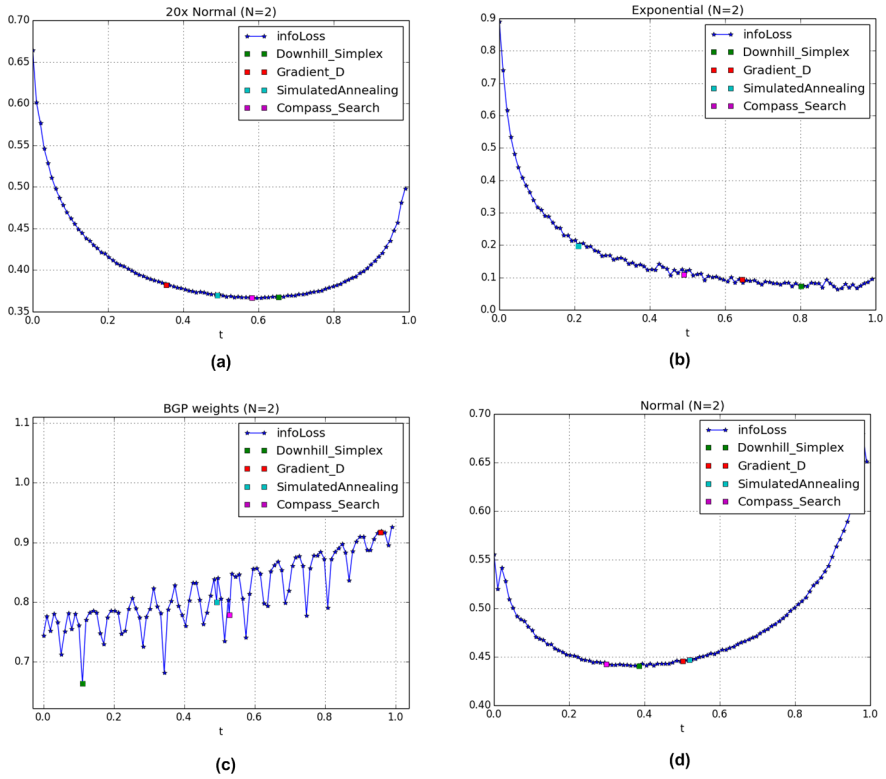


Figure A.3: The value of the objective function of information loss for various output bin partitions (for $N = 2$ bins) controlled by the value of t defined in equation (A.9). The result of the optimization procedure for each of the optimization methods is also illustrated in this figure.

haustively searched to detect its main characteristics. Thus, in the case of the output space bin separation, and considering the input space separation into two bins static:

$$B^{out} = \{b_1^{out},\ b_2^{out}\} \tag{A.7}$$

and normalizing the space to $[0, 1]$:

$$b_1^{out} + b_2^{out} = 1 \tag{A.8}$$

by setting $b_1^{out} = t$, we have that $b_2^{out} = 1 - t$. Thus, adjusting the value of t the optimization space is traversed and the value of the objective function is measured.

$$b_1^{out} = t \; and \; b_2^{out} = 1 - t \tag{A.9}$$

The result of this procedure, which is illustrated in figure A.3 for each of the four distributions, revealed the existence of multiple local minima in the objective function of equation (A.5). This figure also depicts the results of the four optimization methods applied into each of the four distributions respectively. It is apparent that the Downhill Simplex method achieves better results in most of the distributions.

A more thorough comparison of the optimization methods for various number of bins N is illustrated in figure A.4. It is apparent that in most of the cases the Downhill Simplex method, outperforms all the other methods and finds a minima closer to the global one.

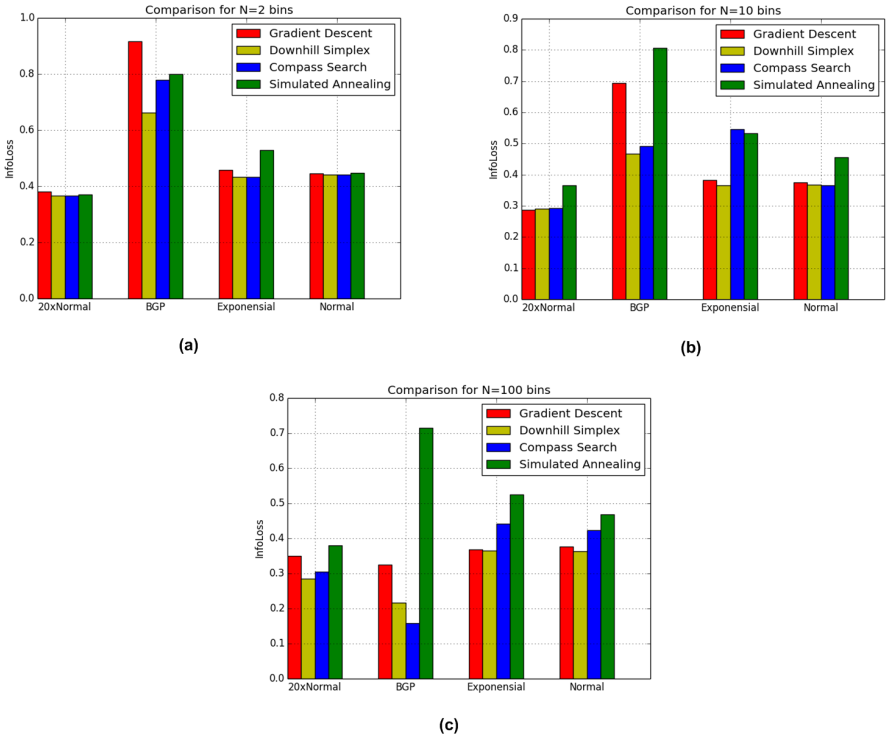


Figure A.4: Comparison of the optimization methods for various number of bins N, and for various input distributions. The Downhill Simplex method outperforms all the other methods in most of the cases.

Complexity of the Downhill Simplex Method of Optimization

Since the Downhill Simplex method of optimization will be utilized for the minimization of entropy loss defined in equation (A.5), it would be useful to find its complexity. However, due to the lack of convergence theory [122] that would allow an approximate definition of the number of iterations required, this calculation is very difficult.

In this context, the worst case complexity of the Downhill Simplex method was experimentally defined. The algorithm was executed multiple times for multiple stopping parameters and multiple dimensions (number of bins N). The stopping parameter used is e, and defines the smallest distance moved in each iteration so that it is fractionally larger than the tolerance e. In the case that the distance moved is smaller than e, the algorithm stops.

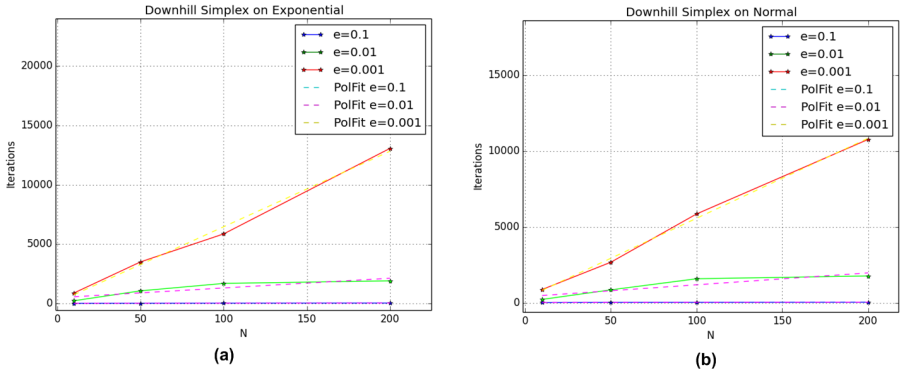


Figure A.5: The relationship of the number of iterations of the Downhill Simplex method and the number of bins N. The dotted lines represent line fits for the respective point. The parameter e is used for stopping the algorithm and represents the smallest step tolerance.

Figure A.5 illustrates the relationship of the number of iterations and the number of bins N, for the exponential and normal distributions. Similar results are found in the other distributions. This figure shows that the number of total iterations for the Downhill Simplex method to terminate, is linearly depended on the value of bins N. Furthermore, the gradient of the line is larger for smaller tolerance parameters e, but still linearly dependant on N.

For each step, the worst case complexity of the Downhill Simplex method is $O(N)$. As a result, combining this fact with the result, the number of iter-

ations is linearly dependant on N, the worst case complexity of the Downhill Simplex method is experimentally found to be $O(N^2)$.

Selection of the Number of bins N

One important aspect of the proposed approach, is the selection of the number of bins N. The answer to this question is very difficult since it depends on multiple parameters that exist in the interaction of the user and the visualization, such as perception issues, specific task, input dataset characteristics (e.g. distribution and size) and the quantization levels of the target visual variable. Thus, a different approach is proposed for the definition of the number of bins N.

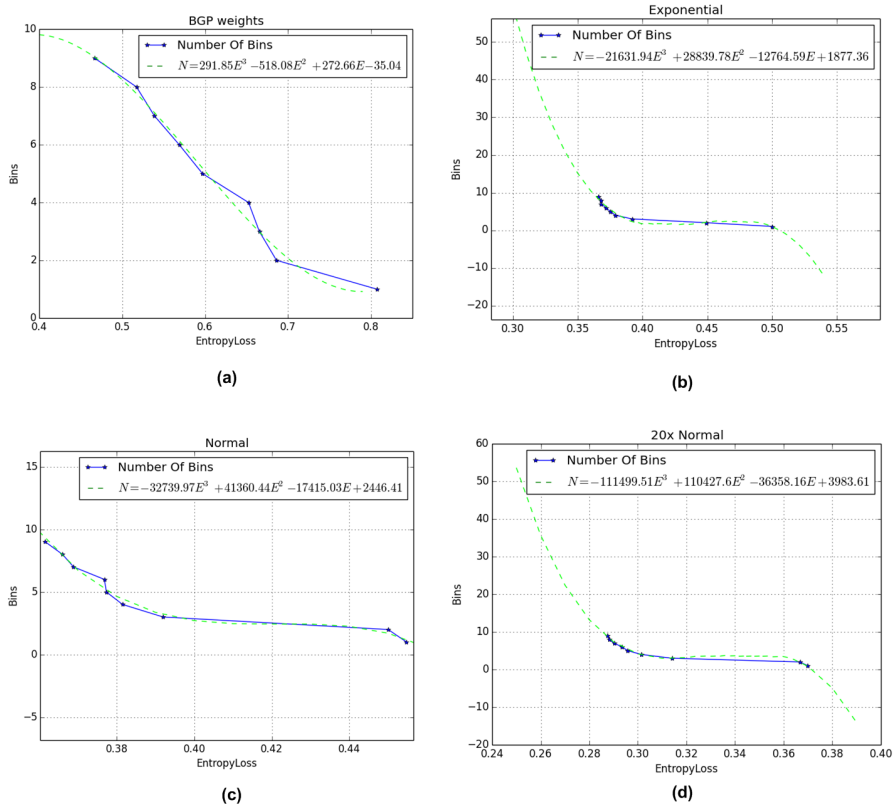


Figure A.6: The relationship of the number of bins and the information loss for four input distributions. Polynomial regression (dot lines) is utilized to find an analytical function describing this relationship.

Specifically, fixing the input dataset characteristics and the quantization

levels of the target visual variable, the user is allowed to select an acceptable information loss level based on the task at hand, as well as the perception of the current visualization. Fixing all these parameters, it is possible to find the optimum number of bins, so as to accommodate the needs of the user. To achieve this result, polynomial regression is utilized, in order to define the relationship of information loss to the number of bins N. This relationship is illustrated in figure A.6 for each of the four test distributions. Furthermore, this figure also illustrates the exact polynomial function found for each of these cases.

Appendix B

BGPGraph: Hierarchical Visualization of BGP Routing Changes

Contents

B.1 Introduction 148
B.2 Graph Creation 148
B.3 Clustering of the AS-Graph 151
B.4 Visualizing AS relationships using edge direction 155
B.5 Interaction with the Graph Hierarchy 158
B.6 Quantification of the information 159
B.7 Applications of the information quantification . . 162

This Section presents BGPGraph [1] [123], a scheme for visualizing Internet Routing changes by utilizing a hierarchical graph visualization scheme. The introduced graph metaphor represents the Autonomous System (AS) topology, while additional intortion is mapped onto the visual features of the graph (color and size). Furthermore, the concept of entropy plot is introduced, for the purpose of providing an overview of the BGP activity over a large period of time and help the analyst detect interesting time windows to focus on. Experimental demonstration in state-of-the-art BGP events, illustrates the flexibility of the proposed method and the significant analytics potential of BGPGraph.

B.1 Introduction

Most of the aforementioned approaches provide a high level view of interactions between autonomous systems due to the highly dynamic nature of the underlying information. Moreover, since the overlaying logic is of higher importance than the visualization itself, simpler visualization techniques such as node link graphs have been used [124] [125]. The proposed framework is developed to visualize global Internet routing changes and help in the visual analysis of BGP routing anomalies. However, the problem is the size of the BGP update messages and the network graph. In the year 2012 the Internet had more that $40,000$ networks (Internet autonomous systems) and over $53,000$ physical links between them. The proposed approach aims to tackle this problem by providing a hierarchical framework for the visualization of the BGP routing changes, allowing therefore the use of high level visualizations in the first steps of the analysis, thus eliminating visual clutter. Moreover, the analyst is capable of seeing in more detail specific parts of the graph that are of particular interest. In addition, the present framework aggregates the number of IPs used by each link so as to deal with the vast number of BGP update messages.

B.2 Graph Creation

The AS-graph is a graph $G(V, E)$ whose set of vertices $V = \{v_i | i \in [1, N]\}$, where N is the total number of ASes, is the set of all the ASes and each vertex represents an AS. The set of edges is $E = \{e_i(v_j, v_k) |\ v_j, v_k \in V\}$, where each edge represents a physical connection between the ASes. The graph is created from the AS-paths of the BGP announcements. For example the announcements:

<prefix>	: <AS-path>
<91.194.12.0/21>	: <3549 1 2 3>
<29.194.12.0/24>	: <3549 5 1 2 4>

were initiated by AS-3 and AS-4 and the monitoring point is AS-3549. These announcements create the graph of Figure B.2(a). Moreover, the set of vertices in this case is $V = \{3549, 1, 2, 3, 4, 5\}$ and the set of Edges is $E =$

$\{(3549, 1), (1, 2), ..., (2, 4)\}$.

Despite the apparent randomness of the AS-graph, there are power-laws that describe its topology. Faloutsos et al. [126] described three power-laws that govern its topology: 1) rank exponent R, 2) outdegree exponent, and 3) eigen exponent. The rank exponent law states that the outdegree d_u of a node (AS) u, is proportional to the rank of node r_u to the power of a constant R:

$$d_u \propto {r_u}^R \tag{B.1}$$

The outdegree exponent law states that the number of nodes f_d with oudegree d, is proportional to the oudegree d to the power of a constant O:

$$f_d \propto d^O \tag{B.2}$$

The final law, eigen exponent, states that the eigen values λ_i of the graph, ordered in decreasing sequence, are proportional to the order i to the power of a constant E:

$$\lambda_i \propto i^E \tag{B.3}$$

The constants R, O, and E uniquely identify the AS-topology and are different in other graph topologies [126].

In addition to the existence of power-laws, it should be noted that the AS-graph has specific characteristics that are apparent in the topology and are unique to this type of graph. Specifically, as it is shown in figure B.4(a) and B.4(b), the graph structure is dominated by a highly interconnected core network, where the higher tier ASes lay, and many leaf nodes that represent the majority of the customer ASes that consume transit services.

In order to construct a meaningful graph, there is a need to calculate the weight values of the vertices and the edges that will be mapped to the visual features of the visualization system. In the present case, the weights represent number of IP addresses. Figure B.1 shows the procedure of this computation. At time T_i, AS-1 owns $m_i^1 = 150$ IPs, AS-2 owns $m_i^3 = 100$ IPs and the edge between them is used by $m_i^2 = 150$ IPs. After some time $\Delta T = T_k - T_i$, these values become $m_k^1 = 200$, $m_k^2 = 200$ and $m_k^3 = 90$. This difference is used to provide the weights of the edges and the vertices since it captures the dynamics of the BGP updates: $weight_1 = \Delta m^1 =$

50, $weight_2 = \Delta m^2 = 50$ and $weight_3 = \Delta m^3 = -10$. In addition, the positive weights are represented with green color and the negative with red. Moreover, the magnitude of the corresponding weight is mapped to the width of the edges or the radius of the vertices, which are represented by two sets, one that defines the quantization levels of the vertex radii and another that defines the quantization levels of the edge widths:

$$\begin{aligned} R = \{r_i | r_{min} \leq r_i \leq r_{max},\ i \in [0, L_r] \ and \\ i \in \aleph\} r_i \neq r_j, for\ i \neq j,\ r_0 = 0\ and\ r_i < r_{i+1} \end{aligned} \tag{B.4}$$

$$\begin{aligned} W = \{w_i | w_{min} \leq w_i \leq w_{max},\ i \in [0, L_w] \ and \\ i \in \aleph\} w_i \neq w_j, for\ i \neq j,\ w_0 = 0\ and\ w_i < w_{i+1} \end{aligned} \tag{B.5}$$

where r_i is the radius of the i_{th} circle in visual degrees, r_{min} is the minimum radius and r_{max} is the maximum radius. L_r is the number of different radius values or alternatively the radius quantization levels. The same notation is used for the edges width set W. Furthermore, it is assumed that the sets are ordered, which is expressed through the relation: $r_i < r_{i+1}$. To define the sets R and W, the visual acuity of the Human Vision System (HVS) needs initially to be measured and quantified. There are many studies and metrics on the acuity of the HVS [52], but the most practical one is the Weber's Law or just-noticeable difference (JND) [127], which is the one utilized here.

Figure B.2 shows a simple example of how the visualizations are generated. At time T_0 the AS-graph is empty (Figure B.2(a)). All the ASes own 0 IPs and all the edges are used by 0 IPs. After some time $\Delta T = T_i - T_0$ (Figure B.2(b)) two announcements are made, that change the corresponding weights of the AS-graph. For example, the prefix <91.194.12.0/21> of the first announcement, consists of $2^{32-21} = 2^{11} = 2048$ IPs, thus the corresponding weight is set to $2048 - 0 = 2048$ IPs. In order to reduce the artificial escalation of the feature space, the visualization system utilizes the square root of the weight instead of the actual weight. In this case $\sqrt{2048} \approx 45$. Again, after some time $\Delta T = T_k - T_i$ one more announcement is made by AS-4 (Figure B.2(c)). In this case AS-4 hijacks part of AS-3 prefix. This is shown in the visualization by positive weight (green color) in the path <3549 5 1 2 4> and negative weight (red color) in the path <3549 1 2 3>. The orange color represents the monitoring point.

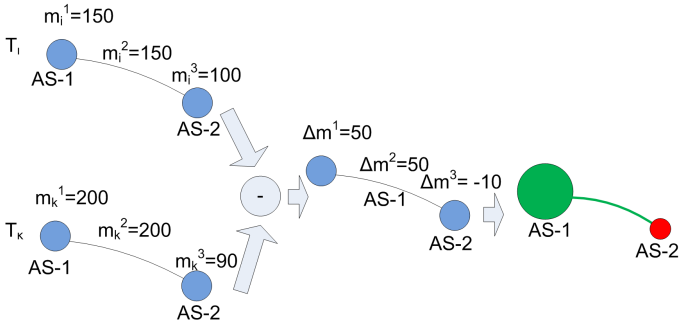


Figure B.1: Computation of the edges and vertices weights. At time T_i, AS-1 owns $m_i^1 = 150$ IPs, AS-2 owns $m_i^3 = 100$ IPs and the edge between them is used by $m_i^2 = 150$ IPs. After some time $\Delta T = T_k - T_i$, these values change and become $m_k^1 = 200$, $m_k^2 = 200$ and $m_k^3 = 90$. This difference constitutes the weights of the edges and the vertices: $weight_1 = \Delta m^1 = 50$, $weight_2 = \Delta m^2 = 50$ and $weight_3 = \Delta m^3 = -10$. The magnitude of the corresponding weight is mapped to the radius of the vertex or the width of edge. Green color represents positive and red color negative weight values.

The reason for using the difference of IPs as weights of the edges and the vertices, lies in the nature of the Internet routing protocol. When a source AS sends information to a destination AS, this information follows a path consisting of ASes and links. As it was mentioned earlier, if a link or an AS fails (for various reasons such as misconfiguration, hardware problem etc), the Internet is built in such a way that the router searches for another path to reach the destination. Thus there is a transfer of traffic from one path to another. Another example of traffic transfer is the case of BGP hijacks. In general there are two types of BGP hijacks [128]: hijack of prefix ownership and AS-path hijack. In both these cases there is a transfer of traffic from one path to the other. This transition of traffic is captured well using the proposed IPs difference metrics. Thus the visualization shows these events in a practical and meaningful way.

B.3 Clustering of the AS-Graph

Since the AS-Graph is too large to be visualized at once, a clustering method is utilized to produce a hierarchy of coarse to fine graphs that are easier to visualize and perceive in a hierarchical manner. Each cluster represents a

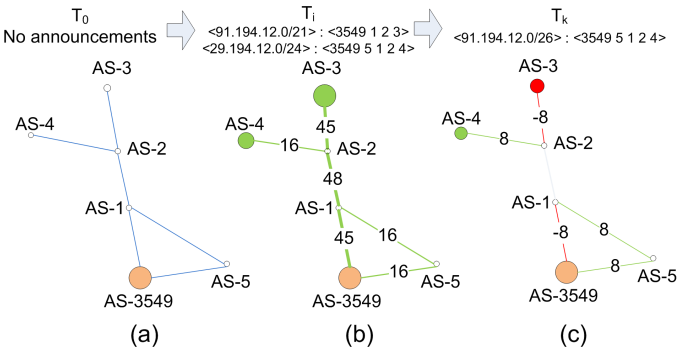


Figure B.2: Example announcements. (a) This is an empty graph. All the ASes own 0 IPs and the edges are used by 0 IPs. (b) Two announcements are made. The weights of the edges that are shown are the square roots of the actual weights $\sqrt{Weight}$. (c) AS-4 hijacks part of AS-3 prefix. This is shown by positive weight (green color) in the path <3549 5 1 2 4> and negative weight (red color) in the path <3549 1 2 3>.

collection of ASes. The proposed hierarchical clustering is based on the work of [129]. First of all, the position of the AS-Graph vertices is calculated using a force-directed algorithm.

The notation $G(V, E)$ represents the AS-Graph which is defined in Section B.2. Furthermore, the notation $G^l\left(V^l, E^l\right)$ represents the graph of the l level of the hierarchy. The superscript is used in general to represent the level. V^l is the set of vertices and E^l is the set of edges of the l level. In this case $G^0\left(V^0, E^0\right) \equiv G(V, E)$. In addition, the notation v_j^l, represents the j_{th} vertex of level l ($v_j^l \in V^l$), which is also a cluster of ASes ($v_j^l \subseteq V$).

To compute the clusters of the $(l+1)$ level of the hierarchy, the set of candidate vertex pairs S^l, is constructed for the graph $G^l\left(V^l, E^l\right)$. To populate the set S^l a proximity graph is initially constructed, which is simply a graph, where two vertices are connected by an edge if and only if the vertices satisfy particular geometric requirements. Different proximity graphs can be formulated with respect to different metrics, but the Euclidean metric is used most frequently and is also adopted in the proposed method. The notation $P^l(V_p^l, E_p^l)$ is used to represent the proximity graph of graph $G^l\left(V^l, E^l\right)$. The proximity graph used is the Urquhart graph [130], which is easily constructed by removing the longest edge from each triangle in the Delaunay triangulation [131]. Furthermore, the Urquhart graph is a good approxima-

tion of the relative neighborhood graph [132]. Two vertices $v_i^l, v_j^l \in V^l$, are neighbors in the relative neighborhood graph if and only if, for every other vertex $v_k^l \in V^l$:

$$\begin{aligned} \|v_{i-pos}^l - v_{j-pos}^l\| \leq \max\{ \|v_{i-pos}^l - v_{k-pos}^l\|, \\ \|v_{j-pos}^l - v_{k-pos}^l\| \}, \forall v_k^l \in V^l, \ and \ i \neq j \end{aligned} \tag{B.6}$$

where $v_{i-pos}^l \in \Re^2$ is the position of vertex $v_i^l \in V^l$.
After the proximity graph $P^l(V_p^l, E_p^l)$ is calculated, the set of candidate pairs S^l is defined:

$$\begin{aligned} S^l = \{s_i^l | s_i^l = \{v_k^l, v_j^l\} \ and \ v_k^l, v_j^l \in V^l\} \\ iff \ \exists e_u^l(v_k^l, v_j^l) \in E_p^l \end{aligned} \tag{B.7}$$

where $e_p^l(v_k^l, v_j^l)$ are edges from vertex v_k^l to vertex v_j^l in the Urquhart graph.

After constructing the set of candidate pairs S^l, the set of disjoint pairs $\widetilde{S}_{dis}^l \subseteq S^l$ is defined, where all pairs of elements $s_i^l, s_j^l \in \widetilde{S}_{dis}^l$, do not have a vertex in common: $s_i^l \cap s_j^l = \emptyset$.

$$\begin{aligned} \widetilde{S}_{dis}^l \subseteq S^l, such \ that \ s_i^l \cap s_j^l = \emptyset, \\ \forall s_i^l, s_j^l \in \widetilde{S}_{dis}^l, \ i \neq j \end{aligned} \tag{B.8}$$

Set $\widetilde{S}_{dis}^l$ is iteratively constructed and for every vertex v_i^l, the sets of candidate pairs S^l including it, are evaluated. The vertex is considered as a pair with the vertex that maximizes the weighted sum of the following measures:

1. Geometric proximity: $\frac{1}{\|v_{i-pos}^l - v_{j-pos}^l\|}$
2. Similarity of neighborhood: $\frac{|N_i^l \cap N_j^l|}{|N_i^l \cup N_j^l|}$
3. Degree: $\frac{1}{\deg_i^l * \deg_j^l}$

where N_i^l is the set of vertices that have a common edge with $v_i^l \in V^l$ and represents the neighborhood of vertex v_i^l. Also, deg_i^l is the degree of vertex v_i^l and is defined by $deg_i^l = |N_i^l|$. The measures are normalized to the range

$[0, 1]$. The vertices that are unmatched at the end of the process are added to the set of disjoint pairs $\widetilde{S}^l_{dis}$ as single elements:

$$S^l_{dis} = \widetilde{S}^l_{dis} \cup \left(V^l - \widetilde{S}^l_{dis}\right) \tag{B.9}$$

As a result each element $s^l_i \in S^l_{dis}$ represents a cluster of ASes: $s^l_i \subseteq V^0 \equiv V$.

Afterwards, from the elements of disjoint pairs set S^l_{dis}, the set of clusters of the level $(l+1)$ are created. To construct the graph $G^{l+1}\left(V^{l+1}, E^{l+1}\right)$ the following sets are defined:

- $V^{l+1} = \{v^{l+1}_i | \, v^{l+1}_i = s^l_i, \, \forall s^l_i \in S^l_{dis}\}$
- $E^{l+1} = \{e^{l+1}_j(v^{l+1}_k, v^{l+1}_p) | \, \forall e^0_h(v^0_c, v^0_x), \, v^0_c \in v^{l+1}_k \; and \; v^0_x \in v^{l+1}_p\}$

This procedure is repeated for each level. The position of a vertex $v^l_u \in V^l$ is computed by finding the average position of the AS nodes that it is comprised of in the original graph $G\,(V, E)$:

$$v^l_{u-pos} = \frac{1}{n} \sum_{i=1}^{n} v^0_{i-pos}, \forall v^0_i \in v^l_u \in V^l \tag{B.10}$$

where $v^l_{u-pos} \in \Re^2$ is the position of the vertex $v^l_u \subseteq V$, $v^0_{i-pos} \in \Re^2$ is the position of the vertex $v^0_i \in V$, and n is the number of ASes that vertex v^l_u contains.

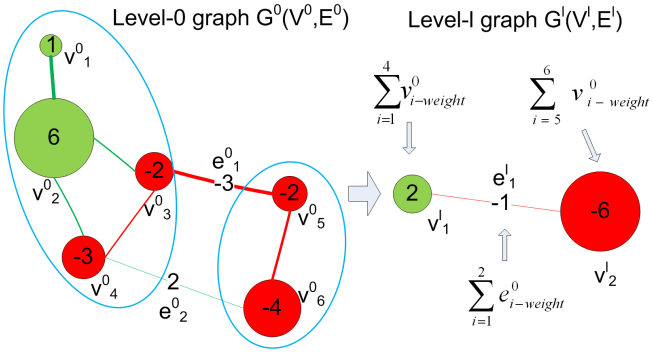


Figure B.3: The mapping of the weights to the Graph Hierarchy. The numbers that are shown represent weights and the blue ellipsoids represent clusters. The mapping is calculated using equations (B.11) and (B.12).

Since what is visualized are the edge and vertex weights derived from the BGP announcements, the following equations are used to provide vertex

and edge weighting for all the levels of the hierarchy:

$$v^l_{u-weight} = \sum_i v^0_{i-weight}, \forall\, v^0_i \in v^l_u \subseteq V^0 \tag{B.11}$$

$$\begin{aligned} e^l_p\Big(v^l_h, v^l_j\Big)_{weight} = \sum_i \sum_k e^0_m\big(v^0_i, v^0_k\big)_{weight} \\ , \forall v^0_i \in v^l_h \, and \, \forall v^0_k \in v^l_j \end{aligned} \tag{B.12}$$

where $v^l_{u-weight}$ is the weight of the vertex $v^l_u \in V^l$, $e^l_p\Big(v^l_h, v^l_j\Big)_{weight}$ is the weight of the edge $e^l_p \in E^l$ that connects vertex $v^l_h \in V^l$ to vertex $v^l_h \in V^l$, and $e^0_m\big(v^0_i, v^0_k\big)_{weight}$ is the weight of the edge $e^0_m \in E^0$ from the vertex $v^0_i \in v^l_h \subseteq V^0$ to the vertex $v^0_k \in v^l_j \subseteq V^0$. Figure B.3 depicts this procedure. The blue ellipsoids represent the clusters and the numbers represent weights.

The problem with the presented clustering approach is that information is lost at the upper levels of the hierarchy. This lost information has two aspects: the first aspect, is the weights of the edges that are lost, and the second is the position of the nodes that changes continuously. The weights of the edges represents the actual information that is visualized, while the position of the nodes is secondary information that only helps in the preservation of the user's mental image of the graph, as he traverses through the levels of the clustering hierarchy. Equation (B.10) tries to minimize the secondary information loss by positioning the clusters on the average position of all the ASes that the corresponding cluster is comprised of. Another aspect of the clustering that aims at the minimization of the secondary information loss is the use of the Urquhart proximity graph which suggests vertex clusters according to their positional proximity (equation (B.7)). On the other hand, the primary information loss is minimized using the approach presented in Section B.7.1, with the help of the entropy measures developed in Section B.6.

The results of the application of the clustering algorithm on the AS-Graph of 2005 are shown in Figure B.4.

B.4 Visualizing AS relationships using edge direction

BGP allows the ASes to select routes and propagate reachability information to others. These selections are governed by are constrained by the contrac-

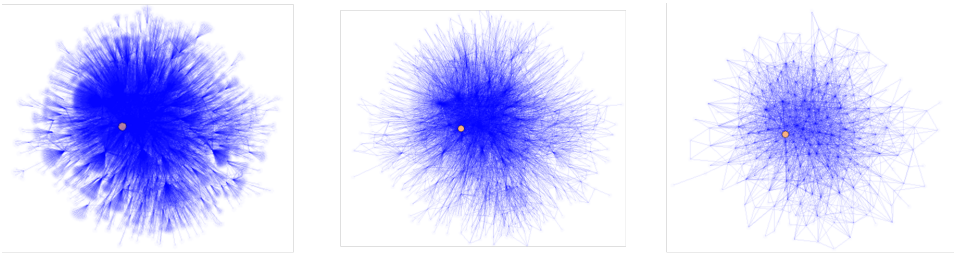


Figure B.4: Cluster Hierarchy of the AS-Graph. As it is shown the number of vertices is reduced as the level increases, but the topology of the original graph (a), is preserved. The monitoring point is the orange node.

tual commercial agreements between them. For example, an AS does not provide transit services between its provider ASes. These policies define the relationships that exist between two ASes and are an important aspect of Internet structure.

Multiple approaches have been proposed for the discovery of these relationships [133] [134], but the first approach was proposed by Gao Lixin [133]. This method infer AS relationships from BGP routing tables. Gao defined three types of AS relationships:

1. Provider-to-Costumer:A Provider offers transit connectivity to IP traffic of its Customers.

2. Peer-to-Peer: Two ASes can establish an agreement for mutual exchange of traffic on a quid pro quo basis. The involved peers forward to each other only traffic regarding either themselves or their customers

3. Sibling-to-Sibling: Two ASes that administratively belong to the same organization can be connected through a direct link. Two siblings exchange only IP traffic of their own origin.

It should be emphasized that only the Costumer-to-Provider relationships is directed [133].

Utilized the algorithm proposed by Gao, the AS relationships can be identified and visualized in the BGPGraph approach. Thus, the AS-Graph directed, and the direction of the edges represents Provider-to-Costumer relationships. These relationships are visualized using directed edges in the

graph, while the rest of the relationships are visualized using undirected edges. An example of directed and undirected edges is illustrated in figure B.5.

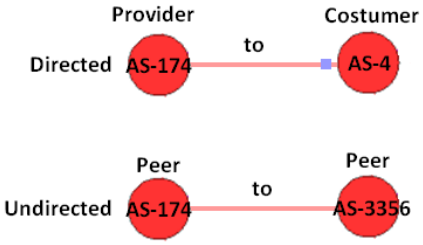


Figure B.5: Visualizing the AS relationships using directed and undirected edges. Directed edges illustrate Provider-to-Costumer relationships, and undirected edges illustrate Peer-to-Peer and Sibling-to-Sibling relationships.

The aforementioned relationships are characteristics of the AS graph, i.e. level-0 of the clustering hierarchy, in which each edge connects two ASes. At higher levels of the clustering hierarchy, the nodes of the graph represent clusters of ASes and the edges connections between ASes in different clusters. Thus, edge direction between clusters is not directly defined. In order to solve this issue, a simple approach is proposed. Specifically, since each edge of level-l greater than level-0 is a combination of multiple level-0 edges, a majority vote is proposed in order to calculate the direction of the corresponding edge. In detail:

$$R\left(e_p^l\left(v_h^l, v_j^l\right)\right) = \arg\max_t N\left(t, [v_i^0, v_k^0]\right), \ where\ v_i^0 \in v_h^l\,,\ v_k^0 \in v_j^l, \\ and\ t \in [CP, PC, PP, SS] \tag{B.13}$$

where $R(e)$ is the relationship the edge e represents, $[v_i^0, v_k^0]$ is the set of all edges from cluster v_h^l to cluster v_j^l, t is the type of relationship, i.e. CP=Customer- Provider, PC=Provider- Customer, PP=Peer- Peer, and SS=Sibling- Sibling. Furthermore, l represents level of clustering hierarchy, and $N(t, [v_i^0, v_k^0])$ returns the number of edges from the input set that have type t.

An example of the proposed directed graph visualization is presented later, in figure 5.20.

B.5 Interaction with the Graph Hierarchy

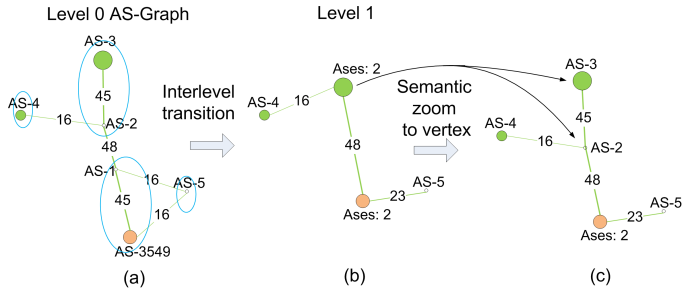


Figure B.6: Interaction methods with the Graph hierarchy. The vertices inside of the blue ellipsoids represent the set of disjoint pairs S^0_{dis}. The orange color represents the monitoring point.

In order to interact with the Graph hierarchy several methods are provided to the user. Some basic graph interaction methods are included, such as brushing, zooming or moving of the vertices. But the most important interaction techniques are:

- "Interlevel transition"
- "Semantic zoom to a vertex"

Both these methods are shown in Figure B.6. In Figure B.6(a) an example AS-graph is depicted. The vertices inside of the blue ellipsoids represent the set of disjoint pairs S^0_{dis} selected by the clustering algorithm. The first interaction method, "Interlevel transition", is shown in Figure B.6(a) and B.6(b). Starting from $level - 0$ (Figure B.6(a)) the user decides to move to $level - 1$ (Figure B.6(b)). In general the user can move from any $level - k$ to any $level - l$, where $k, l \in \mathbb{N}$. The position of each vertex is defined in equation (B.10).

The second interaction method, "Semantic zoom to a vertex", is shown in Figure B.6(b) and B.6(c). As explained in Section B.3, the clusters of the level $(l + 1)$ are constructed from the set of disjoint pairs S^l_{dis}. In other words each cluster can be considered a union of two clusters of the previous level. The position of each vertex is defined in equation (B.10). In the case of Figure B.6(b), starting from level 1 the user decides to "semantic zoom" to a vertex. With this method, the selected cluster "breaks" into the two

constituting clusters of the lower level (Figure B.6(c)). This procedure can be repeated, up to the point that the vertices that are visualized, represent only one AS each.

As it was mentioned earlier, large change in the position of the vertices during the interaction procedure could cause a confusion to the analyst. In order to preserve the mental image of the analyst concerning the vertex positions, the hierarchical clustering algorithm was designed in such a way that the positional changes of each vertex due to the interaction of the user, are minimized. In order to achieve this, the position of each vertex is defined as the average position of all the ASes that this cluster is comprised of (equation (B.10)). Furthermore, the use of the Urquhart proximity graph makes sure that only the vertices that have neighboring positions, are selected in the clustering procedure. This way, the "Interlevel transition" interaction technique, in the case that the source and target levels $level-k$ and $level-l$ are relatively close, presents graphs that have similar structure as it is depicted for example in Figure B.4. Moreover, since the vertices that are presented using the "Semantic zoom to a vertex" interaction technique have only one level difference, the user's mental picture during the interaction procedure with BGPGraph, is also preserved.

B.6 Quantification of the information

In this Section the information content of the proposed visualization method is quantified using entropy measures. This procedure is critical since it provides a metric that could quantify the quality of a visualization system and lead to comparisons and optimizations.

B.6.1 Entropy of the input signal

The input signal represents the input information that the proposed approach uses for visualization purposes. In the respective case, the input signal is composed of the weights that directly reflect the BGP announcements. The calculation of these weights is presented in Section B.2 and the information content of the input signal is measured with respect to these weights.

Since these weights are calculated on per edge and vertex basis (equations (B.11) and (B.12)), it is possible to measure the information content of each

edge and vertex respectively. For each edge $e_j^l \in E$, the weights occurred during a user defined period, are used to calculate its corresponding entropy:

$$H^{in}(e_j^l) = -\sum_{i=1}^{K} \frac{k_i}{k_{total}} \log\left(\frac{k_i}{k_{total}}\right) \tag{B.14}$$

where K is the number of different weight instances that have occurred for the edge $e_j^l \in E^l$ over time, k_i is the number of occurrences of the i_{th} weight and $k_{total} = \sum_{i=1}^{K} k_i$ the total number of weight occurrences for this edge. The superscript "in" represents the input signal. The same procedure is followed to find the entropy $H^{in}(v_j^l)$ of each vertex.

The above mentioned entropies reflect the information content of each edge and vertex of the graph distinctly. To find a metric that could reflect the information content of the entire graph a different approach is followed. Just like before, the weight occurrences are measured, but this time with respect to the entire graph that is currently visualized and not the weight variance of each individual edge or vertex over time. Because of the hierarchical clustering algorithm, along with the two interaction techniques, the user is able to visualize each level of the hierarchy separately or visualize a hybrid graph comprised of vertices end edges from different levels of the hierarchy (an example is depicted in Figure B.6(c)). Due to this fact, a new notation is introduced $G(V^c, E^c)$, which represents the graph that is currently visualized by the system, along with all the respective weights. The graph $G(V^c, E^c)$ can represent either a level of the hierarchy $c = l$, or a hybrid graph comprised of vertices end edges from different levels. This way, the entropy of the edge weights of the entire graph that is currently visualized is defined as:

$$H_G^{in}(E^c) = -\sum_{i=1}^{Y} \frac{y_i}{y_{total}} \log\left(\frac{y_i}{y_{total}}\right) \tag{B.15}$$

where Y is the number of different edge weight instances that are visualized, y_i is the number of occurrences of the i_{th} weight and $y_{total} = \sum_{i=1}^{Y} y_i$ the total number of weight occurrences, but with respect to the edge weights of the entire graph. The same procedure is repeated in order to find the entropy $H_G^{in}(V^c)$ of the vertex weights of the entire graph.

B.6.2 Entropy of the output signal

In general a visualization system uses a mapping function to map the input data to the visualization features: $F : \Re^n \rightarrow V^m$, where $\Re^n$ is the input signal in the space of real numbers and has n features, and V^m is the visualization space (scatterplots, graphs, glyphs etc) that has m features. In the proposed scheme, the output signal is the visualized graph. Because of various factors, such as the display capacity, visual clutter and the limitations of the HVS, a transfer of all the information of the input signal to the output, is usually not possible. As explained in Section B.2, in the context of the proposed approach, the edge weights are mapped on the width and color of the edges and the vertex weights are mapped to the radius and color of the vertices. So the visualization space V^m has four features ($m = 4$), the width and color of edges and the radius and color of vertices.

Furthermore, in equations (B.4) and (B.5), the set of the vertex radius values R and the set of edge width values W were defined. In addition $|W| = L_w$ quantization levels were defined for the edges width and $|R| = L_r$ quantization levels were defined for the vertices radius. The role of the mapping function F is to map the edge and vertex weights to the elements of the sets W and R. Figure B.7 shows the mapping function $F : \Re^n \rightarrow V^m$, that maps the edge weight to the edge width. A similar mapping function is used for the mapping of the vertex weight to the vertex radius.

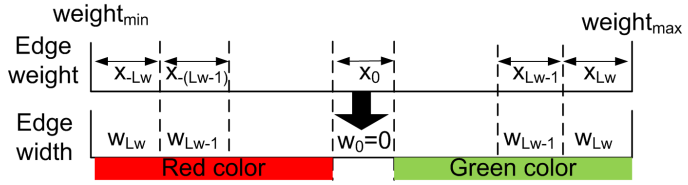


Figure B.7: The mapping function $F : \Re^n \rightarrow V^m$ from the edge weight to the edge width. Green color represents positive weight values and red negative.

In order to formulate the mapping function F, a variable X_i is initially defined as follows:

$$X_i = \left\{ \begin{array}{c} 0 \quad , i < -L_w \\ \sum_{j=-L_w}^{i} x_j \quad , -L_w \leq i \leq L_w \\ \sum_{j=-L_w}^{L_w} x_j \quad , i > L_w \end{array} \right\} \tag{B.16}$$

Afterwards, the mapping function F is defined as:

$$F(e_{weight}) = |i| \ , if \ X_{i-1} \leq e_{weight} < X_i \tag{B.17}$$

where e_{weight} is the corresponding edge weight that is mapped to width $w_k \in W$, for $k = F(e_{weight})$.

To find the entropy of the output signal, the equations (B.14) and (B.15) are used, but the difference is that the weights are firstly mapped to the elements of the sets W and R through the mapping function F. This way the corresponding entropies $H^{out}(e_i^l, F)$ (entropy of an edge), $H^{out}(v_j^l, F)$ (entropy of a vertex), $H_G^{out}(E^c, F)$ (entropy of the edges weights of the entire graph), $H_G^{out}(V^c, F)$ (entropy of the vertices weights of the entire graph) are calculated. The superscript "out" represents the output signal. It is apparent that the value of the output entropy depends on the mapping function $F : \Re^n \rightarrow V^m$ that inevitably induces information loss.

B.7 Applications of the information quantification

This Section presents the applications of the information quantification (Section B.6) for two purposes: 1) Minimize the information loss induced by the clustering procedure, and 2) Detect BGP routing anomalies.

B.7.1 Minimization of information loss induced by the clustering algorithm

This Section presents a method to minimize the information loss induced by the clustering algorithm using the entropy measures developed in Section B.6. The clustering algorithm described in Section B.3 did not take into account the information content of the input signal and as a result the selection of clusters is not optimal in that respect, but rather random.

Each cluster represents a collection of ASes and its weight is the sum of the weights of the ASes it is comprised of (equations (B.11) and (B.12)). Thus, the information of the vertex weights is preserved in an aggregated form. The information that is omitted is related to the weights of the edges among vertices of the same cluster (that are not visualized, as shown in Figure B.3). By carefully selecting the edges that should be preserved in consecutive layers of the hierarchy, the information loss can be minimized. This is achieved by avoiding specific pairs of ASes to be in the same cluster.

The term "forbidden pairs" is used for this case and is defined as the set of N pairs of ASes whose edges exhibit the highest entropy.

Table B.1: The edges with the biggest entropy and their corresponding rate of the total edge entropy

Number of edges M	30	100	200	3800
Rate of total entropy	25%	40%	50%	99.9%

Table B.1 shows the results of the entropy analysis on the AS-Graph of the year 2005 [91]. It shows the first M edges with the highest entropy and the percentage of their information content with respect to the total entropy. It is worth noting that the AS-Graph of 2005 has over $27,000$ edges and as it is shown, the first $3,800$ of them provide almost all the information. It is worth noting that the algorithm of Section B.3 is in fact a "forbidden pairs" algorithm for $M = 0$.

The "forbidden pairs" approach described here, aims to reduce the size of the graph, while also minimizing information loss. Compared with the simple approach of Section B.3 the size of the graphs at each level is slightly larger, a fact which depends on the value of M. Therefore, the decision on the value of M depends on the particular analysis. If preview of a lot of data is performed then there is a need for high abstraction in the visualization, thus a small M should be used. On the other hand, if in depth analysis is needed, large M should be used. So, M is a user defined variable which depends on the intentions of the user and the level of the hierarchy currently being visualized.

In order to compare the "forbidden pairs" implementations for various values of M, the following procedure is followed:

1. Find the entropy of each edge $H^{in}(e_j^0),\ e_j^0 \in E^0$, using equation (B.14)

2. For each level l of the hierarchy, construct the set $J^l = \{e_k^0(v_i^0, v_j^0) |\ v_i^0, v_j^0 \in v_h^l, \forall v_h^l \in V^l,\ i \neq j\}$. In other words, J^l is the set of edges that are omitted.

3. For each level l, find the total entropy of the edges that are lost $H^{in}_{(lost)}(J^l) = \sum_j H^{in}(e_j^0),\ \forall e_j^0 \in J^l$

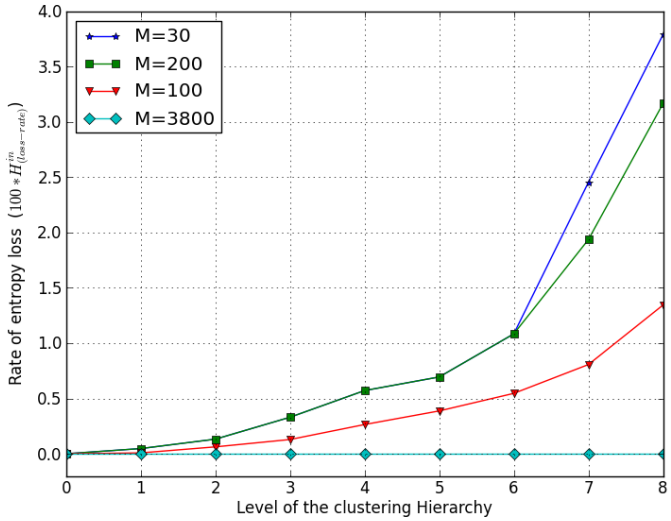


Figure B.8: The relationship between the entropy loss rate $En^{in}_{(loss-rate)}(J^l)$ for various M values and the levels of the hierarchy.

4. The rate of the entropy that is lost at the level l is: $H^{in}_{(loss-rate)}(J^l) = H^{in}_{(lost)}(J^l) / H^{in}(E^0)$

The comparison of the 'forbidden pairs" implementations for various values of M, using the above mentioned method is shown in Figure B.8.

It is worth noting that the entropy calculation is performed using equation (B.14), which measures the entropy of each edge by taking as input its weight variance over time. This calculation might favor irrelevant information, such as highly dynamic behavior. Due to the nature of BGP, it is difficult to discriminate noise from suspicious behavior. As a consequence, the algorithm presented here preserves the edges with the highest entropy, while the user is responsible for discriminating between false positives and actual hijacking events.

B.7.2 Using Entropy to detect events

The entropy measures proposed in Section B.6 can be used to construct plots that summarize the entropy change of the AS-graph over a specific period of time. These plots are called entropy plots. Using the entropy plots, the user can select to focus on specific periods of time where the information is

high and as a result something important might be happening. Afterwards, the user is presented with the AS-Graph of the selected period of time so as to perform his analysis on a detailed graph.

The entropy plots are basically bar plots, in which the length of each bar represents the magnitude of the corresponding input entropy $H_G^{in}(E^l)$ or $H_G^{in}(V^l)$ of the AS-Graph $G^l\left(V^l, E^l\right)$, where l represents the level of the hierarchy. The values of the input entropies are calculated over a period of time using predefined time steps ΔT (as presented in Section B.2). In other words, for every ΔT, the input entropies of the graph are calculated and a new bar is added to the corresponding entropy plot. The calculation procedure of the entropy plot is depicted in Figure B.9. It is apparent that the entropy plot is separated into two areas: the upper plot which represents the bar plot of the vertices entropy $H_G^{in}(V^l)$, and the lower plot which represents the bar plot of the edges entropy $H_G^{in}(E^l)$.

The entropy is not a perceptually linear metric. For instance, the difference between the two entropies $H_1 = 100\ bits$ and $H_2 = 101\ bits$ is one bit. But in fact, entropy H_2 with the addition of one bit represents twice as much information as H_1. This could potentially cause a misinterpretation of the magnitude of the event shown by the entropy plot. In order make the entropy metric perceptually consistent, the length of each bar of the entropy plot represents the following values: $2^{H_G^{in}(E^l)}$ in the case of the entropy of edges and $2^{H_G^{in}(V^l)}$ in the case of the entropy of vertices. This way, the lengths of the two bars of the previous example become $l_1 = 2^{H_1}$ and $l_2 = 2^{H_2} = 2 * l_1$, which is in fact a perceptual scaling.

The calculation of the entropies of the plot depends on two variables: the selected time step ΔT, and the level of the hierarchy l. As it was presented in Section B.7.1, the clustering algorithm was designed in such a way that the information loss at the higher levels of the hierarchy is minimized, while also preserving the important edges. In other words, the higher levels of the clustering hierarchy contain the important information in a compact form, while they also drop any irrelevant information. This important property of the clustering algorithm can be utilized by the entropy plot by using high levels for the calculation of the plots's bars, so as to enhance the accuracy and reduce the cluttering that the irrelevant information might induce. Furthermore, the second variable, the time step ΔT, has a different effect on the entropy plot. By increasing ΔT, the information content calculated within

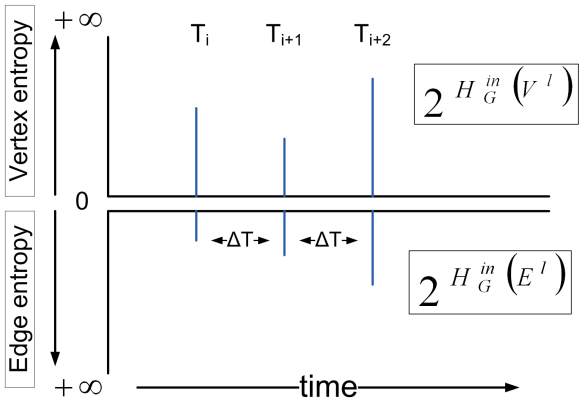


Figure B.9: The Entropy plot concept. The length of each bar represents the magnitude of the edges or vertices entropy, at the predefined time window.

the time windows also increases, which results in smaller ratios between the information content of different time windows. The smaller ratios have a direct impact onto the ratios of the bars, by making their length differences smaller. This effect hinders the goal of the entropy plots, which is to help the user focus onto interesting time windows. Generally, the value of ΔT depends onto the selected time period that is visualized by the entropy plots. The higher the time period is, the higher the ΔT must be, so as to reduce the number of the bars that are visualized. On the other hand, it is important that the value of ΔT is relatively small, so as to reduce the entropy aggregation effect.