JYVÄSKYLÄ STUDIES IN COMPUTING
185

Tuomo Sipola

Knowledge Discovery Using Diffusion Maps

JYVÄSKYLÄN YLIOPISTO

JYVÄSKYLÄ STUDIES IN COMPUTING 185

Tuomo Sipola

Knowledge Discovery Using Diffusion Maps

Esitetään Jyväskylän yliopiston informaatioteknologian tiedekunnan suostumuksella
julkisesti tarkastettavaksi yliopiston Agora-rakennuksen auditoriossa 1
joulukuun 16. päivänä 2013 kello 12.

Academic dissertation to be publicly discussed, by permission of
the Faculty of Information Technology of the University of Jyväskylä,
in building Agora, Auditorium 1, on December 16, 2013 at 12 o'clock.

UNIVERSITY OF JYVÄSKYLÄ

JYVÄSKYLÄ 2013

Knowledge Discovery Using Diffusion Maps

JYVÄSKYLÄ STUDIES IN COMPUTING 185

Tuomo Sipola

Knowledge Discovery Using Diffusion Maps

UNIVERSITY OF JYVÄSKYLÄ

JYVÄSKYLÄ 2013

Editors
Timo Männikkö
Department of Mathematical Information Technology, University of Jyväskylä
Pekka Olsbo, Sini Tuikka
Publishing Unit, University Library of Jyväskylä

URN:ISBN:978-951-39-5540-3
ISBN 978-951-39-5540-3 (PDF)

ISBN 978-951-39-5539-7 (nid.)
ISSN 1456-5390

Jyväskylä University Printing House, Jyväskylä 2013

ABSTRACT

Sipola, Tuomo
Knowledge Discovery Using Diffusion Maps
Jyväskylä: University of Jyväskylä, 2013, 48 p. (+included articles)
(Jyväskylä Studies in Computing
ISSN 1456-5390; 185)
ISBN 978-951-39-5539-7 (nid.)
ISBN 978-951-39-5540-3 (PDF)
Finnish summary
Diss.

This work is devoted to the study of knowledge discovery using diffusion maps. Data manipulation and storage have become increasingly affordable enabling the discovery of hidden information from massive datasets. With ever growing databases it is important to find efficient data mining methods. This research concerns high-dimensional data that contains a high number of measured variables from the point of view of analysis methods, but also from the human perspective. Dimensionality reduction is a process where new features are extracted from high-dimensional data so that their number is smaller than in the input data while the information content stays the same. Diffusion map is a dimensionality reduction method suitable for data exhibiting nonlinear behavior. This research shows that diffusion map data mining technology can be brought to diverse application areas, e.g., to clustering and to system health monitoring. Several studies concerning these data mining tasks are presented. Firstly, clustering using diffusion maps is demonstrated with text mining and brain imaging data. These studies show that the methodology is suitable for discovering knowledge and finding structure in datasets coming from quite different sources. Secondly, system health monitoring using diffusion maps is presented in network security and mechanical engineering scenarios. The related studies show that abnormal behavior in the systems can be found with the proposed methodologies.

Keywords: knowledge discovery, data mining, clustering, anomaly detection, dimensionality reduction, manifold learning, diffusion maps

Author

Tuomo Sipola
Department of Mathematical Information Technology
University of Jyväskylä
Finland

E-mail: `tuomo.sipola@jyu.fi`
`tuomo.sipola@iki.fi`

Supervisors

Prof. Amir Averbuch
School of Computer Science
Tel Aviv University
Israel

Prof. Tapani Ristaniemi
Department of Mathematical Information Technology
University of Jyväskylä
Finland

Reviewers

Prof. Hannu Koivisto
Department of Automation Science and Engineering
Tampere University of Technology
Finland

Dr. Erkki Laitinen
Department of Mathematical Sciences
University of Oulu
Finland

Opponent

Dr. Amaury Lendasse
Department of Information and Computer Science
Aalto University
Finland

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my supervisors, Prof. Amir Averbuch and Prof. Tapani Ristaniemi, for their guidance and support. Their encouragement and support has been invaluable.

I am indebted to my numerous collaborators, Dr. Fengyu Cong, Mr. Antti Juvonen, Mr. Joel Lehtonen, Prof. Asoke Nandi, Mr. Paavo Nieminen, Mr. Ilkka Pölönen, Mr. Gil Shabat, Mr. Yaniv Shmueli, and all the other colleagues who have helped me. Thank you all!

I am grateful to the Faculty of Information Technology of the University of Jyväskylä for giving me the opportunity to work on my dissertation. During my studies I participated in the Finnish Doctoral Programme in Computational Sciences (FICS), which also funded a conference visit, for which I am thankful. I thank the Foundation of Nokia Corporation (Nokian säätiö) and the Finnish Foundation for Technology Promotion (Tekniikan edistämissäätiö) for partly funding my research.

Finally, I would like to thank my family and friends who have kept me company after the working hours and given me a perspective to life beyond the academic world.

Jyväskylä
December 2, 2013

Tuomo Sipola

LIST OF FIGURES

LIST OF TABLES

CONTENTS

LIST OF INCLUDED ARTICLES

PI Tuomo Sipola, Antti Juvonen and Joel Lehtonen. Anomaly detection from network logs using diffusion maps. *IFIP Advances in Information and Communication Technology*, Vol. 363, pp. 172–181, 2011.

PII Tuomo Sipola, Antti Juvonen and Joel Lehtonen. Dimensionality reduction framework for detecting anomalies from network logs. *Engineering Intelligent Systems*, Vol. 20, Iss. 1–2, pp. 87–97, 2012.

PIII Antti Juvonen and Tuomo Sipola. Adaptive framework for network traffic classification using dimensionality reduction and clustering. *Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), 2012 4th International Congress on*, pp. 274–279, St. Petersburg, Russia, 2012.

PIV Antti Juvonen and Tuomo Sipola. Combining conjunctive rule extraction with diffusion maps for network intrusion detection. *The Eighteenth IEEE Symposium on Computers and Communications (ISCC 2013)*, Split, Croatia, 2013.

PV Paavo Nieminen, Ilkka Pölönen and Tuomo Sipola. Research literature mapping using article clustering. *Journal of Informetrics*, Vol. 7, Iss. 4, pp. 874–886, 2013.

PVI Tuomo Sipola, Tapani Ristaniemi and Amir Averbuch. Gear classification and fault detection using a diffusion map framework. *Reports of the Department of Mathematical Information Technology Series B. Scientific Computing*, No. B 6/2013, University of Jyväskylä, Jyväskylä, Finland, 2013.

PVII Yaniv Shmueli, Tuomo Sipola, Gil Shabat and Amir Averbuch. Using affinity perturbations to detect web traffic anomalies. *Proceedings of the 10th International Conference on Sampling Theory and Applications (SampTA 2013)*, pp. 444–447, Bremen, Germany, 2013.

PVIII Tuomo Sipola, Fengyu Cong, Tapani Ristaniemi, Vinoo Alluri, Petri Toiviainen, Elvira Brattico and Asoke K. Nandi. Diffusion map for clustering fMRI spatial maps extracted by independent component analysis. *IEEE International Workshop on Machine Learning for Signal Processing (MLSP 2013)*, Southampton, United Kingdom, 2013.

1 INTRODUCTION

> Quid sit futurum cras, fuge quaerere, et
> quem fors dierum cumque dabit, lucro
> adpone [...]
>
> *Horace Carm. 1.9*

This chapter presents the motivation behind the research concerning knowledge discovery. The related research objectives are also presented. Next, the structure of the dissertation and research work are detailed. The articles included in this dissertation and author's contributions to them are introduced individually. Finally, other published articles by the author are mentioned.

1.1 Research motivation

Data manipulation and storage have become increasingly affordable enabling the discovery of hidden information from massive datasets. Recent advances in these technologies have enabled new ways to automatically gather huge volumes of data. In everyday lives mobile phones, home appliances, cars and social media produce data that contain information about behavior and interactions of humans. The same massive growth in data gathering and storing is also present in business and decision making. Industrial equipment, electrical meters and shipping crates contain advanced sensors that also provide valuable information. Some of the potential topics where modern data analysis could add value include healthcare, public sector, retail, manufacturing and personal location data (Lohr, 2012; Sagiroglu and Sinanc, 2013; Myllymäki et al., 2011).

With ever growing databases new businesses have begun to find the power of data analysis, because in the modern world new data are produced at an increasing speed. Companies such as Amazon, Facebook, Google, Netflix and Yahoo have embraced the new age of big data and use analytics to convert the collected data into economic value (Myllymäki et al., 2011). The amount of scientific data alone grows exponentially and requires special measures from the analy-

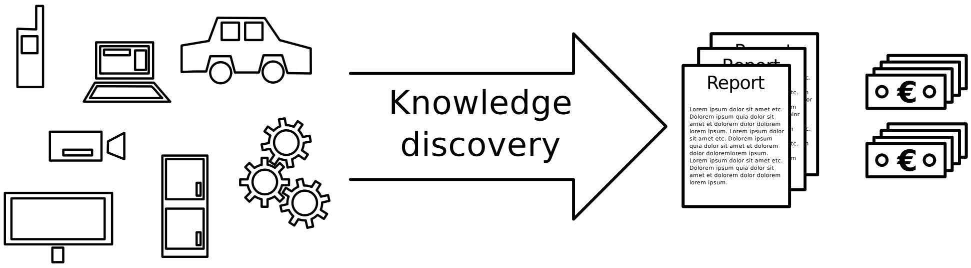
Knowledge discovery
Report
Lorem ipsum dolor sit amet etc. Dolorem ipsum quia dolor sit amet et dolorem dolor dolorem lorem ipsum. Lorem ipsum dolor sit amet etc. Dolorem ipsum quia dolor sit amet et dolorem dolor doloremlorem ipsum. Lorem ipsum dolor sit amet etc. Dolorem ipsum quia dolor sit amet et dolorem dolor dolorem lorem ipsum.
€
€

FIGURE 1 The main goal is to create value by discovering knowledge from existing data sources.

sis point of view (Szalay and Gray, 2006). Moreover, nowadays businesses and people working in general management express interest towards analyzing and understanding their data (Lohr, 2012; Davenport et al., 2012). Figure 1 outlines the main goal of creating value by knowledge discovery. Everything that happens between the data sources and knowledge discovery reporting and business value creation needs to be studied in order to find the most effective ways to analyze data.

In the center of all this is the analysis of huge datasets. There are two big challenges in the analysis. The first one is the big size of datasets, which excludes the possibility of manual work. The other is the unstructured nature of the datasets, which might contain numerical, categorical, textual or multimedia data. The first one can be overcome with the use of advanced data mining methods and other automation. The latter can be solved during preprocessing by restructuring the variables and analyzing different data individually, combining the results, or by converting the data to a unified format (Myllymäki et al., 2011).

The overall process of knowledge discovery in databases (KDD) tries to extract information in order to solve the business problem (Fayyad et al., 1996a,b; Brachman and Anand, 1996). The actual computational part of knowledge discovery is done using data mining methods. Data mining, according to Hand et al. (2001), consists of

- analyzing large data sets,
- finding unsuspected relationships,
- summarizing the data.

Furthermore, their definition calls for results that are understandable and useful. Some famous data mining technique families include data clustering (Jain, 2010; Cîmpanu and Ferariu, 2012) and anomaly detection (Chandola et al., 2009). Data clustering tries to find a structure in the data that separates measurements in a meaningful manner. Anomaly detection on the other hand tries to find outliers that do not ressemble the other measurements in the dataset.

This research concerns specifically high-dimensional data. This kind of data contains an unusually high number of measured variables from the point of view analysis methods, but also from the human perspective. The number of measured variables is high and the mathematical representation of the data has a high dimension. A huge number of variables describe various systems, such as health monitoring, network intrusion detection and process control. Many kinds of variables and quantities can be measured from these systems. As stated above, the objective is to create meaningful groupings of the measurements and to detect anomalous behavior of the systems.

High-dimensional datasets pose several problems. They are sparse, because the data points reside in a much bigger space when the number of dimensions gets bigger. To accurately describe a function, the number of needed samples grows exponentially when the number of variables grows. At the same time classification becomes challenging. This is called the curse of dimesnionality (van der Maaten et al., 2009). High-dimensional datasets are also difficult to visualize with traditional methods. To overcome these problems this dissertation studies dimensionality reduction, and specifically the diffusion map methodology.

1.2 Research questions

The objective of the research is to apply the knowledge discovery process and diffusion map technology to varying real world situations. To achieve this, the dissertation presents various case studies of knowledge discovery and shows that diffusion map methodologies are practically usable in such situations.

The main research questions of this study are as follows:

1. Can diffusion map data mining technology be brought to diverse application areas?
2. How usable diffusion map is for clustering in practical situations?
3. How usable diffusion map is for system health monitoring?

This research is conducted using the design science approach. The goal is to construct an artifact and evaluate it (Hevner et al., 2004). The artifact here is namely the computer program that performs the data mining task. The reports describing the research are part of the evaluation process.

1.3 Structure of the work

The content of this thesis is focused on two topics. Firstly, the theoretical background of knowledge discovery and diffusion maps is discussed with some ideas about implementation. Secondly, the contribution of research articles and case studies is presented. The latter part also covers the concluding remarks.

Chapter 2 introduces the reader to the knowledge discovery process and its main steps. Dimensionality reduction is one way to perform the transformation step in the knowledge discovery process. History and current state of dimensionality reduction is discussed in more detail. Since diffusion map is one dimensionality reduction algorithm, this dissertation focuses on it and related algorithms. The technical aspects of these algorithms are also presented.

Chapter 3 outlines the research contribution of the dissertation. It focuses on the use of diffusion maps in knowledge discovery frameworks. This chapter also discusses the benefits for each case presented in the included articles. Theoretical and practical implications are discussed along with recommendations for further study. Finally, Chapter 4 briefly concludes the dissertation.

1.4 Author's contribution to the included articles

Author's contribution to the included articles lies in the development of the overall knowledge discovery framework for each case, and especially in the use of diffusion map methodology. The relationships between the articles are presented in Figure 2. The first approach to use diffusion map methodology is to monitor system health. Many of the included articles cover this area. Articles PI; PII; PIII; PIV present results from network intrusion detection. In addition, article PVII combines a new kernel update method with diffusion maps in a network log analysis application case. Article PVI concerns fault detection in an industrial environment. The second approach is to use the methodology for explorative data analysis with clustering. Article PV applies diffusion map to text document mining. Finally, article PVIII applies the diffusion map clustering methodology to brain imaging. Each article is discussed in more detail below.

In PI the idea of using diffusion maps for network log analysis is used. In order to detect anomalies from a HTTP log, n-gram feature extraction and diffusion map dimensionality reduction produce a new feature space where spectral clustering is used to separate the normal network traffic from the anomalous behavior. The experimental part consists of analyzing a log file from a real server from a research parter. The author is responsible for the dimensionality reduction approach, its implementation, performed the corresponding experimental tasks and contributed ideas to the general framework and interpretation of the results.

Article PII is direct continuation from PI. It includes more comprehensive test scenarios and deeper discussion. The article proposes a dimensionality reduction framework for anomaly detection in network security context. As earlier, n-gram distribution features are extracted from the HTTP logs. Principal component analysis and diffusion maps reduce the dimensionality of the data matrix, and facilitate anomaly detection. Several data sets from real servers illustrate the usefulness of the approach. The article hilights the adaptiveness of the framework and its use in the application layer of network traffic. Visualization possibilities are also presented. The author designed and implemented the dimensionality reduction and clustering parts of the system, performed the corresponding analysis, and contributed to the design of the overall framework and

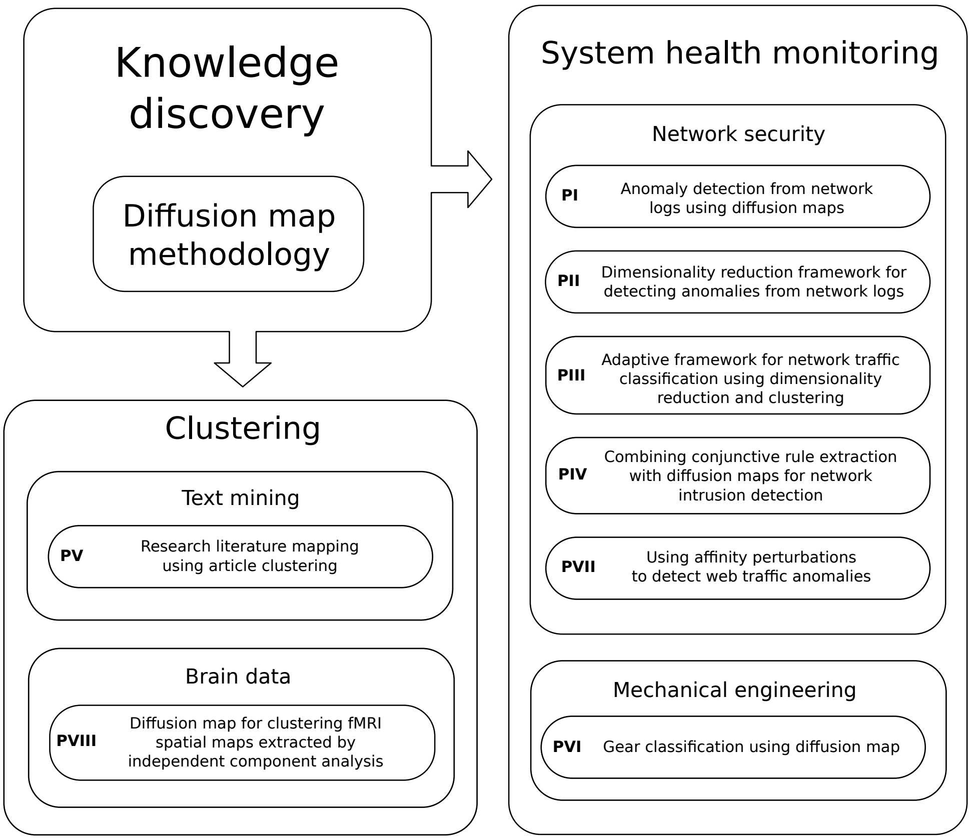
Knowledge discovery
Diffusion map methodology
System health monitoring
Network security
PI Anomaly detection from network logs using diffusion maps
PII Dimensionality reduction framework for detecting anomalies from network logs
PIII Adaptive framework for network traffic classification using dimensionality reduction and clustering
PIV Combining conjunctive rule extraction with diffusion maps for network intrusion detection
PVII Using affinity perturbations to detect web traffic anomalies
Mechanical engineering
PVI Gear classification using diffusion map
Clustering
Text mining
PV Research literature mapping using article clustering
Brain data
PVIII Diffusion map for clustering fMRI spatial maps extracted by independent component analysis

FIGURE 2 Common topics and application areas of the included articles.

interpretation of the results.

In PIII the clustering and visualization capabilities of a framework based on dimensionality reduction are discussed. The experiment uses HTTP logs from a real server, whose traffic is clustered. The author implemented the normalization and dimensionality reduction via diffusion map parts for the proposed framework, performed the corresponding parts of the experiment and assisted in reporting the obtained clustering results.

Article PIV contrasts the proposed rule-based system with signature-based and anomaly detection systems. The main idea is to use unsupervised learning using diffusion maps to learn the labeling of the data. This labeling is then used to learn conjunctive rules. These rules are easy to understand and matching them to newly arriving data is fast. The author contributed to the idea of using conjunctive rules with diffusion maps, designed parts of the framework, implemented the diffusion map algorithm and clustering, and implemented and adapted the rule extraction algorithm with the other author.

In PV the knowledge discovery process is applied to research literature. The proposed methodology is an automatic way of identifying the structure and topics of current research literature. The data in question, i.e. article titles and keywords, were collected from publicly available sources. The metadata was then analyzed using diffusion map dimensionality reduction and clustering. As a result, the article presents a snapshot of current topics in data mining research articles. The author contributed to web scraping, data mining design and implementation, data analysis, result presentation and interpretation.

In report PVI the constructed framework detects gear faults and monitors system health. The diffusion map training is combined with the Nyström extension for out-of-sample data. The constructed framework was used to detect faults in advance, and in some cases earlier than traditional manual monitoring would have. The author carried out the research under the supervision of the co-authors. The author's contribution covers the implementation of the system, finding a sufficiently accurate prediction model, and in addition reporting and evaluating the results.

Article PVII combines recursive power iterations algorithm with diffusion maps. This way the diffusion map model can be updated within a short amount of time in an online scenario. The sliding window technique creates a diffusion map that continuously updates itself with new data while dropping old data. The update algorithm solves low-dimensional coordinates when the distance matrix is perturbed. The author's contribution includes the initial idea of combining the methods, implementation of the system and carrying out the experiments.

In PVIII the methodology is applied to brain imaging data. The data comes from measurements where people listen to music while their brains are measured using functional magnetic resonance imaging. Spatial maps, i.e. represenstations of the brain, are clustered into two groups using diffusion maps. The author created the clustering framework and carried out the clustering part of the data analysis.

1.5 Other published articles

In addition to the included articles, the author has also researched other areas of nonlinear data anlysis. The results have been published in the following articles:

- Cong, F., Sipola, T., Huttunen-Scott, T., Xu, X., Ristaniemi, T. & Lyytinen, H. 2009.Hilbert-Huang versus Morlet wavelet transformation on mismatch negativity of children in uninterrupted sound paradigm. Nonlinear Biomedical Physics 3 (1).
- Cong, F., Sipola, T., Xu, X., Huttunen-Scott, T., Lyytinen, H. & Ristaniemi, T. 2010. Concatenated trial based Hilbert-Huang transformation on event-related potentials. In Proc. International Joint Conference on Neural Networks 2010 (IEEE World Congress on Computational Intelligence), 1379–1383.

2 THEORETICAL FOUNDATION

Interea repetunt caecis obscura latebris
verba datae sortis secum, inter seque volutant.
Ovid Met. 1, 388–389

In this chapter the knowledge discovery process and data mining topics are discussed in more detail. The history of dimensionality reduction is presented and a mathematical explanation of the diffusion map algorithm is given. Supporting algorithms like Nyström extension are also explained.

2.1 Knowledge discovery process

Knowledge discovery is a high-level term for the whole process of deriving actionable knowledge from databases. Presenting data mining as a part of the knowledge discovery process places the technical challenges in the broader scope. The knowledge discovery process from databases (KDD) suggests the steps that are needed to extract business knowledge from available data (Fayyad et al., 1996a,b,c; Brachman and Anand, 1996).

This view is echoed in literature, for example in Abbass et al. (2002, pp. 72, 162). The main diffrerences between the process models are in the bundling of tasks to higher level steps. However, Pechenizkiy et al. (2008) argue that in practice different frameworks need to be combined. Cios et al. (2007) cite Fayyad and call the process academic. They also present a more industry-oriented process and compare many others. Larose (2006, ch. 7) also refers to an industrial framework where the actual data preparation and modeling phases are just part of the whole process. Hand et al. (2001) focus on the data mining part of their process but present a high-level pipeline. Han et al. (2011) also follow a similar data mining process, having databases in mind, as well as Mitra and Acharya (2003, p. 5). Kantardzic (2011) introduces a technical process. The main stages of some of the processes are presented in Table 1.

TABLE 1 Data mining processes (Fayyad et al., 1996a; Hand et al., 2001; Cios et al., 2007; Han et al., 2011).

Fayyad et al. 1996	Hand et al. 2001	Cios et al. 2007	Han et al. 2011
learn domain		business understanding	
create target data	select target data	data understanding	data cleaning
cleaning and preprocessing	preprocess		data integration
			data selection
reduction and projection	transform	data preparation	transformation
choose function			
choose algotithms			
data mining	data mining	modeling	data mining
interpretation	interpretation	evaluation	evaluation
using information		deployment	presentation

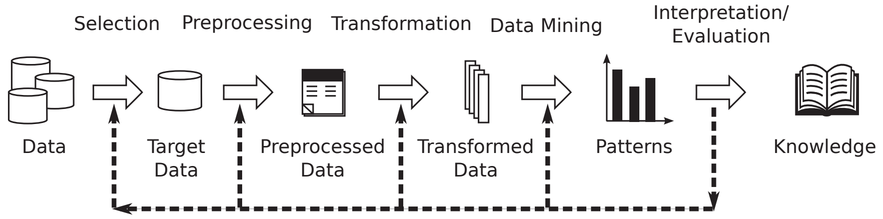

FIGURE 3 Steps of the knowledge discovery process according to Fayyad et al. (1996a).

This research takes the more academic view, focusing on the data mining part. The process includes five steps that are shown in Figure 3. Below is an overview of them (Fayyad et al., 1996a).

Data selection In the data selection step the most relevant data sources for the task are selected. Usually there is an abundance of data sources and sometimes features, and a substantial challenge is to narrow down the sources that might contain important information. This is dependent on the goal of the knowledge discovery task.

Preprocessing Once the target data is defined, it needs to be preprocessed. Besides preprocessing, data cleaning is also a relevant. Converting and collecting the data to correct formats takes effort. Noise removal is a typical preprocessing step. Incomplete data entries and known large changes need to be accounted for. Combining and cleaning various databases is a rather mechanical process but sometimes poses problematic situations. The amount of work needed at this stage is usually underestimated in practical work.

Transformation Preprocessed data is transformed to a more suitable form for clustering and classification. This involves feature extraction and selection, dimensionality reduction and other transformations. The selected features depend on the data mining case, as does the number of needed final features.

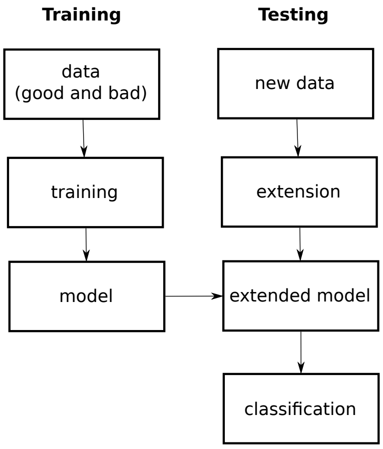

FIGURE 4 Machine learning: training creates a model, while testing classifies new data using the model.

Data mining The data mining step itself tries to extract patterns from the transformed data. Summarization, classification, regression and clustering are some common tasks at this stage. The goal of the knowledge discovery process is matched with the relevant data mining methods. Often, this includes exploratory analysis of the data, which helps in deciding the most suitable models and parameters for the methods. The end products of this stage are rules, decision trees, regressions and clustering of the data.

Interpretation Finally, the business value of the results should be understood. In this step, interpretating the patterns creates the actual final result of the knowledge discovery process. Evaluation of the obtained results is also important because some assumptions might have been wrong, the data might have been insufficient or there might have been some problem during the previous steps. Not all results have business significance even if they present some new information that has scientific or statistic novelty.

This research focuses on the transformation and data mining steps of the knowledge discovery process. The other steps are intimately connected to the utilized data, but the data mining methods are usually separate modules that can be described as independent systems. This is not to say, however, that in all cases the selection of data mining method is independent of the data and knowledge discovery problem. Correct tools should be used with differing datasets.

The actual data mining step usually uses a machine learning method. A supervised machine learning system is first trained using known data labeling (hence supervision), and then the performance of the system is tested. The testing results reveal the quality of the learned model, provided that the testing material adequately reflects the data in a real situation. This idea of training and testing is illustrated in Figure 4.

2.2 Data mining

Data mining has a very broad definition encompassing vast areas of research. One definition for data mining is being "the science of extracting useful information from large data sets of databases" (Hand et al., 2001, p. i). Witten et al. (2011, p. 4) also define data mining: "Data mining is about solving problems by analyzing data already present in databases." They emphasize the view that data mining has a lot to do with the recent availability of huge databases. In general, the goal is to explain the data and make predictions based on it by finding and and describing patterns in it (Witten et al., 2011; Chakrabarti and Cox, 2008). Moreover, data mining tries to find hidden predictive information (Wang, 2003, p. vii) and make sense of large amounts of mostly unsupervised data in some domain with a data driven, and not model driven, approach (Cios et al., 2007, p. 3). The challenge usually lies in finding a model for unsupervised data without *a priori* knowledge (Cios et al., 2007, pp. 5–7).

Generally, according to one view, data mining can be exploratory analysis, descriptive modeling, predictive modeling, discovering patterns and rules or retrieval by content (Padhy et al., 2012). The data mining algortihms can be seen as having three components (Fayyad et al., 1996b):

- The model, where its function (end product) and representational form (technical side) are relevant.
- The preference criterion on which model to select over another.
- The search algorithm to find models and parameters for the data.

Furthermore, the more common models include classification, regression, clustering, rule generation, summarization, dependency modeling, link analysis and sequence analysis (Fayyad et al., 1996b; Mitra and Acharya, 2003).

Feature selection takes a subset of features for further analysis at the beginning of the data mining pipeline. Feature subset generation can be divided to three strategies: complete search finds optimal features, sequential search adds or removes features and thus gives up the whole search, and random search tries to be efficient while avoiding local optima. There are various strategies to evaluate subsets: distance measure, information measure, dependency measure and consistency measure. In addition, a wrapper-type evaluation uses predictive accuracy or cluster goodness (Guyon and Elisseeff, 2003; Liu and Yu, 2005).

Clustering algorithms divide the data into groups whose members are similar in some sense but dissimilar between the groups. The main categories of clustering algorithms could be thougt as hierarchical and partitional (Jain et al., 1999; Berkhin, 2006; Jain, 2010). More detailed categories of clustering algorithms include hierarchical, partitional (including *k*-means, graph), neural network, kernel (including support vector machines, SVM) and sequential. New challenges are met with large-scale datasets, visualization possibilities and validity of the clustering (Xu and Wunsch, 2009). A recent categorization of clustering algrotihms is provided below (Cîmpanu and Ferariu, 2012):

- Distance based clustering analyzes dissimilarity of data points using distance metrics. Such methods include, e.g., *k*-means, PAM and CLARA.
- Density based clustering creates clusters around dense areas and leaves the other points outside. An example is the DBSCAN algorithm.
- Model based algorithms try to find a model by which the clustering was generated. Examples include BIRCH and SOM.
- Grid based methods divide the feature space to grid-cells.

Classification tries to place newly arriving data points to some of the known categories. There are numerous algorithms dedicated to this problem, many from the early years of computing, inclusing logistic regression (Hastie et al., 2001; Duda et al., 2012), Bayesian methods (Williams and Barber, 1998; Cheng and Greiner, 1999), kernel methods, decision trees, rule learning, neural networks, support vector machines and nearest neighbor methods (Hastie et al., 2001; Duda et al., 2012).

Support vector machines (SVM) are a type of supervised learning algorithm that performs linear classification of new data points by dividing the feature space with a hyperplane. Kernel constructions may be used to create nonlinear classification (Steinwart and Christmann, 2008; Ben-Hur and Weston, 2010).

Neural networks are also a big area of machine learning research. There are various neural network architectures that connect the artificial neurons. The number of inputs depends on the data, while the output depends what kind of labeling is needed (Lu et al., 1996; Anthony and Bartlett, 2009). As an example of the versatility of neural networks, self-organizing maps (SOM) are designed for unsupervised dimensionality reduction and clustering (Kohonen, 2001).

2.3 Dimensionality reduction

Dimensionality reduction is a process where new features are extracted from the data so that their number is smaller than in the input data while the information content of the data stays the same. This is achieved with a function that finds new coordinates for the data points in a lower-dimensional space. This process is called mapping. Many dimensionality reduction methods are manifold learning or spectral embedding algorithms that are based on the eigen-decomposition of a similarity matrix (Bengio et al., 2006). A simple example of mapping is an ordinary navigation map where the surface of a 3D object is put on a 2D page, as in Figure 5. Following the same idea, this can be translated to high-dimensional datasets, as in Figure 6, where more than three dimensions are projected to a two-dimensional space.

Perhaps the most famous dimensionality reduction method is the principal component analysis (PCA) (Jolliffe, 2002; Abdi and Williams, 2010). It finds the eigenvectors of the covariance matrix. These eigenvectors are then used to map the data points to a space where the axis directions contain maximal variance.

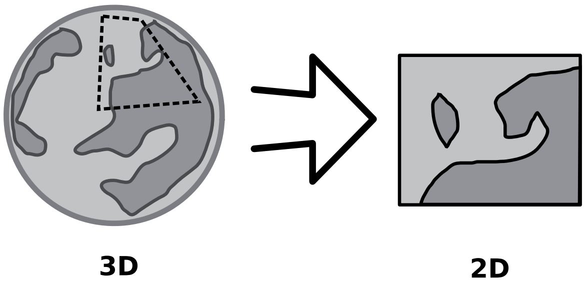

FIGURE 5 Mapping a 3D object, such as the Earth, to a 2D map.

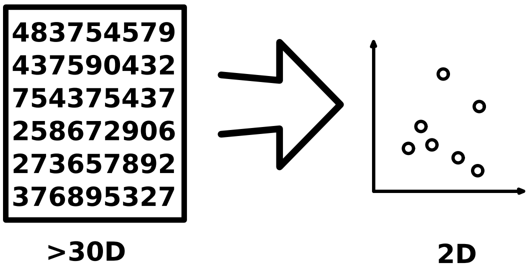

FIGURE 6 When considering high-dimensional data, e.g. 30 dimensions, the goal is to reduce the presentation of the information to only a few dimensions.

PCA is a linear method but useful in most practical cases or for initial global dimensionality reduction. Another variant is the minor component analysis (MCA) which considers the eigenvectors that explain the least amount of variance of the covariance matrix (Luo et al., 1997; Cirrincione et al., 2002).

Kernel PCA adds a kernel function that transforms the data before doing component analysis (Mika et al., 1998; Müller et al., 2001). This adds nonlinear capabilities to the PCA analysis. Kernel functions can be useful in separating clusters that have nonlinear boundaries. When used with isotropic kernels, kernel PCA is a form of metric multidimensional scaling (Williams, 2002). Kernel PCA and spectral embedding methods are special cases of a more general learning problem (Bengio et al., 2004).

Multidimensional scaling (MDS) tries to find coordinates by minimizing the difference between the distance in the high dimensional data point distances and the low-dimensional point distances. Classically the low-dimensional distances are measured using the Euclidean norm. The optimization problem has analytical solutions for some cases but at other times numerical solvers are used. Lately more generalized versions of MDS have been found (Kruskal, 1964; Borg, 2005; Bronstein et al., 2006).

Isomap uses the foundation laid by MDS but incorporates the geodesic distance to the optimization. After finding the neighborhoods for the data points, Isomap finds the geodesic distances, and finally applies MDS to find the low-dimensional coordinates (Tenenbaum et al., 2000; Yang, 2002; Saxena et al., 2004;

Choi and Choi, 2007). Locally linear embedding (LLE) starts from the assumption that enough densely sampled data lie on a locally linear patch of a manifold. First, it finds the local neighbors. Secondly, it reconstructs the neighborhood with linear weights. Finally, LLE minimizes the embedding cost function yielding the low-dimensional coordinates (Roweis and Saul, 2000; de Ridder et al., 2003; Donoho and Grimes, 2003; Chang and Yeung, 2006). Laplacian eigenmaps work in a similar manner. After finding the neighbors and applying a possibly nonlinear kernel to the connected neighbors, the last step involves using the Laplacian to solve the eigenvectors (Belkin and Niyogi, 2001, 2003; Belkin et al., 2006; Belkin and Niyogi, 2007).

Spectral clustering (Kannan et al., 2004; von Luxburg, 2007; Filippone et al., 2008) is closely related to the other spectral embedding algorithms. It uses the Laplacian of the similarity matrix graph when calculating the eigenvectors that are used as low-dimensional coordinates. Starting from the previously mentioned methods, theory has been developed around different ways of creating the distance matrices and finding the cost function for the optimization. The eigenvectors act as the separating feature for the clusters in the low-dimensional space. The eigenvalues are the solutions to the normalized cut problem, which finds small weights between clusters but strong internal ties. This spectral clustering has probabilistic interpretation: grouping happens through similarity of transition probabilities between clusters (Meila and Shi, 2001; Shi and Malik, 2000). The normalized Laplacian spectral clustering seems to be the better choice over unnormalized (von Luxburg et al., 2005) and converge under general conditions (von Luxburg et al., 2008). Further developments in spectral clustering have been made, for example non-redundant views (Niu et al., 2010; Kumar and Iii, 2011) and subspace clustering, where an optimized sparse presentation is used as a basis for the similarity matrix in spectral clustering (Elhamifar and Vidal, 2009, 2013; Soltanolkotabi et al., 2013).

2.4 Diffusion maps in data mining

Diffusion map Ξ is a function from multi-dimensional space to a space with lower dimensions. It can be placed to the existing taxonomy of dimension reduction methods (Lee and Verleysen, 2007; van der Maaten et al., 2009) as a nonlinear geometric method that preserves the diffusion distance global property in the high-dimensional space as Euclidean distance in the low-dimensional space. Diffusion map reduces dimensionality from n dimensions to m dimensions:

$$\Xi : \mathbb{R}^n \to \mathbb{R}^m. \tag{1}$$

In short, the low-dimensional coordinates of the diffusion map are the eigenvectors of a transition matrix between the measurement points. The name diffusion map comes from the fact that the Euclidean distances in the low-dimensional space preserve the diffusion distances in the high-dimensional space (Coifman et al., 2005; Coifman and Lafon, 2006; Lafon et al., 2006; Nadler et al., 2005, 2008). Diffusion maps are suitable for analysis of gemetry and probability distributions of empirical data. With correct normalization it is capable of analyzing dynamical systems that exhibit different time scales (Nadler et al., 2006).

There are broadly two ways to use the diffusion map for data mining:

- Stand-alone diffusion map cluster analysis.
- Supervised learning diffusion map analysis with out-of-sample extension.

If diffusion map is used as a stand-alone data mining tool, the end result is usually produced only once to explore the data set. Analysis consists usually of clustering or anomaly detection. Diffusion map has a rigorous justification for *k*-means clustering (Lafon and Lee, 2006). Moreover, localized diffusion folders create a multilevel clustering which overcomes the scaling problem of many datasets (David, 2009; David et al., 2010; David and Averbuch, 2012). The implemented system can be used again for future datasets, but adapting it for high volume streaming data is challenging. If new results are needed when new data streams in, performing the same analysis successively is an option. Recently diffusion distance has been defined for changing data and a global distance between graphs has been established (Coifman and Hirn, 2013).

The supervised learning and extension approach firstly builds a model of the data and then uses some faster method to extend newly arriving data to the model. Extension methods are discussed in Section 2.7. Here the end result is a classification system. This diffusion map methodology for data mining is divided into two parts. The first part is training, where a model is built upon known behavior data of the system. The second part is extending new data to the model. An unsupervised variant is possible but detecting the correct classes automatically is a huge challenge. The two steps of training and testing are presented from the mathematical point of view in Figure 7. For an implementation, refer to Appendix 1. The mathematical details are discussed in Section 2.5.

2.5 Mathematics of diffusion maps

Let each measured data point be a real vector $\mathbf{x}_i \in \mathbb{R}^n$, $i = 1 \dots N$. This vector is called also the feature vector. Here N is the number of measurement and n is the number of measured features. Each element in the vector corresponds to a measurement from the same sample. For example, the ith measurement would be:

$$\mathbf{x}_i = [x_{i1}\ x_{i2}\ \dots\ x_{in}] \,. \tag{2}$$

The dataset is created by collecting the measurements to matrix X where each measurement $\mathbf{x}_i$ is a row. The columns correspond to the measured features.

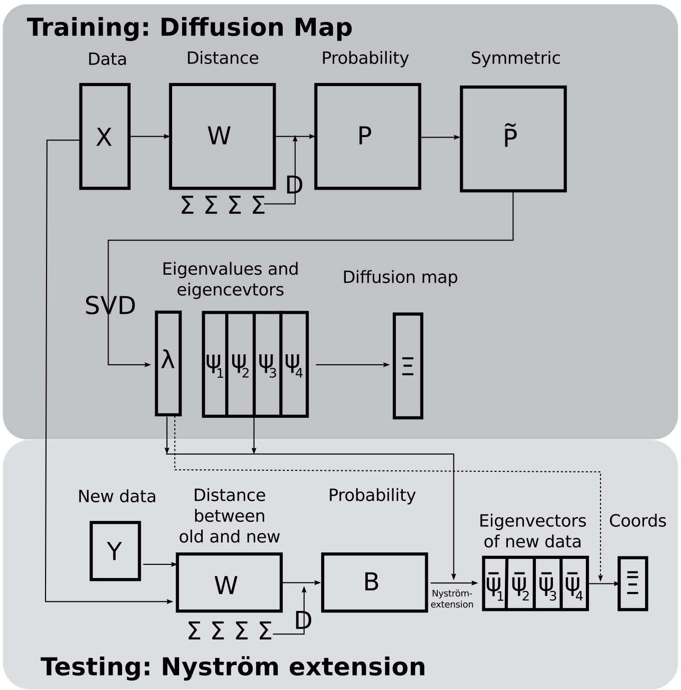

FIGURE 7 Block diagram of diffusion map methodology.

In order to reduce the effect of high dimensions to the sparsity of the data, we define a kernel function $w(\mathbf{x}_i, \mathbf{x}_j)$. It defines an affinity between the two vectors $\mathbf{x}_i$ and $\mathbf{x}_j$. The kernel function satisfies the following properties:

– symmetric $w(\mathbf{x}_i, \mathbf{x}_j) = w(\mathbf{x}_j, \mathbf{x}_i)$,
– non-negative $w(\mathbf{x}_i, \mathbf{x}_j) \geq 0 \; \forall \; \mathbf{x}_i, \mathbf{x}_j \in X$,
– positive semi-definite $\sum_{i=1}^{N} \sum_{j=1}^{N} c_i c_j w(\mathbf{x}_i, \mathbf{x}_j) \geq 0 \; \forall \; \mathbf{x}_i, \mathbf{x}_j \in X, c_i \ldots c_N \in \mathbb{R}$.

The measurements can be thought as points in a graph whose distances are defined by the kernel.

This kernel defines the weight matrix W, whose entries are $W_{ij} = w(\mathbf{x}_i, \mathbf{x}_j)$. These entries tell the distance between two points. The matrix W is N times N because distance from each point is calculated to every other point.

The most used kernel is the Gaussian kernel. Any other kernel function can be used. Measure-based Gausssian correlation kernel replaces the manifold assumption with the more general measure assumption and combines the local distances with the distribution of the data. The spectral properties of a diffusion map utilizing this kind of kernel are similar to the diffusion map (Bermanis et al., 2013c,b).

Here the Gaussian kernel takes the parameter ϵ, which defines the neighborhood for the points, the smaller it is, the more it accentuates the differences of high distances:

$$W(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(\frac{-||\mathbf{x}_i - \mathbf{x}_j||^2}{\epsilon}\right). \tag{3}$$

At this point different families of diffusions may be constructed using renormalization $w^\alpha(\mathbf{x}_i, \mathbf{x}_j) = q^{-\alpha}(\mathbf{x}_i) w(\mathbf{x}_i, \mathbf{x}_j) q^{-\alpha}(\mathbf{x}_j)$ (where q is the degree of the point) and applying the weighted graph Laplacian normalization as shown below. Because the case of $\alpha = 0$ normalizes graph Laplacian on isotropic weights, the equation is reduced to identity (Coifman and Lafon, 2006).

Degree of a point expresses how many close connections a data point has in the graph. The degree of a point is defined as follows:

$$d(\mathbf{x}_i) = \int_X w(\mathbf{x}_i, \mathbf{x}_j) \mathrm{d}\mathbf{x}_j. \tag{4}$$

In practice in a discrete setting the degree is the sum of the weights between the data point and the rest of the points:

$$d(\mathbf{x}_i) = \sum_{\mathbf{x}_j \in X} w(\mathbf{x}_i, \mathbf{x}_j). \tag{5}$$

To create a normalization matrix, the degrees of each points can be collected to the diagonal of a matrix $D_{ii} = \sum_{j=1}^{N} W_{ij}$, while the rest of the matrix is filled with zeros. This means that each row is summed to get the degree of the point corresponding to that row.

Now the normalized graph Laplacian can be constructed (Chung, 1997). This corresponds to the transition probability between each data point. In other words, the probability of travelling from one point to the another. Intuitively the closer points are more probable travel destinations than points that are far away. Moreover, the normalized graph Laplacian

$$p(\mathbf{x}_i, \mathbf{x}_j) = \frac{w(\mathbf{x}_i, \mathbf{x}_j)}{d(\mathbf{x}_i)} \tag{6}$$

preserves the eigenvalues (Nadler et al., 2008). Each entry in the weight matrix is divided by the corresponding row sum. In matrix form this is $P = D^{-1}W$. This transition matrix is symmetrized using the conjugate matrix $\tilde{P}$ of P. Each entry in $\tilde{P}$ can be expressed as:

$$\tilde{p}(\mathbf{x}_i, \mathbf{x}_j) = p(\mathbf{x}_i, \mathbf{x}_j) \sqrt{\frac{d(\mathbf{x}_i)}{d(\mathbf{x}_j)}}. \tag{7}$$

In matrix form the conjugate matrix is $\tilde{P} = D^{\frac{1}{2}} P D^{-\frac{1}{2}}$. This, however, can be combined with the definition of P: $\tilde{P} = D^{\frac{1}{2}} D^{-1} W D^{-\frac{1}{2}} = D^{-\frac{1}{2}} W D^{-\frac{1}{2}}$. Thus, the individual elements of $\tilde{P}$ are calculated by

$$\tilde{p}(\mathbf{x}_i, \mathbf{x}_j) = \frac{w(\mathbf{x}_i, \mathbf{x}_j)}{\sqrt{d(\mathbf{x}_i) d(\mathbf{x}_j)}}. \tag{8}$$

Singular value decomposition (SVD) (Kalman, 1996) $\tilde{P} = U\Lambda U^*$ finds the eigenvalues $\Lambda = \mathrm{diag}([\lambda_1, \lambda_2, \ldots, \lambda_n])$ and eigenvectors $U = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_n]$ for the symmetric matrix $\tilde{P}$. This spectral decomposition is expressed as:

$$\tilde{p}(\mathbf{x}_i, \mathbf{x}_j) = \sum_l \lambda_l \mathbf{u}_l(\mathbf{x}_i)\mathbf{u}_l(\mathbf{x}_j). \tag{9}$$

The eigenvalues for P are the same as for $\tilde{P}$. Finally, the left and right eigenvectors for P are found with $\Phi = D^{\frac{1}{2}}U$ and $\Psi = D^{-\frac{1}{2}}U$ (Nadler et al., 2008), and for individual points with

$$\phi(\mathbf{x}_i) = \mathbf{u}(\mathbf{x}_i)\sqrt{d(\mathbf{x}_i)} \qquad \psi(\mathbf{x}_i) = \frac{\mathbf{u}(\mathbf{x}_i)}{\sqrt{d(\mathbf{x}_i)}}. \tag{10}$$

Each transition probability is expressed by the sum of products of an eigenvalue and eigenvectors, with the parameter t expressing the number of steps used to make that transition:

$$p^t(\mathbf{x}_i, \mathbf{x}_j) = \sum_l \lambda_l^t \psi_l(\mathbf{x}_i)\phi_l(\mathbf{x}_j). \tag{11}$$

The low-dimensional coordinates Ξ are created using $\Xi = \Psi\Lambda$. Only a few of these coordinates are needed to represent the data to a certain degree of error (Coifman and Lafon, 2006). The data can be reconstructed using Equation 11 while iterating over only the first m eigenpairs. The first eigenvector is constant, so only the following eigenvectors and eigenvalues are used. This way we get the following function that maps the data points to a lower-dimensional space:

$$\Xi_m^t : \mathbf{x}_i \longrightarrow \begin{pmatrix} \lambda_1^t \psi_1(\mathbf{x}_i) \\ \lambda_2^t \psi_2(\mathbf{x}_i) \\ \lambda_3^t \psi_3(\mathbf{x}_i) \\ \vdots \\ \lambda_m^t \psi_m(\mathbf{x}_i) \end{pmatrix}. \tag{12}$$

Now, for each n-dimensional data point $\mathbf{x}_i$, there is a corresponding m-dimensional coordinate, where $m \ll n$. The effect of left out dimensions can be seen in Equation 11. The smaller later eigenvalues cause that part of the sum go to near zero. The number of selected dimensions depends on how fast the eigenvalues decay. The first eigenvectors retain most of the information in the data, which is why the later eigenvectors are left out. Some information is lost but the error is bounded and lower dimensionality facilitates clustering.

2.6 Choosing parameters

Choosing the parameters, namely ϵ is not a trivial task. The ϵ characterizes the neighborhood of the data points in the Gaussian kernel. This selection is also called choosing a bandwith. A large ϵ means that many points will be included in the neighborhood, which means that the diffusion distance will be higher because it is easier to traverse to another data point. Conversely, a small ϵ causes the neighborhoods to contain only one point making the graph scarcely connected and the transition probabilities to be low. A value between the extremes is desirable (Schclar et al., 2010). There are many heuristics for estimating this parameter.

The median method takes the middle value of pairwise distances between the data points (Schclar et al., 2010):

$$\epsilon = \text{median}\{\|\mathbf{x}_i - \mathbf{x}_j\|\}_{\mathbf{x}_i,\mathbf{x}_j \in X}. \tag{13}$$

Average smallest distance in the neighborhood finds a neighborhood size that includes at least one neighbor for each data point and then takes the average of such sizes (Lafon, 2004):

$$\epsilon = \frac{1}{N}\sum_{i=1}^{N} \min_{j:\mathbf{x}_j \neq \mathbf{x}_i} \{\|\mathbf{x}_i - \mathbf{x}_j\|^2\}_{\mathbf{x}_i,\mathbf{x}_j \in X}. \tag{14}$$

The max-min measure tries to tune the kernel to describe the infinitesimal connectivity of the data. This means choosing the smallest ϵ while keeping the local connectivity, with $\alpha \geq 1$ assuring that there is at least one point in the neighborhood (Keller et al., 2010; Schclar, 2008):

$$\epsilon = \alpha \max\{\min\{\|\mathbf{x}_i - \mathbf{x}_j\|^2\}\}_{\mathbf{x}_i,\mathbf{x}_j \in X}. \tag{15}$$

The theoretical foundations lie in finding the intrinsic dimensionality in submanifolds of $\mathbb{R}^n$ (Hein and Audibert, 2005) and properties of the manifold Laplacian (Singer, 2006). These findings have lead to the use of the sum of all weights in the transition distance matrix W (Coifman et al., 2008; Singer et al., 2009). In Equation 16 the weights in the affinity matrix are summed up. This sum is larger if the neighborhood is large, and thus there are more distances significantly greater than 0. The reverse is true: smaller neighborhood means less distances and consequently the sum is smaller.

$$L = \sum_{i=1}^{n}\sum_{j=1}^{n} W_{i,j} \tag{16}$$

The stability of eigenproblem and neighborhood size with changing ϵ has been studied in detail. Echoing some of the above methods the neighborhood stability approach ensures that the number of local average points is large enough (Lee and Wasserman, 2010):

$$\epsilon = \min\{\epsilon : \text{median}\{N_1, \ldots, N_n\} \geq k\}, \tag{17}$$

where $N_i = \#\{\mathbf{x}_j : \|\mathbf{x}_i - \mathbf{x}_j\| \leq \sqrt{2\epsilon}\}$. Another way of choosing the bandwidth is via the eigen-stability analysis using signal-to-noise ratio for some $K_n \geq 1$ (Lee and Wasserman, 2010):

$$\epsilon_0 = \inf\{\epsilon : SNR(\epsilon) \geq K_n\}. \tag{18}$$

A graph-based approach aims for a well-connected graph. The longest edge of the minimal spanning tree of a fully connected graph would be the ϵ. However, this approach yields too large values when there are outliers or several tight clusters far away from each other (von Luxburg, 2007).

2.7 Extension of new measurements

Once a low-dimensional representation is obtained from the training stage, it would be beneficial to extend newly arriving data to the model. The goal is to interpolate the coordinates of unknown points based on the coordinate mapping of the known points. Many dimensionality reduction methods and spectral clustering can use a general framework for out-of-sample extension (Bengio et al., 2004). This is the same as the Nyström method that is a popular method to extend new data (Fowlkes et al., 2004; Belongie et al., 2002). The features selected during the training are the only ones needed. These new measurements are normalized using the same normalization as during the training.

Let a new data point be $\mathbf{y}_j \in \mathbb{R}^n$. Then the distance between the new points and each training point are collected in a matrix $\bar{W}$. This function uses the same ϵ as the one in training phase. The new kernel is found by

$$\bar{W}_{ij} = \exp\left(-\frac{||\mathbf{x}_i - \mathbf{y}_j||^2}{\epsilon}\right). \tag{19}$$

Diagonal matrix $\bar{D}_{ii} = \sum_{i=1}^{N} \bar{W}_{ij}$ contains the column sums of $\bar{W}$. Now we can create the transition probability matrix $B = \bar{W}\bar{D}^{-1}$. The following matrix multiplication produces new eigenvectors for the new point, the eigenvectors Ψ and eigenvalues Λ are the same as in training: $\bar{\Psi} = B^T\Psi\Lambda^{-1}$. These new eigenvectors now extend the new point to the diffusion coordinates: $\bar{\Xi} = \bar{\Psi}\Lambda$. The last two steps can be combined:

$$\bar{\Xi} = B^T\Psi. \tag{20}$$

Matrix $\bar{\Xi}$ now contains the extended coordinate approximations in its columns for the new points $\mathbf{y}_j$.

Beyond this straightforward method there are several others. Geometric harmonics provide a multiscale extension scheme for empirical functions. The method decomposes the function defined on a manifold into eigenfunctions, then they are extended using the Nyström method (Lafon, 2004; Coifman et al., 2005; Coifman and Lafon, 2006). Adaptive kernels combined with Nyström method offer another alternatives to extension. This density-weighted version uses the normalized data histogram as coefficients (Zhang and Kwok, 2009). Multiscale function extension is based on the mutual distances and uses the hierarchical decomposition of the Gaussian kernel. The data point distribution defines an adaptive grid on the data (Bermanis et al., 2013a). Generalized out-of-sample extension does not depend on the method used in training. The extension method learns the mapping again and learns the low-dimensional neighborhood of the new data point. Consequently, it can be used with any manifold learning method (Strange and Zwiggelaar, 2011). Sparse representation is an alternative to the more traditional out-of-sample extensions and is shown to work with LLE (Raducanu and Dornaika, 2013).

3 RESEARCH CONTRIBUTION

> καὶ σὺ μὲν οὕτω χαῖρε, Διὸς τέκος αἰγιόχοιο:
> αὐτὰρ ἐγὼ καὶ σεῖο καὶ ἄλλης μνήσομ᾽ ἀοιδῆς.
> *Hymn to Athena, HH 28*

This chapter presents the case studies and their results. First the system health monitoring in mechanical engineering and network security contexts are discussed. Then the clustering cases concerning text mining and brain data are mentioned.

3.1 Gear fault detection

The goal of PVI is to estimate the usefulness of dimensionality reduction methods in gear fault detection. The approach has similarities with other recent research (Huang et al., 2013). The goal of training a model and classifying test samples is met since almost all the gears are classified correctly according to their labels. This proves that the training is succesful and separates the good gears from the bad. More importantly, measurements from totally different gears can be extended into the model.

The challenges of spectral methods in general need some addressing. The proposed method works because, after slight filtering, the good and bad gears are separable in the lower dimensions. However, the high computational cost could be a problem in a more real-time system (Chandola et al., 2009, p. 38).

The differing physical location of gear units makes it difficult to separate behaviors. Better training data and more detailed labeling could prevent this kind of misclassification. Vastly differing operating environments and behaviors might also cause misclassifications. The classification of a gear time series itself is an ambiguous concept. However, this study shows that gears in normal condition and gears that are going to break down behave differently and can be separated from each other.

3.2 Network anomalies

Dynamic web services are vulnerable to a multitude of intrusions that could be previously unknown. Legitimate features can be used for unwanted access (Mukkamala and Sung, 2003). Server logs contain vast amounts of information about network traffic, and finding attacks from these logs improves the security of the services. Intrusion detection systems analyze these logs to identify malicious traffic (di Pietro and Mancini, 2008). Differing traffic can be found using anomaly detection (Chandola et al., 2009). The next paragraphs summarize the main techniques used to detect abnormalities from network logs and the main results with real-world datasets.

The goal of PI is to find security attacks from network data. The proposed anomaly detection scheme includes *n*-gram feature extraction (Damashek et al., 1995), dimensionality reduction and spectral clustering style linear clustering (Shi and Malik, 2000; Meila and Shi, 2001). It could be used for query log analysis in real situations. In practice the boundary between normal and anomalous might not be as clear as in this example. However, the relative strangeness of the sample could indicate how severe an alert is. The data in question is rather sparse and the discriminating features are quite evident from the feature matrix. This is the merit of the *n*-gram feature extraction which creates a feature space that separates the normal behavior in a good manner. The features describe the data clearly, and they are easy to process afterwards. The presented anomaly detection method performs well on real data. As an unsupervised algorithm this approach is well suited to finding previously unknown intrusions. This method could be applied to offline clustering as well as extended to a real-time intrusion detection system.

These results are elaborated in PII. The dimensionality reduction framework adapts to the log data. It assumes that only few variables are needed to express the interesting information, and finds a coordinate system that describes the global structure of the data. These coordinates could be used for further analysis of characteristics of anomalous activities. The practical results show that abnormal behavior can be found from HTTP logs. The main benefits of this framework include:

- The amount of log lines that needs to be inspected is reduced. This is useful for system administrators trying to identify intrusions. The number of interesting log lines is low compared to the total number of lines in the log file.
- The unsupervised nature and adaptiveness of the framework. The proposed methods adapt to the structure of the data without training or previous knowledge. This makes it suitable for exploration and analysis of data without prior examples or attack signatures. This means that the framework may also detect zero-day attacks.
- It works on the application layer in the network. The attacks themselves must in some way target the actual applications running on the computer.

These logs might be more available than pure low-level network packet data.

- Visualization of text log data. It is much easier to analyze the structure of traffic using visualizations than it is to read raw textual logs.

The feature extraction from the web log is currently done with *n*-grams. However, this is only one method for it and other text-focused features might better describe the dataset. Furthermore, the dimensionality reduction scheme could be developed to adapt to this kind of data more efficiently, and the quality of the reduction could also be evaluated. Finally, automated root cause detection would make the system more usable in practice.

In PIII a framework for preprocessing, clustering and visualizing web server log data is presented and used for anomaly detection, visualization and explorative data analysis. The results indicate that there are traffic structures that can be visualized from HTTP query information. Traffic clustering can give new information about the users. They could be categorized with more accuracy, and individual advertising or content could be offered. Using data mining methods, underlying structure and anomalies are found from HTTP logs and these results can be visualized and analyzed to find patterns and anomalies.

Article PIV deals with extracting rules from the clustering results provided by a diffusion map training framework. Modern data mining technology in network security context does not always create understandable results for the end users. Therefore, this so-called black box system is not a desirable end goal. Simple conjunctive rules (Craven and Shavlik, 1994; Ryman-Tubb and d'Avila Garcez, 2010) are easier to understand, and rule extraction from the complex data mining techniques might facilitate user acceptance. The main benefit of this framework is that the final output is a set of rules. No black box implementation is needed as the end result is a simple and easy to understand rule matching system. The training data may contain intrusions and anomalies, provided that the clustering step can differentiate them. In addition, rule matching is a fast operation compared to more complex algorithms. The proposed framework is useful in situations where high-dimensional datasets need to be used as a basis for anomaly detection and quick classification. Such datasets are common nowadays in research environments as well as in industry, because collecting data is widespread. Our example case has been network security, which bears real benefits to anyone using modern communication networks. The provided tools are useful for network administrators who are trying to understand anomalous behavior in their networks.

In PVII another approach is taken to create a more online system. The training phase is computationally expensive in machine learning algorithms. Evolving datasets require updating the training. The proposed method updates the training profile using the recursive power iterations algorithm (Shmueli et al., 2012) and a sliding window algorithm for online processing. The algorithms assume that the data is modeled by a kernel method that includes spectral decomposition. A web server request log where an actual intrusion attack is known to happen is used to illustrate the online processing. Continuous update of the kernel prevents the problem of multiple costly trainings.

3.3 Automated literature mapping

Research in PV follows the knowledge discovery process creating a literature mapping framework based on article clustering. The goal is to analyze topics of current interest in a particular field of science. This work has similar goals to other literature mapping and clustering studies (Szczuka et al., 2012; Leydesdorff et al., 2013). The proposed framework ressembles the ones that use multidimensional scaling (Boyack et al., 2005; Waltman et al., 2010). The article also presents case study example in the field of data mining literature. Metadata are collected from high impact journals. The analysis uses a word occurrence matrix as basis. Diffusion map with hierarchical agglomerative clustering finds groups of similar articles from this sparse matrix. The clustering enables a researcher to get a quick overview of the topics published in the selected body of literature. The results from this study include frequency tables of the occurring words, structural view obtained from the clustering and journal article distribution among the clusters. Currently the output of our method is a snapshot of current published articles. Combining a longitudinal point of view might reveal long-term trends in research literature.

3.4 Clustering brain imaging data

Article PVIII presents a nonlinear analysis method for clustering independent components (ICA) of functional magnetic resonance (fMRI) imaging. In order to gain understanding about the human brain, various technologies have recently been introduced, such as functional magnetic resonance imaging (fMRI), which measures blood oxygenation level. It detects changes that are believed to be related to neurotransmitter activity. The method localises brain function well, and thus is useful in detecting differences in subject brain responses (Matthews and Jezzard, 2004; Huettel et al., 2004). Deeper understanding about the simultaneous activities in the brain begins with a decomposition of the data. Independent component analysis (ICA) has been extensively used to analyze fMRI data. It tries to decompose the data into multiple components that are mixed in the original data (Calhoun et al., 2009). The input data consists of ICA components. The proposed clustering is based on diffusion map manifold learning, which reduces the dimensionality of the data and enables clustering algorithms to perform their task. The two-dimensional clustering derived from the 209,633-dimensional feature space provides a new tool to compare the components. The results show that the proposed methodology separates groups of similarly behaving spatial maps. Results from diffusion map spectral clustering are similar to hierarchical agglomerative clustering and *k*-means clustering. Small sample size and good separation of clusters make the clustering problem easier. Moreover, the visualization obtained from diffusion map offers an interpretation for clustering.

3.5 Discussion

To characterize the diffusion map method and its motivations, several qualifications may be used. Below the methodology is placed in the field of dimensionality reduction methods using twelve proposed characteristics (Lee and Verleysen, 2007). This categorization places diffusion maps naturally within the neighborhood of nonlinear PCA-like spectral methods, many of which are discussed in Section 2.3 of this dissertation.

1. Diffusion maps are categorized as hard dimensionality reduction methods, meaning that the initial dimensionality is in the magnitude of hundreds or thousands. The huge number of dimensions is not a problem since the use of diffusion distances finds the high-dimensional data structure, although too small sample sizes might not capture the variety of data behavior. Diffusion maps are also usable with soft dimensionality reduction if a nonlinear approach is needed. Large sample sizes may actually become a problem since the distance matrix construction takes computational time and memory.
2. Diffusion maps take the traditional modeling approach using the transition probability connection between the observed and latent variables. The method finds the latent variables (i.e. the low-dimensional coordinates) starting from the observed ones.
3. The nonlinearity of diffusion maps has already been stated. The connection between the latent variables and observed data indeed might be more complex than just a linear one. This makes diffusion maps capable of recovering more information in certain cases than linear methods. This, however, comes with a cost in the complexity of the method.
4. The mathematics behind diffusion maps use a continuous model, much like PCA. As the name suggests, diffusion map is a mapping from the high-dimensional space to a low-dimensional embedded space.
5. The mapping is implicit in nature. The model unfolds the manifold where the data resides in the high-dimensional space. There is no direct association with each data points, but a mapping. Extensions of new data points may use the same manifold information.
6. When using diffusion maps, external estimation of the dimensionality is needed. The dimensionality is a metaparameter given by the user. This embedding dimension is used to select the number of dimensions in the latent variable coordinates. Using the eigencap in the eigenvalues of diffusion map is a possibility to decide the dimensionality, but since the variance is not linked to it in the same way as with PCA, this method is not equivalent. The interesting information might be captured in some other low-dimensional coordinates.
7. Following the example of PCA, diffusion maps are incremental and produce layered embeddings. This means that the coordinates do not change if a dimension is dropped from the resulting low-dimensional space. This is a

common feature of spectral methods, which are based on eigendecomposition. The embeddings are not independent because they are not optimized for the dimensionality.

8. Diffusion maps produce a single coordinate system using the connectivity of the manifold where the data lies. The global structure of the data and its overall information content is considered. The local patch approach dividing the manifold into pieces has recently been applied in the context of diffusion maps (Salhov et al., 2012; Wolf and Averbuch, 2013).
9. There is no mandatory vector quantization in diffusion map methods. The overabundance of data is problem with large kernel sizes, in that case initial data distribution prototypes created with vector quantization would be beneficial. In practice, finding the representative samples for larger datasets is a problem.
10. In Section 2.4 two approaches of diffusion map for data mining are presented. The clustering version, or batch algorithm works with all the observations, the dataset being available at once. Diffusion maps alone are suitable only for such batch processing, unless used with sliding window techniques. However, online processing of newly arriving data is possible with extension methods that are presented in Section 2.7. This does not update the training model, except for the densities, but makes it possible to classify new data points.
11. Diffusion maps are based on a closed form solution of the objective. This means that it can be categorized as an dimensionality reduction method based on exact optimization. The goal of finding diffusion distance preserving low-dimensional coordinates is achieved.
12. The pairwise distances are used as the type of criterion to be optimized. The Euclidean distances between the embedded points are as close as possible to the disffusion distances between the dataset points.

A categorizing hierarchy of unsupervised data analysis methods is presented in Figure 8 (Lee and Verleysen, 2007). Diffusion maps are placed in the geometric distance-based nonlinear dimensionality method part of the hierarchy.

From the knowledge discovery perspective, diffusion maps are a modular part of the process. Its specific mathematical features provide low-dimensional representations similar to other spectral methods. As a hard dimensionality reduction method, diffusion mapping could be useful for unfolding initial nonlinear structures in the data. On the other hand, using them after an initial linear transformation, the low-level nonlinearities could be found. In the future more work on the combination of methodologies is needed, especiallly clustering methods and their theoretical usefulness in spectral clustering tasks.

The various case studies and their topics show that the introduced methodology can be used for knowledge discovery as part of the transformation and data mining steps. The datasets in the case studies differ in their source, case specific data mining feature extraction and end result needs. As previously presented, diffusion map framework was useful in finding a meaningful low-dimensional

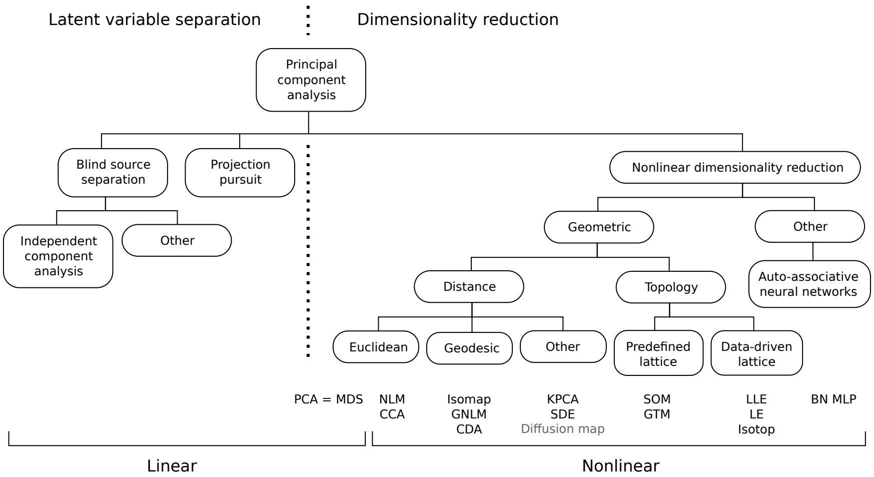

FIGURE 8 Hierarchy of some unsupervised data analysis methods used for latent variable separation and dimensionality reduction (Lee and Verleysen, 2007).

representation of the datasets. Its use for visualization is also helpful.

Stemming from the Gaussian kernel definition, the ability to change the neighborhood in transition probability sense is an advantage because the shape of the embedding can be controlled. However, finding the correct embedding is not always easy. This estimation of the different metaparameters of diffusion map, namely ϵ, t and the number m of relevant eigenvectors still needs more analytical models or heuristic estimates. However, the situation with many applications is that only the domain knowledge will reveal the best values from the knowledge discovery point of view.

Diffusion map methods find the optimal solution for the distance preserving problem. However, this does not mean that the solution is always relevant to the knowledge discovery problem. In such cases it has been a step in an explorative data mining process, which might lead to another direction. The case-specific meaningful information might not be in the eigenvectors corresponding to the largest eigenvalues or in the least correlated eigenvectors. This emphasizes the importance of domain area expertise.

Diffusion maps are usable with moderately sized datasets. In practice, the size of the pairwise distance matrix is a real drawback. This can be overcome to some extent with extension methods but then the representative data points must be found for training. Ultimately this comes to the question of sampling the data so that it represents the underlying phenomenon in a reliable way.

Another concern is that datasets do not usually exhibit nonlinear behavior that would necessiate the use of nonlinear methods, at least on the global level. A basic PCA procedure finds roughly the same structure. However, the kernel approach gives means to modify the mapping for the current task. Moreover, the overall mathematical framework facilitates new kernel inventions that are mathematically sound and more adapted to the data.

4 CONCLUSION

As the storage and manipulation of data become less expensive and more available, the need to understand collected data drives the development of new data mining methodologies. This dissertation discusses the use of diffusion map dimensionality reduction methodology to discover knowledge in several application cases.

There are two main perspectives in this study. The first one is the knowledge discovery process. It is a useful tool for data analysis and makes the flow from the data sources to the interpreted knowledge an understandable task. The second perspective is the diffusion map framework and also its place in the data transformation and data mining parts of the knowledge discovery process. Diffusion map is a nonlinear and distance preserving method, which make it suitable for complex data behavior. The mathematical theory behind diffusion maps enables their use in practical situations to solve data mining problems. Combining the two perspectives creates a data mining pipeline that is both practical and on such an abstraction level that it is easier to understand.

The practical cases include fault detection and system monitoring. Network anomaly detection, gear fault detection, automated research literature mapping and brain imaging clustering show that the introduced methodology can be used to create knowledge about these complex datasets. The datasets in the case studies differ in their source, case specific data mining feature extraction and end result needs. Diffusion map framework was useful in finding meaningful low-dimensional representations of the datasets.

The estimation of the different metaparameters of diffusion map still needs more analytical models or heuristic estimates. However, the situation with many applications is that only domain knowledge will reveal the best values from the knowledge discovery point of view. In the future more work on the combination of methodologies is needed, especiallly clustering methods and their usefulness in spectral clustering tasks. There are new possibilities in bringing diffusion maps to dynamic online environments and also finding more accurate kernels to describe the data.

YHTEENVETO (FINNISH SUMMARY)

Tämä väitöskirja, *Tietämyksen löytäminen diffuusiokuvauksia käyttäen*, käsittelee korkeaulotteisen datan hyödyntämistä. Aineistojen varastointi ja käsittely on yhä edullisempaa ja yleisempää, mikä on mahdollistanut piilossa olevan tietämyksen löytämisen massiivisista tietovarastoista. Tämä mahdollisuus ajaa ymmärtämään kerättyä dataa ja kehittämään tiedonlouhintamenetelmiä. Väitöskirja tutkii diffuusiokuvausten käyttöä aineiston ulottuvuuksien vähentämiseksi ja tiedon löytämiseksi erilaisissa käytännön tapauksissa. Erityisesti mielenkiinto kohdistuu korkeaulotteisiin aineistoihin, joissa on suuri määrä mitattuja muuttujia analyysimenetelmien ja myös ihmisen näkökulmasta. Ulottuvuuden pienentäminen on menetelmä, jossa korkeaulotteisesta aineistosta erotetaan uusia piirteitä, joiden lukumäärä on pienempi kuin syötteessä, kuitenkin siten, että informaatiosisältö pysyy samana. Diffuusiokuvaus on ulottuvuuden pienentämiseen tarkoitettu menetelmä, joka soveltuu epälineaarisen aineiston analysointiin.

Tutkimuksessa on kaksi näkökulmaa. Ensimmäinen on tietämyksen löytäminen aineistosta. Tämä prosessimalli on hyödyllinen data-analyysin työkalu ja kuvaa analyysin kulun tietolähteestä tulkittuun tietämykseen asti ymmärrettävänä tehtävänä. Toinen näkökulma on diffuusiokuvausmenetelmät ja niiden sijoittuminen tiedon muokkaamisen ja tiedonlouhinnan osioihin tietämyksen löytämisen prosessissa. Diffuusiokuvaus on epälineaarinen ja etäisyyden säilyttävä menetelmä, mikä tekee siitä hyödyllisen monimutkaisesti käyttäytyvää dataa käsiteltäessä. Matemaattinen teoria diffuusiokuvauksen takana mahdollistaa sen käytännöllisen hyödyntämisen tiedonlouhintaongelmia ratkottaessa. Näiden kahden näkökulman yhdistäminen luo tiedonlouhintatavan, joka on käytännöllinen, ja joka toimii sellaisella abstraktiotasolla, että kokonaisuus on helpompi ymmärtää.

Esitetyt esimerkkitapaukset jakautuvat ryhmittelyyn ja kunnonseurantaan. Ryhmittelyä diffuusiokuvauksen avulla käytettiin automatisoituun kirjallisuuden kartoittamiseen ja aivokuvantamiseen. Kunnonvalvontalähtökohtaa käytettiin vaihteiden vianseurantaan ja tietoverkkoliikenteen poikkeavuuksien havaitsemiseen. Tulokset osoittavat menetelmien soveltuvan tietämyksen löytämiseen monimutkaisista aineistoista. Tapausten aineistot eroavat lähteensä, tapauskohtaisen piirteiden erotuksen ja tarvittavien tulosten osalta. Diffuusiokuvaus osana järjestelmää oli hyödyllinen löytämään merkityksellisiä matalaulotteisia kuvauksia.

REFERENCES

Abbass, H., Sarker, R. & Newton, C. 2002. Data mining: a heuristic approach. Idea Group.

Abdi, H. & Williams, L. J. 2010. Principal component analysis. Wiley Interdisciplinary Reviews: Computational Statistics 2 (4), 433–459.

Anthony, M. & Bartlett, P. L. 2009. Neural network learning: Theoretical foundations. Cambridge; New York: Cambridge University Press.

Belkin, M., Niyogi, P. & Sindhwani, V. 2006. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. The Journal of Machine Learning Research 7, 2399–2434.

Belkin, M. & Niyogi, P. 2001. Laplacian eigenmaps and spectral techniques for embedding and clustering. In NIPS, Vol. 14, 585–591.

Belkin, M. & Niyogi, P. 2003. Laplacian eigenmaps for dimensionality reduction and data representation. Neural computation 15 (6), 1373–1396.

Belkin, M. & Niyogi, P. 2007. Convergence of Laplacian eigenmaps. Advances in Neural Information Processing Systems 19, 129.

Belongie, S., Fowlkes, C., Chung, F. & Malik, J. 2002. Spectral partitioning with indefinite kernels using the Nyström extension. In A. Heyden, G. Sparr, M. Nielsen & P. Johansen (Eds.) Computer Vision — ECCV 2002, Vol. 2352. Springer Berlin Heidelberg. Lecture Notes in Computer Science, 531-542.

Ben-Hur, A. & Weston, J. 2010. A user's guide to support vector machines. In Data mining techniques for the life sciences. Springer, 223–239.

Bengio, Y., Delalleau, O., Le Roux, N., Paiement, J.-F., Vincent, P. & Ouimet, M. 2004. Learning eigenfunctions links spectral embedding and kernel PCA. Neural Computation 16 (10), 2197–2219.

Bengio, Y., Delalleau, O., Roux, N. L., Paiement, J.-F., Vincent, P. & Ouimet, M. 2006. Spectral dimensionality reduction. In Feature Extraction. Heidelberg: Springer Berlin. Studies in Fuzziness and Soft Computing, 519–550.

Bengio, Y., Paiement, J.-F., Vincent, P., Delalleau, O., Le Roux, N. & Ouimet, M. 2004. Out-of-sample extensions for LLE, Isomap, MDS, eigenmaps, and spectral clustering. Advances in neural information processing systems 16, 177–184.

Berkhin, P. 2006. A survey of clustering data mining techniques. In Grouping multidimensional data. New York; Berlin: Springer, 25–71.

Bermanis, A., Averbuch, A. & Coifman, R. R. 2013a. Multiscale data sampling and function extension. Applied and Computational Harmonic Analysis 34 (1), 15–29.

Bermanis, A., Wolf, G. & Averbuch, A. 2013b. Diffusion-based kernel methods on Euclidean metric measure spaces. (Submitted).

Bermanis, A., Wolf, G. & Averbuch, A. 2013c. Measure-based diffusion kernel methods. In 10th International Conference on Sampling Theory and Applications (SampTA). Bremen, Germany: IEEE.

Borg, I. 2005. Modern multidimensional scaling: Theory and applications. New York: Springer.

Boyack, K. W., Klavans, R. & Börner, K. 2005. Mapping the backbone of science. Scientometrics 64 (3), 351–374.

Brachman, R. J. & Anand, T. 1996. Advances in knowledge discovery and data mining. In U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth & R. Uthurusamy (Eds.) Advances in knowledge discovery and data mining. American Association for Artificial Intelligence, 37–57.

Bronstein, A. M., Bronstein, M. M. & Kimmel, R. 2006. Generalized multidimensional scaling: a framework for isometry-invariant partial surface matching. Proceedings of the National Academy of Sciences of the United States of America 103 (5), 1168–1172.

Calhoun, V. D., Liu, J. & Adalı, T. 2009. A review of group ICA for fMRI data and ICA for joint inference of imaging, genetic, and ERP data. Neuroimage 45 (1), S163–S172.

Chakrabarti, S. & Cox, E. 2008. Data mining: Know it all. Elsevier/Morgan Kaufmann Publishers. Know It All.

Chandola, V., Banerjee, A. & Kumar, V. 2009. Anomaly detection: A survey. ACM Computing Surveys (CSUR) 41 (3), 15.

Chang, H. & Yeung, D.-Y. 2006. Robust locally linear embedding. Pattern recognition 39 (6), 1053–1065.

Cheng, J. & Greiner, R. 1999. Comparing Bayesian network classifiers. In Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence. Morgan Kaufmann Publishers Inc., 101–108.

Choi, H. & Choi, S. 2007. Robust kernel Isomap. Pattern Recognition 40 (3), 853–862.

Chung, F. R. K. 1997. Spectral Graph Theory.

Cios, K., Pedrycz, W. & Swiniarski, R. 2007. Data mining: A knowledge discovery approach. Springer.

Cirrincione, G., Cirrincione, M., Herault, J. & van Huffel, S. 2002. The MCA EXIN neuron for the minor component analysis. Neural Networks, IEEE Transactions on 13 (1), 160–187.

Coifman, R., Shkolnisky, Y., Sigworth, F. & Singer, A. 2008. Graph Laplacian tomography from unknown random projections. Image Processing, IEEE Transactions on 17 (10), 1891–1899.

Coifman, R. R., Lafon, S., Lee, A. B., Maggioni, M., Nadler, B., Warner, F. & Zucker, S. W. 2005. Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps. In Proceedings of the National Academy of Sciences of the United States of America, Vol. 102. National Acad Sciences, 7426–7431.

Coifman, R. R. & Hirn, M. J. 2013. Diffusion maps for changing data. Applied and Computational Harmonic Analysis.

Coifman, R. R. & Lafon, S. 2006. Diffusion maps. Applied and Computational Harmonic Analysis 21 (1), 5–30.

Coifman, R. R., Lafon, S., Lee, A. B., Maggioni, M., Nadler, B., Warner, F. & Zucker, S. W. 2005. Geometric diffusions as a tool for harmonic analysis and structure definition of data: Multiscale methods. In Proceedings of the National Academy of Sciences of the United States of America, Vol. 102. National Acad Sciences, 7432–7437.

Coifman, R. R. & Lafon, S. 2006. Geometric harmonics: A novel tool for multiscale out-of-sample extension of empirical functions. Applied and Computational Harmonic Analysis 21 (1), 31–52.

Craven, M. & Shavlik, J. W. 1994. Using sampling and queries to extract rules from trained neural networks. In ICML. Citeseer, 37–45.

Cîmpanu, C. & Ferariu, L. 2012. Survey of data clustering algorithms. Buletinul Institutului Politehnic din Iaşi LVIII (LXII) (3), 23–42.

Damashek, M. et al. 1995. Gauging similarity with n-grams: Language-independent categorization of text. Science 267 (5199), 843–848.

Davenport, T. H., Patil, D. et al. 2012. Data scientist: the sexiest job of the 21st century. Harvard Business Review 90 (10), 70–77.

David, G., Averbuch, A. & Coifman, R. R. 2010. Hierarchical clustering via localized diffusion folders. In Proceedings of the Association for the Advancement of Artificial Intelligence (AAAI) Fall Symposium Series.

David, G. & Averbuch, A. 2012. Hierarchical data organization, clustering and denoising via localized diffusion folders. Applied and Computational Harmonic Analysis 33 (1), 1–23.

David, G. 2009. Anomaly Detection and Classification via Diffusion Processes in Hyper-Networks. Tel-Aviv University. Ph. D. Thesis.

Donoho, D. L. & Grimes, C. 2003. Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data. Proceedings of the National Academy of Sciences 100 (10), 5591–5596.

Duda, R. O., Hart, P. E. & Stork, D. G. 2012. Pattern classification. New York: John Wiley & Sons.

Elhamifar, E. & Vidal, R. 2009. Sparse subspace clustering. In Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. IEEE, 2790–2797.

Elhamifar, E. & Vidal, R. 2013. Sparse subspace clustering: Algorithm, theory, and applications. IEEE Transactions on Pattern Analysis and Machine Intelligence 35 (11).

Fayyad, U., Piatetsky-Shapiro, G. & Smyth, P. 1996a. From data mining to knowledge discovery in databases. AI Magazine 17 (3), 37.

Fayyad, U., Piatetsky-Shapiro, G. & Smyth, P. 1996b. The KDD process for extracting useful knowledge from volumes of data. Communications of the ACM 39, 27–34.

Fayyad, U. M., Piatetsky-Shapiro, G., Smyth, P. et al. 1996c. Knowledge discovery and data mining: Towards a unifying framework. In KDD, Vol. 96, 82–88.

Filippone, M., Camastra, F., Masulli, F. & Rovetta, S. 2008. A survey of kernel and spectral methods for clustering. Pattern recognition 41 (1), 176–190.

Fowlkes, C., Belongie, S., Chung, F. & Malik, J. 2004. Spectral grouping using the Nyström method. Pattern Analysis and Machine Intelligence, IEEE Transactions on 26 (2), 214 -225.

Guyon, I. & Elisseeff, A. 2003. An introduction to variable and feature selection. The Journal of Machine Learning Research 3, 1157–1182.

Han, J., Kamber, M. & Pei, J. 2011. Data Mining: Concepts and Techniques. Elsevier Science & Technology. The Morgan Kaufmann Series in Data Management Systems.

Hand, D., Mannila, H. & Smyth, P. 2001. Principles of data mining. MIT Press. Adaptive computation and machine learning.

Hastie, T., Tibshirani, R. & Friedman, J. 2001. The elements of statistical learning, Vol. 1. Springer New York.

Hein, M. & Audibert, J. 2005. Intrinsic dimensionality estimation of submanifolds in $\mathbb{R}^d$. In Proceedings of the 22nd international conference on Machine learning. ACM, 289–296.

Hevner, A. R., March, S. T., Park, J. & Ram, S. 2004. Design science in information systems research. MIS quarterly 28 (1), 75–105.

Huang, Y., Zha, X. F., Lee, J. & Liu, C. 2013. Discriminant diffusion maps analysis: A robust manifold learner for dimensionality reduction and its applications in machine condition monitoring and fault diagnosis. Mechanical Systems and Signal Processing 34 (1), 277–297.

Huettel, S. A., Song, A. W. & McCarthy, G. 2004. Functional magnetic resonance imaging, Vol. 1. Sinauer Associates Sunderland.

Jain, A. K., Murty, M. N. & Flynn, P. J. 1999. Data clustering: A review. ACM Comput. Surv. 31 (3), 264–323.

Jain, A. K. 2010. Data clustering: 50 years beyond k-means. Pattern Recognition Letters 31 (8), 651–666.

Jolliffe, I. T. 2002. Principal Component Analysis. New York, Berlin, Heidelberg: Springer-Verlag.

Kalman, D. 1996. A singularly valuable decomposition: The SVD of a matrix. College Math Journal 27 (1).

Kannan, R., Vempala, S. & Vetta, A. 2004. On clusterings: Good, bad and spectral. Journal of the ACM (JACM) 51 (3), 497–515.

Kantardzic, M. 2011. Data Mining: Concepts, Models, Methods, and Algorithms. John Wiley & Sons.

Keller, Y., Coifman, R., Lafon, S. & Zucker, S. 2010. Audio-visual group recognition using diffusion maps. Signal Processing, IEEE Transactions on 58 (1), 403–413.

Kohonen, T. 2001. Self-organizing maps, Vol. 30. Berlin; New York: Spriinger.

Kruskal, J. B. 1964. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. Psychometrika 29 (1), 1–27.

Kumar, A. & Iii, H. D. 2011. A co-training approach for multi-view spectral clustering. In Proceedings of the 28th International Conference on Machine Learning (ICML-11), 393–400.

Lafon, S., Keller, Y. & Coifman, R. 2006. Data fusion and multicue data matching by diffusion maps. Pattern Analysis and Machine Intelligence, IEEE Transactions on 28 (11), 1784 -1797.

Lafon, S. & Lee, A. B. 2006. Diffusion maps and coarse-graining: A unified framework for dimensionality reduction, graph partitioning, and data set parameterization. Pattern Analysis and Machine Intelligence, IEEE Transactions on 28 (9), 1393–1403.

Lafon, S. S. 2004. Diffusion maps and geometric harmonics. Yale University. Ph. D. Thesis.

Larose, D. 2006. Data mining methods and models. Wiley-Interscience.

Lee, A. & Wasserman, L. 2010. Spectral connectivity analysis. Journal of the American Statistical Association 105 (491), 1241–1255.

Lee, J. J. A. & Verleysen, M. 2007. Nonlinear dimensionality reduction. New York; London: Springer.

Leydesdorff, L., Carley, S. & Rafols, I. 2013. Global maps of science based on the new web-of-science categories. Scientometrics 94 (2), 589–593.

Liu, H. & Yu, L. 2005. Toward integrating feature selection algorithms for classification and clustering. Knowledge and Data Engineering, IEEE Transactions on 17 (4), 491–502.

Lohr, S. 2012. The age of big data. New York Times 11.

Lu, H., Setiono, R. & Liu, H. 1996. Effective data mining using neural networks. Knowledge and Data Engineering, IEEE Transactions on 8 (6), 957–961.

Luo, F.-L., Unbehauen, R. & Cichocki, A. 1997. A minor component analysis algorithm. Neural Networks 10 (2), 291 - 297.

von Luxburg, U., Belkin, M. & Bousquet, O. 2008. Consistency of spectral clustering. The Annals of Statistics 36 (2), 555–586.

von Luxburg, U., Bousquet, O. & Belkin, M. 2005. Limits of spectral clustering. Advances in Neural Information Processing Systems (NIPS) 17, 857–864.

von Luxburg, U. 2007. A tutorial on spectral clustering. Statistics and computing 17 (4), 395–416.

van der Maaten, L. J. P., Postma, E. O. & van Den Herik, H. J. 2009. Dimensionality reduction: A comparative review. Journal of Machine Learning Research 10, 1–41.

Matthews, P. & Jezzard, P. 2004. Functional magnetic resonance imaging. Journal of Neurology, Neurosurgery & Psychiatry 75 (1), 6–12.

Meila, M. & Shi, J. 2001. A random walks view of spectral segmentation. In 8th International Workshop on Artificial Intelligence and Statistics (AISTATS).

Mika, S., Schölkopf, B., Smola, A. J., Müller, K.-R., Scholz, M. & Rätsch, G. 1998. Kernel PCA and de-noising in feature spaces. In NIPS, Vol. 11, 536–542.

Mitra, S. & Acharya, T. 2003. Data mining: multimedia, soft computing, and bioinformatics. John Wiley.

Mukkamala, S. & Sung, A. H. 2003. A comparative study of techniques for intrusion detection. In Tools with Artificial Intelligence, 2003. Proceedings. 15th IEEE International Conference on. IEEE, 570–577.

Myllymäki, P., Ahtikari, J., Puolamäki, K., Carlsson, C., Sahala, S., Saarnio, R. & Kurki, P. 2011. Strategic Research Agenda for Data to Intelligence (D2I). TIVIT (ICT SHOK).

Müller, K.-R., Mika, S., Rätsch, G., Tsuda, K. & Schölkopf, B. 2001. An introduction to kernel-based learning algorithms. Neural Networks, IEEE Transactions on 12 (2), 181–201.

Nadler, B., Lafon, S., Coifman, R. & Kevrekidis, I. 2008. Diffusion maps – a probabilistic interpretation for spectral embedding and clustering algorithms. In A. Gorban, B. Kégl, D. Wunsch & A. Zinovyev (Eds.) Principal Manifolds for Data Visualization and Dimension Reduction, Vol. 58. Springer Berlin Heidelberg. Lecture Notes in Computational Science and Enginee, 238–260.

Nadler, B., Lafon, S., Coifman, R. R. & Kevrekidis, I. G. 2006. Diffusion maps, spectral clustering and reaction coordinates of dynamical systems. Applied and Computational Harmonic Analysis 21 (1), 113–127.

Nadler, B., Lafon, S., Coifman, R. R. & Kevrekidis, I. G. 2005. Diffusion maps, spectral clustering and eigenfunctions of Fokker-Planck operators. In in Advances in Neural Information Processing Systems 18.

Niu, D., Dy, J. G. & Jordan, M. I. 2010. Multiple non-redundant spectral clustering views. In Proceedings of the 27th international conference on machine learning (ICML-10), 831–838.

Padhy, N., Mishra, P. & Panigrahi, R. 2012. The survey of data mining applications and feature scope. International Journal of Computer Science 2 (3).

Pechenizkiy, M., Puuronen, S. & Tsymbal, A. 2008. Does relevance matter to data mining research? Data Mining: Foundations and Practice 118, 251–275.

di Pietro, R. & Mancini, L. V. 2008. Intrusion detection systems. Springer.

Raducanu, B. & Dornaika, F. 2013. Embedding new observations via sparse-coding for non-linear manifold learning. Pattern Recognition.

de Ridder, D., Kouropteva, O., Okun, O., Pietikäinen, M. & Duin, R. P. 2003. Supervised locally linear embedding. In Artificial Neural Networks and Neural Information Processing—ICANN/ICONIP 2003. Springer, 333–341.

Roweis, S. T. & Saul, L. K. 2000. Nonlinear dimensionality reduction by locally linear embedding. Science 290 (5500), 2323–2326.

Ryman-Tubb, N. F. & d'Avila Garcez, A. 2010. Soar — sparse oracle-based adaptive rule extraction: Knowledge extraction from large-scale datasets to detect credit card fraud. In Neural Networks (IJCNN), The 2010 International Joint Conference on. IEEE, 1–9.

Sagiroglu, S. & Sinanc, D. 2013. Big data: A review. In Collaboration Technologies and Systems (CTS), 2013 International Conference on. IEEE, 42–47.

Salhov, M., Wolf, G. & Averbuch, A. 2012. Patch-to-tensor embedding. Applied and Computational Harmonic Analysis 33 (2), 182–203.

Saxena, A., Gupta, A. & Mukerjee, A. 2004. Non-linear dimensionality reduction by locally linear Isomaps. In Neural Information Processing. Springer, 1038–1043.

Schclar, A., Averbuch, A., Rabin, N., Zheludev, V. & Hochman, K. 2010. A diffusion framework for detection of moving vehicles. Digital Signal Processing 20 (1), 111–122.

Schclar, A. 2008. A diffusion framework for dimensionality reduction. Soft Computing for Knowledge Discovery and Data Mining, 315.

Shi, J. & Malik, J. 2000. Normalized cuts and image segmentation. Pattern Analysis and Machine Intelligence, IEEE Transactions on 22 (8), 888–905.

Shmueli, Y., Wolf, G. & Averbuch, A. 2012. Updating kernel methods in spectral decomposition by affinity perturbations. Linear Algebra and its Applications 437 (6), 1356–1365.

Singer, A. 2006. From graph to manifold Laplacian: The convergence rate. Applied and Computational Harmonic Analysis 21 (1), 128–134.

Singer, A., Erban, R., Kevrekidis, I. G. & Coifman, R. R. 2009. Detecting intrinsic slow variables in stochastic dynamical systems by anisotropic diffusion maps. Proceedings of the National Academy of Sciences 106 (38), 16090–16095.

Soltanolkotabi, M., Elhamifar, E. & Candes, E. 2013. Robust Subspace Clustering. (arXiv preprint arXiv:1301.2603).

Steinwart, I. & Christmann, A. 2008. Support vector machines. New York: Springer.

Strange, H. & Zwiggelaar, R. 2011. A generalised solution to the out-of-sample extension problem in manifold learning. In Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence.

Szalay, A. & Gray, J. 2006. 2020 computing: Science in an exponential world. Nature 440 (7083), 413–414.

Szczuka, M., Janusz, A. & Herba, K. 2012. Semantic clustering of scientific articles with use of DBpedia knowledge base. In Intelligent Tools for Building a Scientific Information Platform. Springer, 61–76.

Tenenbaum, J. B., de Silva, V. & Langford, J. C. 2000. A global geometric framework for nonlinear dimensionality reduction. Science 290 (5500), 2319–2323.

Waltman, L., van Eck, N. J. & Noyons, E. 2010. A unified approach to mapping and clustering of bibliometric networks. Journal of Informetrics 4 (4), 629–635.

Wang, J. 2003. Data mining: Opportunities and challenges. Idea Group Pub.

Williams, C. K. & Barber, D. 1998. Bayesian classification with Gaussian processes. Pattern Analysis and Machine Intelligence, IEEE Transactions on 20 (12), 1342–1351.

Williams, C. K. 2002. On a connection between kernel PCA and metric multidimensional scaling. Machine Learning 46 (1-3), 11–19.

Witten, I., Frank, E. & Hall, M. 2011. Data Mining: Practical Machine Learning Tools and Techniques. Elsevier Science & Technology. Morgan Kaufmann series in data management systems.

Wolf, G. & Averbuch, A. 2013. Linear-projection diffusion on smooth Euclidean submanifolds. Applied and Computational Harmonic Analysis 34 (1), 1–14.

Xu, R. & Wunsch, D. 2009. Clustering. Hoboken, New Jersey: John Wiley & Sons.

Yang, M.-H. 2002. Extended Isomap for pattern classification. In AAAI/IAAI, 224–229.

Zhang, K. & Kwok, J. T. 2009. Density-weighted Nyström method for computing large kernel eigensystems. Neural Comput. 21 (1), 121–146.

APPENDIX 1 IMPLEMENTATION

Listing 1 provides a basic implementation of the diffusion map algorithm. It is written in syntax compatible with Matlab or Octave. With a matrix-oriented programming language most of the steps can be expressed as matrix multiplications.

```
function [V, l] = diffusion_map(x, epsilon, kernel, a, p, t)
% Creates a dimension-reduced version of input matrix x.
% Compares all the time points to each other and maps
% using the eigenvalues and eigenvectors.
% Needs pdist (in Octave statistics package or Matlab statistics toolbox).
% 2012 Tuomo Sipola (tuomo.sipola@jyu.fi)
%  [V, l] = diffusion_map(x, epsilon, kernel, a, p, t)
% Parameters:
%  x      Input data which should be in format time x parameters.
%  epsilon Epsilon value for the kernel.
%  kernel Distance function, 'euclidean' (default). Optional.
%  a      The alpha value for diffusion families, 0, 0.5 or 1. Optional.
%  p      To which power the distance is put.
%  t      How many time steps to use, default 1. Optional.
% Returns:
%  V      Eigenvectors in format values x vectors.
%  l      Eigenvalues in descending order.

if nargin < 3
   kernel = 'euclidean';
end
if nargin < 4
   a = 0;
end
if nargin < 5
   p = 2;
end
if nargin < 6
   t = 1;
end

% Create the kernel.
K = squareform(pdist(x, kernel));
K = exp( - K.^p ./ epsilon );

% Create different families of diffusions.
if a > 0
   Q = diag(sum(K));
   W = Q^-a * K * Q^-a;
else
   W = K;
end
```

```

% Calculate the integral of weights.
d = sum(W);

% Create a Markov probability matrix and symmetrize it.
D2 = diag(1./sqrt(d));
S = D2 * W * D2;

% Calculate the 32 first eigenvalues and eigenvectors.
[U, L, T] = svds(S, 32);
l = diag(L.^t);

% Take right eigenvectors and normalize with the constant first one.
V = D2 * U;
V = V / V(1,1);
```

LISTING 1 Diffusion map implementation

ORIGINAL PAPERS

PI

ANOMALY DETECTION FROM NETWORK LOGS USING DIFFUSION MAPS

by

Tuomo Sipola, Antti Juvonen and Joel Lehtonen 2011

IFIP Advances in Information and Communication Technology, Vol. 363,
pp. 172–181

Anomaly Detection from Network Logs Using Diffusion Maps

Tuomo Sipola, Antti Juvonen, and Joel Lehtonen*

Department of Mathematical Information Technology
University of Jyväskylä, Finland
tuomo.sipola@jyu.fi antti.juvonen@jyu.fi joel.lehtonen@iki.fi

Abstract. The goal of this study is to detect anomalous queries from network logs using a dimensionality reduction framework. The fequencies of 2-grams in queries are extracted to a feature matrix. Dimensionality reduction is done by applying diffusion maps. The method is adaptive and thus does not need training before analysis. We tested the method with data that includes normal and intrusive traffic to a web server. This approach finds all intrusions in the dataset.

Keywords: intrusion detection, anomaly detection, n-grams, diffusion map, data mining, machine learning

1 Introduction

The goal of this paper is to present an adaptive way to detect security attacks from network log data. All networks and systems can be vulnerable to different types of intrusions. Such attacks can exploit e.g. legitimate features, misconfigurations, programming mistakes or buffer overflows [15]. This is why *intrusion detection systems* are needed. An intrusion detection system gathers data from the network, stores this data to logfiles and analyzes it to find malicious or anomalous traffic [19]. Systems can be vulnerable to previously unknown attacks. Because usually these attacks differ from the normal network traffic, they can be found using anomaly detection [2].

In modern networks clients request and send information using queries. In HTTP traffic these queries are strings containing arguments and values. It is easy to manipulate such queries to include malicious attacks. These injection attacks try to create requests that corrupt the server or collect confidential information [18]. Therefore, it is important to analyze data collected from logfiles.

An *anomaly* is a pattern in data that is different from the well defined normal data [2]. In network data, this usually means an intrusion. There are two main approaches for detecting intrusions from network data: *misuse detection* and *anomaly detection* [19]. Misuse detection means using predefined attack signatures to detect the attacks, which is usually accurate but detecting new types of

[*] Now with C2 SmartLight Oy.

attacks is not possible. In anomaly detection the goal is to find actions that somehow deviate from normal traffic. This way it is possible to detect previously unknown attacks. However, not all anomalous traffic is intrusive. This means there might be more false alarms. Different kinds of machine learning based methods, such as self-organizing maps and support vector machines, have been used in anomaly detection [20, 23]. Information about other anomaly detection methods can be found in the literature [19]. *Unsupervised anomaly detection* techniques are most usable in this case, because no normal training data is required [2].

This study takes the approach of dimensionality reduction. Diffusion map is a manifold learning method that maps high-dimensional data to a low-dimensional diffusion space [5]. It provides tools for visualization and clustering [6]. The basic idea behind any manifold learning method is the eigen-decomposition of a similarity matrix. By unfolding the manifold it reveals the underlying structure of the data that is originally embedded in the high-dimensional space [1]. Diffusion maps have been applied to various data mining problems. These include vehicle classification by sound [21], music tonality [10], sensor fusion [12], radio network problem detection [25] and detection of injection attacks [8]. Advantages of this approach are that the dimensionality of the data is reduced and that it can be used unsupervised [2].

2 Method

2.1 Feature extraction

First let us define an n-gram as a consecutive sequence of n characters [7]. For example, the string *ababc* contains unique 2-grams *ab*, *ba* and *bc*. The 2-gram *ab* appears twice, thus having frequency of 2. A list of tokens of text can be represented with a vector consisting of n-gram frequencies [7]. Feature vector describing this string would be $x_{ababc} = [2,\ 1,\ 1]$. The only features extracted are n-gram frequencies. Furthermore, syntactic features of the input strings might reveal the differences between normal and anomalous behavior. Computed n-grams can extract features that describe these differences.

The frequencies are collected to a feature matrix X whose rows correspond to lines in logfiles and columns to features. These n-gram frequencies are key-value fields, variable-length by definition. Key strings are ignored and 2-grams are produced from each parameter value. The count of occurrences of every occurring 2-gram is summed. In practice n-gram tables produced from real-life data are very sparse, containing columns in which there are only zero occurrences. To minimize the number of columns, the processing is done in two passes. If a column contains no variation between entries, that column is not present in the final numeric matrix X. That makes it reasonable to use diffusion maps to process n-gram tables directly with no further preprocessing.

2.2 Dimensionality reduction

The number of extracted features is so large that dimensionality reduction is performed using diffusion maps. It is a manifold learning method that embeds the original high-dimensional space into a low-dimensional diffusion space. Anomaly detection and clustering are easier in this embedded space [6].

The recorded data describe the behavior of the system. Let this data be $X = \{x_1, x_2, \ldots, x_N\}, x_i \in \mathbb{R}^n$. Here N is the number of samples and n the dimension of the original data. In practice the data is a $N \times n$ matrix with features as columns and each sample as rows.

At first, an affinity matrix W is constructed. This calculation takes most of the computation time. The matrix describes the distances between the points. This study uses the common Gaussian kernel with Euclidean distance measure, as in equation 1 [6, 16].

$$W_{ij} = \exp\left(-\frac{||x_i - x_j||^2}{\epsilon}\right) \tag{1}$$

The affinity neighborhood is defined by ϵ. Choosing the parameter ϵ is not trivial. It should be large enough to cover the local neighborhood but small so that it does not cover too much of it [21].

The rows of the affinity matrix are normalized using the diagonal matrix D, which contains the row sums of the matrix W on its diagonal.

$$D_{ii} = \sum_{j=1}^{N} W_{ij} \tag{2}$$

P expresses normalization that represents the probability of transforming from one state to another. Now the sum of each row is 1.

$$P = D^{-1}W \tag{3}$$

Next we need to obtain the eigenvalues of this transition probability matrix. The eigenvalues of P are the same with the conjugate matrix in equation 4. The eigenvectors of P can be derived from $\tilde{P}$ as shown later.

$$\tilde{P} = D^{\frac{1}{2}} P D^{-\frac{1}{2}} \tag{4}$$

If we substitute the P in equation 4 with the one in equation 3, we get the symmetric probability matrix $\tilde{P}$ in equation 5. It is called the normalized graph Laplacian [4] and it preserves the eigenvalues [16].

$$\tilde{P} = D^{-\frac{1}{2}} W D^{-\frac{1}{2}} \tag{5}$$

This symmetric matrix is then decomposed with singular value decomposition (SVD). Because $\tilde{P}$ is a normal matrix, spectral theorem states that such a matrix is decomposed with SVD: $\tilde{P} = U \Lambda U^*$. The eigenvalues on the diagonal of $\Lambda = \mathrm{diag}([\lambda_1, \lambda_2, \ldots, \lambda_N])$ correspond to the eigenvalues of the same matrix $\tilde{P}$ because it is symmetric. Matrix $U = [u_1, u_2, \ldots, u_N]$ contains in its columns the N eigenvectors u_k of $\tilde{P}$. Furthermore, because $\tilde{P}$ is conjugate with P, these two matrices share their eigenvalues. However, to calculate the right eigenvectors v_k of P, we use equation 6 and get them in the columns of $V = [v_1, v_2, \ldots, v_N]$ [16].

$$V = D^{-\frac{1}{2}}U \tag{6}$$

The coordinates of a data point in the embedded space using eigenvalues in Λ and eigenvectors in V are in the matrix Ψ in equation 7. The rows correspond to the samples and the columns to the new embedded coordinates [6].

$$\Psi = V\Lambda \tag{7}$$

Strictly speaking, the eigenvalues should be raised to the power of t. This scale parameter t tells how many time steps are being considered when moving from data point to another. Here we have set it $t = 1$ [6].

With suitable ϵ the decay of the spectrum is fast. Only d components are needed for the diffusion map for sufficient accuracy. It should be noted that the first eigenvector v_1 is constant and is left out. Using only the next d components the diffusion map for original data point x_i is presented in equation 8. Here $v_k(x_i)$ corresponds to the ith element of kth eigenvector [6].

$$\Psi_d : x_i \rightarrow [\lambda_2 v_2(x_i), \lambda_3 v_3(x_i), \ldots, \lambda_{d+1} v_{d+1}(x_i)] \tag{8}$$

This diffusion map embeds the known point x_i to a d-dimensional space. Dimension of the data is reduced from n to d. If desired, the diffusion map may be scaled by dividing the coordinates with λ_1.

2.3 Anomaly detection

After obtaining the low-dimensional presentation of the data it is easier to cluster the samples. Because spectral methods reveal the manifold, this clustering is called spectral clustering. This method reveals the normal and anomalous samples [13]. Alternatively, k-means or any other clustering method in the low-dimensional space is also possible [17]. Another approach is the density-based method [25].

Only the first few low-dimensional coordinates are interesting. They contain most of the information about the manifold structure. We use only the dimension corresponding to second eigenvector to determine the anomality of the samples. At 0, this dimension is divided into two clusters. The cluster with more samples is considered normal behavior. Conversely, the points in the other cluster are considered anomalous [22, 11, 13]. The second eigenvector acts as the separating feature for the two clusters in the low-dimensional space. The second eigenvalue is the solution to the normalized cut problem, which finds small weights between clusters but strong internal ties. This spectral clustering has probabilistic interpretation: grouping happens through similarity of transition probabilities between clusters [22, 14].

3 Results

3.1 Data acquisition

The data is acquired from a large real-life web service. The logfiles contain mostly normal traffic, but they also include anomalities and actual intrusions. The logfiles are from several Apache servers and are stored in *combined log format*. Listing below provides an example of a single logline. It includes information about the user's IP-address, time and timezone, the HTTP request including used resource and parameters, Apache server response code, amount of data sent to the user, the web page that was requested and used browser software.

```
127.0.0.1 - - [01/January/2011:00:00:01 +0300]
"GET /resource?parameter1=value1&parameter2=value2 HTTP/1.1"
200 2680 "http://www.address.com/webpage.html"
"Mozilla/5.0 (SymbianOS/9.2;...)"
```

The access log of a web site contains entries from multiple, distinct URLs. Most of them point to static requests like images, CSS files, etc. We are not focused to find anomalies at those requests because it is not possible to inject code via static requests unless there are major deficiencies in the HTTP server itself. Instead, we are focused in finding anomalies from dynamic requests because those requests are handled by the Web application, which is run behind the HTTP server.

To reach this goal, the access log entries are grouped by the resource URL. That is the part between host name and parameters in the HTTP URL scheme. Those resources containing only HTTP GET requests with no parameters are ignored. Each remaining resource is converted to a separate numerical matrix. In this matrix, a row represents a single access log entry, and a column represents an extracted feature.

Feature extraction is done in two passes. In the first pass the number of features is determined, and in the second pass the resulting matrix is produced. In our study we extracted the number of occurrences of 2-grams produced from HTTP GET parameters. These frequencies are normalized with logarithm in order to scale them. This ensures that the distances between the samples are comparable.

3.2 Data analysis

To measure the effectiveness of the method the data is labeled so that classification accuracy can be measured. However, this labeling is not used for training the diffusion map. The class labels are not input for the method.

Diffusion map reveals the structure of the data, and all the anomalies are detected. The n-gram features of the data are mapped to a lower dimensions. Figure 1 shows the resulting low-dimensional diffusion space with $\epsilon = 100$. The normal behavior lies in the dense area to the lower right corner. Anomalous points are to the left of 0.

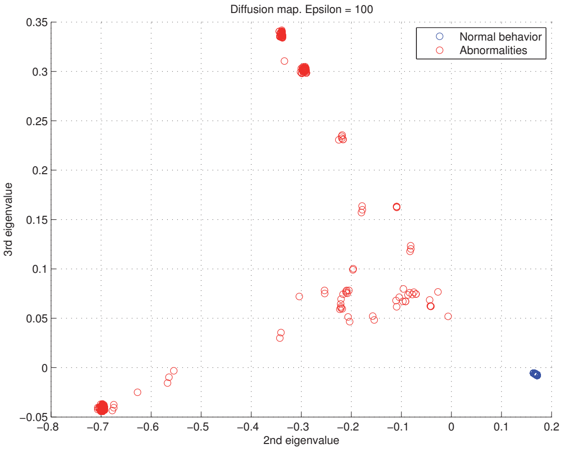

Fig. 1. Two-dimensional diffusion map of the dataset.

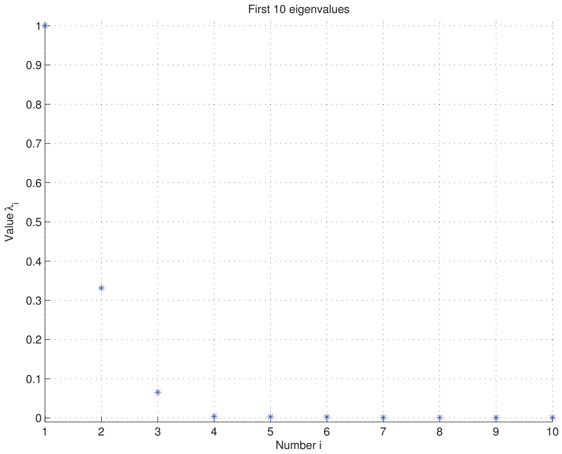

Fig. 2. Eigenvalues of transition matrix with $\epsilon = 100$.

Figure 2 shows that the eigenvalues converge rapidly with $\epsilon = 100$. This means that the first few eigenvalues and eigenvectors cover most of the differences observed in the data. The first value is 1 and corresponds to the constant eigenvector that is left out in the analysis. Eigenvalues $\lambda_2 = 0.331$ and $\lambda_3 = 0.065$ cover large portions of the data when compared to the rest that have values below 0.005.

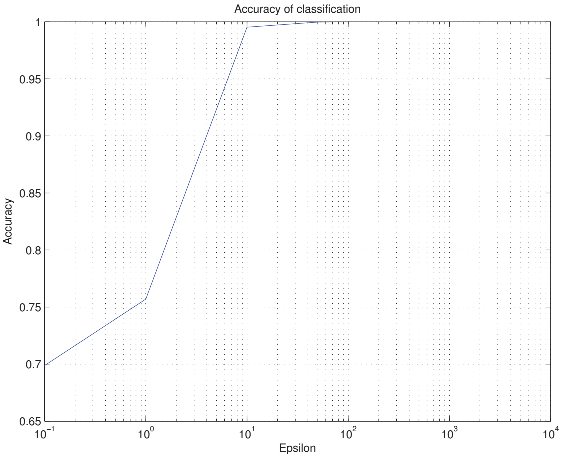

Fig. 3. Accuracy of classification changes when the parameter ϵ is changed.

Classification is tested with different values of ϵ, which defines the neighborhood for diffusion map. Accuracy of classification is defined as $accuracy = (tp + tn)/(tp + fp + fn + tn)$. Figure 3 shows how the accuracy of classification changes when ϵ is changed. Higher values of ϵ result in better accuracy. Precision of classification is defined $precision = tp/(tp+fp)$. The precision stays at 1 once any anomalies are detected, which means that all the anomalies detected are real anomalies regardless of the accuracy [9, p. 361].

For comparison, principal component analysis (PCA) is performed on the same normalized feature matrix [9, p. 79]. Results are very similar to the diffusion map approach, because of the simple structure of the feature matrix. Furthermore, PCA reaches the same accuracy and precision as diffusion map. The low-dimensional presentation is also very similar. Figure 4 shows the first two coordinates of PCA.

We also apply support vector machines (SVM) to the same data [9, p. 337–344]. LIBSVM implementation is used [3]. We use one-class SVM with RBF kernel function. A subset of the data is used in the model selection for SVM (500 lines randomly selected). Then the rest of the data is used to test the method.

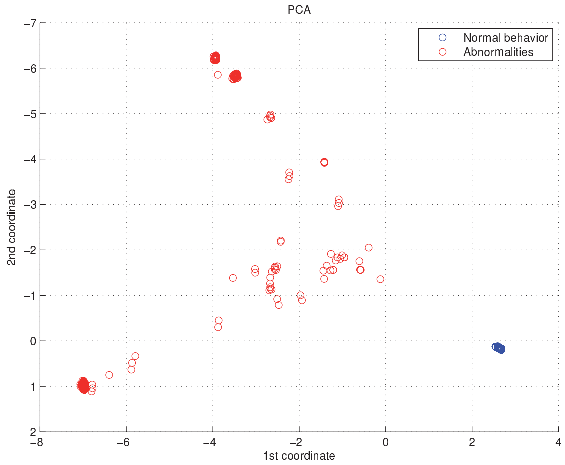

Fig. 4. PCA of the dataset, first two coordinates. The Y-axis of this figure has been reversed for better visual comparison with diffusion map.

The data labels are unknown, so the training data is not ”clean” and contains some intrusions as well. It is possible to find the right parameters (ν and γ) for model selection if pre-specified true positive rate is known. The parameters which give a similar cross-validation accuracy can be selected [3]. However, this kind of information is not available. Fully automatic parameter selection for OC-SVM could be achieved by using more complicated methods, such as evolving training model method [24]. In this study the parameter selection is done manually. At best the accuracy is 0.999 and precision 0.998.

4 Conclusion

The goal of this study is to find security attacks from network data. This goal is met since all the known attacks are found. The proposed anomaly detection scheme could be used for query log analysis in real situations. In practice the boundary between normal and anomalous might not be as clear as in this example. However, the relative strangeness of the sample could indicate how severe an alert is.

The diffusion map framework adapts to the log data. It assumes that the data lies on a manifold, and finds a coordinate system that describes the global structure of the data. These coordinates could be used for further analysis of characteristics of anomalous activities.

Because all the methods perform extremely well, the data in question is rather sparse and the discriminating features are quite evident from the feature matrix.

This is the merit of n-gram feature extraction which creates a feature space that separates the normal behavior in a good manner. The features describe the data clearly, and they are easy to process afterwards.

One advantage of the diffusion map methodology is that it has only one meta-parameter, ϵ. It can be estimated with simple interval search. If for some reason the threshold sensitivity needs to be changed, ϵ gives the flexibility to adapt to the global structure. For comparison, the SVM we used has two parameters, ν and γ. Searching the best parameters for the application gets more difficult as the number of parameters increases.

The presented anomaly detection method performs well on real data. As an unsupervised algorithm this approach is well suited to finding previously unknown intrusions. This method could be applied to offline clustering as well as extended to a real-time intrusion detection system.

Acknowledgements

The authors thank Professors Amir Averbuch, Timo Hämäläinen and Tapani Ristaniemi for their continued support. Thanks are extended to Juho Knuuttila and Kristian Siljander for useful discussions and ideas.

References

1. Bengio, Y., Delalleau, O., Roux, N.L., Paiement, J.F., Vincent, P., Ouimet, M.: Feature Extraction, chap. Spectral Dimensionality Reduction, pp. 519–550. Studies in Fuzziness and Soft Computing, Springer Berlin, Heidelberg (2006)
2. Chandola, V., Banerjee, A., Kumar, V.: Anomaly detection: A survey. ACM Comput. Surv. 41(3), 1–58 (2009)
3. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines (2001), software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`
4. Chung, F.R.K.: Spectral Graph Theory, p. 2. AMS Press, Providence, R.I (1997)
5. Coifman, R.R., Lafon, S., Lee, A.B., Maggioni, M., Nadler, B., Warner, F., Zucker, S.W.: Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps. In: Proceedings of the National Academy of Sciences of the United States of America. vol. 102, p. 7426 (2005)
6. Coifman, R.R., Lafon, S.: Diffusion maps. Applied and Computational Harmonic Analysis 21(1), 5–30 (2006)
7. Damashek, M.: Gauging similarity with n-grams: Language-independent categorization of text. Science 267(5199), 843 (1995)
8. David, G.: Anomaly Detection and Classification via Diffusion Processes in Hyper-Networks. Ph.D. thesis, Tel-Aviv University (2009)
9. Han, J., Kamber, M.: Data mining: concepts and techniques. Morgan Kaufmann (2006)
10. İzmirli, Ö.: Tonal-atonal classification of music audio using diffusion maps. In: 10th International Society for Music Information Retrieval Conference (ISMIR 2009) (2009)
11. Kannan, R., Vempala, S., Vetta, A.: On clusterings: Good, bad and spectral. J. ACM 51, 497–515 (May 2004)

12. Keller, Y., Coifman, R., Lafon, S., Zucker, S.: Audio-visual group recognition using diffusion maps. Signal Processing, IEEE Transactions on 58(1), 403–413 (2010)
13. von Luxburg, U.: A tutorial on spectral clustering. Statistics and Computing 17, 395–416 (2007)
14. Meila, M., Shi, J.: Learning segmentation by random walks. In: NIPS. pp. 873–879 (2000)
15. Mukkamala, S., Sung, A.: A comparative study of techniques for intrusion detection (2003)
16. Nadler, B., Lafon, S., Coifman, R., Kevrekidis, I.G.: Diffusion maps – a probabilistic interpretation for spectral embedding and clustering algorithms. In: Barth, T.J., Griebel, M., Keyes, D.E., Nieminen, R.M., Roose, D., Schlick, T., Gorban, A.N., Kégl, B., Wunsch, D.C., Zinovyev, A.Y. (eds.) Principal Manifolds for Data Visualization and Dimension Reduction, Lecture Notes in Computational Science and Engineering, vol. 58, pp. 238–260. Springer Berlin Heidelberg (2008)
17. Ng, A.Y., Jordan, M.I., Weiss, Y.: On spectral clustering: Analysis and an algorithm. In: Advances in Neural Information Processing Systems 14. pp. 849–856. MIT Press (2001)
18. Nguyen-Tuong, A., Guarnieri, S., Greene, D., Shirley, J., Evans, D.: Automatically hardening web applications using precise tainting. In: Sasaki, R., Qing, S., Okamoto, E., Yoshiura, H. (eds.) Security and Privacy in the Age of Ubiquitous Computing, IFIP Advances in Information and Communication Technology, vol. 181, pp. 295–307. Springer Boston (2005)
19. Patcha, A., Park, J.: An overview of anomaly detection techniques: Existing solutions and latest technological trends. Computer Networks 51(12), 3448–3470 (2007)
20. Ramadas, M., Ostermann, S., Tjaden, B.: Detecting anomalous network traffic with self-organizing maps. In: Vigna, G., Jonsson, E., Kruegel, C. (eds.) Recent Advances in Intrusion Detection. pp. 36–54. Springer (2003)
21. Schclar, A., Averbuch, A., Rabin, N., Zheludev, V., Hochman, K.: A diffusion framework for detection of moving vehicles. Digital Signal Processing 20(1), 111–122 (2010)
22. Shi, J., Malik, J.: Normalized cuts and image segmentation. Pattern Analysis and Machine Intelligence, IEEE Transactions on 22(8), 888 –905 (2000)
23. Tran, Q., Duan, H., Li, X.: One-class support vector machine for anomaly network traffic detection. China Education and Research Network (CERNET), Tsinghua University, Main Building 310 (2004)
24. Tran, Q.A., Zhang, Q., Li, X.: Evolving training model method for one-class svm. In: Systems, Man and Cybernetics, 2003. IEEE International Conference on. vol. 3, pp. 2388–2393 (2003)
25. Turkka, J., Ristaniemi, T., David, G., Averbuch, A.: Anomaly detection framework for tracing problems in radio networks. In: Proc. to ICN 2011 (2011)

PII

DIMENSIONALITY REDUCTION FRAMEWORK FOR DETECTING ANOMALIES FROM NETWORK LOGS

by

Tuomo Sipola, Antti Juvonen and Joel Lehtonen 2012

Engineering Intelligent Systems, Vol. 20, Iss. 1–2, pp. 87–97

Dimensionality Reduction Framework for Detecting Anomalies from Network Logs

Tuomo Sipola Antti Juvonen Joel Lehtonen[*]

tuomo.sipola@jyu.fi antti.k.a.juvonen@jyu.fi joel.lehtonen@iki.fi

Department of Mathematical Information Technology
University of Jyväskylä
Jyväskylä, Finland

Abstract

Dynamic web services are vulnerable to a multitude of intrusions that could be previously unknown. Server logs contain vast amounts of information about network traffic, and finding attacks from these logs improves the security of the services. In this research features are extracted from HTTP query parameters using 2-grams. We propose a framework that uses dimensionality reduction and clustering to identify anomalous behavior. The framework detects intrusions from log data gathered from a real network service. This approach is adaptive, works on the application layer and reduces the number of log lines that needs to be inspected. Furthermore, the traffic can be visualized.

Keywords: intrusion detection, anomaly detection, n-grams, diffusion map, data mining, machine learning

1 Introduction

The goal of this paper is to present an adaptive way to detect security attacks from network log data. All networks and systems can be vulnerable to different types of intrusions. Such attacks can exploit e.g. legitimate features, misconfigurations, programming mistakes or buffer overflows [1]. This is why *intrusion detection systems* are needed. An intrusion detection system gathers data from the network, stores these data to log files and analyzes them to find malicious or anomalous traffic [2]. Systems can be vulnerable to previously unknown attacks, commonly known as zero-day attacks [3]. Because usually these attacks differ from the normal network traffic, they can be found using anomaly detection [4].

In modern networks clients request and send information using queries. In HTTP traffic these queries are strings containing arguments and values. It is easy to manipulate such queries to include malicious attacks. These injection attacks try to create requests that corrupt the server or collect confidential information [5]. Therefore, it is important to analyze the collected data in log files. Most intrusion detection systems

[*] Now with C2 SmartLight Ltd.

The original print version appeared as: Sipola, T., Juvonen, A., Lehtonen, J. (2012). Dimensionality Reduction Framework for Detecting Anomalies from Network Logs. *Engineering Intelligent Systems* 20(1), 87–97.

analyze TCP packet data. There are not many application layer IDS systems available. Because the HTTP log data are very different from network packet data, they both need to be analyzed. Different attacks can be performed on different layers.

An *anomaly* is a pattern in data that is different from the well defined normal data [4]. In network data, this usually means an intrusion. There are two main approaches for detecting intrusions from network data: *misuse detection* and *anomaly detection* [2]. Misuse detection means using predefined attack signatures to detect the attacks, which is usually accurate but detecting new types of attacks is not possible. In anomaly detection the goal is to find actions that somehow deviate from normal traffic. This way it is possible to detect previously unknown attacks. However, not all anomalous traffic is intrusive. This means there might be more false alarms. Different kinds of machine learning based methods, such as self-organizing maps and support vector machines, have been used in anomaly detection [6, 7]. Information about other anomaly detection methods can be found in the literature [2]. *Unsupervised anomaly detection* techniques are most usable in this case, because no normal training data are available [4]. These techniques work without prior knowledge of attack patterns. This kind of adaptive framework is suitable for *a posteriori* network log analysis.

This study takes the approach of dimensionality reduction. Because the number of dimensions of the feature space grows large and sparse when analyzing textual information, such as log files, this is one of the most feasible techniques. Furthermore, the sparsity of data suggests about the underlying low dimensional structure. Almost the same amount of information can be represented with lower number of dimensions. Diffusion map is a manifold learning method that maps high-dimensional data to a low-dimensional diffusion space [8]. It provides tools for visualization and clustering [9]. The basic idea behind any manifold learning method is the eigendecomposition of a similarity matrix. By unfolding the manifold it reveals the underlying structure of the data that is originally embedded in the high-dimensional space [10]. Diffusion maps have been applied to various data mining problems. These include vehicle classification by sound [11], music tonality [12], sensor fusion [13], radio network problem detection [14, 15] and detection of injection attacks [16]. In addition to the advantage of reduced number of dimensions, the approach can be used for unsupervised learning [4].

2 Related research

Kruegel and Vigna [17] analyzed the parameter values of HTTP queries. The static queries with no parameters were removed. The underlying assumption is that attack patterns differ from normal traffic and that this difference can be expressed quantitatively. They used several different analyzing methods, such as attribute length and character distribution. The learning was based on previous data. The data were not labeled. The analysis of character distribution is similar to our research, because essentially the characters are *n*-grams with the length 1. We use 2-grams for higher detection rates, but we will also get more dimensions in the data matrices.

Hubballi et al. [18] used layered higher order *n*-grams for detecting intrusions. However, this analysis was not done on application layer data, but on the network packet payloads. Higher order *n*-grams are *n*-grams where $n > 2$. This means that the method is computationally more expensive, but rare events might be detected more accurately. The *n*-grams are organized into bins based on their frequency. The analysis starts with 1-grams, and it moves to higher *n*-grams incrementally to get higher accuracy. In the research the number of distinct and unique *n*-grams went up almost linearly as n increased. Therefore, using higher order n-grams might not be as complex in practice as it could be theoretically. For example, the theoretical maximum number for 3-grams in ascii-characters is 256^3, which is considerably higher than the case with 2-grams.

Dimensionality reduction has been discussed in the context of anomaly detection from networks. Ringberg et al. studied the IP packet data and tried to detect anomalies using principal component analysis. They also identified the main challenges when using principal component dimensionality reduction approach. The finding about large anomalies contaminating the subspace is relevant also to our research. However, their network architecture is more complex than ours [19]. Callegari et al. analyzed similar packet data [20]. These studies used low-level IP packet datasets that need specific aggregation before they can be processed. Our research concerns the application level log data, which is text, while the IP packet datasets are numeric. In addition, we compare the results of principal component analysis and diffusion maps.

Diffusion maps have been applied in the network security context. David explored the use of diffusion map methods to find injection attacks in hyper-networks. His data included SQL injection examples that used a similar feature extraction as our research. The n-gram feature extraction was applied to tokenized SQL [16]. Our research, in contrast, focuses on the raw textual queries. Furthermore, David and Averbuch used a localized diffusion folder approach to classify network protocols, among other examples. Their data contains low-level features such as duration and the number of bytes [21]. However, our data comes from the application layer of the network, specifically web server logs. These are different from the low-level network features and contain lots of textual information in the form of queries. Moreover, we use the theoretical framework of spectral clustering as the basis of our research.

3 Methodology

Straightforward numerical methods are difficult to apply to textual data such as log files. Therefore, log data must be transformed into feature space. This mapping of textual information to numerical matrix enables mathematical analysis of the original log lines.

However, this leads to a large number of dimensions in the feature space. For efficient analysis, classification and visualization the number of dimensions must be reduced. This gives the opportunity to use a multitude of classification algorithms.

The proposed method consists of the following steps:

1. Removing lines that do not contain parameters.
2. Feature extraction from the log line using 2-grams.
3. Dimensionality reduction of the features.
4. Classifying the lines either as normal or attack.

After these steps the log file can be visualized as a figure where the attacks are more easily seen than from a text file. Furthermore, the suspected attack lines can be inspected in more depth. This facilitates finding abnormal activities because only these suspected lines are inspected, instead of thousands in the original log.

3.1 Feature extraction

The features include 2-grams from HTTP query parameters. The log files are simple text files where each line represents one query sent from the client to the server. Extracting the true intention of the query is challenging, and the text needs to be converted to a more machine-friendly format. The feature extraction essentially means converting this textual data into numerical matrices.

First let us define an *n*-gram as a consecutive sequence of *n* characters [22]. *N*-gram is a substring with length of *n*. For example, the string *ababc* contains unique 2-grams *ab*, *ba* and *bc*. The 2-gram *ab* appears twice, thus having frequency of 2. A list of tokens of text can be represented with a vector consisting of *n*-gram frequencies [22]. Feature vector describing this string would be $x_{ababc} = [2,\ 1,\ 1]$. The only features extracted are *n*-gram frequencies. Furthermore, syntactic features of the input strings might reveal the differences between normal and anomalous behavior. Computed *n*-grams can extract features that describe these differences. It is assumed that an anomalous query contains some text in the parameter part that differs from normal behaviour. This means that it must contain some *n*-grams that appear rarely in the data.

Here is an example of constructing the feature matrix using the *n*-gram analysis process with two words, *anomaly* and *analysis*. From these words we get the unique 2-grams `an`, `no`, `om`, `ma`, `al`, `ly`, `na`, `ys`, `si` and `is`. From this information we can construct a matrix with the *n*-gram frequencies.

an	no	om	ma	al	ly	na	ys	si	is
1	1	1	1	1	1	0	0	0	0
1	0	0	0	1	1	1	1	1	1

The feature matrix is constructed in this way, including all the different strings that appear in the parameter fields on each query. Each log line corresponds to a row in the matrix. From this we can see that there are 10 unique 2-grams in this example. The logs are ascii-coded, so they can contain 256 different characters. The theoretical maximum number for unique 2-grams using ascii-characters is 256^2, but in practice we did not get even near to that number. However, in a very varied and big dataset the number of dimensions could get very high.

The frequencies are collected to a feature matrix X. These *n*-gram frequencies are key-value fields, variable-length by definition. Key strings are ignored and 2-grams are produced from each parameter value. The count of occurrences of every occurring 2-gram is summed. In practice *n*-gram tables produced from real life data are very sparse, containing columns in which there are only zero occurrences. To minimize the number of columns, the processing is done in two passes, first determining the number of unique *n*-grams and then analyzing the frequencies. If a column contains no variation between entries, that column is not present in the final numeric matrix X. Therefore, only the columns that actually contain some useful information about the features are included in the analysis.

With this preprocessing technique it is possible to use *n*-grams whose value of *n* is higher than 2. However, the number of unique *n*-grams will increase and therefore the number of dimensions will increase as well.

3.2 Dimensionality reduction

The number of extracted features is so large that dimensionality reduction is performed using principal component analysis and diffusion map. Diffusion map is a manifold learning method that embeds the original high-dimensional space into a low-dimensional diffusion space. Anomaly detection and clustering are easier in this embedded space [9].

The recorded data describe the behavior of the system. Let this data be $X = \{x_1, \dots, x_N\}$, $x_i \in \mathbb{R}^n$. Here N is the number of samples and n the dimension of the original data. In practice the data are in a $N \times n$ matrix with features as columns and each sample as rows.

3.2.1 Principal component analysis

Principal component analysis (PCA) tries to extract orthogonal components maximizing their variance from the data. This simplifies the representation of the information within the data and also facilitates the analysis of the structure and features in the data. The principal components are linear combinations of the original features. The first principal component contains the largest amount of variance. PCA reveals the most information in terms of variance, but this does not necessarily mean that it separates different clusters in an optimal way [23, 24, 25].

PCA performs the eigendecomposition on the covariance matrix C of the centered data matrix X_c. The decomposition $C = U\Lambda U^*$ gives the eigenvectors in U that map the points in X to a low-dimensional space. This mapping can be calculated with $X_{PCA} = XU$. Another approach is to take the singular value decomposition (SVD) of the original matrix X. One way to interpret this is as rotation of axes to find the most important features. The new principal components are in the direction of most variance in the data and thus represent the most differentiating combination of features [23, 24, 25].

As with many dimensionality reduction methods using eigendecompositions, the number of selected components becomes a problem. One way to do this is to seek for the eigengap, i.e. a big change of eigenvalues. This way the eigenvalues reveal the principal components that cover most of the variance [23, 24, 25].

PCA is a linear method and has difficulties finding nonlinear dependencies between features. It has initial assumptions that restrict its use for latent variable separation and nonlinear dimensionality reduction [25].

3.2.2 Diffusion map

At first, an affinity matrix W is constructed. This calculation takes most of the computation time. The matrix describes the distances between the points. This study uses the common Gaussian kernel with Euclidean distance measure, as in equation 1 [9, 26].

$$W_{ij} = \exp\left(-\frac{||x_i - x_j||^2}{\varepsilon}\right) \tag{1}$$

The affinity neighborhood is defined by ε. Choosing the parameter ε is not trivial. It should be large enough to cover the local neighborhood but small so that it does not cover too much of it [11].

The rows of the affinity matrix are normalized using the diagonal matrix D, which contains the row sums of the matrix W on its diagonal.

$$D_{ii} = \sum_{j=1}^{N} W_{ij} \tag{2}$$

P expresses normalization that represents the probability of transforming from one state to another. Now the sum of each row is 1.

$$P = D^{-1}W \tag{3}$$

Next we need to obtain the eigenvalues of this transition probability matrix. The eigenvalues of P are the same with the conjugate matrix in equation 4. The eigenvectors of P can be derived from $\tilde{P}$ as shown later.

$$\tilde{P} = D^{\frac{1}{2}}PD^{-\frac{1}{2}} \tag{4}$$

If we substitute the P in equation 4 with the one in equation 3, we get the symmetric probability matrix $\tilde{P}$ in equation 5. It is called the normalized graph Laplacian [27] and it preserves the eigenvalues [26].

$$\tilde{P} = D^{-\frac{1}{2}}WD^{-\frac{1}{2}} \tag{5}$$

This symmetric matrix is then decomposed with singular value decomposition (SVD). Because $\tilde{P}$ is a normal matrix, spectral theorem states that such a matrix is decomposed with SVD: $\tilde{P} = U\Lambda U^*$. The singular values of this symmetric square matrix equal to its eigenvalues, which lie on the diagonal of $\Lambda = \mathrm{diag}([\lambda_1, \lambda_2, \ldots, \lambda_N])$. Matrix $U = [u_1, u_2, \ldots, u_N]$ contains in its columns the N eigenvectors u_k of $\tilde{P}$. Furthermore, because $\tilde{P}$ is conjugate with P, these two matrices share their eigenvalues. However, to calculate the right eigenvectors v_k of P, we use equation 6 and get them in the columns of $V = [v_1, v_2, \ldots, v_N]$ [26].

$$V = D^{-\frac{1}{2}}U \tag{6}$$

The coordinates of a data point in the embedded space using eigenvalues in Λ and eigenvectors in V are in the matrix Ψ in equation 7. The rows correspond to the samples and the columns to the new embedded coordinates [9].

$$\Psi = V\Lambda \tag{7}$$

Strictly speaking, the eigenvalues should be raised to the power of t. This scale parameter t tells how many time steps are being considered when moving from data point to another. Here we have set it $t = 1$ [9].

With suitable ε the decay of the spectrum is fast. Only d components are needed for the diffusion map for sufficient accuracy. It should be noted that the first eigenvector v_1 is constant and is left out. Using only the next d components the diffusion map for original data point x_i is presented in equation 8. Here $v_k(x_i)$ corresponds to the ith element of kth eigenvector [9].

$$\Psi_d : x_i \longrightarrow [\lambda_2 v_2(x_i), \lambda_3 v_3(x_i), \ldots, \lambda_{d+1} v_{d+1}(x_i)] \tag{8}$$

This diffusion map embeds the known point x_i to a d-dimensional space. Dimension of the data are reduced from n to d. If desired, the diffusion map may be scaled by dividing the coordinates with λ_1.

3.3 Anomaly detection

After obtaining the low-dimensional presentation of the data it is easier to cluster the samples. Because spectral methods reveal the manifold, this clustering is called spectral clustering. This method reveals the normal and anomalous samples [28]. Alternatively, *k*-means or any other clustering method in the low-dimensional space is also possible [29]. Another approach is the density-based method [14].

Only the first few low-dimensional coordinates are interesting. They contain most of the information about the manifold structure. For diffusion map we use only the dimension corresponding to second eigenvector to determine the anomality of the samples. At 0, this dimension is divided into two clusters. The cluster with more samples is considered normal behavior. Conversely, the points in the other cluster are considered anomalous [30, 31, 28]. The second eigenvector acts as the separating feature for the two clusters in the low-dimensional space. The second eigenvalue is the solution to the normalized cut problem, which finds small weights between clusters but strong internal ties. This spectral clustering has probabilistic interpretation: grouping happens through similarity of transition probabilities between clusters [30, 32]. For PCA we use the first principal component in a similar way.

In practice the border between the normal and anomalous behavior might be unclear. This is the case especially with unsupervised learning, or when exploring the data for the first time. The normal cluster is usually very dense, and most of the data points lie within that cluster. The other points can be interpreted as deviating from the normal state, and thus anomalous.

4 Case 1: Validation with labeled data

4.1 Data acquisition

The data are acquired from a large real life web service. Let us call this dataset “A”. This case has been presented in an earlier publication [33]. The log files contain mostly normal traffic, but they also include anomalities and actual intrusions. The log files are from several Apache servers and are stored in *combined log format*. Listing below provides an example of a single log line. It includes information about the user’s IP address, time and timezone, the HTTP request including used resource and parameters, Apache server response code, amount of data sent to the user, the web page that was requested and used browser software.

```
127.0.0.1 - - [01/January/2011:00:00:01 +0300]
"GET /resource?parameter1=value1&parameter2=value2 HTTP/1.1"
200 2680 "http://www.address.com/webpage.html"
"Mozilla/5.0 (SymbianOS/9.2;...)"
```

The access log of a web site contains entries from multiple, distinct URLs. Most of them point to static requests like images, CSS files, etc. We do not focus on finding anomalies from those requests because it is not possible to inject code via static requests unless there are major deficiencies in the HTTP server itself. Instead, we focus on finding anomalies from dynamic requests because those requests are handled by the web application, which is run behind the HTTP server.

To reach this goal, the access log entries are grouped by the resource URL. That is the part between host name and parameters in the HTTP URL scheme. Resources containing only HTTP GET requests with no parameters are ignored. Each remaining resource is converted to a separate numerical matrix. In this matrix, a row represents a single access log entry, and a column represents an extracted feature.

Feature extraction is done in two passes. In the first pass the number of features is determined, and in the second pass the resulting matrix is produced. In our study we extracted the number of occurrences of 2-grams produced from HTTP GET parameters. In the example above, the parameter values form a string `value1value2`. This string is then analyzed for 2-gram frequencies. These frequencies are normalized with logarithm in order to scale them. This ensures that the distances between the samples are comparable.

4.2 Data analysis

To measure the effectiveness of the method the data are labeled so that classification accuracy can be measured. However, this labeling is not used for training the diffusion map. The class labels are not input for the method.

Diffusion map reveals the structure of the data, and all the anomalies are detected. The *n*-gram features of the data are mapped to lower dimensions. Figure 1 shows the resulting low-dimensional diffusion space with $\varepsilon = 100$. The normal behavior (N=2999) lies in the dense area to the upper left corner. Anomalous points (N=1293) are to the right of 0.

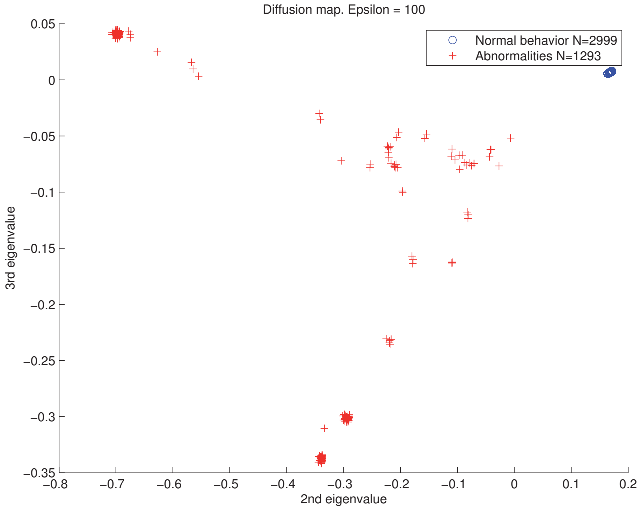

Figure 1: Two-dimensional diffusion map of the dataset A.

Figure 2 shows that the eigenvalues converge rapidly with $\varepsilon = 100$. This means that the first few eigenvalues and eigenvectors cover most of the differences observed in the data. The first value is 1 and corresponds to the constant eigenvector that is left out in the analysis. Eigenvalues $\lambda_2 = 0.331$ and $\lambda_3 = 0.065$ cover large portions of the data when compared to the rest that have values below 0.005.

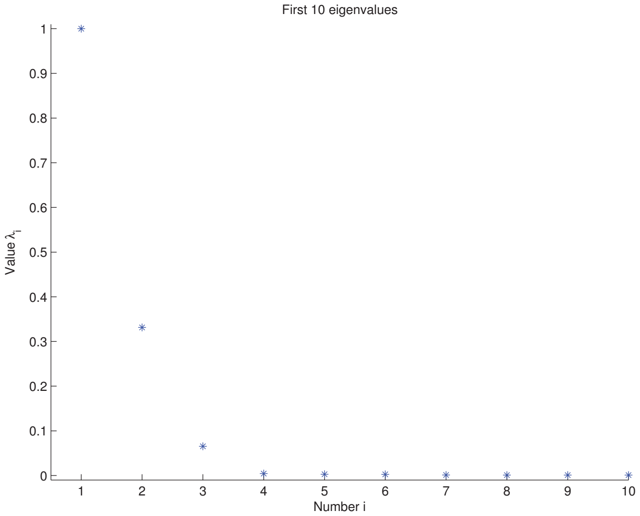

Figure 2: Eigenvalues of transition matrix with $\varepsilon = 100$ (dataset A).

Classification is tested with different values of ε, which defines the neighborhood for diffusion map. Accuracy of classification is defined as $accuracy = (tp+tn)/(tp+fp+fn+tn)$. Figure 3 shows how the accuracy of classification changes when ε is changed. Higher values of ε result in better accuracy. Precision of classification is defined $precision = tp/(tp+fp)$. The precision stays at 1 once any anomalies are detected, which means that all the anomalies detected are real anomalies regardless of the accuracy [23, p. 361].

For comparison, principal component analysis (PCA) is performed on the same normalized feature matrix [23, p. 79]. Results are very similar to the diffusion map approach, because of the simple structure of the feature matrix. This suggests that data points are linearly dependent. Furthermore, PCA reaches the same accuracy and precision as diffusion map. The low-dimensional presentation is also very similar. Figure 4 shows the first two coordinates of PCA.

5 Case 2: Unknown data

5.1 Data acquisition

After testing the methods with known data, we now analyze data that is totally unknown. We call this dataset “B”. This is the realistic situation with the web service that we are trying to analyze. There is no previous information about any attacks or other anomalies. The goal is to find a small amount of interesting lines that can then be analyzed more accurately. The number of log lines is so big that it is impossible to check all the lines manually. This is why anomaly detection is needed.

We start with relatively new dataset that has about 10 million lines. However, the lines with no parameters in the HTTP queries can be filtered out, because they are

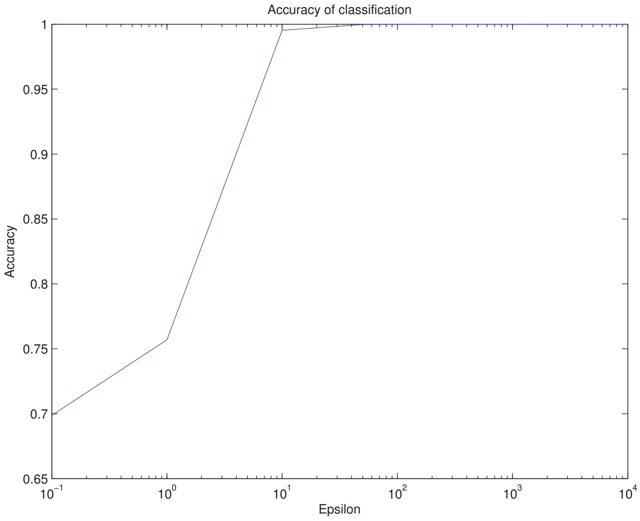

Figure 3: Accuracy of classification changes when the parameter ε is changed (dataset A).

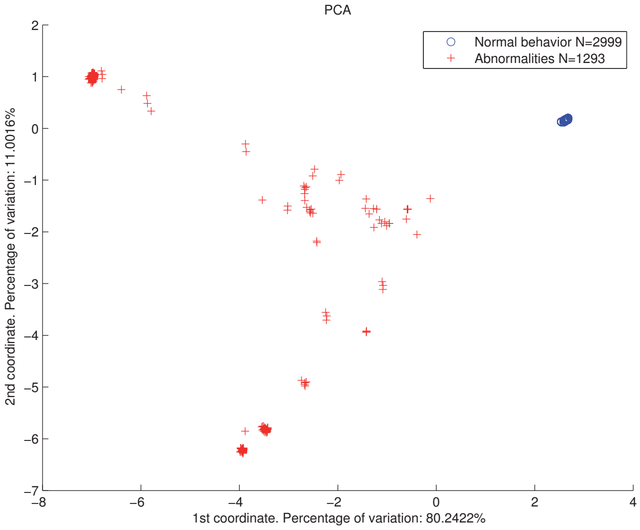

Figure 4: PCA of dataset A, first two coordinates.

not a big security risk. This leaves us with 2.5 million lines. These lines are then divided into different files based on the HTTP request URI. The entire log file cannot be analyzed at once, because different resources have very different parameters. The normal parameters for one single URI are usually quite similar, however. It makes more sense trying to analyze them individually. The number of different resources is quite high, more than 80 000, but many of the resources include just a few lines. In this case, it is sufficient to analyze some of the most frequently used resources. In addition to finding anomalies, this will give us more information about the web service traffic in general. The traffic can also be visualized.

After preprocessing the number of unique 2-grams in the most used resource URI was more than 1100. This means that dimensionality reduction is definitely needed, but the number of dimensions is not close to the theoretical maximum of 256^2 2-grams.

The new log file acquired for this case is in a different format than the log file analyzed in the first case. Some additional information, such as time in UNIX time, is also included. However, this information is not used in this analysis. The features extracted are the same as in the previous case. The feature matrix is calculated only from the parameter values. In this example the string to be analyzed would be `value`.

```
1305167880 111.222.111.222 965 633 29112
GET /path/to/resource.png?parameter=value HTTP/1.1
/path/to/resource.png parameter=value
/full/path/to/resource.png 200 + -
image/png,image/*;q=0.8,*/*;q=0.5 GB2312,utf-8;q=0.7,*;q=0.7
gzip,deflate fi-fi,fi;q=0.5 http://example.com Mozilla/5.0
(Windows; U; Windows NT 5.1; fi-FI; rv:1.9.2.15)
Gecko/20110101 Firefox/3.6.15
```

Feature matrix is constructed and logarithmic scaling applied in the same way as presented earlier with the dataset A. Even though the number of lines in the log file is quite high, the preprocessing phase takes only less than 10 minutes. Everything is written into temporary files to save memory. If more memory is available, the preprocessing could be changed to use it and it would get faster.

5.2 Data analysis

We choose two commonly used resources for analysis from the whole log data B. These resources are called “B1” and “B2”. We aim to find possible intrusion attempts from them. Dimensionality reduction is performed with both PCA and diffusion map. The results are then compared. Choosing ε for diffusion map is done differently than for dataset A. The sum $L = \sum_{i,j} W_{ij}$ plotted using logarithmic scale reveals the desirable linear region for ε [34, 35]. The value is chosen from that range, however, because even small changes of ε in that area affect the resulting embedding drastically, some human discretion must be used. Classification is done using spectral clustering.

Dataset B1 turns out to be a simple case where most of the data points are similar. The few deviations are easy to find from the feature matrix. First B1 is analyzed using PCA. Figure 5 shows that most of the normal behavior (N=14206) is concentrated to a very dense cluster. Our classifier assumes the points (N=87) to the right of the normal cluster to be anomalous. This clustering is feasible because the log lines contain actual previously unknown intrusions, although not all anomalies are intrusive. The anomalous points also seem to form clusters. These could indicate different types of attacks that happen frequently. This information could be used to further protect the service in the future.

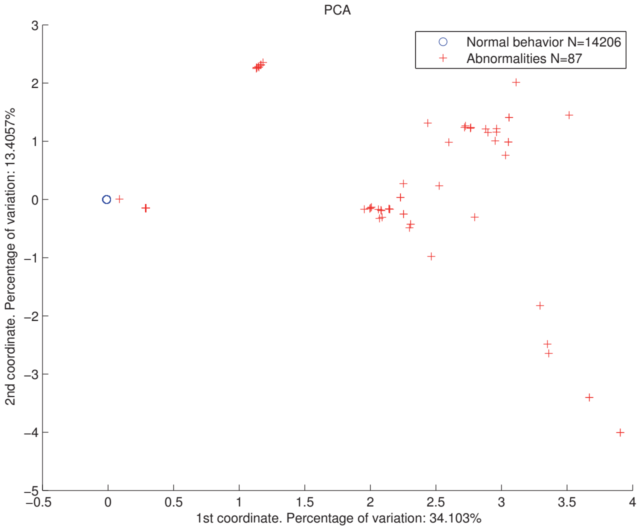

Figure 5: PCA of dataset B1.

Dataset B1 is also analyzed with diffusion map. The classification results are similar to PCA, even though the Figure 6 looks different. The normal behavior (N=14216) is concentrated leaving the anomalies (N=77) to the right of zero based on the first coordinate. The most differing anomalies are very far from the normal cluster. Some anomalies are very close to the normal data points. This means that the border between anomalous and normal traffic is not very clear. For this reason, 10 intrusion attempts that PCA detected were not discovered by diffusion map. This explains the difference in the number of found anomalies. Finding an optimal ε value would improve the result. However, this is a difficult task because of the unsupervised approach. Even though the low dimensional picture of PCA does not look as clear as the diffusion map, the result for PCA is better due to 10 false negatives that diffusion map fails to detect.

Dataset B2 contains more difficult and complex queries. This set is an example where the low dimensional representation by PCA and diffusion map are clearly different. Figure 7 shows the PCA of this dataset. The structure of the dataset is seen from the figure but the exact location of anomalies is difficult to find. This is because even the normal query lines include long and dynamically changing strings. The sparse left side is actually normal traffic, but there seems to be a lot of variation in the normal traffic alone. The anomalies found by diffusion map are situated in the upper right corner of the PCA representation. Data points do not form a distinct cluster, making anomaly detection and clustering very difficult with this representation. The used simple spectral clustering clearly does not work in this case. Further clustering with more advanced algorithms might reveal what types of queries the log file contains. Most variance is captured by the first principal component. However, two first principal components do not contain most of the total cumulative variance. Even this kind of visualization

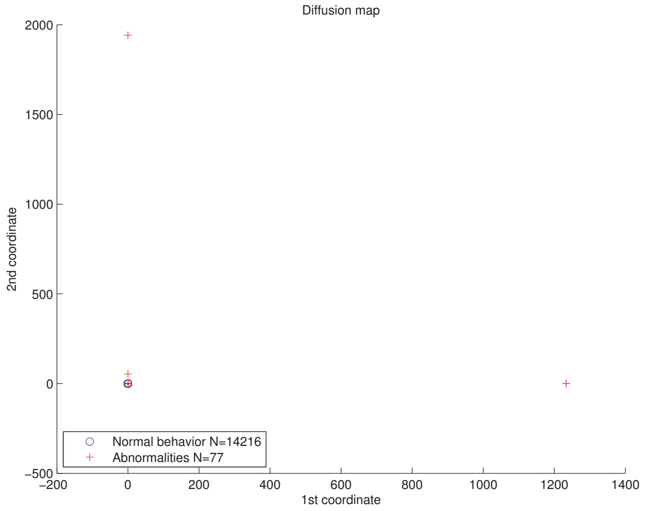

Figure 6: Diffusion map of dataset B1, $\varepsilon = 7$.

facilitates the analysis of huge text files (N=21406).

Diffusion map finds anomalies from dataset B2. The first two coordinates capture almost all of the difference between normal and anomalous queries (Figure 8). In addition, the clusters are very clearly separated and the normal traffic is easy to distinguish. This dataset shows a clear difference between PCA and diffusion map results. The anomalous cluster (N=173) contains the points on the far left and the anomalous points near the normal cluster. Again, the normal cluster (N=21233) is very dense. The found anomalies contain 88 real intrusions. The intrusions are related to injecting malicious SQL queries or scripts into the HTTP query. Some non-intrusive queries are also included, but they can be manually screened afterwards. The number of log lines is small enough so that system administrator can inspect the anomalous lines and easily find the intrusion attempts. Anomaly detection seems to find attacks from a large and varying dataset. The anomalous traffic forms two distinct clusters, one of which contains the intrusions. Diffusion map with a correctly selected ε helps in finding anomalies and automatically detecting normal cluster. Larger values of ε make the diffusion map behave more like PCA. These approachees are more suitable for visual inspection and multicluster analysis.

6 Conclusion

The goal of this study is to find security attacks from network data. The proposed anomaly detection scheme could be used for query log analysis in real life situations. We concentrate on web server log data, which contains text queries that are the focus of our analysis. In these kinds of practical situations the boundary between normal traffic and intrusions is not always very clear. However, the relative strangeness of the sample could indicate how severe an alert is.

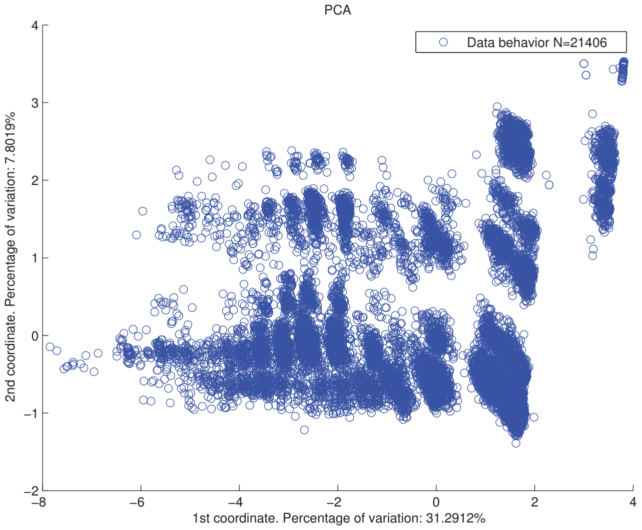

Figure 7: PCA of dataset B2.

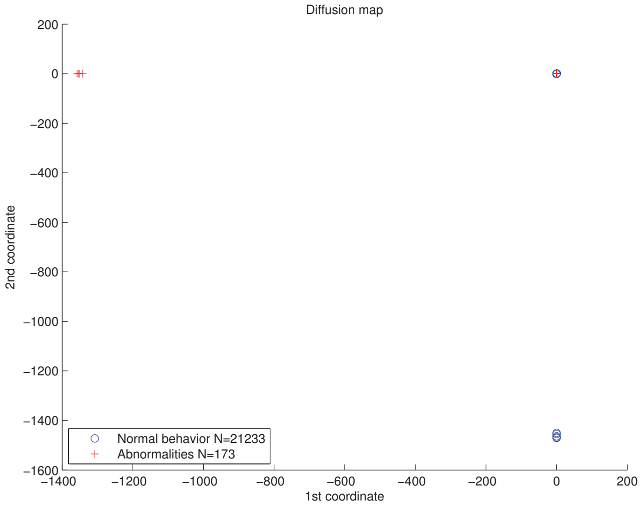

Figure 8: Diffusion map of dataset B2, $\varepsilon = 7$.

The dimensionality reduction framework adapts to the log data. It assumes that only few variables are needed to express the interesting information, and finds a coordinate system that describes the global structure of the data. These coordinates could be used for further analysis of characteristics of anomalous activities.

The main benefits of this framework include:

- The amount of log lines that needs to be inspected is reduced. This is useful for system administrators trying to identify intrusions. The number of interesting log lines is low compared to the total number of lines in the log file.

- The unsupervised nature and adaptiveness of the framework. The proposed methods adapt to the structure of the data without training or previous knowledge. This makes it suitable for exploration and analysis of data without prior examples or attack signatures. This means that the framework also detects zero-day attacks.

- It works on the application layer in the network. The attacks themselves must in some way target the actual applications running on the computer. These logs might be more available than pure low-level network packet data.

- Visualization of text log data. It is much easier to analyze the structure of traffic using visualizations than it is to read raw textual log.

The data in question are rather sparse and the discriminating features are quite evident from the feature matrix. This is the merit of *n*-gram feature extraction which creates a feature space that separates the normal behavior in a good manner. The features describe the data clearly, and they are easy to process afterwards. Still, an attacker might take advantage of the features used by the intrusion detection system. If the *n*-gram frequencies of the attack query are similar enough to normal behavior, the currently proposed system could not detect the attacks. Also, if most of the traffic in a single log file consists of attack queries, they will be considered to be normal. This might be a problem in rarely used services.

One advantage of the diffusion map methodology is that it has only one metaparameter, ε. There exists estimation methods for finding the optimal value. If for some reason the threshold sensitivity needs to be changed, ε gives the flexibility to adapt to the global structure. However, the quality of the results is sensitive to changes of this parameter. Values that are too small or large give non-desirable results.

The presented anomaly detection framework performs well on real data. Several actual intrusions are detected. As an unsupervised algorithm this approach is well suited for finding previously unknown intrusions. This method could be applied to offline systems, as well as extended to a real-time intrusion detection system.

There are several points in this framework that could benefit from further research. The feature extraction from the web log is currently done with *n*-grams. However, this is only one method for it and other text-focused features might better describe the dataset. Furthermore, the dimensionality reduction scheme could be developed to adapt to this kind of data more efficiently, and the quality of the reduction could also be evaluated. The classification method may be improved or changed altogether to another algorithm. Finally, automated root cause detection would make the system more usable in practice.

Acknowledgements

The authors thank Professors Amir Averbuch, Timo Hämäläinen and Tapani Ristaniemi for their continued support. Thanks are extended to Juho Knuuttila and Kristian Siljander for useful discussions and ideas.

References

[1] Mukkamala, S. and Sung, A. *A comparative study of techniques for intrusion detection* (2003).

[2] Patcha, A. and Park, J. *An overview of anomaly detection techniques: Existing solutions and latest technological trends*. *Computer Networks*, 51(12):3448–3470 (2007).

[3] Pietro, R. and Mancini, L. *Intrusion detection systems*. Advances in information security. Springer (2008).

[4] Chandola, V., Banerjee, A., and Kumar, V. *Anomaly detection: A survey*. *ACM Comput. Surv.*, 41(3):1–58 (2009).

[5] Nguyen-Tuong, A., Guarnieri, S., Greene, D., Shirley, J., and Evans, D. *Automatically hardening web applications using precise tainting*. In R. Sasaki, S. Qing, E. Okamoto, and H. Yoshiura, editors, *Security and Privacy in the Age of Ubiquitous Computing*, volume 181 of *IFIP Advances in Information and Communication Technology*, pp. 295–307. Springer Boston (2005).

[6] Ramadas, M., Ostermann, S., and Tjaden, B. *Detecting anomalous network traffic with self-organizing maps*. In G. Vigna, E. Jonsson, and C. Kruegel, editors, *Recent Advances in Intrusion Detection*, pp. 36–54. Springer (2003).

[7] Tran, Q., Duan, H., and Li, X. *One-class support vector machine for anomaly network traffic detection*. *China Education and Research Network (CERNET), Tsinghua University, Main Building*, 310 (2004).

[8] Coifman, R.R., Lafon, S., Lee, A.B., Maggioni, M., Nadler, B., Warner, F., and Zucker, S.W. *Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps*. In *Proceedings of the National Academy of Sciences of the United States of America*, volume 102, p. 7426 (2005).

[9] Coifman, R.R. and Lafon, S. *Diffusion maps*. *Applied and Computational Harmonic Analysis*, 21(1):5–30 (2006).

[10] Bengio, Y., Delalleau, O., Roux, N.L., Paiement, J.F., Vincent, P., and Ouimet, M. *Feature Extraction*, chapter Spectral Dimensionality Reduction, pp. 519–550. Studies in Fuzziness and Soft Computing. Springer Berlin, Heidelberg (2006).

[11] Schclar, A., Averbuch, A., Rabin, N., Zheludev, V., and Hochman, K. *A diffusion framework for detection of moving vehicles*. *Digital Signal Processing*, 20(1):111–122 (2010).

[12] İzmirli, Ö. *Tonal-atonal classification of music audio using diffusion maps*. In *10th International Society for Music Information Retrieval Conference (ISMIR 2009)* (2009).

[13] Keller, Y., Coifman, R., Lafon, S., and Zucker, S. *Audio-visual group recognition using diffusion maps*. *Signal Processing, IEEE Transactions on*, 58(1):403–413 (2010).

[14] Turkka, J., Ristaniemi, T., David, G., and Averbuch, A. *Anomaly detection framework for tracing problems in radio networks*. In *Proc. to ICN 2011* (2011).

[15] Chernogorov, F., Turkka, J., Ristaniemi, T., and Averbuch, A. *Detection of sleeping cells in LTE networks using diffusion maps*. In *Vehicular Technology Conference (VTC Spring), 2011 IEEE 73rd*, pp. 1–5. IEEE (2011).

[16] David, G. *Anomaly Detection and Classification via Diffusion Processes in Hyper-Networks*. Ph.D. thesis, Tel-Aviv University (2009).

[17] Kruegel, C. and Vigna, G. *Anomaly detection of web-based attacks*. In *Proceedings of the 10th ACM conference on Computer and communications security*, pp. 251–261. ACM (2003).

[18] Hubballi, N., Biswas, S., and Nandi, S. *Layered higher order n-grams for hardening payload based anomaly intrusion detection*. In *Availability, Reliability, and Security, 2010. ARES'10 International Conference on*, pp. 321–326. IEEE (2010).

[19] Ringberg, H., Soule, A., Rexford, J., and Diot, C. *Sensitivity of PCA for traffic anomaly detection*. *ACM SIGMETRICS Performance Evaluation Review*, 35(1):109–120 (2007).

[20] Callegari, C., Gazzarrini, L., Giordano, S., Pagano, M., and Pepe, T. *A novel PCA-based network anomaly detection*. In *Communications (ICC), 2011 IEEE International Conference on*, pp. 1–5. IEEE (2011).

[21] David, G. and Averbuch, A. *Hierarchical data organization, clustering and denoising via localized diffusion folders*. *Applied and Computational Harmonic Analysis* (2011).

[22] Damashek, M. *Gauging similarity with n-grams: Language-independent categorization of text*. *Science*, 267(5199):843 (1995).

[23] Han, J. and Kamber, M. *Data mining: concepts and techniques*. Morgan Kaufmann (2006).

[24] Abdi, H. and Williams, L. *Principal component analysis*. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(4):433–459 (2010).

[25] Lee, J. and Verleysen, M. *Nonlinear dimensionality reduction*. Springer Verlag (2007).

[26] Nadler, B., Lafon, S., Coifman, R., and Kevrekidis, I.G. *Diffusion maps – a probabilistic interpretation for spectral embedding and clustering algorithms*. In T.J. Barth, M. Griebel, D.E. Keyes, R.M. Nieminen, D. Roose, T. Schlick, A.N. Gorban, B. Kégl, D.C. Wunsch, and A.Y. Zinovyev, editors, *Principal Manifolds for Data Visualization and Dimension Reduction*, volume 58 of *Lecture Notes in Computational Science and Engineering*, pp. 238–260. Springer Berlin Heidelberg (2008).

[27] Chung, F.R.K. *Spectral Graph Theory*, p. 2. AMS Press, Providence, R.I (1997).

[28] von Luxburg, U. *A tutorial on spectral clustering*. *Statistics and Computing*, 17:395–416 (2007).

[29] Ng, A.Y., Jordan, M.I., and Weiss, Y. *On spectral clustering: Analysis and an algorithm*. In *Advances in Neural Information Processing Systems 14*, pp. 849–856. MIT Press (2001).

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

[31] Kannan, R., Vempala, S., and Vetta, A. *On clusterings: Good, bad and spectral*. *J. ACM*, 51:497–515 (2004).

[32] Meila, M. and Shi, J. *Learning segmentation by random walks*. In *NIPS*, pp. 873–879 (2000).

[33] Sipola, T., Juvonen, A., and Lehtonen, J. *Anomaly detection from network logs using diffusion maps*. In L. Iliadis and C. Jayne, editors, *Engineering Applications of Neural Networks*, volume 363 of *IFIP Advances in Information and Communication Technology*, pp. 172–181. Springer Boston (2011).

[34] Hein, M. and Audibert, J. *Intrinsic dimensionality estimation of submanifolds in* $\mathbb{R}^d$. In *Proceedings of the 22nd international conference on Machine learning*, pp. 289–296. ACM (2005).

[35] Coifman, R., Shkolnisky, Y., Sigworth, F., and Singer, A. *Graph Laplacian tomography from unknown random projections*. *Image Processing, IEEE Transactions on*, 17(10):1891–1899 (2008).

PIII

ADAPTIVE FRAMEWORK FOR NETWORK TRAFFIC CLASSIFICATION USING DIMENSIONALITY REDUCTION AND CLUSTERING

by

Antti Juvonen and Tuomo Sipola 2012

Ultra Modern Telecommunications and Control Systems and Workshops
(ICUMT), 2012 4th International Congress on, pp. 274–279, St. Petersburg,
Russia

Adaptive Framework for Network Traffic Classification Using Dimensionality Reduction and Clustering

Antti Juvonen, Tuomo Sipola
Department of Mathematical Information Technology
University of Jyväskylä
Jyväskylä, Finland
{*antti.k.a.juvonen, tuomo.sipola*}*@jyu.fi*

***Abstract*—Information security has become a very important topic especially during the last years. Web services are becoming more complex and dynamic. This offers new possibilities for attackers to exploit vulnerabilities by inputting malicious queries or code. However, these attack attempts are often recorded in server logs. Analyzing these logs could be a way to detect intrusions either periodically or in real time. We propose a framework that preprocesses and analyzes these log files. HTTP queries are transformed to numerical matrices using n-gram analysis. The dimensionality of these matrices is reduced using principal component analysis and diffusion map methodology. Abnormal log lines can then be analyzed in more detail. We expand our previous work by elaborating the cluster analysis after obtaining the low-dimensional representation. The framework was tested with actual server log data collected from a large web service. Several previously unknown intrusions were found. Proposed methods could be customized to analyze any kind of log data. The system could be used as a real-time anomaly detection system in any network where sufficient data is available.**

***Keywords*-intrusion detection; anomaly detection; n-grams; diffusion map; k-means; data mining; machine learning**

I. Introduction

Most web servers log their traffic. This log data is rarely used, but it could be analyzed in order to find anomalies or to visualize the traffic structure. Acquiring the data does not require any modifications to the actual web service, because data logging is usually done by default. Different kinds of log files are created, but for this study the most interesting log is the one containing HTTP queries.

One important application for network traffic analysis is anomaly detection. This is done using *intrusion detection systems* (IDS) [1]. Many of these analyze the transport layer, mostly TCP packet data. However, we try to find anomalies and other information from application layer log files. HTTP queries include this information. Many attacks, such as SQL injections, can be detected from this layer.

Log files are in textual form. Therefore, some preprocessing is needed to transform query strings into numerical matrices. This can be done using information about n-gram analysis, which is described in section III-A. Calculating the frequencies of individual substrings in the data results in a numerical data matrix.

After preprocessing, many data mining methods can be used to visualize and analyze the logs. We perform dimensionality reduction and clustering. After visualizing the results it is possible to interpret the findings and make more detailed analysis about the web service traffic.

We propose a framework that processes textual log files in order to visualize them. We are trying to find patterns and anomalies using only log files containing HTTP queries. The framework is adaptive, and individual parts of it can be changed. For example, the choice of dimensionality reduction method or clustering algorithm can be done based on current needs.

The proposed methods use data mining principles, and they work as an IDS and network traffic visualization and analysis tool. Using the framework, we are trying to find whether the textual HTTP query logs actually include some information about the traffic structure. This information could then be used to classify users and individual queries and to find anomalies and intrusion attempts.

II. Related work

We have previously researched log data preprocessing and anomaly detection [2], [3]. This research focused on finding intrusions from log data. We now extend this methodology to further analyze and cluster the structure of the traffic. This is done by adding more accurate clustering algorithms into the framework.

Principal component analysis has been widely used in network intrusion detection and traffic analysis. Xu et al. used PCA and support vector machine to reduce dimensions and classify network traffic in order to find intrusions [4]. Taylor et al. used PCA and clustering analysis to find network anomalies and perform traffic screening [5].

Diffusion methods have been applied in network traffic analysis. These studies have concentrated on low-level IP packet features. These features are numerical and the network architecture differs from our study [6] [7]. Network server logs have also been analyzed using diffusion maps and spectral clustering [2] [3].

This is the authors' postprint version of the article. The original print version appeared as: A. Juvonen and T. Sipola, "Adaptive Framework for Network Traffic Classification Using Dimensionality Reduction and Clustering," in *Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), 2012 4th International Congress on*. IEEE 2012, pp. 274–279. Available online at http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6459678&isnumber=6459638

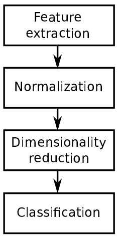

Figure 1. The data mining process

III. Methodology

Our overall approach is rooted in the data mining process [8], [9]. This approach is method-centric as our research is focused on the data processing and not business aspects. The data mining process of our study flows as follows:

1) Data selection.
2) Extract n-gram features from the text data.
3) Normalize the feature matrix.
4) Reduce the number of dimensions to obtain low-dimensional features.
5) Classify or cluster the low-dimensional data presentation.
6) Interpret the found patterns or anomalies.

The process is presented in figure 1.

A. Feature extraction

The log files are in text format. Therefore, it is necessary to transform the log lines into numerical vectors which then can be used in further mathematical analysis. We use n-gram analysis to process log files into numerical matrices. It has been used e.g. in judging similarity in text documents [10], analyzing protein sequences [11] and detecting malicious code [12].

N-grams are consecutive sequences of n characters [10]. Each log line corresponds to a feature vector containing the frequencies of each individual n-gram found in the data. The list of n-grams appearing in the data can be found using n-character-wide sliding window moved along the string one character at a time [10].

Let us consider the following example. Having two strings containing the words `anomaly` and `analysis`, we can construct the feature matrix in the following way:

an	no	om	ma	al	ly	na	ys	si	is
1	1	1	1	1	1	0	0	0	0
1	0	0	0	1	1	1	1	1	1

In this study, 2-grams are used. However, it is possible to use longer n-grams as well. This will of course results in more dimensions in the matrix, because there are more unique n-grams. The theoretical maximum number of individual 2-grams using ASCII-characters is $256^2 = 65536$, but in practice this is usually not the case. This is due to the fact that many characters are never actually used [10].

B. Normalization

Normalization ensures that the features of the input data are in the same scale. We use logarithm for this purpose. To avoid complex numbers, the input must be above zero. The normalization function for a point x_i in the dataset is

$$f_n(x_i) = \log(x_i - X_{min} + 1),$$

where X_{min} is the minimum of all the values in the dataset.

C. Principal Component Analysis

Principal Component Analysis (PCA) [13] is perhaps the best-known dimensionality reduction technique. It has many practical applications, such as computer vision and image compression [14].

The PCA process is explained in more detail in [14]. First we must substract the mean from the original data to make the data have zero mean. Then the covariance matrix must be calculated. From the covariance matrix we can then calculate eigenvalues and the corresponding eigenvectors. If we choose d eigenvectors that contain most of the variance, we get a lower dimension representation of the original data with d dimensions. This is done by choosing the d eigenvectors as columns for a matrix, and multiplying the mean-centered data with this matrix. For visualization purposes it is necessary to choose either 2 or 3 dimensions, ie. eigenvectors.

Calculating PCA is relatively simple, but it will only work in linear cases. If the dataset is non-linear, some other dimensionality reduction method must be used. PCA can also give inaccurate results if there are outliers in the data.

D. Diffusion Map

Diffusion map (DM) reduces the dimensions while retaining the diffusion distances in the high-dimensional space as Euclidean distances in the low-dimensional space. This reduction is non-linear. The goal is to move from n-dimensional space to a low-dimensional space with d dimensions, when $d \ll n$ [15].

One measurement $x_i \in \mathbb{R}^n$ in this study corresponds to one line in the log file. Given the dataset $X = \{x_1,\ x_2,\ x_3,\ \ldots\ x_N\}$ the affinity matrix $W(x_i, x_j) = \exp\left(\frac{-||x_i - x_j||^2}{\epsilon}\right)$ describes the affinities between measurements. Here we have used the Gaussian kernel. Matrix $P = W^{-1}K$ represents the transition probabilities between the measurements. Next, the matrix D collects the row sums to its diagonal. Using the singular value decomposition (SVD) of matrix $\tilde{P} = D^{-\frac{1}{2}}WD^{-\frac{1}{2}}$ we obtain the eigenvectors v_k and eigenvalues λ_k.

The diffusion map maps the measurements x_i to low dimensions by giving each high-dimensional point coordinates in the low dimensions: $x_i \rightarrow [\lambda_1 v_1(x_i),\ \lambda_2 v_2(x_i) \ldots\ \lambda_d v_d(x_i)]$. These new coordinates lose some of the information contained in the original dataset. However, the accuracy is usually good enough for later classification. Even though there is loss of information, the classification problem becomes easier.

E. Traffic clustering using k-means algorithm

We use cluster analysis to divide network traffic into meaningful groups. In this way we can capture the natural structure of the data [16].

K-means algorithm was introduced in 1955 and huge number of other clustering algorithms have been introduced since then, but k-means method is still widely used [17]. It is a prototype-based clustering technique [16]. Given the original data $X = x_i$, where $i = 1, .., n$, the goal is to cluster the data points into k clusters. The mean of cluster k is now μ_k, and the mean squared error (MSE) between a data point and the cluster mean is $||x_i - \mu_k||^2$. This leads to an optimization problem where the MSE for each datapoint in each cluster must be minimized.

The problem can be solved following these steps [18]:

1) Select initial centers for k clusters.
2) Assign each datapoint to its closest cluster centroid.
3) Compute the new cluster centers by calculating the mean of the datapoints in each cluster.

Steps 2 and 3 are repeated until a stopping criterion is met. Usually this means that the partitioning has not changed since the last iteration, and thus a local optimum solution for the problem has been found.

Choosing the number of clusters is not trivial, but there are many methods for calculating the number of clusters, such as Davies-Bouldin index, described in [19]. This algorithm takes into account both scatter within a cluster and separation between different clusters. Davies-Bouldin index is used in this study to determine the number of clusters for each resource.

The algorithm can give different results depending on the initialization, because it only finds the local optimal solution. This can happen especially when using random initialization. However, this problem can be overcome by running k-means multiple times and choosing the clustering results that gives the smallest squared error [17]. There are also many other algorithms for choosing the initial cluster centroids.

IV. Experimental setup

Figure 2 shows the architecture of the web service that was analyzed. It contains many servers that offer the same service to users using load balancing. Proprietary log files were acquired from this service. These files then need to be preprocessed into numerical matrices. The data and this process are described in this section.

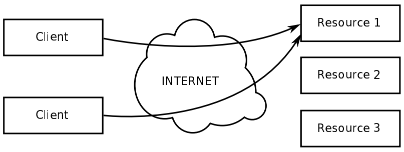

Figure 2. Experiment architecture.

A. Data acquisition

The data have been collected from a large web service. Apache web servers are used, and they log data using Combined Log Format, example of a single log line:

```
127.0.0.1 - -
[01/January/2012:00:00:01 +0300]
"GET /resource.php?parameter1=value1
&parameter2=value2
HTTP/1.1"
200 2680
"http://www.address.com/webpage.html"
"Mozilla/5.0
(SymbianOS/9.2;...)"
```

For this analysis, the HTTP query part is used because it contains the only information that a user can input. This offers possibilities for attackers. The other information, such as time, can be used when further analyzing individual log lines (e.g. for finding anomalies or attacks). On the other hand, HTTP query parameters and their values are dynamic and changing, offering valuable information about this dynamic web service. Analyzing this information will explain a lot about the structure of the traffic. The parameter values in data used in this study were dynamic and changing, and also not always human-readable. Therefore, analyzing these fields has to be done automatically with mathematical methods.

B. Data preprocessing

The first step is to select the data for analysis. The original log file contains approximately 4 million log lines. However, most of these lines contain only static queries. Static lines do not contain changing parameter values. These lines do not offer a lot of information, because they are practically identical in the used dataset. In addition, static lines do not contain information about user input, meaning it is not possible to detect attacks from those log lines alone. On the other hand, dynamic web resources are changing and also vulnerable, so dynamic lines containing parameters and parameter values are interesting and can offer more information about the web service. Therefore, static log lines are filtered out, leaving only approximately 221 000 lines to be inspected and clustered. This data selection reduces the size considerably and creates a database of the most interesting aspects of the log files.

After the first filtering stage, log files are divided into smaller files according to resource URI. This is because different resources accept different parameter values, so they do not have much to do with each other. This makes anomaly detection from full data very difficult and inaccurate. However, traffic structure inside single resource is more consistent. After this division, smaller logfiles can be analyzed independently. It makes sense to further analyze the largest log files, because some of the resources contain only a few lines. These lines have to be omitted.

Finally, in order to create data matrices out of textual log data, n-gram analysis is performed. This process is explained in III-A.

V. Results

For this research, 3 relatively large resources are selected for further analysis and clustering. Resource 1 contains 10935 lines and 414 dimensions, and is the simplest in terms of HTTP query parameters. Resource 2 contains only 2982 lines, but the number of dimensions is 3866, which makes analysis challenging. Also, the parameters are clearly not human-readable, i.e. it is impossible to say anything about the queries by looking at the parameter string alone. Resource 3 is the largest, including 21406 lines and 991 dimensions.

All the resources are analyzed using the proposed framework. The feature data are normalized with the logarithm function. PCA and diffusion map reduce the dimensionality of the normalized feature matrices. Clustering then reveals the structure of the data and facilitates the interpretation of the log files.

Resource 1 contains 10935 lines and 414 dimensions. The results for diffusion map and principal component analysis are presented in figures 3 and 4, respectively. This resource is a simple example, mainly useful in validating that the methods do give satisfactory results. The only difference is that DM separates the data points more clearly. Due to this separation we get 3 clusters, instead of 2 as in PCA. The biggest cluster contains varying parameter values. The parameters in smaller clusters are almost the same within that cluster. However, this behavior is easy to see directly from the log lines. The framework visualizes the traffic well, but in this case we do not obtain any new information about the data.

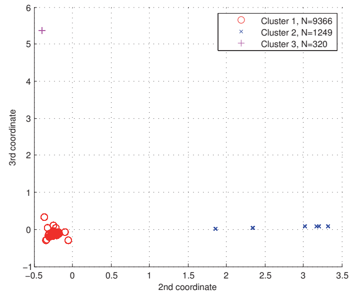

Figure 3. Resource 1, diffusion map.

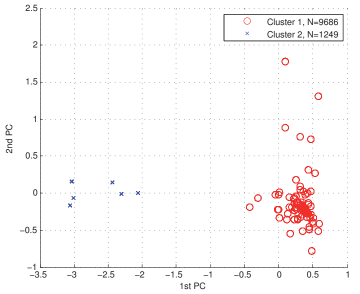

Figure 4. Resource 1, PCA.

Resource 2 is the smallest in this research, containing only 2982 requests. However, the number of dimensions is 3866. This means that there are more dimensions than data points, which is always a problem in classification tasks. Despite that, we obtained clear results. The DM and PCA results presented in figures 5 and 6. The results are essentially identical, figures look slightly different but the clustering is exactly the same. This might mean that variables are linearly dependent, otherwise PCA would not work well. The log lines themselves are not human-readable, containing error tickets that have a seemingly random code as the parameter value. However, as can be seen from the figures, there are clearly two distinct clusters that can be seen using both dimensionality reduction methods. This behavior was not previously known and requires more detailed analysis with the administrator of the web service.

Resource 3 is the largest with 21406 lines and 996 dimensions. It also shows that DM (in figure 7) and PCA (in figure 8) can sometimes give very different results. Normal parameter values in this resource are long and varied. This results in PCA not being able to clearly distinguish any clusters. For this reason, k-means clustering was not performed for resource 3 PCA datapoints. However, with DM the results are very meaningful. Normal traffic clearly forms it's own cluster, while 2 other groups are apparent. Cluster 2 with 5 datapoints does not contain anything malicious, but is slightly different from other normal datapoints. The most interesting finding in this data is cluster 3, which contains 4 lines. All of these lines contain an SQL injection attack, where an attacker tried to include malicious SQL queries as parameter values. The 2nd DM coordinate clearly separates attacks from rest of the data, meaning that in this case only one dimension is needed for anomaly detection.

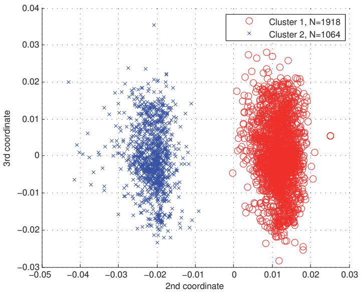

Figure 5. Resource 2, diffusion map.

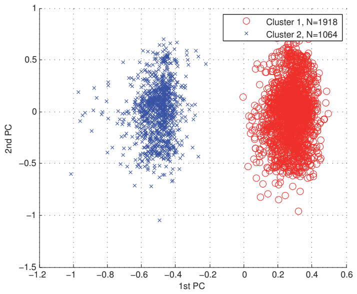

Figure 6. Resource 2, PCA.

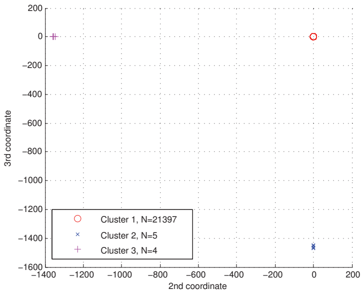

Figure 7. Resource 3, diffusion map.

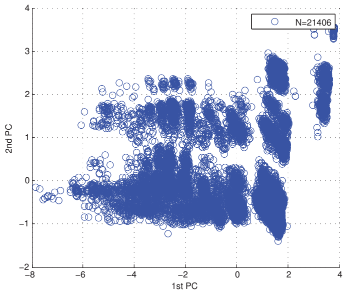

Figure 8. Resource 3, PCA.

In all of the figures, except PCA for resource 3, it can be seen that the separation of clusters is clear. A simple clustering method such as spectral clustering or decision tree could be used.

VI. Conclusion

We presented a framework for preprocessing, clustering and visualizing web server log data. This framework was used for anomaly detection, visualization and explorative data analysis based only on application layer data. Individual parts of the architecture can be changed for different results. For example, k-means clustering can be replaced with hierarchical linkage clustering method.

The results clearly indicate that there are traffic structures that can be visualized from HTTP query information. The data forms distinct clusters and contains anomalies as well. The sensitivity for outliers creates some problems for PCA, which means that it can be challenging to use it for anomaly detection. Diffusion maps give good results, but more research would have to be done to get more information about performance issues. In some cases the results for PCA and DM are nearly identical, while in other cases they differ greatly. PCA is faster but cannot be used with non-linear data. DM seems to work in most situations but can be too slow.

Traffic clustering can give new information about the users of a web service. This information could be used to categorize users more accurately. This gives opportunities for more accurate advertising or offering better content for users. Finding anomalies gives information about possible intrusion attempts and other abnormalities.

To make the framework more usable, it should be automatic and work in real-time. More research is needed to find the most generally usable algorithms for each phase in the architecture. In addition, log data tends to be high in volume, so performance issues might become a problem. For dimensionality reduction the number of dimensions is not trivial. Also, the number of clusters must be determined depending on the chosen clustering algorithm. Real-time functioning requires changes in preprocessing and limits the dimensionality reduction options. For this purpose, PCA might be a good method, since projection of new points into lower dimensions is simply a matter of matrix multiplication. However, the limitations mentioned previously still apply.

Using data mining methods, underlying structure and anomalies are found from HTTP logs and these results can be visualized and analyzed to find patterns and anomalies.

References

[1] K. Scarfone and P. Mell, "Guide to intrusion detection and prevention systems (idps)," *NIST Special Publication*, vol. 800, no. 2007, p. 94, 2007.

[2] T. Sipola, A. Juvonen, and J. Lehtonen, "Anomaly detection from network logs using diffusion maps," in *Engineering Applications of Neural Networks*, ser. IFIP Advances in Information and Communication Technology, L. Iliadis and C. Jayne, Eds. Springer Boston, 2011, vol. 363, pp. 172–181.

[3] ——, "Dimensionality reduction framework for detecting anomalies from network logs," *Engineering Intelligent Systems*, 2012, forthcoming.

[4] X. Xu and X. Wang, "An adaptive network intrusion detection method based on pca and support vector machines," *Advanced Data Mining and Applications*, pp. 731–731, 2005.

[5] C. Taylor and J. Alves-Foss, "Nate: N etwork analysis of a nomalous t raffic e vents, a low-cost approach," in *Proceedings of the 2001 workshop on New security paradigms*. ACM, 2001, pp. 89–96.

[6] G. David, "Anomaly Detection and Classification via Diffusion Processes in Hyper-Networks," Ph.D. dissertation, Tel-Aviv University, 2009.

[7] G. David and A. Averbuch, "Hierarchical data organization, clustering and denoising via localized diffusion folders," *Applied and Computational Harmonic Analysis*, 2011.

[8] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, "The KDD process for extracting useful knowledge from volumes of data," *Commun. ACM*, vol. 39, pp. 27–34, November 1996. [Online]. Available: http://doi.acm.org/10.1145/240455.240464

[9] ——, "From data mining to knowledge discovery in databases," *AI magazine*, vol. 17, no. 3, p. 37, 1996.

[10] M. Damashek, "Gauging similarity with n-grams: Language-independent categorization of text," *Science*, vol. 267, no. 5199, p. 843, 1995.

[11] M. Ganapathiraju, D. Weisser, R. Rosenfeld, J. Carbonell, R. Reddy, and J. Klein-Seetharaman, "Comparative n-gram analysis of whole-genome protein sequences," in *Proceedings of the second international conference on Human Language Technology Research*. Morgan Kaufmann Publishers Inc., 2002, pp. 76–81.

[12] T. Abou-Assaleh, N. Cercone, V. Keselj, and R. Sweidan, "N-gram-based detection of new malicious code," in *Computer Software and Applications Conference, 2004. COMPSAC 2004. Proceedings of the 28th Annual International*, vol. 2. IEEE, 2004, pp. 41–42.

[13] H. Hotelling, "Analysis of a complex of statistical variables into principal components." *Journal of educational psychology*, vol. 24, no. 6, pp. 417–441, 1933.

[14] L. Smith, "A tutorial on principal components analysis," *Cornell University, USA*, vol. 51, p. 52, 2002.

[15] R. R. Coifman and S. Lafon, "Diffusion maps," *Applied and Computational Harmonic Analysis*, vol. 21, no. 1, pp. 5–30, 2006.

[16] P. Tan, M. Steinbach, and V. Kumar, "Cluster analysis: Basic concepts and algorithms," *Introduction to data mining*, pp. 487–568, 2006.

[17] A. K. Jain, "Data clustering: 50 years beyond k-means," *Pattern Recognition Letters*, vol. 31, no. 8, pp. 651–666, 2010.

[18] A. K. Jain and R. C. Dubes, *Algorithms for clustering data*. Prentice Hall., 1988.

[19] D. Davies and D. Bouldin, "A cluster separation measure," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, no. 2, pp. 224–227, 1979.

PIV

COMBINING CONJUNCTIVE RULE EXTRACTION WITH DIFFUSION MAPS FOR NETWORK INTRUSION DETECTION

by

Antti Juvonen and Tuomo Sipola 2013

The Eighteenth IEEE Symposium on Computers and Communications (ISCC 2013), Split, Croatia

Combining Conjunctive Rule Extraction with Diffusion Maps for Network Intrusion Detection

Antti Juvonen, Tuomo Sipola
Department of Mathematical Information Technology
University of Jyväskylä
Jyväskylä, Finland
{antti.k.a.juvonen, tuomo.sipola}@jyu.fi

***Abstract*—Network security and intrusion detection are important in the modern world where communication happens via information networks. Traditional signature-based intrusion detection methods cannot find previously unknown attacks. On the other hand, algorithms used for anomaly detection often have black box qualities that are difficult to understand for people who are not algorithm experts. Rule extraction methods create interpretable rule sets that act as classifiers. They have mostly been combined with already labeled data sets. This paper aims to combine unsupervised anomaly detection with rule extraction techniques to create an online anomaly detection framework. Unsupervised anomaly detection uses diffusion maps and clustering for labeling an unknown data set. Rule sets are created using conjunctive rule extraction algorithm. This research suggests that the combination of machine learning methods and rule extraction is a feasible way to implement network intrusion detection that is meaningful to network administrators.**

Keywords—Intrusion detection, anomaly detection, n-gram, rule extraction, diffusion map, data mining, machine learning.

I. Introduction

Web services and networks have become more and more complex in the past years. This means that services and servers face new threats and attacks. *Intrusion detection systems* (IDS) are used to detect these attacks. An IDS works generally using one of two detection principles, *signature-based* and *anomaly-based* detection [1]. Signatures are predetermined attack rules that can be used to trigger an alarm when a user's behavior matches the signature. Previous information about intrusions is required for creating these rules. This leads to a low rate of false alarms, but new and unknown threats cannot be detected. On the other hand, anomaly-based detection systems try to detect traffic that deviates from the normal behavior. New attacks can be detected but this methodology will also lead to some false alarms. Both principles can also be combined to a so-called *hybrid intrusion detection system* [2]. Figure 1 shows a simplified block diagram of the different intrusion detection approaches, demonstrating how our system relates to other approaches.

Information security reseachers have been interested in intrusion detection systems extensively, and surveys describing advances in the field have been published [3], [4]. Many machine learning methods, such as self-organizing maps [5] and support vector machines [6] have been used to cluster data and detect anomalies in these systems. Various hybrid systems combining signature and anomaly-based detection have been used [2], [7]. A two-stage adaptive hybrid system for IP level intrusion detection has also been recently devised. A probabilistic classifier detects anomalies and a hidden Markov model narrows down attacker addresses [8]. Recently genetic algorithms have been widely used in anomaly detection and misuse detection [9], [10]. Artificial immune systems have raised the interest of intrusion detection researchers [11]. More traditional methods such as k nearest neighbors are also still researched because they can be combined with other methods, for example Dempster-Shafer theory of evidence [12]. A distributed environment has been proposed where intelligent agents analyze the network connections using data mining with association rule mining [13]. Moreover, in our previous work we have researched intrusion detection using dimensionality reduction and clustering to find anomalies from network traffic [14], [15].

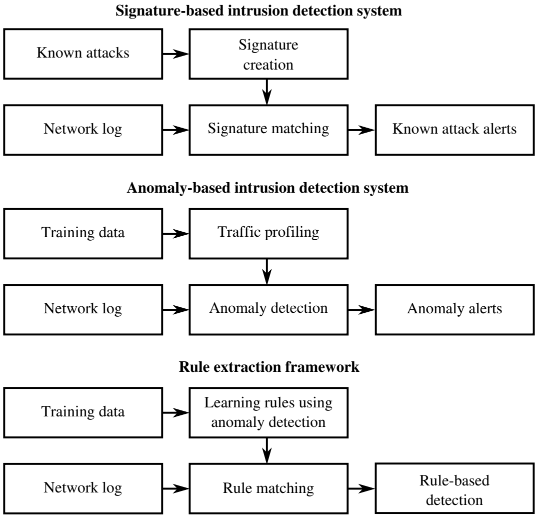

Fig. 1. Different IDS principles.

The problem with deploying anomaly detection systems in the commercial sector is that some algorithms, such as neural networks, work like a black box [16]. The systems are automated and it is difficult to know exactly how the decisions are made. To overcome this problem, *rule extraction* methods have been proposed [17]. These rules can be directly applied to the original data for efficient web traffic filtering. In addition,

This is the authors' postprint version of the article. The original print version appeared as: A. Juvonen and T. Sipola, "Combining conjunctive rule extraction with diffusion maps for network intrusion detection," in *In The Eighteenth IEEE Symposium on Computers and Communications (ISCC 2013)*. IEEE 2013.

this symbolic knowledge can be read and inspected by humans. This can lead to a better understanding of the data and will aid user acceptance especially in real-life company networks.

One way of extracting these rules is taking a decompositional approach [18]. This can be achieved, e.g., by decomposing a neural network architecture. However, methods of this type are algorithm dependent and the rules themselves may not be sufficiently comprehensible [16]. Another way to extract rules is by using pedagogical approach [17]. This approach takes only the input data and output results into account. Therefore, it is not specific to any particular classification method. Any suitable algorithm can be used to find anomalies or cluster data. Also, the produced rules are directly related to original data and can therefore be easily understood. Because of these reasons, we take the pedagogical approach in our system. Various methods have been used to create different kinds of rule sets and trees. Recent research seems to focus on methods based on heuristic algorithms or creating intelligent wrapper methods [19]. A less researched option is to use conjunctive rules [17]. These rule extraction methods should not be confused with association rule mining [20].

We propose a framework for detecting network anomalies and extracting rules from a data set. Figure 1 shows how it differs from other common approaches. This framework is a supplementary module for signature-based intrusion detection systems, such as next generation firewalls. In this approach, network logs or other similar data is collected and preprocessed to extract features and form numerical matrices to be analyzed further. The dimensionality of this data is reduced for more efficient clustering. After clustering the data to normal and anomalous traffic, the obtained clustering is used to create labels for the data. Subsequently, this information is used to create a rule set for the high-dimensional features. This rule set can then be used to classify traffic and detect intrusions in real time. The proposed framework enables rule creation in an unsupervised manner for previously unknown data. Our contribution is combining unsupervised data analysis with rule extraction techniques to create an online anomaly detection system.

II. Methodology

The proposed framework uses training data to create a rule set which can then be used to classify testing data or actual network traffic data. Thus, our approach is divided into two phases: *rule set learning* and *traffic classification*. The first phase takes the approach of learning the clustering of the data using dimensionality reduction and creating conjunctive rules to describe these clusters in the initial feature space. These rules will then be used to classify new incoming traffic in the second phase. This process is described in Figure 2, which shows the needed input data sets, produced rule set and classification results.

The rule set learning phase aims to find rules that describe the training data. This is done by clustering and labeling the training data set. The resulting rule set classifies data according to the obtained clustering. Architecture of the rule set learning process is shown in Figure 3, which shows the labeling and rule extraction phases in more detail. The methods in individual modules are not fixed, meaning that the specific methods can be changed if better alternatives are found. The rule set learning phase consists of the following steps:

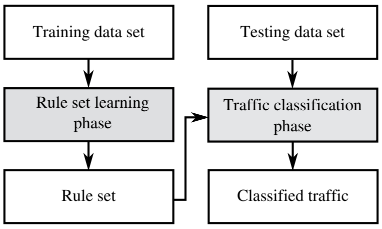

Fig. 2. Block diagram of the overall process.

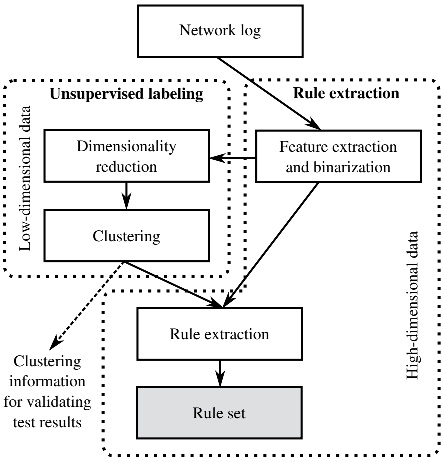

Fig. 3. Block diagram of the rule set learning process.

- Feature extraction from training data
- Unsupervised labeling
    - Dimensionality reduction
    - Clustering
- Rule extraction

In the traffic classification phase new incoming traffic is preprocessed and classified using the generated rule set. Because of the conjunctive nature of the rules simple matching is sufficient. This phase validates how well the rules apply to data that was not part of the training data set. The steps are as follows:

- Feature extraction from testing data
- Classification by rule matching

The following subsections describe the methods used in previously mentioned phases in detail.

A. Feature extraction

Network log files consist of text lines that need to be converted to numeric feature vectors. An n-gram is a consecutive sequence of n characters that represents extracted semantic information [21]. Our study uses 2-gram features generated from the network logs. This approach produces a rather sparse feature matrix [14]. The rule extraction algorithm works with symbolic conjunctive rules. This means that only nominal and binarized data can be used. Converting data to this kind of format ensures that the feature matrix may be used with the overall learning pipeline.

The feature matrix consists of binary values representing whether an n-gram is present in a specific log line or not. Let us consider the following example. Having two strings containing the words `anomaly` and `analysis`, we can construct the feature matrix in the following way:

an	no	om	ma	al	ly	na	ys	si	is
1	1	1	1	1	1	0	0	0	0
1	0	0	0	1	1	1	1	1	1

In this study, 2-grams are used. However, it is possible to use longer n-grams as well. This will result in more dimensions in the matrix, because there will be more unique n-grams, slowing down the process. For the purposes of this research, 2-grams contained enough information for separating normal and anomalous traffic. Also, using $n = 1$ will give the character distribution. Single characters may not contain enough semantic information, and therefore higher values of n are often used.

B. Dimensionality reduction and clustering

Clustering high-dimensional data is facilitated by dimensionality reduction. We employ diffusion map training to identify the attacks in the training data set. The features describing the dataset are numerous and sometimes hard to interpret together. Therefore, a dimensionality reduction approach using diffusion map is taken. Diffusion map training produces a low-dimensional model of the data, which reveals the internal structure of the dataset and facilitates anomaly detection. In addition, it can cope with non-linear dependencies in the data. Diffusion map retains the diffusion distance in the initial feature space as the Euclidean distance in the low-dimensional space [22], [23], [24].

One log line is represented by feature vector $x_i \in \mathbb{R}^n$. The whole data set is $X = \{x_1,\ x_2,\ x_3,\ \dots\ x_N\}$, from which the affinity matrix

$$W(x_i, x_j) = \exp\left(\frac{-||x_i - x_j||^2}{\epsilon}\right)$$

can be calculated. As seen, the Gaussian kernel is used for the distance matrix and the bandwith parameter ϵ is selected from the optimal region in the weight matrix sum [25]. D, which collects W's row sums on its diagonal, and the transition matrix $P = D^{-1}W$ form the symmetric matrix

$$\tilde{P} = D^{\frac{1}{2}} P D^{-\frac{1}{2}} = D^{-\frac{1}{2}} W D^{-\frac{1}{2}}.$$

The singular value decomposition (SVD) of $\tilde{P}$ yields the eigenvectors v_k and eigenvalues λ_k. Now, the low-dimensional coordinates corresponding each original log line are found: $x_i \rightarrow [\lambda_1 v_1(x_i),\ \lambda_2 v_2(x_i) \dots\ \lambda_d v_d(x_i)]$. Most of the information is retained in the first eigenvectors and less meaningful ones are left out. Some information is lost because not all eigenvectors are used, but lower dimensionality makes clustering easier.

The k-means method is used to group the data points into clusters. This method is simple and well-known clustering algorithm and it has been used in various data mining tasks. The algorithm description and examples of use can be found in literature [26], [27], [28]. The k-means method relies heavily on the parameter k which determines the number of clusters. Silhouette expresses the quality of clustering for each data point. The optimal number of clusters for the k-means is determined using average silhouette value [29]. An alternative clustering method could be used.

The obtained clustering is believed to describe behavior of the data. If the high-dimensional features can differentiate normal and intrusive behavior, this should be apparent from the resulting low-dimensional clusters. The actual nature of the clusters should be confirmed with domain area experts.

If performance becomes an issue with larger data sets, the learning process could be expanded with out-of-sample extension. However, representative selection of training data is usually a more challenging problem.

C. Rule extraction

A rule is a way to determine the class of a data point based on certain conditions. Ideally a rule would be easily interpretable by a network administrator. All the possible rules span such a huge space that it is not feasible to go through all of them. This means that a sub-optimal but efficient method needs to be used. Such systems have been applied with neural networks [17], [16] and support vector machines [30], [31], [32].

Conjunctive rule is a logical expression containing truth values about the inclusion of binary features. These rules tell whether a symbol should be included, excluded or if it does not matter. Let us assume that we have binary features a, b, c, d, e. Thus, the feature matrix contains five columns corresponding to each binary feature. For the sake of example we have a rule set containing three rules:

$$\begin{aligned} r_1 =& \neg a && \text{for class } c_1, \\ r_2 =& a \wedge b \wedge c \wedge \neg d \wedge e && \text{for class } c_1, \\ r_3 =& a \wedge b \wedge \neg c && \text{for class } c_2. \end{aligned}$$

The rule set for class c_1 would be expressed in logical form as $R_1 = r_1 \vee r_2$. In practice, there are usually multiple rules for each class. Note that in rule r_1, values of features b, c, d and e do not matter. Similarly, for r_3 values of d and e can be anything.

For implementation purposes, the rules are expressed as vectors. The length of these vectors is equivalent to the number of features. The logical truth values are converted to 1 and -1. The values that do not matter are expressed as 0. In the previous example, the rules would be vectors of length 5. Rule r_1 is expressed as a vector $(-1 \quad 0 \quad 0 \quad 0 \quad 0)$. It is easy to match feature vectors to this kind of rule vectors. Note that in this research a rule symbol corresponds to an n-gram feature as described in II-A.

The conjunctive rule extraction algorithm [17] finds rule-based classifier that approximates the clustering obtained in the unsupervised labeling step. Conjunctive rule extraction is presented in Algorithm II.1. Note that a rule r consists of symbols $r = s_1 \wedge s_2 \wedge s_3 \wedge \ldots \wedge s_n$.

Algorithm II.1: Conjunctive rule extraction.

```
Input: data points $E$, classes $C$
Output: rules $R_c$ that cover $E$ with classification $C$
  repeat
    $e$ := get new training observation from $E$
    $c$ := get the classification of $e$ from $C$
    if $e$ not covered by the rules $R_c$ then
      $r$ := use $e$ as basis for new rule $r$
      for all symbols $s_i$ in $r$ do
        $r'$ = $r$ with symbol $s_i$ dropped
        if all instances covered by $r$ are of the same class
        as $e$ then
          $r$ := $r'$
        end if
      end for
      add rule $r$ to the rule set $R_c$
    end if
  until all training data analyzed
```

The obtained rules separate the training data into the clusters. These rules can now be matched to new incoming data points. Their performance depends on how well the training data covers the behavior of the data. If the point matches one of the rules, the exact type of the abnormal or normal state can be interpreted. If a data point does not fall under any of the rules, then it can be considered abnormal.

Created rules are valid for the classification task while the essential profile of the data remains the same. This is often not the case for extended periods of time, especially for network traffic or similar data. Therefore, rules can be recreated periodically, e.g., daily.

III. Results

This section contains the classification results using real-world network log data. The goal is to perform preliminary validation on real data to test the feasibility of rule extraction in a practical IDS application. The previously described framework was implemented and applied to this data. Data acquisition and analysis are presented below. These results illustrate that the rule set learning phase works on a data that comes from a real-world source.

A. Data acquisition and processing

We use the same network log database that has been used in our previous related research [15]. The data comes from a real-life web server used by a company. Different kinds of intrusion attempts and other abnormal log lines are included in the data. We examine two log files that correspond to different resource URIs. The servers are using Apache server software, which logs network traffic using Combined Log Format. A single log line contains information about the HTTP query:

```
127.0.0.1 - -
[01/January/2012:00:00:01 +0300]
"GET /resource.php?parameter1=value1
&parameter2=value2
HTTP/1.1"
200 2680
"http://www.address.com/webpage.html"
"Mozilla/5.0
(SymbianOS/9.2;...)"
```

The HTTP GET request part of the log line might contain information about SQL injections and other kinds of attacks. This request part is preprocessed using the methods described in Section II-A. Consequently, we get a binary matrix representing whether an n-gram is present in a specific log line or not. The resulting data points are then clustered into normal and anomalous clusters as described in Section II-B. Because the data set is unlabeled, the unsupervised labeling is performed for the whole data set. This is information is used for test result validation as shown in Figure 3.

B. Data analysis

The first data set for initial testing contains 4292 log lines. After preprocessing we find that there are 490 unique 2-grams in the data, resulting in 4292×490 sized feature matrix. Each datapoint now has a label (normal or anomalous) based on the clustering results. This information can be used to extract the rules. We select randomly 2000 data points for rule creation. The whole data set contains 2292 log lines that are not present during rule set learning phase. These remaining lines are our testing data set.

First, we discover that the used algorithm creates 6 rules, 2 for the normal traffic cluster and 4 for the anomalous one. After testing the rules with the whole dataset, all the data points except one match the correct rules. One anomalous data point is not covered by any rule. All of the normal traffic data points match one of the rules. In this case the system works with almost 100% accuracy, which means that the training data represents the testing data well enough.

The second data set contains 10935 log lines. In this data, 414 unique n-grams are found, resulting in a matrix of size 10935×414. After dimensionality reduction, the number of clusters k is determined using the average silhouette value, as described in Section II-B. Figure 4 shows that the data seems to form 4 clusters that are found using k-means algorithm. For rule set learning phase, 8000 data points are used. Other 2935 are used for traffic classification testing. Figure 5 shows all of the data points after dimensionality reduction and clustering used for unsupervised labeling step. As we can see from this visualization, cluster c_4 contains clearly more points than the others.

Rule extraction from the training set produces 15 rules describing 3 of the classes. One class is not featured in the training data and therefore no rules were generated for this class. The testing data set does not contain any samples belonging to class c_1. Out of the 493 data points of class c_3,

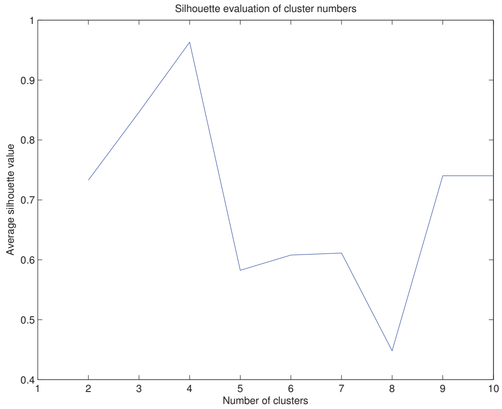

Fig. 4. Optimal cluster number for k-means.

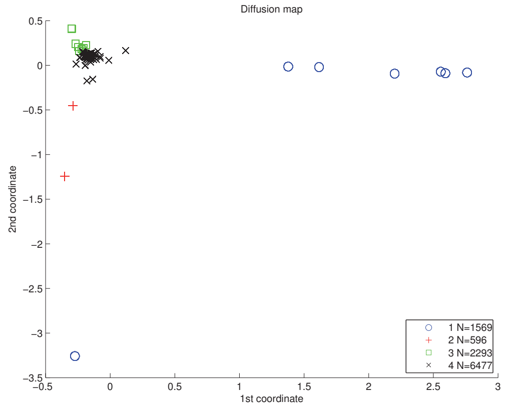

Fig. 5. Two-dimensional visualization of diffusion map of the whole data set.

the extracted rules successfully identify 349 (71%). Test data set contains 2742 data points of class c_4, out of which 1990 are found using the rules (73%). The reason these percentage figures are so low is that the training data differs from the testing data too much. However, the conjunctive rule extraction algorithm always covers the whole training data with 100% identification rate.

IV. Conclusion

Using modern data mining technology in network security context can become problematic when facing end-user needs. Even if the technology produces tangible results, the user rarely has understanding of the methodology. Therefore, this so-called black box system is not a desirable end goal. Simple conjunctive rules are easier to understand, and rule extraction from the complex data mining techniques might facilitate user acceptance. In this research, we have combined rule extraction methodology with diffusion map training framework in order to produce a rule-based network security system.

The main benefit of this framework is that the final output is a set of rules. No black box implementation is needed as the end result is a simple and easy to understand rule matching system. The training data may contain intrusions and anomalies, provided that the clustering step can differentiate them. In addition, rule matching is a fast operation compared to more complex algorithms.

The experimental data sets in this study are suitable for rule generation. The number of created rules is not too high for practical purposes and the accuracy with the first data set is high enough. Data points that do not match any rule could still be flagged as an anomaly in a practical intrusion detection system. The most important thing is to recognize normal traffic accurately. However, if new data points introduced after rule generation are very different from the training data set, the accuracy of classification using the rules might suffer considerably. Periodical rule updating will solve this issue. The second test data set demonstrates how important it is to have a training set that corresponds to the real situation as accurately as possible. If some types of data points are not featured in the rule generation phase, corresponding rules are not generated and these points will not be classified correctly. With proper training data the generated rules give much better accuracy. The created clustering may not represent reality but it is convenient while actual data labels are unknown. Another concern is overfitting of the rules, but the rules can be generalized to mitigate this problem.

The proposed framework is useful in situations where high-dimensional data sets need to be used as a basis for anomaly detection and quick classification. Such data sets are common nowadays in research environments as well as in industry, because collecting data is wide-spread. Our example case has been network security, which bears real benefits to anyone using modern communication networks. The provided tools are useful for network administrators who are trying to understand anomalous behavior in their networks.

Future topics include dynamic rule update as systems evolve, rule set optimization and using the rule set to filter real-time data sets. The modular structure of the framework enables these additions to be implemented conveniently. The applicability of the system to a wider network security context should also be tested, meaning cooperation with other security systems and components such as next-generation firewalls and other signature-based systems.

Acknowledgment

This research was supported by the Foundation of Nokia Corporation. Thanks are extended to Kilosoft Group Oy.

References

[1] K. Scarfone and P. Mell, “Guide to intrusion detection and prevention systems (idps),” *NIST Special Publication*, vol. 800, no. 2007, p. 94, 2007.

[2] M. A. Aydın, A. H. Zaim, and K. G. Ceylan, “A hybrid intrusion detection system design for computer network security,” *Computers & Electrical Engineering*, vol. 35, no. 3, pp. 517–526, 2009.

[3] A. Lazarevic, V. Kumar, and J. Srivastava, “Intrusion detection: A survey,” *Managing Cyber Threats*, pp. 19–78, 2005.

[4] F. Sabahi and A. Movaghar, “Intrusion detection: A survey,” in *Systems and Networks Communications, 2008. ICSNC’08. 3rd International Conference on*. IEEE, 2008, pp. 23–26.

[5] M. Ramadas, S. Ostermann, and B. Tjaden, "Detecting anomalous network traffic with self-organizing maps," in *Recent Advances in Intrusion Detection*, G. Vigna, E. Jonsson, and C. Kruegel, Eds. Springer, 2003, pp. 36–54.

[6] Q. Tran, H. Duan, and X. Li, "One-class support vector machine for anomaly network traffic detection," *China Education and Research Network (CERNET), Tsinghua University, Main Building*, vol. 310, 2004.

[7] H. Om and A. Kundu, "A hybrid system for reducing the false alarm rate of anomaly intrusion detection system," in *Recent Advances in Information Technology (RAIT), 2012 1st International Conference on*, march 2012, pp. 131–136.

[8] R. Rangadurai Karthick, V. Hattiwale, and B. Ravindran, "Adaptive network intrusion detection system using a hybrid approach," in *Communication Systems and Networks (COMSNETS), 2012 Fourth International Conference on*, jan. 2012, pp. 1–7.

[9] L. Li, G. Zhang, J. Nie, Y. Niu, and A. Yao, "The application of genetic algorithm to intrusion detection in mp2p network," in *Advances in Swarm Intelligence*, ser. Lecture Notes in Computer Science, Y. Tan, Y. Shi, and Z. Ji, Eds. Springer Berlin Heidelberg, 2012, vol. 7331, pp. 390–397.

[10] M. Goyal and A. Aggarwal, "Composing signatures for misuse intrusion detection system using genetic algorithm in an offline environment," in *Advances in Computing and Information Technology*, ser. Advances in Intelligent Systems and Computing, N. Meghanathan, D. Nagamalai, and N. Chaki, Eds. Springer Berlin Heidelberg, 2012, vol. 176, pp. 151–157.

[11] A. Parashar, P. Saurabh, and B. Verma, "A novel approach for intrusion detection system using artificial immune system," in *Proceedings of All India Seminar on Biomedical Engineering 2012 (AISOBE 2012)*, ser. Lecture Notes in Bioengineering, V. Kumar and M. Bhatele, Eds. Springer India, 2013, pp. 221–229.

[12] D. Dave and S. Vashishtha, "Efficient intrusion detection with knn classification and ds theory," in *Proceedings of All India Seminar on Biomedical Engineering 2012 (AISOBE 2012)*, ser. Lecture Notes in Bioengineering, V. Kumar and M. Bhatele, Eds. Springer India, 2013, pp. 173–188.

[13] I. Brahmi, S. Yahia, H. Aouadi, and P. Poncelet, "Towards a multiagent-based distributed intrusion detection system using data mining approaches," in *Agents and Data Mining Interaction*, ser. Lecture Notes in Computer Science, L. Cao, A. Bazzan, A. Symeonidis, V. Gorodetsky, G. Weiss, and P. Yu, Eds. Springer Berlin Heidelberg, 2012, vol. 7103, pp. 173–194.

[14] T. Sipola, A. Juvonen, and J. Lehtonen, "Anomaly detection from network logs using diffusion maps," in *Engineering Applications of Neural Networks*, ser. IFIP Advances in Information and Communication Technology, L. Iliadis and C. Jayne, Eds. Springer Boston, 2011, vol. 363, pp. 172–181.

[15] ——, "Dimensionality reduction framework for detecting anomalies from network logs," *Engineering Intelligent Systems*, vol. 20, pp. 87–97, 2012.

[16] N. Ryman-Tubb and A. d'Avila Garcez, "SOAR – Sparse oracle-based adaptive rule extraction: Knowledge extraction from large-scale datasets to detect credit card fraud," in *Neural Networks (IJCNN), The 2010 International Joint Conference on*. IEEE, 2010, pp. 1–9.

[17] M. W. Craven and J. W. Shavlik, "Using sampling and queries to extract rules from trained neural networks," in *In Proceedings of the Eleventh International Conference on Machine Learning*. Morgan Kaufmann, 1994, pp. 37–45.

[18] A. d'Avila Garcez, K. Broda, and D. Gabbay, "Symbolic knowledge extraction from trained neural networks: A sound approach," *Artificial Intelligence*, vol. 125, no. 1, pp. 155–207, 2001.

[19] D. Martens, B. Baesens, and T. Van Gestel, "Decompositional rule extraction from support vector machines by active learning," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 21, no. 2, pp. 178–191, 2009.

[20] J. Hipp, U. Güntzer, and G. Nakhaeizadeh, "Algorithms for association rule mining–a general survey and comparison," *ACM SIGKDD Explorations Newsletter*, vol. 2, no. 1, pp. 58–64, 2000.

[21] M. Damashek, "Gauging similarity with n-grams: Language-independent categorization of text," *Science*, vol. 267, no. 5199, p. 843, 1995.

[22] R. Coifman, S. Lafon, A. Lee, M. Maggioni, B. Nadler, F. Warner, and S. Zucker, "Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 102, no. 21, pp. 7426–7431, 2005.

[23] R. R. Coifman and S. Lafon, "Diffusion maps," *Applied and Computational Harmonic Analysis*, vol. 21, no. 1, pp. 5–30, 2006.

[24] B. Nadler, S. Lafon, R. Coifman, and I. Kevrekidis, "Diffusion maps, spectral clustering and reaction coordinates of dynamical systems," *Applied and Computational Harmonic Analysis*, vol. 21, no. 1, pp. 113–127, 2006.

[25] R. Coifman, Y. Shkolnisky, F. Sigworth, and A. Singer, "Graph laplacian tomography from unknown random projections," *Image Processing, IEEE Transactions on*, vol. 17, no. 10, pp. 1891–1899, oct. 2008.

[26] P. Tan, M. Steinbach, and V. Kumar, "Cluster analysis: Basic concepts and algorithms," *Introduction to data mining*, pp. 487–568, 2006.

[27] A. K. Jain, "Data clustering: 50 years beyond k-means," *Pattern Recognition Letters*, vol. 31, no. 8, pp. 651–666, 2010.

[28] A. K. Jain and R. C. Dubes, *Algorithms for clustering data*. Prentice Hall., 1988.

[29] P. J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *Journal of Computational and Applied Mathematics*, vol. 20, no. 0, pp. 53 – 65, 1987.

[30] H. Núñez, C. Angulo, and A. Català, "Rule extraction from support vector machines," in *In European Symposium on Artificial Neural Networks Proceedings*, 2002, pp. 107–112.

[31] N. Barakat and J. Diederich, "Learning-based rule-extraction from support vector machines," in *The 14th International Conference on Computer Theory and applications ICCTA'2004*, 2004.

[32] N. Barakat and A. P. Bradley, "Rule extraction from support vector machines: A review," *Neurocomputing*, vol. 74, no. 1–3, pp. 178–190, 2010.

PV

RESEARCH LITERATURE MAPPING USING ARTICLE CLUSTERING

by

Paavo Nieminen, Ilkka Pölönen and Tuomo Sipola 2013

Journal of Informetrics, Vol. 7, Iss. 4, pp. 874–886

Research literature clustering using diffusion maps

Paavo Nieminen, Ilkka Pölönen*, Tuomo Sipola

Department of Mathematical Information Technology, P.O. Box 35 (Agora), FI-40014 University of Jyväskylä, Finland

Abstract

We apply the knowledge discovery process to the mapping of current topics in a particular field of science. We are interested in how articles form clusters and what are the contents of the found clusters. A framework involving web scraping, keyword extraction, dimensionality reduction and clustering using the diffusion map algorithm is presented. We use publicly available information about articles in high-impact journals. The method should be of use to practitioners or scientists who want to overview recent research in a field of science. As a case study, we map the topics in data mining literature in the year 2011.

Keywords: knowledge discovery process, literature mapping, data mining, clustering, diffusion map

1. Introduction

A task that researchers in any field of science face is to gain an understanding of what others are doing on the field and how it is currently developing. This is a necessary step when relating the researcher's own work to the bigger picture. The research presented here originates from our interest to answer the following basic questions:

1. What main topics are discussed in current data mining research literature?
2. What are the most frequently mentioned methods in the literature?
3. Which journals publish the different topics within the field of data mining?

Very soon we found out that data mining is a rapidly expanding branch of science with a large number of articles published about it each year. Therefore, gaining a general view about the publication space turns out to be, in practice, quite challenging.

A rigorous way to create a secondary study would be to perform a systematic literature review. Originating from medical sciences, systematic reviews can be used also in other disciplines, exemplified by the adaptation to software engineering by Kitchenham

*Corresponding author.

Email addresses: paavo.j.nieminen@jyu.fi (Paavo Nieminen), ilkka.polonen@jyu.fi (Ilkka Pölönen), tuomo.sipola@jyu.fi (Tuomo Sipola)

Preprint submitted to Journal of Informetrics *September 16, 2013*

(2004). A systematic literature review creates a synthesis about a specific phenomenon by conglomerating the evidence published in primary research papers. There is also a lighter version of systematic literature review called mapping study, or scoping review, that intends to identify groups of current literature and identify gaps for further, more detailed, literature review (Budgen et al., 2008). Mapping study, even if lighter than a systematic review, is still a laborious task to do for a massive body of literature.

As data mining methodologies facilitate the handling of huge data masses, it would seem natural to use them to summarize the research literature itself. After all, a definition of data mining, according to Hand et al. (2001, p. 1), is "*the analysis of (often large) observational data sets to find unsuspected relationships and to summarize the data in novel ways that are both understandable and useful to the data owner.*" As it turns out, others have followed a similar way of thinking and studied the creation of automated tools for literature surveys. For example, Cohen et al. (2006), and later Matwin et al. (2010), use machine learning algorithms to assess the relevance of articles in order to reduce the workload of experts who maintain systematic reviews common in evidence-based medicine.

Our goal greatly resembles those pursued by researchers in the field of scientometrics, which is commonly defined as the quantitative study of science. Ivancheva (2008) provides a categorization of scientometrics methodology for research subjects, information types and method classes. Our research subject can be seen as *science by itself* because we try to understand the structure of a field of science. The field is limited, focused and concrete, so the information type of this research is *operational*. Finally, in the classification of Glänzel (2003) our work positions itself in *structural scientometrics* trying to map the research area.

Classical methods used in science mapping, for example in planning of research policies or finding out structures in scientific communities, include those of co-citation analysis (Small, 1973) and co-word analysis (Callon et al., 1983). Co-citation analysis looks for structure in research literature by analyzing the frequency that an article is cited together with another one in later works. Co-word analysis is based on the idea that the text in scientific publications connects key concepts to each other. In co-word analysis, connections between the concepts emerge from the network of co-occuring words instead of the network of citations made between authors.

For the goal of mapping literature, metadata could be used instead of the full research papers. Metadata is usually more readily available and, additionally, it should contain less noise because it is very focused in content and limited in form. There are existing metadata and article databases for certain fields of science. Some of the more notable examples are CiteSeerX[1], DBLP[2], arXiv[3] and PubMed[4]. CiteSeerX is an online database that collects article metadata focusing primarily on the computer and information sciences. DBLP is a database for computer science focusing on authors. ArXiv covers mathematics, computer science, nonlinear sciences, quantitative biology

[1] http://citeseerx.ist.psu.edu/

[2] http://www.informatik.uni-trier.de/~ley/db/

[3] http://arxiv.org/

[4] http://www.ncbi.nlm.nih.gov/pubmed/

and statistics. PubMed archives biomedical literature citations. There are also existing software frameworks to collect information about scientific articles, for example that of CiteSeerX (Teregowda et al., 2010), using web spider technology and various heuristics to collect metadata and citations. The original article databases, and the metadata repositories, can be accessed via web browser interfaces and in some cases also machine-readable interfaces such as the OAI protocol[5]. A major interdisciplinary database with a significant role in the development of scientometrics is the Thomson Reuters (formerly known as ISI) Web of Knowledge (WoK)[6].

To utilize these databases efficiently, computational methods are required. Current work about literature database analysis seems to focus on analyzing citations. One example of such a system is CiteSpace that finds trends and patterns in scientific literature. It was tested with mass-extinction research and terrorism research (Chen, 2006). There have also been schemes for recommending research papers using citation data with subspace clustering based analysis (Agarwal et al., 2005).

Journal interdisciplinarity has been studied with citation reports by clustering using bi-connected graphs (Leydesdorff, 2004). Leydesdorff & Rafols (2009) used factor analysis to cluster the ISI subject categories. Later, these results were replicated for the revised list of categories (Leydesdorff et al., 2013). The methods can be used to produce global maps of sciences, which are two-dimensional illustrations of global literature, in which subsets such as the publications of researchers or companies can be positioned and compared with each other (Rafols et al., 2010).

Tseng & Tsay (2013) present a data processing pipeline that identifies subfields of science. With Dice coefficient similarity and multi-stage clustering, they cluster journals. They believe that articles form topics or categories which in turn form subfields. The research uses manual cluster labeling, but the task is assisted with text mining. The results include subfield descriptions and visualizations of topical maps.

Crimmins et al. (1999) use their framework to discover information from the Internet. They collect frequently occurring phrases, citation and metainformation, summarizing the results into a contingency table. The framework provides clustering and principal component analysis capabilities. Clustering and visualization produce maps that facilitate the understanding of the searched information. This kind of approach seems reasonable also in the context of scientific articles, because there is a similar graph-like structure.

As further examples, clustering frameworks for more traditional text mining have been used to analyze large text databases. Bravo-Alcobendas & Sorzano (2009) clustered biomedical papers using non-negative matrix factorization and k-means algorithms. Aljaber et al. (2010) used various clustering methods to examine literature concerning high energy physics and genomics. Their datasets are from knowledge discovery competitions and workshop tasks[7]. They show that the combination of citation information and extracted features from full article text produces an efficient way to capture the content of scientific papers.

[5] `http://www.openarchives.org/OAI/2.0/openarchivesprotocol.htm`

[6] `http://wokinfo.com/`

[7] KDD Cup 2003, TREC 2006 and 2007 Genomics Tracks

We view computer-assisted literature mapping as a special case of the process of knowledge discovery in databases, as described by Fayyad et al. (1996a,b), and we shall continue using terminology related to their description, which is presented in Figure 1. The steps from raw data to the goal (knowledge to be discovered) involve selection, preprocessing, transformation and mining of the data, as well as representing and interpreting the discovered patterns. The goal in our case is not so much to aid in matters of policy, but to help a researcher gain an initial understanding of what others are currently doing in the same research field. Therefore, we are interested in applying data mining to the concepts (keywords) being discussed in the literature rather than the authors and their affiliations. The electronic articles that reside in databases owned by journal publishers form the bulk of raw data. Consequently, we want keyword vectors to be the transformed data.

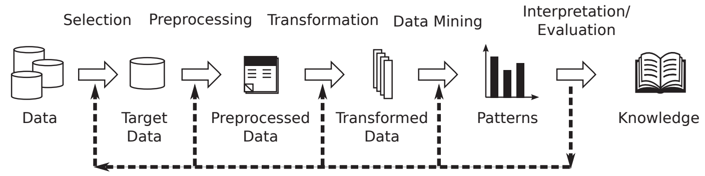

Figure 1: Steps of the knowledge discovery process after Fayyad et al. (1996a).

The technical data mining steps used by Szczuka et al. (2012) in their document grouping and concept identification system are similar to those used in our approach. However, we build upon the clustering approach by using a diffusion map dimensionality reduction step. In addition, our case study analyzes a somewhat larger number of articles. These articles are a subset of scientific literature, and are selected using a specified procedure. Our features are based on the publicly available metadata, while Szczuka et al. use the whole text of the articles, which is feasible when they are easily available.

In this paper, we propose a knowledge discovery and data mining method to create a global view of current topics in a particular field of science using publicly available information about publications in high-impact journals. We compare recent articles using their keywords and title words using a diffusion map data mining approach. The purpose is to find the current snapshot state and structure of the research field based on the data. Maps of science are mostly built upon citation information, but the interests of this article lie in the content of the articles, not connections of citations. In Section 2 we describe the details of our approach, and adapt it to our case study in Section 3. Section 4 presents and discusses the results regarding the data mining literature case study. Section 5 provides a summary of this research.

2. Methodology

We present a clustering framework, which is designed to be useful when searching for a general overview of topics covered in a body of text documents. The major steps in our metadata-based clustering framework follow the adapted knowledge discovery process (Fayyad et al., 1996a,b). The adapted steps include:

1. Selection of relevant literature.
2. Dataset formation (preprocessing and transformation).
3. Data mining the article set with dimensionality reduction and clustering.
4. Interpretation of the summaries obtained from the previous step.

Later on, in Section 3, we present our procedure using data mining literature as an example. However, the steps are in no way limited to any specific field of research that one might want to study.

2.1. Selection of relevant literature

The first step of the process, i.e., selection of the relevant research literature, is important because it defines the publication space. These steps could be automated but at least some initial query from the user must restrict the search. We suggest the following general steps:

1. Identify journals that are likely to be relevant to the field of interest.
2. Focus on the most relevant journals within the identified ones.
3. Decide on further restrictions, e.g., dates of publication.

How this selection is done depends on the research goals. Subsequently, in Section 3, we make suggestions that are based on our experiences and could be used when the goal is similar.

2.2. Article dataset formation

After selecting the body of literature to be studied, metadata needs to be gathered and preprocessed. The main steps, which should mostly be automated, include the following:

1. Gathering data.
2. Normalization of data.
3. Feature extraction.
4. Construction of feature matrix.

Gathering data may be done in various ways, e.g., web scraping, accessing public databases or using public APIs. Data normalization consists of unifying notational conventions and spelling. Feature extraction gathers numerical features from the available textual information. As a final step, a feature matrix is constructed for data mining. The dimensions of this matrix are $n_{\text{articles}} \times n_{\text{features}}$.

2.3. Data mining

In what follows we describe our data mining and analysis steps consisting of article clustering, keyword frequency counting and computation of journal distribution within clusters.

2.3.1. Article clustering

After preprocessing and matrix formation, the data is clustered in order to look for the most dominating groups of topics. The overall procedure of article clustering consists of two steps:

1. Dimensionality reduction using diffusion map.
2. Clustering using agglomerative method with Ward distance.

The first step produces an eigenvector presentation of the transition matrix of the data. This presentation reduces noise in the data, makes the clustering easier and enables visualization. The second step is a simple clustering task.

The binary matrix obtained from data formation step can be of high dimensionality, for example in the order of thousands. In bibliometrics and scientometrics this problem is commonly solved with a combination of hierarchical clustering and multidimensional scaling (MDS) for dimensionality reduction (Boyack et al., 2005; Waltman et al., 2010). Our approach is fundamentally the same, but instead of MDS we employ the diffusion map algorithm (Coifman & Lafon, 2006). It finds a low-dimensional representation using the singular value decomposition of a transition probability matrix based on some chosen distance function. Thus, the high-dimensional data points become embedded in a lower-dimensional space. The dimensionality reduction yields a space where the Euclidean distance corresponds to the diffusion distance in the original space (Coifman & Lafon, 2006; Nadler et al., 2008).

Let us consider a dataset $X = \{x_1, x_2, \ldots, x_n\}, x_i \in \{0, 1\}^p$, that consists of vectors of binary digits, where n is the number of data points and p is the number of measured features. The initial step of the diffusion map algorithm calculates the affinity kernel matrix W, which has data vector distances as its elements:

$$W_{ij} = \exp\left(-\frac{\mathrm{dist}(x_i, x_j)}{\epsilon}\right),$$

where $\mathrm{dist}(x_i, x_j)$ is the similarity measure of Jaccard (1901). Our algorithm uses this for the initial distance matrix between the articles, because only the non-zero elements should contribute to the distance metric. A kernel is used in order to bring close points closer and to increase the distance to distant points.

The row sum diagonal matrix $D_{ii} = \sum_{j=1}^{n} W_{ij}, i \in 1 \ldots n$ is used to normalize the W matrix: $P = D^{-1}W$. This matrix represents the transition probabilities between the data points. The conjugate matrix $\tilde{P} = D^{\frac{1}{2}} P D^{-\frac{1}{2}}$ is created in order to find the eigenvalues of P. With substitution we get

$$\tilde{P} = D^{-\frac{1}{2}} W D^{-\frac{1}{2}}.$$

This normalized graph Laplacian (Chung, 1997) preserves the eigenvalues (Nadler et al., 2008). Singular value decomposition (SVD) $\tilde{P} = U \Lambda U^*$ finds the eigenvalues $\Lambda = \mathrm{diag}([\lambda_1, \lambda_2, \ldots, \lambda_n])$ and eigenvectors $U = [u_1, u_2, \ldots, u_n]$ for $\tilde{P}$. The eigenvalues for P are the same as for $\tilde{P}$. The eigenvectors for P are found with $V = D^{-\frac{1}{2}} U$ (Nadler et al., 2008). The low-dimensional coordinates Ψ are created using $\Psi = V\Lambda$. Only a few of these coordinates are needed to represent the data to a certain degree of error (Coifman & Lafon, 2006).

Basically, the row-stochastic Markov matrix P corresponds to modes of a random walk on the data. It should be noted that the eigen-analysis is based on the distance matrix rather than the data matrix. The use of the kernel brings the neighborhood closer to the point. Points that are close to each other on the graph are also close in the embedded space. Diffusion map has a fundamental difference to principal component analysis (PCA) and multi-dimensional scaling (MDS) there: it also reveals nonlinear relationships between features in embedded space. Linear projections (PCA and MDS) cannot show these.

Diffusion map facilitates the clustering by simplifying the representation of data. Therefore, simple clustering methods can be used to find relevant structure of the data. For clustering the articles using the low-dimensional coordinates, we apply agglomerative clustering using the Ward method for cluster distances. The agglomerative hierarchical clustering scheme is discussed in Everitt et al. (2001, ch. 4) and Hastie et al. (2011, p. 523). The number of clusters is determined using the silhouette measure; the number yielding the highest average silhouette for a clustering is chosen, as recommended by Rousseeuw (1987). When compared to the brief overview by Waltman et al. (2010) our combination of diffusion map dimensional reduction and clustering seems to be unique in the field of science mapping, although it is previously shown to be both theoretically sound and applicable to many real-world tasks, including document clustering (Lafon & Lee, 2006).

The clustering usually has a dense center forming one cluster and a few sparser clusters that stand out. For this reason, the clustering was repeated using only the remaining center, which we call the residual cluster. We end up with an overall iterative data clustering method that includes the following steps:

1. Dimensionality reduction using diffusion maps.
2. Agglomerative clustering with optimal silhouette.
3. Take small clusters as results, and remove them from further analysis.
4. Continue from step 1 using the big residual cluster until stopping criterion is met.

2.3.2. *Keyword frequency and journal distributions*

Simple keyword analysis helps to identify the topics that have been discussed the most in the examined set of articles. The number of how many articles include each keyword is counted. A simple sum over all the articles yields overall keyword frequencies. In our case study, the purpose of this step was to find out the most common methods and applications in current literature.

As yet another additional piece of information, we compute the distribution of journals in the clusters. Each article in a cluster belongs to a single journal and it is easy to create a frequency table. This table supports the knowledge discovery task by showing the relations between the generated clusters and the journals.

2.4. *Interpretation*

The data mining analysis step produces summaries of the data which need to be interpreted by the user. They can be presented in the form of visualizations, tables and lists. The evaluation of the results depends on the initial search goals. It is up to the user to decide whether the obtained clustering, structural visualization and found categories are sensible. We do this verification by comparing the results with published expert opinions.

2.5. *Comparison with other scientometric methods*

A short comparison with other analysis methods is provided, because the reader might not be familiar with our approach.

Traditional co-word analysis compares word pairs found in literature. The pairs are created from the body of literature and the co-occurrence frequencies are collected to a matrix (Callon et al., 1983). These words and their relations are believed to define concepts in the scientific field. The concepts can be connected and clustered using graph algorithms. However, the approach described in our research clusters *articles*, not word pair concepts. We measure the distances between articles using keywords. Naturally, word co-occurrence plays a part also in our method via the chosen Jaccard distance metric and the diffusion process modelled by the dimension reduction algorithm.

OpenOrd is a highly scalable citation graph based method for science mapping (formerly known as as VxOrd or DrL), used by Boyack et al. (2005). OpenOrd uses state of the art graph algorithms to produce (x, y) -coordinates and pruned edge distances for the articles being examined. Standard clustering methods, such as k-means can then be used to find structure in the data. Albeit similar, our method differs in three major ways. First, we use keywords instead of citations in the similarity matrix computation. Second, the optimization problem being solved is different. Visualization methods try to optimize for clarity, while diffusion map aims to retain the diffusion distance. Third, the dimensionality of our output space can be more than two, as our main goal is clustering rather than distance visualization.

3. Adaptation for the case study

This section presents an adaptation of the methodology for the case study. The abstract steps introduced in Section 2 are now applied to current data mining literature. Figure 2 shows the adapted knowledge discovery steps to fit the task of mining specified literature. Each step now contains more phases and the detailed execution has to be determined. The redefined steps are as follows:

1. Selection of relevant literature using impact factors and manual screening of journals.
2. Dataset formation (automatic preprocessing and transformation), including web scraping, filtering, normalization and title conversion.
3. Data mining with dimensionality reduction and clustering.
4. Interpretation of the summaries obtained from the previous step and comparison with published expert opinions.

These steps are detailed in the following subsections and the motivation behind them is discussed.

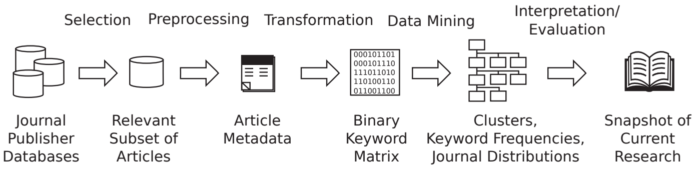
Selection
Preprocessing
Transformation
Data Mining
Interpretation/ Evaluation
000101101
000101110
111011010
110100110
011001100
Journal Publisher Databases
Relevant Subset of Articles
Article Metadata
Binary Keyword Matrix
Clusters, Keyword Frequencies, Journal Distributions
Snapshot of Current Research

Figure 2: Our adaptation of the knowledge discovery process for mapping research literature based on Fayyad et al. (1996a), cf. Figure 1.

3.1. Selection of relevant literature

In this practical case the literature selection step in the methodology is specialized to find the most relevant journals and articles. In order to limit our data to only articles concerning the field of data mining, we used the following restrictions:

1. Selecting journals that are listed in WoK.
2. Limiting the WoK subject categories.
3. WoK impact factor over a threshold.
4. Further voting about the relevance to data mining.
5. Limiting the target time frame.

To identify relevant journals, we suggest using the impact factor metric published yearly in the Journal Citation Report[8] of WoK. Impact factor (Garfield, 1972) is a numerical value, that provides a quantitative tool for ranking journals based on their impact to a field of science. The impact factor is computed by dividing the number of citations made to the articles of a journal by the total number of articles published during a time window. Longer-term impact factors and trend graphs are available from WoK, but we restricted our scope to one-year impact factors in order to get a recent picture of the quickly developing field of data mining, with the newest journals included. Impact factors of 2010 were the most recent ones available when starting our work.

Despite its limitations and pitfalls, discussed, for example, in Seglen (1997), impact factor is regarded as a *de facto* tool for assessing the relevance of journals. Therefore, we chose to restrict our study to journals with impact factor higher than the arbitrarily selected threshold of 1.0 in order to focus only on the most cited research. Comparing impact factors might not generalize to very interdisciplinary topics, because the metric is not comparable across fields of science due to different citation cultures. However, in our case of data mining, we expect the topic to be covered mostly by journals focusing on computer science, statistics and mathematics, between which we expect the citation culture to be similar.

The Thomson Reuters Web of Knowledge divides the listed journals to 176 subject categories. Not all of these categories are related to the field of science that is in focus. In our study we selected the following categories, which in our opinion should contain

[8] `http://wokinfo.com/products_tools/analytical/jcr/`

most of the work related to the field of data mining: “computer science, artificial intelligence”; “computer science, information systems”; “computer science, interdisciplinary applications”; “mathematics, applied”; “mathematics, interdisciplinary applications”; “statistics & probability”.

In Tables 1 and 2, we list the journals that were initially identified as the candidate data sources for this study, i.e., they were listed in the WoK, had an impact factor of at least 1.0 and included one of the words *Data Mining (dm)*, *Data Engineering (de)*, *Knowledge Discovery (kd)*, *Knowledge Engineering (ke)* or *Data Analysis (da)* in their editorial statements or public scope definitions. The technical filtering did not seem to single out the most data mining related journals, perhaps due to the term data mining being a buzzword used more than it factually should. So finally, to focus on the most relevant ones, we voted for inclusion of journals based on inspection of the journals’ editorial statements and preliminary browsing of their content. The threshold of inclusion was that at least two of the three authors regarded the journal relevant. In Table 1, we show the journals that were finally selected for inclusion in this study, based on subjective evaluation of each journal’s relevance to our research questions. In Table 2, we list the journals that were initially identified but finally rejected. The last column of the tables shows the number of relevance votes that each journal received from the authors.

In our case study, the purpose was to get a snapshot of recent publications, so we chose to restrict our study to the articles published during the year 2011.

3.2. Article dataset formation

We built our database using web scraping to collect data directly from the journal databases via the public WWW interfaces provided by the publishers. Other sources could be added for further studies. For this study the scraper reads the WWW pages of the journal publishers and yields a database entry for each article, including the title, keywords and name of the journal where the article has been published. All published titles from each journal will be listed at this stage, including many non-essential ones, such as editorial comments, letters to the editor, book and software reviews and calls for papers. These non-essential titles are then automatically filtered out based on words contained in the title. Our approach does not extract keywords from the text. Instead, it uses the available metadata and assumes that they are correctly entered by the authors.

Also, some further pre-processing was found to be necessary because of varying formats and conventions found in the data sources. Notational conventions were occasionally found to differ also between different articles within a journal. These discrepancies necessitate a technical cleaning step, where HTML tags are removed, Greek letters and mathematical symbols are converted to corresponding $\LaTeX$ expressions, and the separating characters in keyword lists are heuristically chosen. In order to further normalize the keyword lists, we created an automatic tool that converts plurals to singular form, and British English spellings into their American English equivalents.

Feature extraction from the metadata is straightforward. The occurrence of keywords describes the contents of an article, which means that a binary feature vector can be used to represent an article. While inspecting the author-defined lists of keywords,

Table 1: Selected journals after relevance vote.

Selected journal	Scope	Publisher	rel.
ACM Transactions on Information Systems	dm,kd	ACM	2
Applied Soft Computing	dm	Elsevier	2
Bayesian Analysis	dm	ISBA	3
Computational Statistics & Data Analysis	dm,da	Elsevier	3
Computer Journal	dm	Oxf.UP	3
Data Mining and Knowledge Discovery	dm,kd,da	Springer	3
Fuzzy Sets and Systems	da	Elsevier	2
Genetic Programming and Evolvable Machines	dm	Springer	2
IEEE Transactions on Knowledge and Data Engineering	de	IEEE	2
Information Sciences	de,ke	Elsevier	3
International Journal of Approximate Reasoning	da	Elsevier	2
International Journal of Information Technology & Decision Making	dm	World Sc.	2
International Journal of Innovative Computing Information and Control	dm,kd, da	ICIC	2
Journal of Computational and Graphical Statistics	da	ASA	2
Knowledge and Information Systems	dm,de, kd,ke	Springer	2
The Knowledge Engineering Review	ke	Cambr.UP	2
Machine Learning	dm	Springer	3
Pattern Analysis and Applications	ke	Springer	2
Pattern Recognition Letters	dm	Elsevier	3
Statistics and Computing	dm,da	Springer	3

Table 2: Excluded journals after relevance vote.

Excluded journal	Scope	Publisher	rel.
ACM Transactions on Database Systems	dm	ACM	0
ACM Transactions on Internet Technology	dm,kd	ACM	0
ACM Transactions on the Web	dm	ACM	0
Artificial Intelligence in Medicine	ke	Elsevier	0
Computer-aided Civil and Infrastructure Engineering	de	Wiley	0
Computers in Industry	ke	Elsevier	0
Data & Knowledge Engineering	de,ke	Elsevier	0
Electronic Commerce Research and Applications	dm	Elsevier	0
Environmental Modelling & Software	dm	Elsevier	0
Expert Systems with Applications	kd	Elsevier	1
Information Systems	dm	Elsevier	1
Integrated Computer-Aided Engineering	kd	IOS Press	0
Journal of Database Management	dm,ke	IGI Publ.	1
Journal of Hydroinformatics	ke	IWA Publ.	0
Journal of Molecular Modeling	da	Springer	0
Journal of Quality Technology	ke	ASQ	0
Journal of Web Semantics	kd	Elsevier	0
Psychometrika	da	Springer	0
SAR and QSAR in Environmental Research	da	Taylor&Fr.	0
Stata Journal	da	StataCorp	1
World Wide Web – Internet and Web Information Systems	dm	Springer	0

we found out that the keywords, even after normalization, were quite different from each other, even when the articles could have been related to similar topics based on their titles. To improve the situation, our system augments the list of keywords in the following way:

1. List all of the original keywords (for example “face recognition”).
2. Add to the list also split, i.e., single-word, versions of the original keywords (for example “face” and “recognition”).
3. Remove common English stopwords (such as “a”, “the”, “in”, “and”, . . .) from the list.
4. Remove also some additional words very common in scientific titles (such as “using”, “based”, “novel”, “new”, . . .).

Each article is then judged by the software to be related to a keyword in the list if the keyword is found within the title or within one of the keywords of the specific article. For example, an article with the title “About face recognition” would be related to the keywords “face recognition”, “face” and “recognition”. The information is stored as a binary matrix where each row corresponds to an article and each column to a keyword in our augmented keyword list. A non-zero element means that the keyword is found from the title or keyword list of the article.

At the end of this step, we automatically remove singleton keywords and articles, i.e., keywords that appear only once and articles that contain no keywords common with any other article. Such singleton words are irrelevant in analyzing connections between the articles. In our case study, the final keyword list contained 11,844 words or phrases, and with 2,511 articles the size of the matrix was $2,511 \times 11,844$. After removal of singleton words and articles, 4,187 keywords and 2,499 articles remained. The data matrix of size $2,499 \times 4,187$ was sparse; only 0.3% of its values were ones instead of zeros.

3.3. *Data mining*

The data mining step follows closely the article clustering approach presented in Section 2.3.1. Figure 3 shows the clustering results for our case study at the first iteration level. The visualization uses the first three dimensions, although empirically chosen first six dimensions were used in the analysis. These coordinates in the figure correspond to the three largest eigenvalues obtained from the diffusion map algorithm.

The iterative approach clusters the articles into several categories, which can be used to analyze the structure of the dataset. The obtained clusters vary considerably in size. Inspection of the keywords and titles in the clusters reveal that the separated clusters have high semantic cohesion. The iteration is stopped when the total size of the separated clusters becomes smaller than 2% of the original data. We conjecture that the most important clusters according to the keyword vectors are found during the first few iterations.

4. Results of the case study

This section presents the results of our case study with key findings and answers to the original research questions: the main topics, most frequent methods and journal

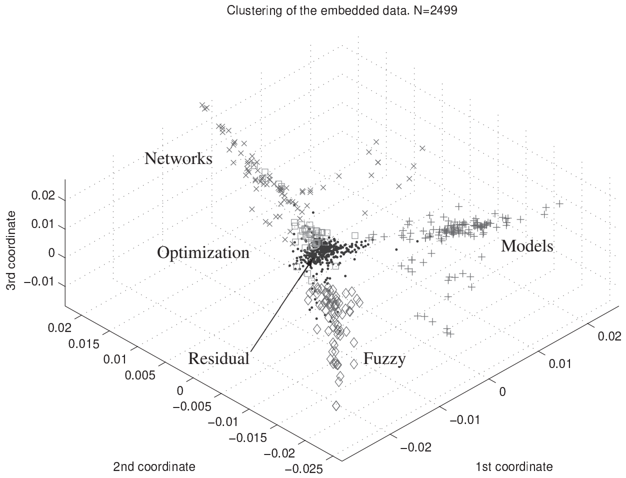

Figure 3: Low-dimensional embedding of the article dataset. Each point corresponds to one article. Different clusters are marked differently, and given their interpreted names. The dense residual cluster can be seen in the middle. For visual clarity, only one third of the data (randomly selected) is plotted in this figure.

identities in the field of data mining, taking into account the restrictions set by the article selection process. We find that the most convenient order is to report the findings from simple keyword frequency counts first, and then to continue with the results from clustering and journal distributions.

4.1. Keyword frequency analysis

Of the 4187 keywords only some were obviously related to data mining methods. This led to a subjective screening of the keywords. The most common method-related keywords and their frequencies were *fuzzy* (327), *optimization* (198), *classification* (172), *clustering* (119) and *Bayesian* (112). All of these are rather general method families.

The more specific method families were not mentioned as frequently. The following list includes notable examples of these method-related keywords: *neural network* (63), *genetic algorithm* (62), *stochastic* (53), *particle swarm optimization* (42) *support vector machine* (40), *fuzzy logic* (36), *feature extraction* (30), *feature selection* (30), *pattern recognition* (27), *evolutionary algorithm* (26), *self-organizing* (23), *decision tree* (19), *genetic programming* (18), *reinforcement learning* (17), *hidden Markov model* (17), *PCA* (16), *differential evolution* (15), *self-organizing map* (14), *dimensional reduction* (14), *least squares* (13), *kernel method* (13), *Kalman* (13), *fuzzy clustering* (12), *k-means* (11), *manifold learning* (11), *feature detection* (8), *c-means* (8) and *independent component analysis* (6).

Some other findings, that were omitted from the above list, are worthy of a short discussion. There were 63 articles that had *data mining* itself as a keyword. The frequencies of keywords *linear* (125) and *non-linear* (61) tell something about the expected result that linear methods are studied or used more widely. Four often mentioned application areas were *face recognition* (32), *wireless sensor network* (30), *image segmentation* (23) and *text analysis* (12).

4.2. Structural view using clustering

The iterative clustering resulted in 19 clusters on five iteration levels and a final residual cluster of size 598. Therefore, 76% of the data falls within these 19 identified clusters. Figure 4 illustrates the levels of the iterative clustering process. The clusters are manually labeled from the most common keywords inside them. On the highest level, the original data of 2,499 articles was clustered into four smaller clusters and a residual cluster of 1,459 articles. We chose a descriptive name for each cluster by examining the 10 most common keywords in the cluster.

Thus, the highest level revealed the following clusters (number of articles in parentheses): *Models* (388), *Networks* (241), *Fuzzy* (239) and *Optimization* (172). The *Models* cluster included also keywords such as *Bayesian*, *fuzzy*, *Markov* and *regression*. The *Networks* cluster covered both *neural networks* and *sensor networks*. The *Fuzzy* cluster included topics such as *fuzzy sets* and *fuzzy logic*. Finally, the *Optimization* cluster included *particle swarm optimization* and topics related to *evolutionary* and *genetic algorithms*.

The second level was obtained by clustering the residual cluster (1,459 articles) of the first level. Clusters on this level were named *Images*, *Learning*, *Face/Pattern*

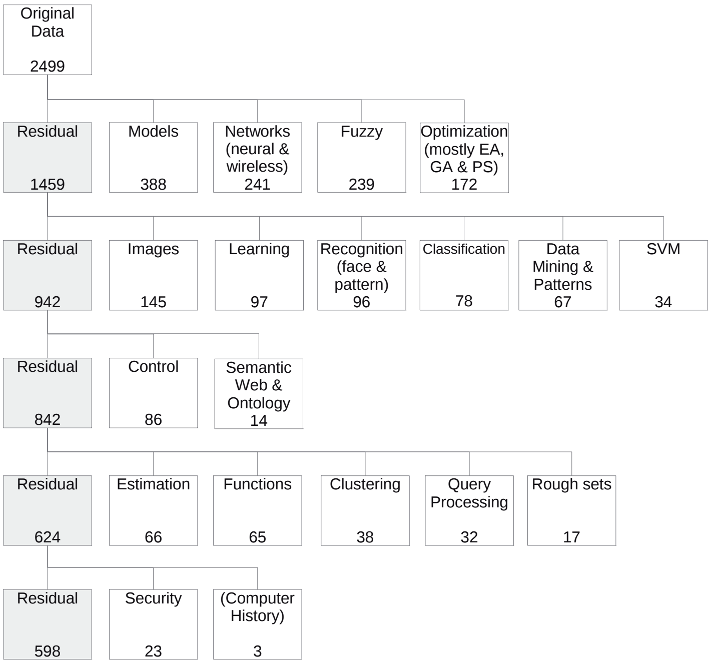

Figure 4: Clusters found during the first five iterations of the algorithm. The numbers tell how many articles fall into the respective clusters. Names are given by inspection of cluster contents.

recognition, *Classification*, *Data mining & Patterns*, and *SVM*. Like on the first level, the descriptive names were chosen on the basis of the 10 most common keywords in the clusters. For example, common keywords in the *Images* cluster contained *image segmentation*, *image retrieval* and *classification*.

The third level extracted two new clusters that we call *Control* and *Semantic web & Ontology*. The fourth level revealed the clusters of *Estimation*, *Functions*, *Clustering*, *Query Processing* and *Rough Sets*. The fifth level yielded one more larger cluster, *Security*, and a very small cluster *Computer History*. The ending criterion was met on this level.

4.3. *Journal distribution*

The number of articles is not uniformly distributed among the journals, as shown in Table 3. It is also seen that each journal has its own areas of interest with respect to the clusters identified by this study. For example, Pattern Recognition Letters publishes articles related to the clusters *Recognition* and *Images*; in contrast, articles published in Fuzzy Sets and Systems belong to the *Fuzzy* cluster. On the other hand, journals like International Journal of Innovative Computing, Information and Control (IJICIC) and Information Sciences relate to almost all the clusters in the taxonomy discovered by our framework.

4.4. *Discussion*

This case study presented one viewpoint to understand recent data mining literature. This discussion compares our results to the expert opinion. The advancement of the field has been of interest to the community, and accordingly some overviews have been made. Among recent overview literature there are some interesting papers, such as the one by Kriegel et al. (2007) where the authors envision the major challenges in data mining and knowledge discovery today and especially in the future. Venkatadri & Reddy (2011) give a general overview of current and future trends in data mining. In a similar manner, Kumar & Bhardwaj (2011) review potential future application areas. Wu et al. (2008) give a list of top data mining algorithms based on the opinions of an expert panel. We contribute to this discussion by the quantitative results presented above. Although interesting and enlightening reading, the current reviews and position papers seem to be somewhat restricted in their scope of selected literature, whereas our study attempts to sample the current state of the leading data mining research holistically with an objective, structured and more unbiased method that is based on a methodically selected subset of literature.

The definition of current data mining research is, to an extent, a question of opinion. However, our results seem to adhere to the opinions of other data mining experts. The findings in Section 4.1 about methods are quite similar to KDnuggets poll answers[9], where "academic" persons' most used algorithms in data mining in 2011 were genetic algorithms, support vector machines and association rules. In their brief review, Venkatadri & Reddy (2011) recognize neural networks, fuzzy logic and genetic

[9] `http://www.kdnuggets.com/polls/2011/algorithms-analytics-data-mining.html`

Table 3: Journal distribution in clusters

	ACM Trans. Inf. Syst.	Appl. Soft. Comput.	Bayesian Anal.	Comput. Stat. Data An.	Data Min. Knowl. Disc.	Fuzzy Set Syst.	Genet. Program. Evol. M.	Inform. Sciences	Int. J. Approx. Reason.	IJITDM	IJICIC	J. Comput. Graph. Stat.	IEEE T. Knowl. Data En.	Knowl. Inf. Syst.	Mach. Learn.	Pattern Anal. Appl.	Pattern Recogn. Lett.	Stat. Comput.	Comput. J.	Knowl. Eng. Rev.
Models	4	15	27	92	6	10	1	35	12	8	51	14	11	4	16	7	33	27	13	2
Networks	1	31	1	2	3	4	2	30	7	1	67	0	14	7	5	1	4	2	59	0
Fuzzy	0	34	0	2	0	64	0	57	18	9	45	0	4	2	0	0	4	0	0	0
Optimization	0	42	1	4	0	0	12	38	0	2	43	0	8	2	3	2	3	1	9	2
Images	1	5	0	5	1	0	1	15	0	1	36	0	2	0	0	8	63	0	7	0
Learning	1	6	0	0	3	0	1	17	2	2	7	1	8	6	21	1	16	0	3	2
Recognition	0	5	0	0	1	0	0	2	1	1	20	0	3	1	0	1	58	0	3	0
Classification	1	4	0	3	3	1	0	8	5	2	11	0	8	5	1	3	20	2	0	1
Data Mining & Patterns	1	5	0	0	6	0	0	14	1	2	6	0	14	15	3	0	0	0	0	0
SVM	0	4	1	3	0	0	0	7	0	0	5	1	1	1	1	1	9	0	0	0
Control	0	3	0	8	0	0	0	7	0	0	63	0	0	0	0	0	2	0	3	0
Semantic Web & Ontology	0	0	0	0	0	0	0	1	1	1	1	0	1	2	0	0	0	0	0	7
Estimation	0	0	0	21	0	1	0	7	2	1	10	3	1	1	0	1	13	5	0	0
Functions	0	1	0	10	0	9	0	15	14	2	8	0	2	2	0	0	2	0	0	0
Clustering	0	0	0	2	0	0	0	6	0	0	4	2	3	3	0	0	16	1	1	0
Query Processing	0	0	0	0	0	0	0	2	0	0	0	0	27	1	0	0	0	0	2	0
Rough sets	0	0	0	0	0	3	0	8	3	0	2	0	0	0	0	0	1	0	0	0
Security	0	0	0	0	0	0	0	8	0	0	12	0	0	0	0	0	0	0	3	0
Computer History	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0
Residual	7	8	2	75	8	22	7	71	28	18	116	33	21	23	9	8	57	13	56	16
TOTAL	16	163	32	227	31	114	24	348	94	50	507	54	128	75	59	33	301	51	162	30

programming as the future trends of data mining. Our results in Section 4.2 agree with their findings since corresponding clusters were found already on the first level of our iterative algorithm. Journal distribution analysis in Section 4.3 showed that most journals specialize in just a few topics. However, some journals publishing more diverse topics were also found. The journals adhere to the obtained clustering quite closely, which can help a researcher to select a publication venue.

Overall, our findings seem to agree with the definition of data mining by Hand et al. (2001), which suggests that what is done currently under the label of data mining still studies the problems stemming from the definition given over ten years ago.

4.5. Benefits and limitations

In our study, we did not use an existing benchmark corpus because one main goal of the research was to apply the method to immediately gain new information about recent data mining literature. The method is verified by comparing it to existing expert opinion instead. We wanted to base our study on freely available public data, which excludes full texts in many cases. This unfortunate fact was noted also by some of the researchers we have cited above. The use of full texts would have given a larger feature space and produced more noise. While the main connections might be the same as when using metadata, the additional data mass could have created unforeseen connections between articles that cannot be produced with mere metadata.

To our knowledge this is a unique study of this kind performed on recent data mining literature, which should make the results useful for the data mining community.

5. Conclusion

Following the knowledge discovery process, we created a literature mapping framework based on article clustering. It can be used to analyze topics of current interest in a particular field of science. As a case study, we tested the framework with data mining research literature. Our approach uses publicly available metadata about articles published in high-impact journals. The proposed methodology can be automated, but a more delicate screening may use manual approach in needed steps. In the case study, the data source selection and interpretation included manual work. The methodology is mainly automated and the individual steps can be changed if a more fitting method is discovered. Because of automation the process is less biased than surveys that use opinion-based approach.

The clustering enables a researcher to get a quick overview of the topics published in the selected body of literature. The system may reveal unexpected articles under a topic label, because an article can be connected to the cluster via keywords other than the obvious cluster label. Thus, the structural view could be used as a search strategy that complements a simple keyword search. Also, a starting point for a quick literature review on a topic, for example “Security applications of data mining” which was a cluster found in our case study, could be the articles within the particular cluster. Larger clusters corresponding to more general topics, such as “Optimization in data mining”, could be taken as a basis of a new clustering, in order to find and categorize subtopics. For the goals in our case study, though, the initial granularity was sufficient.

Our methodology should be helpful for individuals and companies trying to gain an understanding of large textual datasets, e.g., personal or company internal documentation. It should be useful also for the application field scientists and companies who want to find methods that are currently used widely.

The clustering framework could be used with many different datasets, large or small. There may be scalability issues with larger datasets due to the dimensionality reduction and clustering methods used. Another problem with a large dataset is that some details could be lost in noise. However, when searching for a general overview, this is not a big problem.

Currently the output of our method is a snapshot of current published articles. Combining a longitudinal point of view might reveal long-term trends in research literature. Our approach could benefit from additional information gained from features extracted from abstracts. Abstracts are usually freely available in addition to keywords and titles, whereas other parts of the articles might not be.

Acknowledgements

We are grateful to the anonymous reviewers whose comments have significantly improved the clarity of this paper. This research was partially supported by the Foundation of Nokia Corporation and the Finnish Foundation for Technology Promotion.

References

Agarwal, N., Haque, E., Liu, H., & Parsons, L. (2005). Research paper recommender systems: A subspace clustering approach. In W. Fan, Z. Wu, & J. Yang (Eds.), *Advances in Web-Age Information Management* (pp. 475–491). Berlin Heidelberg: Springer volume 3739 of *Lecture Notes in Computer Science*.

Aljaber, B., Stokes, N., Bailey, J., & Pei, J. (2010). Document clustering of scientific texts using citation contexts. *Information Retrieval*, *13*, 101–131.

Boyack, K. W., Klavans, R., & Börner, K. (2005). Mapping the backbone of science. *Scientometrics*, *64*, 351–374.

Bravo-Alcobendas, D., & Sorzano, C. (2009). Clustering of biomedical scientific papers. In *Intelligent Signal Processing, 2009. WISP 2009. IEEE International Symposium on* (pp. 205–209).

Budgen, D., Turner, M., Brereton, P., & Kitchenham, B. (2008). Using mapping studies in software engineering. In *Proceedings of PPIG* (pp. 195–204).

Callon, M., Courtial, J.-P., Turner, W. A., & Bauin, S. (1983). From translations to problematic networks: An introduction to co-word analysis. *Social Science Information*, *22*, 191–235.

Chen, C. (2006). CiteSpace II: Detecting and visualizing emerging trends and transient patterns in scientific literature. *Journal of the American Society for Information Science and Technology*, *57*, 359–377.

Chung, F. R. K. (1997). Spectral graph theory. (p. 2). Providence, RI: AMS Press.

Cohen, A. M., Hersh, W. R., Peterson, K., & Yen, P.-Y. Y. (2006). Reducing workload in systematic review preparation using automated citation classification. *Journal of the American Medical Informatics Association: JAMIA*, *13*, 206–219.

Coifman, R. R., & Lafon, S. (2006). Diffusion maps. *Applied and Computational Harmonic Analysis*, *21*, 5–30.

Crimmins, F., Smeaton, A. F., Dkaki, T., & Mothe, J. (1999). Tétrafusion: Information discovery on the internet. *IEEE Intelligent Systems*, *14*, 55–62.

Everitt, B. S., Landau, S., & Leese, M. (2001). *Cluster Analysis*. (4th ed.). London, Arnold; New York: Oxford University Press.

Fayyad, U., Piatetsky-Shapiro, G., & Smyth, P. (1996a). From data mining to knowledge discovery in databases. *AI Magazine*, *17*, 37.

Fayyad, U., Piatetsky-Shapiro, G., & Smyth, P. (1996b). The KDD process for extracting useful knowledge from volumes of data. *Communications of the ACM*, *39*, 27–34.

Garfield, E. (1972). Citation analysis as a tool in journal evaluation. *Science*, *178*, 471–479.

Glänzel, W. (2003). Bibliometrics as a research field. A course on theory and application of bibliometric indicators. Course Handouts.

Hand, D., Mannila, H., & Smyth, P. (2001). *Principles of data mining*. Adaptive computation and machine learning. Cambridge, Mass.: MIT Press.

Hastie, T., Tibshirani, R., & Friedman, J. H. (2011). *The Elements of Statistical Learning*. New York: Springer.

Ivancheva, L. (2008). Scientometrics today: A methodological overview. In *Fourth International Congerence on Webometrics, Informetrics, and Scientometrics & Ninth COLLNET Meeting*.

Jaccard, P. (1901). Étude comparative de la distribution florale dans une portion des Alpes et des Jura. *Bulletin de la Société Vaudoise des Sciences Naturelles*, *37*, 547–579.

Kitchenham, B. (2004). *Procedures for performing systematic reviews*. Technical Report Keele University and NICTA.

Kriegel, H.-P., Borgwardt, K., Kröger, P., Pryakhin, A., Schubert, M., & Zimek, A. (2007). Future trends in data mining. *Data Mining and Knowledge Discovery*, *15*, 87–97.

Kumar, D., & Bhardwaj, D. (2011). Rise of data mining: Current and future application areas. *IJCSI International Journal of Computer Science Issues*, *8*, 256–260.

Lafon, S., & Lee, A. B. (2006). Diffusion maps and coarse-graining: a unified framework for dimensionality reduction, graph partitioning, and data set parameterization. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, *28*, 1393–1403.

Leydesdorff, L. (2004). Clusters and maps of science journals based on bi-connected graphs in the journal citation reports. *Journal of Documentation*, *60*, 371–427.

Leydesdorff, L., Carley, S., & Rafols, I. (2013). Global maps of science based on the new web-of-science categories. *Scientometrics*, *94*, 589–593.

Leydesdorff, L., & Rafols, I. (2009). A global map of science based on the ISI subject categories. *Journal of the American Society for Information Science and Technology*, *60*, 348–362.

Matwin, S., Kouznetsov, A., Inkpen, D., Frunza, O., & O'Blenis, P. (2010). A new algorithm for reducing the workload of experts in performing systematic reviews. *Journal of the American Medical Informatics Association : JAMIA*, *17*, 446–453.

Nadler, B., Lafon, S., Coifman, R., & Kevrekidis, I. G. (2008). Diffusion maps – a probabilistic interpretation for spectral embedding and clustering algorithms. In T. J. Barth, M. Griebel, D. E. Keyes, R. M. Nieminen, D. Roose, T. Schlick, A. N. Gorban, B. Kégl, D. C. Wunsch, & A. Y. Zinovyev (Eds.), *Principal Manifolds for Data Visualization and Dimension Reduction* (pp. 238–260). Berlin Heidelberg: Springer volume 58 of *Lecture Notes in Computational Science and Engineering*.

Rafols, I., Porter, A. L., & Leydesdorff, L. (2010). Science overlay maps: A new tool for research policy and library management. *Journal of the American Society for Information Science and Technology*, *61*, 1871.

Rousseeuw, P. J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, *20*, 53–65.

Seglen, P. O. (1997). Why the impact factor of journals should not be used for evaluating research. *BMJ*, *314*, 498–502.

Small, H. (1973). Co-citation in the scientific literature: A new measure of the relationship between two documents. *Journal of the American Society for Information Science*, *24*, 265–269.

Szczuka, M., Janusz, A., & Herba, K. (2012). Semantic clustering of scientific articles with use of DBpedia knowledge base. In R. Bembenik, L. Skonieczny, H. RybiÅski, & M. Niezgodka (Eds.), *Intelligent Tools for Building a Scientific Information Platform* (pp. 61–76). Springer volume 390 of *Studies in Computational Intelligence*.

Teregowda, P. B., Councill, I. G., Fernández, R. J. P., Khabsa, M., Zheng, S., & Giles, C. L. (2010). Seersuite: developing a scalable and reliable application framework for building digital libraries by crawling the web. In *Proceedings of the 2010 USENIX conference on Web application development* WebApps'10. USENIX Association.

Tseng, Y.-H., & Tsay, M.-Y. (2013). Journal clustering of library and information science for subfield delineation using the bibliometric analysis toolkit: CATAR. *Scientometrics*, *95*, 503–528.

Venkatadri, M., & Reddy, L. C. (2011). A review on data mining from past to the future. *International Journal of Computer Applications*, *15*, 19–22.

Waltman, L., van Eck, N. J., & Noyons, E. C. (2010). A unified approach to mapping and clustering of bibliometric networks. *Journal of Informetrics*, *4*, 629–635.

Wu, X., Kumar, V., Quinlan, J. R., Ghosh, J., Yang, Q., Motoda, H., McLachlan, G., Ng, A., Liu, B., Yu, P., Zhou, Z.-H., Steinbach, M., Hand, D., & Steinberg, D. (2008). Top 10 algorithms in data mining. *Knowledge and Information Systems*, *14*, 1–37.

PVI

GEAR CLASSIFICATION AND FAULT DETECTION USING A DIFFUSION MAP FRAMEWORK

by

Tuomo Sipola, Tapani Ristaniemi and Amir Averbuch 2013

Reports of the Department of Mathematical Information Technology Series B.
Scientific Computing, No. B 6/2013, University of Jyväskylä, Jyväskylä, Finland

Reports of the Department of Mathematical Information Technology
Series B. Scientific Computing
No. B 6/2013

Gear Classification and Fault Detection Using a Diffusion Map Framework

Tuomo Sipola Tapani Ristaniemi
Amir Averbuch

University of Jyväskylä
Department of Mathematical Information Technology
P.O. Box 35 (Agora)
FI–40014 University of Jyväskylä
FINLAND
fax +358 14 260 2771
http://www.mit.jyu.fi/

ISBN 978-951-39-5464-2
ISSN 1456-436X

Gear Classification and Fault Detection Using a Diffusion Map Framework*

Tuomo Sipola† Tapani Ristaniemi‡ Amir Averbuch§

Abstract

A system health monitoring scheme using diffusion map is proposed. Diffusion map reduces the dimensionality of measurement data. This facilitates the comparison of newly arriving measurements to the known training data. The method is trained and tested with real gear monitoring data. The results show that data recordings can be classified as working or broken using dimensionality reduction.

1 Introduction

Modern industry monitoring systems produce high-dimensional data that are difficult to analyze as a whole without dimensionality reduction. The goal of the study is to estimate whether the proposed dimensionality reduction scheme effectively distinguishes working gears from broken ones. System health management has multiple sensors that measure vibration, temperature and oil properties. The early detection of anomalous gear behavior using this sensor data reduces the risk of severe damage. Sensor data are then used to monitor the health of the system, to detect anomalies and to predict problems [3, pp. 15–16].

Anomaly detection methods try to find deviant or atypical measurements from a large datamass [3]. In this study known anomalies are in the training so that they can be contrasted with the normal behavior. An ideal indicator would tell with certainty that a machine works or is going to fail. However, in reality the non-working state is ambiguous and it can be difficult to classify.

Spectral dimensionality reduction methods include principal component analysis (PCA), kernel PCA, multi-dimensional scaling (MDS), Laplacian eigenmaps,

*This research was partially supported by the Foundation of Nokia Corporation and the Finnish Foundation for Technology Promotion.

†Department of Mathematical Information Technology, University of Jyväskylä, PO Box 35 (Agora), FI-40014 University of Jyväskylä, Finland, tuomo.sipola@jyu.fi

‡Department of Mathematical Information Technology, University of Jyväskylä, PO Box 35 (Agora), FI-40014 University of Jyväskylä, Finland, tapani.ristaniemi@jyu.fi

§School of Computer Science, Tel Aviv University, Israel, amir@math.tau.ac.il

isomap and locally linear embedding (LLE). These methods facilitate the analysis of high-dimensional data by mapping the high-dimensional coordinates to a lower dimension. The spectral approach also leads to the concept of spectral clustering [2, 19]. Spectral methods have been used to analyze system operational states [15], motor fault detection [14] and anomaly detection for spacecraft [7].

This study uses diffusion map, which is another spectral dimensionality reduction method. Its mathematical foundation is random walk on Markov transition matrix of the graph of the data [4]. Diffusion map can be classified as a nonlinear distance-preserving dimensionality reduction method that preserves global properties [18]. Furthermore, the Nyström method is used to extend new points, although newer methods such as geometric harmonics exist [6, 5]. A similar study using diffusion map has been made concerning machine condition monitoring [8]. This study presents a way to detect faults in gears by devicing an index to describe how close to the faulty state a gear is. Besides gear fault detection, this method can also be used with other collections of high-dimensional time series data.

2 Method

This method trains a diffusion map that describes the good and bad state of the gears. It then extends newly arriving test measurements to the model and classifies the gear as good or bad. Most of the preprocessing is domain specific, but the dimensionality reduction and classification, that are more universally applicable, are presented here. Figure 1 introduces the overall data processing architecture. The equations are in matrix form. The details behind them are discussed elsewhere [12, 6, 1].

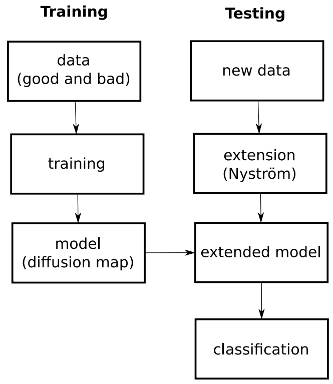

Figure 1: Data processing block diagram.

2.1 Training dimensionality reduction

The underlying assumption in manifold learning methods is that the data is situated on a lower-dimension manifold in the high-dimension measurement data [3, p. 37]. We try to create a function that maps the behavior of high-dimensional points to lower dimensions. Then new measurement points are mapped from high dimensions to this low-dimensional presentation.

Let $x_i \in \mathbb{R}^n, i = 1 \ldots N$ be a measurement in n-dimensional space. The kernel matrix W includes the pairwise distances of these points. The used kernel is the Gaussian kernel using Euclidian distance measure. This is the most computationally intensive step because each point is compared to other points:

$$W_{ij} = \exp\left(-\frac{||x_i - x_j||^2}{\epsilon}\right). \tag{1}$$

Determining ϵ is a problem in itself. The chosen estimation is the median of the distances between the points, $\epsilon = \mathrm{median}\{||x_i - x_j||\}_{x_i, x_j \in \mathbb{R}^n}$ [16]. Depending on the problem, changing this parameter might give more meaningful results.

Matrix $D_{ii} = \sum_{j=1}^{N} W_{ij}$ has the degree of each point on its diagonal. The degree of a point is the sum of weights that connect to other points. This is equal to the sum of kernel matrix rows.

The rows are normalized by these sums. The result can also be understood as transition probabilities between points. These probabilities are collected in matrix P,

$$P = D^{-1}W. \tag{2}$$

However, future calculations on P become easier if a similarity transformation symmetricizes the matrix:

$$\tilde{P} = D^{\frac{1}{2}} P D^{-\frac{1}{2}}. \tag{3}$$

These last two steps can be combined. Substituting P with $D^{-1}W$ yields:

$$\tilde{P} = D^{-\frac{1}{2}} W D^{-\frac{1}{2}}. \tag{4}$$

Such normal matrix is decomposed as:

$$\tilde{P} = U \Lambda U^*. \tag{5}$$

This decomposition is done using singular value decomposition (SVD). The columns of matrix U contain eigenvectors u_k of matrix $\tilde{P}$. Likewise, the diagonal of Λ contains its corresponding eigenvalues. However, the real interest is in the eigenvectors of the transition matrix P. The eigenvalues of P are the same, but the eigenvectors are obtained from V:

$$V = D^{-\frac{1}{2}} U. \tag{6}$$

Recall that the eigenvalues λ are in the diagonal of Λ. The eigenvector v are columns of V. An original data point x_i has a corresponding value on the ith row of the eigenvector. For example, $v_2(x_{236})$ would signify the second eigenvector and its 236th row, corresponding to the 236th sample x_{236} of the original dataset.

The diffusion map itself is a function in the form $\Psi : \mathbb{R}^n \rightarrow \mathbb{R}^d$, when $d \ll n$. We multiply the eigenvectors and eigenvalues to get the diffusion coordinates of the training points:

$$\Psi = V\Lambda. \tag{7}$$

The first eigenvector is constant, so only the following eigenvectors and eigenvalues are used. This way we get the following function that maps the original data points to a lower-dimensional space:

$$\Psi_d : x_i \rightarrow \begin{pmatrix} \lambda_2 v_2(x_i) \\ \lambda_3 v_3(x_i) \\ \lambda_4 v_4(x_i) \\ \vdots \\ \lambda_{d+1} v_{d+1}(x_i) \end{pmatrix}. \tag{8}$$

It has been shown that the diffusion distance in the original space equals to the Euclidean distance in the diffusion space [4]. Thus, the distance measurements in the diffusion space are actually meaningful and can be used in further analysis in this lower-dimensional space.

Later analysis uses only the first few diffusion coordinates. Fast decay of eigenvalues leaves most of the diffusion coordinates rather small compared to the first few. The overall reconstruction of P does not differ much from a reconstruction that uses only the first coordinates. These coordinates capture most of the differences between the data points [4, 11].

2.2 Extension of new measurements

New measurements that are not part of training are extended to the model with Nyström method [6, 1]. The features selected during training are the only ones needed. These new measurements are normalized using the same normalization as during training.

Let a new data point be $y_j \in \mathbb{R}^n$. Then the distance between the new points and each training point are collected in a matrix $\bar{W}$. This function uses the same ϵ as the one in training phase:

$$\bar{W}_{ij} = \exp\left(-\frac{||x_i - y_j||^2}{\epsilon}\right). \tag{9}$$

Diagonal matrix $\bar{D}_{ii} = \sum_{i=1}^{N} \bar{W}_{ij}$ contains the column sums of $\bar{W}$. Now we can create the transition probability matrix B:

$$B = \bar{W} * \bar{D}^{-1}. \tag{10}$$

The following matrix multiplication produces new eigenvectors for the new point. The eigenvectors V and eigenvalues Λ are the same as in training:

$$\bar{V} = B^T V \Lambda^{-1}. \tag{11}$$

These new eigenvectors now extend the new point to the diffusion coordinates:

$$\bar{\Psi} = \bar{V}\Lambda. \tag{12}$$

The last two steps can be combined:

$$\bar{\Psi} = B^T V. \tag{13}$$

Matrix $\bar{\Psi}$ now contains the extended eigenvectors in its columns for the new points y_j.

2.3 Classification of new measurements

Low-dimensional presentation of the data facilitates clustering. The clustering approach here is spectral clustering and it reveals the normal and anomalous areas [19, 9]. Any other clustering, for example k-means, can be used if they provide better results [13, 10, 17]. The used algorithm simply tests whether the sample is to the left or to the right of 0 on the dimension corresponding to the 2nd eigenvector. This provides a classifier that discriminates two states: working or broken.

2.4 Warning levels

For more warning levels, different thresholds can be applied. There are three warning levels: note, warning and damage. These describe the severity of the problem in the gear.

Note means that there is an unusual measurement in the data, but the gear is still in operational state. The sample is not inside the good cluster but is still closer to it than to the bad.

$$\theta_{note} = \min\{\Psi_{1,good}\} \tag{14}$$

Warning level is at $\theta_{warning} = 0$. It describes the border between good and bad clusters. The sample is closer to the bad cluster. This can be seen as a predictive sign that the gear has problems. If the bad cluster goes beyond 0, the middle point between the two clusters can be used.

Damage level is at $\theta_{damage} = \max\{\Psi_{1,bad}\}$. This means that the sample is within the bad cluster.

3 Results

This study uses a dataset consisting of gear monitoring recordings of multiple features. It consists of recordings of 18 good and 20 bad machines labeled by domain specialists. The gears come from different locations where the operational environment varies. However, each gear is of the same type and includes same features. Two of the gears are discarded because they contain empty data due to instrument failures. The dataset is divided to training and testing sets. The training set includes five good and five bad gears. The testing set includes the rest of the gears.

3.1 Preprosessing

The data are sampled at an approximate frequency of one sample per 30 minutes. The recordings last for months. Because there were times when no data were available, linear interpolation is used. This data formed the samples × features matrix.

Instrument failures give unrealistic or missing measurements. Because it is difficult to compare such measurements to ones that do not have unrealistic values, measurements containing missing values are discarded. However, this process might lose some usable information.

3.1.1 RPM filtering

Samples whose rotations per minute (RPM) value is too small are filtered out, because only higher values represent the actual working state of a gear. Lower values are associated with idle state, and those measurements are not interesting when monitoring actual working gears. The RPM values are clustered into two clusters using k-means clustering. The threshold value,

$$threshold_{RPM} = \max\{\min\{RPM_{\mathit{cluster}\ 1}\}, \min\{RPM_{\mathit{cluster}\ 2}\}\}, \tag{15}$$

is calculated and all the samples whose RPM value is below this threshold are removed.

3.1.2 Data scaling

All the data are normalized with logarithm. Other normalizations, like dividing by maximum or dividing by norm, do not give as good separation for this dataset.

3.1.3 Feature selection

There are 136 features. The initial feature selection reduced their number to 20. Some features separate more clearly the two groups from each other. A preliminary feature selection in the original feature space gives these features. One feature is left out at a time. The average Mahalanobis distance between the good and bad machines shows how much that feature describes the difference. The features with smallest averaged Mahalanobis distances are most useful. Small distance reveals that leaving the feature out affects negatively the separation of good and bad. Thus, using the feature separates the groups well in the feature space.

4 Classification results

Five good and five bad gears were used in training. The data has 136 features, 20 of which are used after preliminary feature selection. All the gears, including training gears, were then tested as new incoming data. Table 1 shows that each of the broken test gears had alerts. Table 2 shows that no working gear had warnings, although some of them had notes.

gear	alerts
OO03	2.5703%
*OO06	14.4068%
OO08	42.6573%
OO09	6.6667%
AH01	16.835%
AH02	16.7431%
AH06	2.8777%
*AH11	14.916%
AH18	7.0941%
FE09	5.3495%
FE10	4.4068%
*FE12	16.0083%
CA03	22.7599%
CA04	7.7089%
QU23	6.0469%
*QU32	21.6535%
ET104	0.32841%
*ET403	3.3597%
PH05	9.3694%

Table 1: Broken gear units (alert threshold 0). Asterisk marks training gears.

gear	alerts
*AH10	0%
AH16	0%
AH18	0%
FE01	0%
*FE02	0%
FE03	0%
CA01	0%
LS04	0%
*LS05	0%
MB08	0%
QU32	0%
*PH01	0%
PH03	0%
PH05	0%
PH08	0%
*PH09	0%
PH13	0%

Table 2: Working gear units (alert threshold 0). Asterisk marks training gears.

The following figures illustrate the behavior of broken gears. Normal state does not produce figures of interest because there are no alerts. Figure 2 shows how the newly incoming data is situated in low-dimensional space. Figure 3 shows the alert index, while Figure 4 indicates the accumulating number of alerts. The alerts themselves are in Figure 5. Figures 6, 7, 8, 9 show the same measurements for another gear. It breaks down more slowly but the high number of notes can be seen.

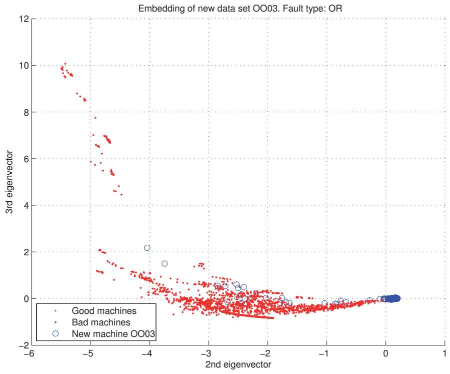

Figure 2: Samples of a broken gear in low-dimensional space.

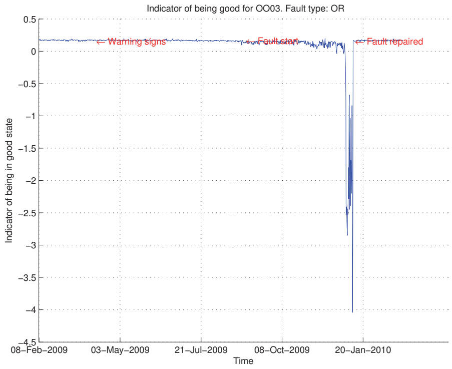

Figure 3: Alert level index of a broken gear. Above 0 is considered normal working state.

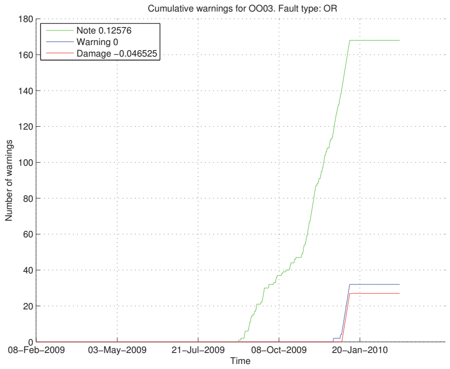

Figure 4: Number of alerts.

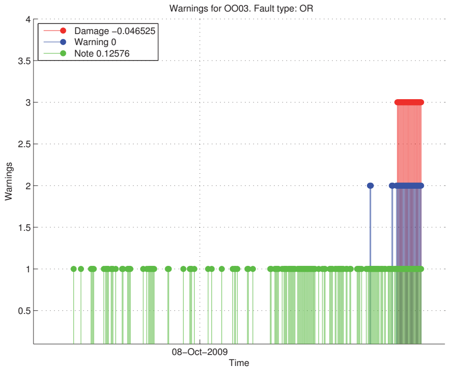

Figure 5: Alerts given by the method.

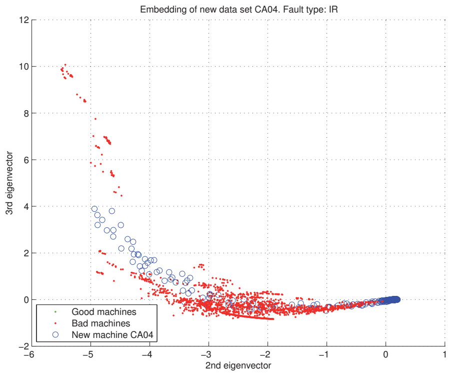

Figure 6: Samples of a broken gear in low-dimensional space.

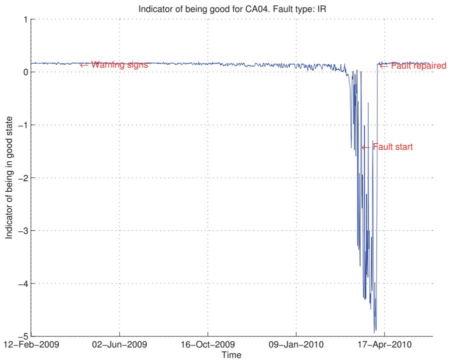

Figure 7: Alert level index of a broken gear. Above 0 is considered normal working state.

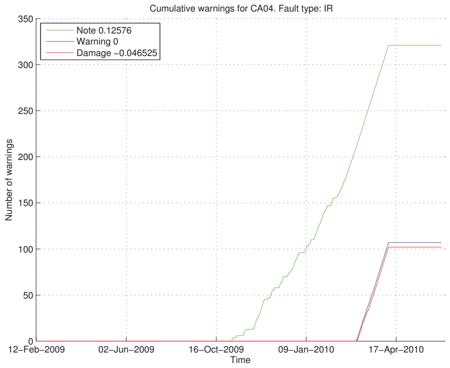

Figure 8: Number of alerts.

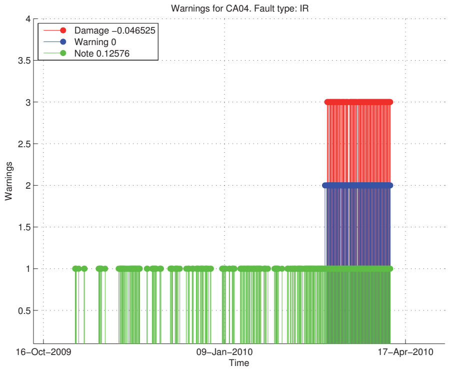

Figure 9: Alerts given by the method.

5 Discussion

The goal of this study is to estimate the usefulness of dimensionality reduction methods in gear fault detection. This goal is met since almost all the gears are classified correctly according to their labels. This proves that the training is successful and separates the good gears from the bad. More importantly, measurements from totally different gears can be extended into the model.

The misclassification of good machine FE01 as bad is probably because of the data interpolation. Further domain analysis revealed that there actually had been a small problem with the gear, and thus raises the question whether it is labeled correctly. The misclassification of bad machine ET104 as good can be explained. Firstly, there are no training gears from this location. ET104 is too close to the good gears in diffusion space. Secondly, domain analysis reveals that this gear has only a small problem. Better training data and more detailed labeling could prevent this kind of misclassification. Vastly different operating environment and behavior of gears in ET1 might also cause this misclassification.

The problems of spectral methods in general need some addressing. The proposed method works because, after slight filtering, the good and bad gears are separable in the lower dimensions. However, the high computational cost could be a problem in a more real-time system. The classification of a gear time series itself is an ambiguous concept. However, this study shows that gears in normal condition and gears that are going to break down behave differently and can be separated from each other.

References

[1] Serge Belongie, Charless Fowlkes, Fan Chung, and Jitendra Malik. Spectral partitioning with indefinite kernels using the Nyström extension. In Anders Heyden, Gunnar Sparr, Mads Nielsen, and Peter Johansen, editors, *Computer Vision – ECCV 2002*, volume 2352 of *Lecture Notes in Computer Science*, pages 51–57. Springer Berlin / Heidelberg, 2002.

[2] Yoshua Bengio, Olivier Delalleau, Nicolas Le Roux, Jean-François Paiement, Pascal Vincent, and Marie Ouimet. *Feature Extraction*, chapter Spectral Dimensionality Reduction, pages 519–550. Studies in Fuzziness and Soft Computing. Springer Berlin, Heidelberg, 2006.

[3] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM Comput. Surv.*, 41:15:1–15:58, July 2009.

[4] Ronald R. Coifman and Stéphane Lafon. Diffusion maps. *Applied and Computational Harmonic Analysis*, 21(1):5–30, 2006.

[5] Ronald R. Coifman and Stphane Lafon. Geometric harmonics: A novel tool for multiscale out-of-sample extension of empirical functions. *Applied and Computational Harmonic Analysis*, 21(1):31 – 52, 2006. Diffusion Maps and Wavelets.

[6] C. Fowlkes, S. Belongie, F. Chung, and J. Malik. Spectral grouping using the Nystrom method. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(2):214 –225, 2004.

[7] Ryohei Fujimaki, Takehisa Yairi, and Kazuo Machida. An approach to spacecraft anomaly detection problem using kernel feature space. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, KDD '05, pages 401–410, New York, NY, USA, 2005. ACM.

[8] Yixiang Huang, Xuan F Zha, Jay Lee, and Chengliang Liu. Discriminant diffusion maps analysis: A robust manifold learner for dimensionality reduction and its applications in machine condition monitoring and fault diagnosis. *Mechanical Systems and Signal Processing*, 34(1):277–297, 2013.

[9] Ravi Kannan, Santosh Vempala, and Adrian Vetta. On clusterings: Good, bad and spectral. *J. ACM*, 51:497–515, May 2004.

[10] Marina Meila and Jianbo Shi. Learning segmentation by random walks. In *NIPS*, pages 873–879, 2000.

[11] Boaz Nadler, Stephane Lafon, Ronald Coifman, and Ioannis Kevrekidis. Diffusion maps, spectral clustering and eigenfunctions of Fokker-Planck operators. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems 18*, pages 955–962. MIT Press, Cambridge, MA, 2006.

[12] Boaz Nadler, Stephane Lafon, Ronald Coifman, and Ioannis G. Kevrekidis. Diffusion maps – a probabilistic interpretation for spectral embedding and clustering algorithms. In Timothy J. Barth, Michael Griebel, David E. Keyes, Risto M. Nieminen, Dirk Roose, Tamar Schlick, Alexander N. Gorban, Balázs Kégl, Donald C. Wunsch, and Andrei Y. Zinovyev, editors, *Principal Manifolds for Data Visualization and Dimension Reduction*, volume 58 of *Lecture Notes in Computational Science and Engineering*, pages 238–260. Springer Berlin Heidelberg, 2008.

[13] Andrew Y. Ng, Michael I. Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems 14*, pages 849–856. MIT Press, 2001.

[14] Lucas Parra, Gustavo Deco, and Stefan Miesbach. Statistical independence and novelty detection with information preserving nonlinear maps. *Neural Computation*, 8(2):260–269, 1996.

[15] Markus Pylvänen, Sami Äyrämö, and Tommi Kärkkäinen. Visualizing time series state changes with prototype based clustering. In Mikko Kolehmainen,

Pekka Toivanen, and Bartlomiej Beliczynski, editors, *Adaptive and Natural Computing Algorithms*, volume 5495 of *Lecture Notes in Computer Science*, pages 619–628. Springer Berlin / Heidelberg, 2009.

[16] A. Schclar, A. Averbuch, N. Rabin, V. Zheludev, and K. Hochman. A diffusion framework for detection of moving vehicles. *Digital Signal Processing*, 20(1):111 – 122, 2010.

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

[18] L. J. P. van der Maaten, E. O. Postma, and H. J. van Den Herik. Dimensionality reduction: A comparative review. *Journal of Machine Learning Research*, 10:1–41, 2009.

[19] Ulrike von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17:395–416, 2007.

PVII

USING AFFINITY PERTURBATIONS TO DETECT WEB TRAFFIC ANOMALIES

by

Yaniv Shmueli, Tuomo Sipola, Gil Shabat and Amir Averbuch 2013

Proceedings of the 10th International Conference on Sampling Theory and Applications (SampTA 2013), pp. 444–447, Bremen, Germany

Using Affinity Perturbations to Detect Web Traffic Anomalies

Yaniv Shmueli
School of
Computer Science
Tel Aviv University
yaniv.shmueli@cs.tau.ac.il

Tuomo Sipola
Department of
Mathematical Information Technology
University of Jyväskylä
tuomo.sipola@jyu.fi

Gil Shabat
School of
Electrical Engineering
Tel Aviv University
gil@eng.tau.ac.il

Amir Averbuch
School of
Computer Science
Tel Aviv University
amir@math.tau.ac.il

***Abstract*—The initial training phase of machine learning algorithms is usually computationally expensive as it involves the processing of huge matrices. Evolving datasets are challenging from this point of view because changing behavior requires updating the training. We propose a method for updating the training profile efficiently and a sliding window algorithm for online processing of the data in smaller fractions. This assumes the data is modeled by a kernel method that includes spectral decomposition. We demonstrate the algorithm with a web server request log where an actual intrusion attack is known to happen. Updating the kernel dynamically using a sliding window technique, prevents the problem of single initial training and can process evolving datasets more efficiently.**

***Index Terms*—perturbation theory, eigenvalue problem, diffusion maps, dimensionality reduction, anomaly detection, web traffic**

I. Introduction

Evolving data that requires frequent updates to the training is a challenging target when extracting constructive information. The computational complexity of the training phase increases with such datasets because an earlier profile may not accurately represent the behavior of current data. Therefore, the extracted profile has to be updated frequently. A straightforward approach for updating the training profile is to repeat the entire computational process that generated the original profile. This paper summarizes a method for efficiently updating the evolving profile.

A common practice in kernel methods is to extract features from a high dimensional dataset, and to form a similarity graph between the features in the dataset. In this research we apply the Diffusion Maps (DM) methodology [1] to a web traffic log. DM finds the embedded coordinates for a low-dimensional representation of the data. This embedding is accomplished by eigenvectors computation of the graph affinity matrix. Changes in the affinity matrix will result in changes in the eigenvectors, and thus will force us to compute them frequently. We use a solution based on the Recursive Power Iteration algorithm combined with the first-order approximation of the perturbed eigenvectors and eigenvalues (eigenpairs) [2]. This enables us to update the dataset profile by considering only the changes in the original dataset, which also requires less computational effort.

Since network data is dynamic and evolving, the embedded low-dimensional space has to be updated as the training data does not adequately represent the incoming data that did not participate in the initial training phase. Even if most of the network log lines in our interest window are unchanged, we will still need to perform the entire computation since we cannot determine the effect of such a change on the embedded space. Therefore, the goal of the paper is to provide an efficient method for updating the embedding coordinates without the need to re-compute the entire SVD again and again over time. We treat the log line feature changes as perturbations from the original network log profile of the feature affinity matrix. By applying a sliding window technique to the incoming network data, we are able to process the data online, and keep embedding it in the low-dimensional space. We test this method on real web traffic data and compare our results to the true classification.

II. Related Work

Traditional computational methods such as the power iteration, inverse iteration and Lanczos methods operate in the same way and compute the eigenpairs of each update of the perturbed matrix. Here, the computation is performed with a random guess as the initial input without taking the unperturbed matrix and its eigenpairs into consideration.

Incremental versions of low-dimensional embedding algorithms have been tailored specifically to fit local linear embeddings (LLE) [3] and ISOMAP [4]. These algorithms use modified manifold learning methods to process the data iteratively. When a new data point arrives, these algorithms add it to the embedding and then efficiently update all the existing data points in the low-dimensional space.

Network security has been one focus among the machine learning community. Kruegel and Vigna studied the parameters of HTTP queries using a training step with unlabeled data with various methods. Their character distribution analysis uses similar feature extraction as our current study [5]. Hubballi et al. described an n-gram approach to detect intrusions from network packets [6]. Ringberg et al. studied IP packets using principal component analysis-based dimensionality reduction [7]. Callegari et al. analyzed similar low-level packet data [8].

Diffusion maps have been also used for network security problems. David studied the use of diffusion map methodology for detecting intrusions in network traffic [9]. Network server logs have also been studied with diffusion maps with an offline approach using n-gram features and spectral clustering [10]. In these works, data analysis was performed in a batch fashion, processing all recordings as a single, offline dataset.

III. Finding a Low-Dimensional Embedded Space

A. Diffusion Maps

Finding a low-dimensional embedded space is an important step in understanding high-dimensional data more profoundly. To better understand the proposed algorithm, we review the DM methodology [1] that performs non-linear dimensionality reduction. Given our web log feature matrix X, we define a weighted graph over the log lines, where the weight between lines i and j is given by the kernel $k(i,j) \triangleq e^{-\frac{\|x_i - x_j\|}{\varepsilon}}$. The degree of a log line (vertex) i in this graph is $d(i) \triangleq \sum_j k(i,j)$. Normalizing the kernel with this degree produces an $n \times n$ row stochastic transition matrix whose cells are $[P]_{ij} = p(i,j) = k(i,j)/d(i)$ for log lines i and j. This defines a Markov process over the network log features.

The dimensionality reduction achieved by this diffusion process is a result of the spectral analysis of the kernel. Thus, it is preferable to work with a symmetric conjugate to P that we denote by A and its cells are denoted by

$$[A]_{ij} = a(i,j) = \frac{k(i,j)}{\sqrt{d(i)}\sqrt{d(j)}} = \sqrt{d(i)}p(i,j)\frac{1}{\sqrt{d(j)}}. \quad (1)$$

The eigenvalues $1 = \lambda_1 \geq \lambda_2 \geq \ldots$ of P and their corresponding eigenvectors v_k $(k = 1, 2, \ldots)$ are derived from the eigenvectors u_k of A. The v_k are used to obtain the desired dimensionality reduction by mapping each i onto the data point $\Psi(i) = (\lambda_2 v_2(i), \lambda_3 v_3(i), ..., \lambda_\delta v_\delta(i))$ for a sufficiently small δ, which depends on the decay of the spectrum of A [1].

In matrix notation, the operator A is defined as $A = D^{-\frac{1}{2}} K D^{-\frac{1}{2}} = D^{\frac{1}{2}} P D^{-\frac{1}{2}}$ where D is the diagonal matrix containing the $d(i)$ value in cell D_{ii}. To retrieve the eigenvectors in columns V of P from the eigenvactors of A, we use the transformation $V = D^{-\frac{1}{2}} U$ where U is the eigenvector column matrix of A. The eigenvectors V obtained for P are scaled by dividing each one by the first value of the first eigenvector.

B. Updating the Embedding

Once we have the DM embedding of the initial matrix A, we need to keep updating the embedding for the next arriving samples. By replacing the oldest samples with the newly arriving ones, we can model such a change as a perturbation matrix $\tilde{A}$ of the matrix A. We assume that the perturbations are sufficiently small, that is, $\|\tilde{A} - A\| < \varepsilon$ for some small ε. Note that $\tilde{A}$ is symmetric since it represents the operator defined in 1. We wish to update the eigenpairs of $\tilde{A}$ based on A and its eigenpairs. We now present the problem in mathematical terms.

Given a symmetric $n \times n$ matrix A where its k dominant eigenvalues are $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_k$ and its eigenvectors are $\phi_1, \phi_2, ..., \phi_k$, respectively, and a perturbed matrix $\tilde{A}$ such that $\|\tilde{A} - A\| < \varepsilon$, find the perturbed eigenvalues $\tilde{\lambda}_1 \geq \tilde{\lambda}_2 \geq ... \geq \tilde{\lambda}_k$ and its eigenvectors $\tilde{\phi}_1, \tilde{\phi}_2, ..., \tilde{\phi}_k$ of $\tilde{A}$ in the most efficient way [2].

In the next section, we explain how such processing can be done using the recursive power iteration (RPI) algorithm.

IV. The Recursive Power Iteration (RPI) Algorithm

A. Eigenpair First-Order Approximation

To efficiently update each eigenpair of the perturbed matrix $\tilde{A}$, we first compute the first-order approximation of each eigenpair. Later, it will be used in our algorithm as the initial guess for the RPI algorithm.

Given an eigenpair (ϕ_i, λ_i) of a symmetric matrix A where $A\phi_i = \lambda_i \phi_i$, we compute the first-order approximation of the eigenpair of the perturbed matrix $\tilde{A} = A + \Delta A$. We assume that the change ΔA is sufficiently small, which results in a small perturbation in ϕ_i and λ_i. We look for $\Delta\lambda_i$ and $\Delta\phi_i$ that satisfy the equation

$$(A + \Delta A)(\phi_i + \Delta\phi_i) = (\lambda_i + \Delta\lambda_i)(\phi_i + \Delta\phi_i). \quad (2)$$

Using the methods described by Shmueli et al. [2], we can obtain the following first-order approximations for the eigenvalues and eigenvectors of $\tilde{A}$

$$\tilde{\lambda}_i = \lambda_i + \phi_i^T [\Delta A] \phi_i \quad (3)$$

and

$$\tilde{\phi}_i = \phi_i + \sum_{j \neq i} \frac{\phi_j^T [\Delta A] \phi_i}{\lambda_i - \lambda_j} \phi_j. \quad (4)$$

B. The Recursive Power Iteration Method

The power iteration method has proved to be effective when calculating the principal eigenvector of a matrix [11]. However, this method cannot find the other eigenvectors of the matrix. In general, an initial guess of the eigenvector is also important to guarantee fast convergence of the algorithm. In Algorithm IV.1, which we call recursive power iteration (RPI), the first-order approximations of the perturbed eigenvectors of $\tilde{A}$ are the initial guess for each power iteration. Once the eigenvector $\tilde{\phi}_i$ is obtained in step i, we transform $\tilde{A}$ into a matrix that has $\tilde{\phi}_{i+1}$ as its principal eigenvector. We iterate this step until we recover the k dominant eigenvectors of $\tilde{A}$.

The correctness of the RPI algorithm and its complexity analysis are given in the original article [2].

The justification for this approach is that the first-order approximation of the perturbed eigenvector is inexpensive, and each RPI step guarantees that this approximation converges to the actual eigenvector of $\tilde{A}$. The first-order approximation should be close to the actual solution we seek and therefore requires fewer iteration steps to converge.

Algorithm IV.1: Recursive Power Iteration Algorithm

```
Input: Perturbed symmetric matrix $\tilde{A}_{n\times n}$, number of
       eigenvectors to calculate $k$, initial eigenvectors
       guesses $\{v_i\}_{i=1}^{k}$, admissible error $err$
Output: Approximated eigenvectors $\left\{\tilde{\phi}_i\right\}_{i=1}^{k}$,
        approximated eigenvalues $\left\{\tilde{\lambda}_i\right\}_{i=1}^{k}$
for $i = 1 \rightarrow k$ do
    $\phi \leftarrow v_i$
    repeat
        $\phi_{next} \leftarrow \frac{\tilde{A}\phi}{\|\tilde{A}\phi\|}$
        $err_{\phi} \leftarrow \|\phi - \phi_{next}\|$
        $\phi \leftarrow \phi_{next}$
    until $err_{\phi} \leq err$
    $\tilde{\phi}_i \leftarrow \phi$
    $\tilde{\lambda}_i \leftarrow \frac{\tilde{\phi}_i^T \tilde{A} \tilde{\phi}_i}{\tilde{\phi}_i^T \tilde{\phi}_i}$
    $\tilde{A} \leftarrow \tilde{A} - \tilde{\phi}_i \tilde{\lambda}_i \tilde{\phi}_i^T$
end for
```

Algorithm V.1: Sliding Window Diffusion Map with RPI

```
Input: Dataset $X$, window width $n$, embedded dimension $k$,
    admissible error $err$.
Output: Anomaly score for points in $X$.
    $\epsilon \leftarrow$ estimate kernel parameter for first window of size $n$.
    $[K]_{ij} \leftarrow \exp\left(-\frac{||x_i - x_j||^2}{\epsilon}\right)$, where $i, j = 1 \dots n$
    $D \leftarrow \mathrm{diag}(\sum_{i=1}^{n}[K]_{ij})$
    $A \leftarrow D^{-\frac{1}{2}} K D^{-\frac{1}{2}}$
    $U, \Lambda, U^T \leftarrow \mathrm{SVD}(A)$
    while new sample $x_t$ available, where $t > n$ do
        $l \leftarrow t \bmod n$
        Replace the row $l$ in $X$ with the new sample $x_t$.
        Update both row $l$ and column $l$ of the affinity matrix
        $K$.
        $D \leftarrow \mathrm{diag}(\sum_{i=1}^{n}[K]_{ij})$
        $\tilde{A} \leftarrow D^{-\frac{1}{2}} K D^{-\frac{1}{2}}$
        $U, \Lambda \leftarrow$ RPI with first-order approximation
        $(\tilde{A}, A, k, U, \Lambda, err)$
        $V \leftarrow D^{-\frac{1}{2}} U$
        $V \leftarrow \frac{V}{V_{1,1}}$
        $\Psi \leftarrow V\Lambda$
        Find anomalies in $\Psi$ and rate all samples in $X$.
        $A \leftarrow \tilde{A}$
    end while
    Return aggregated anomaly scores for each sample in $X$.
```

V. Sliding Window Diffusion Map

Using DM to embed high volumes of data can be computationally intensive. It is even more challenging when the data is generated online and needs to be processed continuously. Therefore, we try to process the incoming data with iterative methodology by using the sliding window model. A sliding window X takes into account the n latest measurements. In practice, it is an $n \times m$ matrix with features on the columns and samples on the rows. The samples are high-dimensional, so the dimensionality of the sliding window is reduced from m to d using DM. This $n \times d$ matrix X_r now contains the low-dimensional representation of the data. This reduction is done each time a new sample appears and the window moves. However, the consecutive update of the DM is a time-consuming process that requires singular value decomposition during each window.

When updating the window, we can replace the oldest measurement with a new one in the matrix X, therefore changing a single row in X. This means that one line and one column of the K matrix in the DM algorithm change. This change can be interpreted as a perturbation to the matrix K, and furthermore to the matrix A, which is defined using the K matrix. The RPI algorithm with first-order approximation solves the eigenvectors for perturbed matrices. This leads us to use the RPI algorithm instead of time-consuming SVD.

Algorithm V.1 outlines the sliding window DM method. First, it solves the eigenvectors for the initial window using SVD. Then the algorithm iteratively process the following windows until no new samples are available.

There are, some practical problems with this approach. First, the RPI algorithm might not be able to solve the eigenvectors for some low-rank matrices. It is possible to prevent this with standard SVD when a low-rank (or otherwise unsuitable) matrix is encountered. Second, the window size itself has to be decided. The changing scales of the data over time introduce a challenge to the sliding window algorithm. The initial window still determines the profile and scale for the beginning of the analysis. Big windows cover a larger representation of the data and thus include a more varied overview of the normal behavior. With smaller windows, the percentage of anomalies within the data might get too big, and detecting the normal state becomes more difficult. Small windows, however, require less computational time since they induce smaller matrices. Optimal window size would therefore be the smallest possible that contains a small enough percentage of anomalies within the data, enabling it to capture the normal samples correctly.

Detecting the anomalies in the low-dimensional representation can be done in various ways. A straightforward approach is to calculate distances between the embedded samples and find the ones that deviate too far from the center of the dataset. This and other spectral clustering methods give good results for datasets that contain clear separation [12], [10]. Similarly, k-means or any other clustering algorithm can find possible normal as well as anomalous behavior in the data. The density of points in the low-dimensional space tells how far they are from the more clustered areas. These methods calculate the distances to neighboring points [9]. All these methods usually need a threshold value for the anomalous region.

In each iteration, we evaluate the anomaly level of the samples within the window. Each sample gets a score if it is classified as an anomaly according to the selected anomaly detection method. The scores of each sample are added as the window moves. This cumulative anomaly score histogram may be used to determine the anomaly level of a point. Scoring is used because locally inside a window some samples might appear anomalous but globally, considering the whole dataset, they are not. Even if the sample looks like an anomaly in some windows, it still gets only a few scores globally.

VI. Experimental Results

For the experiment, we use a labeled proprietary dataset of queries to a web server, which is known to contain some network attacks. These web queries are in Apache combined log text file. To extract numerical features from this text file, only the changing parameter values are used. The frequencies of 2-grams in these parameters are calculated to a matrix. In this matrix, the rows represent the log lines, and the columns represent the different 2-grams we found. The entries in this matrix count how many times each specific 2-gram appeared in the parameters of a log line. See [10] for more information about this dataset and the feature extraction.

The web log we use has 4292 lines and contains 480 different 2-grams. Thus, the feature matrix has dimensions 4292 × 480. The experiment simulates the initial state when n samples, or log lines, have arrived. When a new line arrives, it is added to the current window, while the oldest sample is removed from the matrix. This is continued until no new samples are available. The algorithm tracks only the samples within the window so that the dynamically changing nature of the data can be followed. As the size of the window does not change, the eigenpair problem stays reasonably sized.

Anomaly detection with Euclidian distances finds the most deviating samples within a window. This leads to false alarms when using simple normalized anomaly metrics because inside a window a point might look anomalous. Its local abnormality might be evident, but it should not be classified as one since globally it is just a small deviation from the normal state. This fact promotes thresholding the non-normalized but centered low-dimensional representation $d_k = |\Psi_k - \mathrm{mean}(\Psi_k)|$ within one window using statistical threshold $\theta_k = c \cdot \mathrm{std}(d_k)$, where the parameter c has to be adjusted empirically, for each dimension k in the embedded space.

Figure 1 illustrates the scores each point gets as the sliding window moves. The number of times the data points are classified anomalous are plotted against time. The window width is set to $n = 1000$. This experiment uses only the second eigenpair, $k = 2$, Ψ_2 for the low-dimensional presentation. In our analysis, we use a value of $c = 10$ for the anomaly threshold calculation. These scores themselves indicate in how many windows each sample is considered anomalous: the data points that are considered attacks are clearly seen from 2500 to 3500. Notice that a sample might be considered anomalous in several windows, but in the global view it is not an anomaly. Therefore, we use another threshold, which is the horizontal red line in the figure. With this setup, we manage to reach an accuracy of 92.5% and a precision of 99.7% after tuning the parameters of the algorithm.

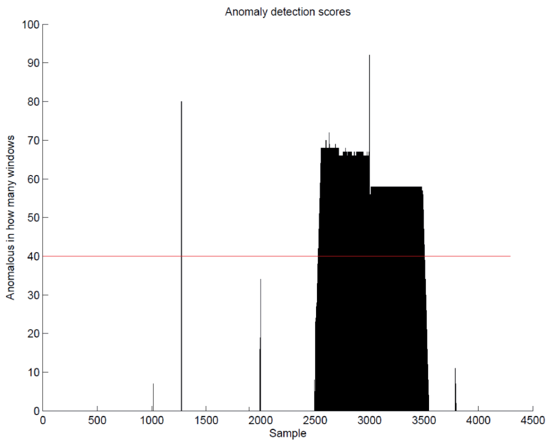

Fig. 1. The scores for each point with window size 1000 using the second eigenvector. The more times the data point is classified anomalous, the higher the score.

Acknowledgments

This research was supported by the Israel Science Foundation (Grant No. 1041/10) and by the Foundation of Nokia Corporation. The authors thank Antti Juvonen for assisting with the data analysis and Tapani Ristaniemi for guidance and support.

References

[1] R. R. Coifman and S. Lafon, "Diffusion maps," *Applied and Computational Harmonic Analysis*, vol. 21, no. 1, pp. 5–30, 2006.

[2] Y. Shmueli, G. Wolf, and A. Averbuch, "Updating kernel methods in spectral decomposition by affinity perturbations," *Linear Algebra and its Applications*, vol. 437, no. 6, pp. 1356–1365, 2012.

[3] O. Kouropteva, O. Okun, and M. Pietikäinen, "Incremental locally linear embedding algorithm," *Image Analysis*, pp. 145–159, 2005.

[4] M. Law and A. Jain, "Incremental nonlinear dimensionality reduction by manifold learning," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 28, no. 3, pp. 377–391, 2006.

[5] C. Kruegel and G. Vigna, "Anomaly detection of web-based attacks," in *Proceedings of the 10th ACM conference on Computer and communications security*. ACM, 2003, pp. 251–261.

[6] N. Hubballi, S. Biswas, and S. Nandi, "Layered higher order n-grams for hardening payload based anomaly intrusion detection," in *Availability, Reliability, and Security, 2010. ARES'10 International Conference on*. IEEE, 2010, pp. 321–326.

[7] H. Ringberg, A. Soule, J. Rexford, and C. Diot, "Sensitivity of PCA for traffic anomaly detection," *ACM SIGMETRICS Performance Evaluation Review*, vol. 35, no. 1, pp. 109–120, 2007.

[8] C. Callegari, L. Gazzarrini, S. Giordano, M. Pagano, and T. Pepe, "A novel PCA-based network anomaly detection," in *Communications (ICC), 2011 IEEE International Conference on*. IEEE, 2011, pp. 1–5.

[9] G. David, "Anomaly Detection and Classification via Diffusion Processes in Hyper-Networks," Ph.D. dissertation, Tel-Aviv University, 2009.

[10] T. Sipola, A. Juvonen, and J. Lehtonen, "Anomaly detection from network logs using diffusion maps," in *Engineering Applications of Neural Networks*, ser. IFIP Advances in Information and Communication Technology, L. Iliadis and C. Jayne, Eds. Springer Boston, 2011, vol. 363, pp. 172–181.

[11] A. Langville and C. Meyer, "Updating Markov chains with an eye on Google's PageRank," *SIAM journal on matrix analysis and applications*, vol. 27, no. 4, pp. 968–987, 2006.

[12] U. von Luxburg, "A tutorial on spectral clustering," *Statistics and Computing*, vol. 17, pp. 395–416, 2007.

PVIII

DIFFUSION MAP FOR CLUSTERING FMRI SPATIAL MAPS EXTRACTED BY INDEPENDENT COMPONENT ANALYSIS

by

Tuomo Sipola, Fengyu Cong, Tapani Ristaniemi, Vinoo Alluri, Petri Toiviainen,
Elvira Brattico and Asoke K. Nandi 2013

IEEE International Workshop on Machine Learning for Signal Processing (MLSP 2013), Southampton, United Kingdom

DIFFUSION MAP FOR CLUSTERING FMRI SPATIAL MAPS EXTRACTED BY INDEPENDENT COMPONENT ANALYSIS

Tuomo Sipola[1,*], *Fengyu Cong*[1,†], *Tapani Ristaniemi*[1], *Vinoo Alluri*[1,2],
Petri Toiviainen[2], *Elvira Brattico*[2,4,5], *Asoke K. Nandi*[3,1,‡]

[1]Department of Mathematical Information Technology,
University of Jyväskylä, Finland
[2]Finnish Centre of Excellence in Interdisciplinary Music Research,
University of Jyväskylä, Finland
[3]Department of Electronic and Computer Engineering,
Brunel University, United Kingdom
[4]Cognitive Brain Research Unit, Institute of Behavioral Sciences,
University of Helsinki, Finland
[5]Brain & Mind Laboratory, Biomedical Engineering and Computational Science,
Aalto University, Espoo, Finland

ABSTRACT

Functional magnetic resonance imaging (fMRI) produces data about activity inside the brain, from which spatial maps can be extracted by independent component analysis (ICA). In datasets, there are n spatial maps that contain p voxels. The number of voxels is very high compared to the number of analyzed spatial maps. Clustering of the spatial maps is usually based on correlation matrices. This usually works well, although such a similarity matrix inherently can explain only a certain amount of the total variance contained in the high-dimensional data where n is relatively small but p is large. For high-dimensional space, it is reasonable to perform dimensionality reduction before clustering. In this research, we used the recently developed diffusion map for dimensionality reduction in conjunction with spectral clustering. This research revealed that the diffusion map based clustering worked as well as the more traditional methods, and produced more compact clusters when needed.

Index Terms— clustering, diffusion map, dimensionality reduction, functional magnetic resonance imaging (fMRI), independent component analysis, spatial maps

1. INTRODUCTION

In order to gain understanding about the human brain, various technologies have recently been introduced, such as electroencephalography (EEG), tomography, magnetoencephalography (MEG) and functional magnetic resonance imaging (fMRI). They provide scientists with data about the temporal and spatial activity inside the brain. Functional magnetic resonance imaging is a brain imaging method that measures blood oxygenation level. It detects changes in this level, that are believed to be related to neurotransmitter activity. This enables the study of brain functioning, pathological trait detection and treatment response monitoring. The method localises brain function well, and thus is useful in detecting differences in subject brain responses [1, 2].

Deeper understanding about the simultaneous activities in the brain begins with a decomposition of the data. Independent component analysis (ICA) has been extensively used to analyze fMRI data. It tries to decompose the data into multiple components that are mixed in the original data. Basically, there are two ways to perform ICA: group ICA and individual ICA [3]. Group ICA is performed on the data matrix including all the participants' fMRI data, and individual ICA is applied on each dataset of each participant. Among datasets of different participants, group ICA tends to need more assumptions which are not required by individual ICA [4]. For individual ICA, if the components for each participant are known, it is expected to find the most common components among the participants. Therefore, clustering spatial maps extracted by ICA is a necessary step for the individual ICA approach to

[*] Tuomo Sipola's work was supported by the Foundation of the Nokia Corporation and the Finnish Foundation for Technology Promotion.

[†] This work was partially supported by TEKES (Finland) grant 40334/10 Machine Learning for Future Music and Learning Technologies.

[‡] Asoke K. Nandi would like to thank TEKES for the award of the Finland Distinguished Professorship.

find common spatial information across different participants in fMRI research.

ICA decomposes the individual datasets and creates components that can be presented with spatial maps. After ICA has been applied, a data matrix of size n by p is produced, where n is the number of spatial maps and p is the number of voxels of each spatial map. The n spatial maps come from different participants, and n is much smaller than p in fMRI research. Clustering the spatial maps is mostly done using the n by n similarity matrix of the n by p data matrix [3, 5]. Surprisingly, it usually works well although such a similarity matrix inherently can just explain a certain amount of the total variance contained in the high-dimensional n by p data matrix [5].

New mathematical approaches for functional brain data analysis should take into account the characteristics of the data analyzed. As stated, spatial maps have high dimensionality p. In machine learning, dimensionality reduction is usually performed on such datasets before clustering. In the small-n-large-p clustering problem, the conventional dimensionality reduction methods, for example, principal component analysis (PCA) [6], might not be suitable for the non-linear properties of the data. In this research, we apply a recently developed non-linear method called diffusion map [7, 8] for dimensionality reduction. The probabilistic background of the diffusion distance metric will give an alternative angle to this dataset by facilitating the clustering task and providing visualization. This paper explores the possibility of using the diffusion map approach for fMRI ICA component clustering.

2. METHODOLOGY

This paper considers a dimensionality reduction approach to clustering of high-dimensional data. The clustering procedure flows as follows:

1. Data normalization with logarithm
2. Neighborhood estimation
3. Dimensionality reduction with diffusion map
4. Spectral clustering

Data normalization should be done if the features are on differing scales. This ensures that the distances between the data points are meaningful. Neighborhood estimation for diffusion map creates the neighborhood where connections between data points are considered. Dimensionality reduction creates a new set of fewer features that still retain most information. Spectral clustering groups similar points together.

We assume that our dataset consists of vectors of real numbers: $X = \{x_1, x_2, \ldots, x_n\}, x_i \in \mathbb{R}^p$. In practice the dataset is a data matrix of size $n \times p$, whose rows represent the samples and columns the features. In this study each row vector is a spatial map and column vector contains the corresponding voxels in different spatial maps.

2.1. Diffusion map

Diffusion map is a dimensionality reduction method that embeds the high-dimensional data to a low-dimensional space. It is part of the manifold learning method family and can be characterized with its use of diffusion distance as the preserved metric [7].

The initial step of the diffusion map algorithm itself calculates the affinity matrix W, which has data vector distances as its elements. Here Gaussian kernel with Euclidean distance metric is used [7, 9]. For ϵ selection, see below. The affinity matrix is defined as

$$W_{ij} = \exp\left(-\frac{||x_i - x_j||^2}{\epsilon}\right),$$

where x_i is the p-dimensional data point. The neighborhood size parameter ϵ is determined by finding the linear region in the sum of all weights in W, while trying different values of ϵ [10, 11]. The sum is

$$L = \sum_{i=1}^{n} \sum_{j=1}^{n} W_{i,j},$$

From the affinity matrix W the row sum diagonal matrix $D_{ii} = \sum_{j=1}^{n} W_{ij}, i \in 1 \ldots n$ is calculated. The W matrix is then normalized as $P = D^{-1}W$. This matrix represents the transition probabilities between the data points, which are the samples for clustering and classification. The conjugate matrix $\tilde{P} = D^{\frac{1}{2}} P D^{-\frac{1}{2}}$ is created in order to find the eigenvalues of P. In practice, substituting P, we get

$$\tilde{P} = D^{-\frac{1}{2}} W D^{-\frac{1}{2}}.$$

This so-called normalized graph Laplacian [12] preserves the eigenvalues [9]. Singular value decomposition (SVD) $\tilde{P} = U \Lambda U^*$ yields the eigenvalues $\Lambda = \mathrm{diag}([\lambda_1, \lambda_2, \ldots, \lambda_n])$ and eigenvectors in matrix $U = [u_1, u_2, \ldots, u_n]$. The eigenvalues of P and $\tilde{P}$ stay the same. It is now possible to find the eigenvectors of P with $V = D^{-\frac{1}{2}} U$ [9].

The low-dimensional coordinates in the embedded space Ψ are created using Λ and V:

$$\Psi = V\Lambda.$$

Now, for each p-dimensional point x_i, there is a corresponding d-dimensional coordinate, where $d \ll p$. The number of selected dimensions depends on how fast the eigenvalues decay. The coordinates for a single point can be expressed as

$$\Psi_d : x_i \rightarrow [\lambda_2 v_2(x_i), \lambda_3 v_3(x_i), \ldots, \lambda_{d+1} v_{d+1}(x_i)]. \quad (1)$$

The diffusion map now embeds the data points x_i while preserving the diffusion distance to a certain bound given that enough eigenvalues are taken into account [7].

2.2. Spectral clustering

Spectral clustering is a method to group samples into clusters by benefitting from the results of spectral methods that reveal the manifold, such as the diffusion map. Spectrum here is understood in the mathematical sense of spectrum of an operator on the matrix P. The main idea is that the dimensionality reduction has already simplified the clustering problem so that the clustering itself in the low-dimensional space is an easy task. This leaves the actual clustering for any clustering method that can work with real numbers [13, 14, 15].

The first few dimensions from the diffusion map represent the data up to a relative precision, and thus contain most of the distance differences in the data [7]. Therefore, some of the first dimensions will be used to represent the data. Threshold at 0 in the embedded space divides the space between the possible clusters, which means that a linear classification can be used. With the linear threshold, the second eigenvector separates the data into two clusters in the low-dimensional space. This eigenvector solves the normalized cut problem, which means that there are small weights between clusters but the internal connections between the members inside the cluster are strong. Clustering in this manner happens through similarity of transition probabilities between clusters [13, 14, 16, 17].

3. RESULTS

The data comes from experiments where participants listened to music. The data analysis was performed on a collection of spatial maps of brain activity. After dimensionality reduction and spectral clustering, the results are presented and compared to more traditional methods.

3.1. Data description

In this research the fMRI data are based on the data sets used by Alluri et al. [18]. Eleven musicians listened to a 512-second modern tango music piece during the experiment. In the free-listening experiment the expectation was to find relevant brain activity significantly correlating with the music stimulus. The stimuli were represented by musical features used in music information retrieval (MIR) [18].

After preprocessing, PCA and ICA were performed on each dataset of each participant, and 46 ICA components (i.e., spatial maps) were extracted for each dataset [19, 20]. Then, temporal courses of the spatial maps were correlated with one musical feature, Brightness [18]. As long as the correlation coefficient was significant (statistical p-value < 0.05), the spatial maps were selected for further analysis. Altogether, $n = 23$ spatial maps were selected from 11 participants. The number of voxels for each spatial map was $p = 209{,}633$. So, the 23 by 209,633 data matrix was used for the clustering to find the common spatial map across the 11 participants.

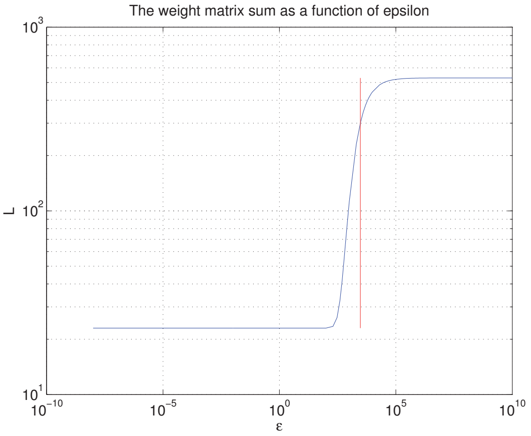

Fig. 1. Selecting ϵ for diffusion map. The red line shows the selected value.

3.2. Data analysis

The data matrix was analyzed using the methodology explained in Section 2. The dimensionality of the dataset was reduced and then the spectral clustering was carried out. The weight matrix sum for ϵ selection is in Figure 1; the used value is in the middle region, highlighted with straight vertical line. Clustering was performed with only one dimension in the low-dimensional space. To compare the results with more traditional clustering methods, the high-dimensional data was clustered with agglomerative hierarchical clustering [21] with Euclidean distances using the similarity matrix [5] and k-means algorithms [21]. The clustering results for two clusters were identical using all the methods.

Figure 2 shows the resulting clustering from the diffusion map. The figure uses the first two eigenpairs for low-dimensional presentation, for these two clusters even one dimension is enough. The spatial maps are numbered and the two clusters are marked with different symbols. The dividing spectral clustering line is at 0 along the horizontal axis, so the point to the right of 0 are in one cluster and to the left another. Two clusters, dense and sparse, are detected using this threshold. The dense cluster, marked with crosses, contains components that are considered to be similar according to this clustering. The traditional PCA and kernel PCA with Gaussian kernel for spectral clustering are compared to the diffusion map [22, 23]. In Figure 3 diffusion map with correct ϵ creates more firm connections, which eases the clustering task. The effect of diffusion distance metric is also seen.

In Figure 4 the dendrogram produced by the agglomerative clustering is shown. The clustering results are the same as with the dimensionality reduction approach. The separation is visible at the highest level and the structure corresponds

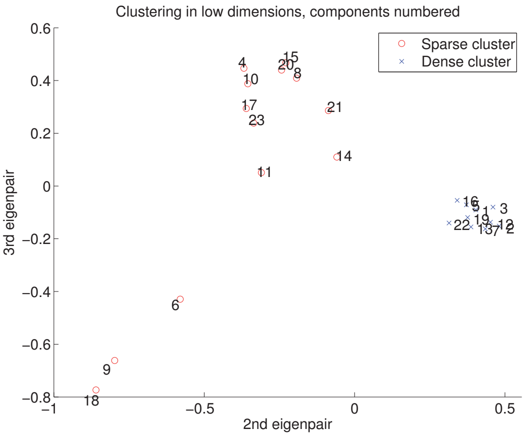

Fig. 2. Diffusion map clustering results.

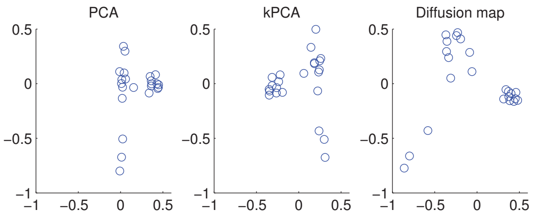

Fig. 3. Dimensionality reduction method comparison. The coordinates have been scaled.

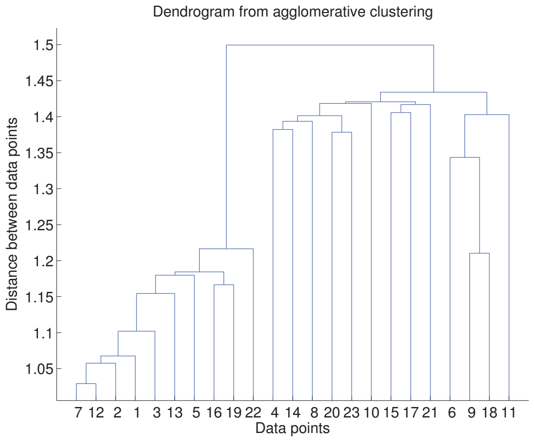

Fig. 4. Dendrogram from the agglomerative clustering.

to the distances seen in Figure 2. All the points in, e.g., the dense cluster in Figure 2 are in the left cluster of Figure 4. This comparison shows the evident separation between the two clusters and also validates the results from diffusion map methodology.

Figure 5 illustrates the kind of spatial maps that are found in the dense cluster. Dark areas along the lateral sides is used to highlight those voxels whose values differed more than three standard deviations from the mean. The numbers marking the slices are their Z-coordinates. The corresponding low-dimensional point is in Figure 2 numbered as 3. It is now possible to inspect the clusters more closely with domain experts.

Figure 6 shows the correlation matrix of all the 23 spatial maps. This is a way to inspect the similarity of the brain activity. The correlation matrix is also the basis of analysis for the hierarchical clustering [5]. In the figure it can be seen that there is some correlation between some of the spatial maps, but not so much between others.

Figures 7 and 8 illustrate the internal structure of the clusters by showing the correlation matrices for the individual clusters. The members in the dense cluster have higher correlation among themselves than the members in the sparse cluster. This information is also seen in Figure 2 where the diffusion distances inside the dense cluster are smaller.

4. DISCUSSION

In this paper we have proposed a theoretically sound nonlinear analysis method for clustering ICA components of fMRI imaging. The clustering is based on diffusion map manifold learning, which reduces the dimensionality of the data and enables clustering algorithms to perform their task. This approach is more suitable for high-dimensional data than just applying clustering methods that are designed for low-dimensional data. The assumption of non-linear nature of brain activity also promotes the use of methods designed for such problems. Particularly, the advantage of diffusion map is in visualizing the distribution of all data samples (n spatial maps with p voxels in each) by using only two coordinates. As seen in the visualization, it becomes more straightforward to determine the compact cluster from the two-dimensional plot derived from the 209,633-dimensional feature space than from the similarity matrix.

The results show that the proposed methodology separates groups of similarly behaving spatial maps. Results from diffusion map spectral clustering are similar to hierarchical agglomerative clustering and k-means clustering. Small sample size and good separation of clusters makes the clustering problem rather simple to solve. Moreover, the visualization obtained from diffusion map offers an interpretation for clustering.

The proposed methodology should be useful for analyzing the function of the brain and understanding which stimuli create similar spatial responses in which group of participants.

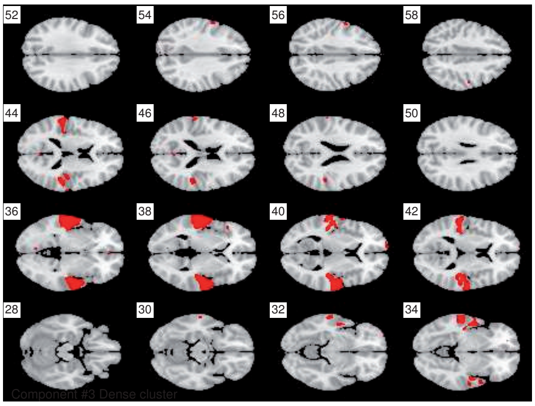

Fig. 5. Example spatial map in the dense cluster, this is data point number 3. Dark lateral areas mark more than three standard deviations from the mean, e.g. in slices 36 and 38.

The domain experts can gain more basis for the interpretation of brain activity when similar activities are already clustered using automated processes suitable for the task. Furthermore, visualization helps to identify the relationships of the clusters.

Diffusion map execution times become increasingly larger if the number of samples goes very high. This can be overcome to a certain degree with out-of-sample extension. Big sample sizes are also a problem with traditional clustering methods. However, diffusion map offers a nonlinear approach, and is suitable for high-dimensional data. Both properties are true for fMRI imaging data.

The analysis could be expanded to more musical features and to bigger datasets in order to further validate its usefulness in understanding the human brain during listening to music. The method is not restricted only to certain kind of stimulus, so it is usable with diverse fMRI experimental setups. Furthermore, situations where traditional clustering fails when processing spatial maps, the proposed methdodology might give more reasonable results.

5. REFERENCES

[1] P M Matthews and P Jezzard, “Functional magnetic resonance imaging,” *Journal of Neurology, Neurosurgery & Psychiatry*, vol. 75, no. 1, pp. 6–12, 2004.

[2] S. A. Huettel, A. W. Song, and G. McCarthy, *Functional Magnetic Resonance Imaging*, Sinauer, Massachusetts, 2nd ed. edition, 2009.

[3] Vince D Calhoun, Jingyu Liu, and Tülay Adalı, “A review of group ICA for fMRI data and ICA for joint inference of imaging, genetic, and ERP data,” *Neuroimage*, vol. 45, no. 1 Suppl, pp. S163, 2009.

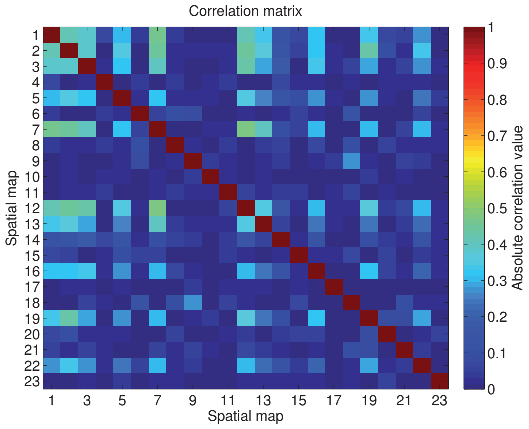

Fig. 6. Absolute correlation values between the spatial maps.

[4] Fengyu Cong, Zhaoshui He, Jarmo Hämäläinen, Paavo H.T. Leppnen, Heikki Lyytinen, Andrzej Cichocki, and Tapani Ristaniemi, “Validating rationale of group-level component analysis based on estimating number of sources in EEG through model order selection,” *Journal of Neuroscience Methods*, vol. 212, no. 1, pp. 165–172, 2013.

[5] Fabrizio Esposito, Tommaso Scarabino, Aapo Hyvarinen, Johan Himberg, Elia Formisano, Silvia Comani, Gioacchino Tedeschi, Rainer Goebel, Erich Seifritz, Francesco Di Salle, et al., “Independent component analysis of fMRI group studies by self-organizing clustering,” *Neuroimage*, vol. 25, no. 1, pp. 193–205, 2005.

[6] Ian Jolliffe, *Principal component analysis*, Springer Verlag, 2002.

[7] Ronald R. Coifman and Stéphane Lafon, “Diffusion maps,” *Applied and Computational Harmonic Analysis*, vol. 21, no. 1, pp. 5–30, 2006.

[8] B. Nadler, S. Lafon, R.R. Coifman, and I.G. Kevrekidis, “Diffusion maps, spectral clustering and reaction coordinates of dynamical systems,” *Applied and Computational Harmonic Analysis*, vol. 21, no. 1, pp. 113–127, 2006.

[9] Boaz Nadler, Stephane Lafon, Ronald Coifman, and Ioannis G. Kevrekidis, “Diffusion maps – a probabilistic interpretation for spectral embedding and clustering algorithms,” in *Principal Manifolds for Data Visualization and Dimension Reduction*, Timothy J. Barth, Michael Griebel, David E. Keyes, Risto M. Nieminen, Dirk Roose, Tamar Schlick, Alexander N. Gorban, Balázs Kégl, Donald C. Wunsch, and Andrei Y.

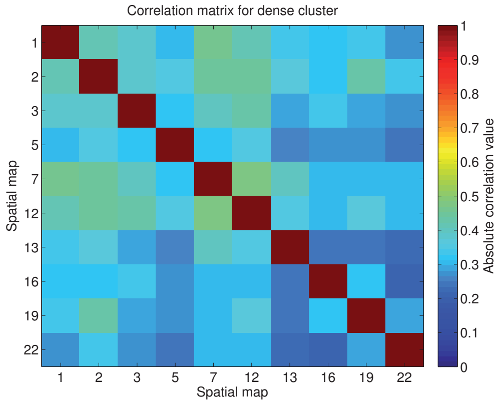

Fig. 7. Absolute correlation values between the spatial maps that belong to the dense cluster.

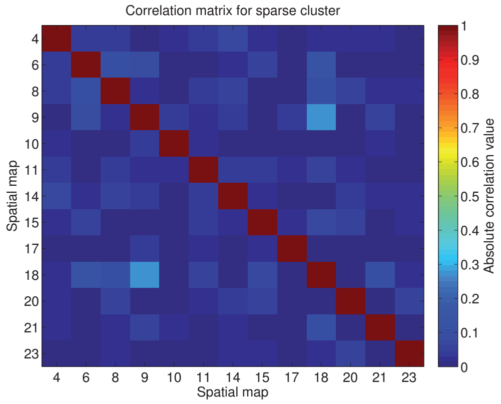

Fig. 8. Absolute correlation values between the spatial maps that belong to the sparse cluster.

Zinovyev, Eds., vol. 58 of *Lecture Notes in Computational Science and Engineering*, pp. 238–260. Springer Berlin Heidelberg, 2008.

[10] R.R. Coifman, Y. Shkolnisky, F.J. Sigworth, and A. Singer, "Graph laplacian tomography from unknown random projections," *Image Processing, IEEE Transactions on*, vol. 17, no. 10, pp. 1891–1899, oct. 2008.

[11] Amit Singer, Radek Erban, Ioannis G. Kevrekidis, and Ronald R. Coifman, "Detecting intrinsic slow variables in stochastic dynamical systems by anisotropic diffusion maps," *Proceedings of the National Academy of Sciences*, vol. 106, no. 38, pp. 16090–16095, 2009.

[12] F. R. K. Chung, *Spectral Graph Theory*, p. 2, AMS Press, Providence, R.I, 1997.

[13] Andrew Y. Ng, Michael I. Jordan, and Yair Weiss, "On spectral clustering: Analysis and an algorithm," in *Advances in Neural Information Processing Systems 14*. 2001, pp. 849–856, MIT Press.

[14] Ravi Kannan, Santosh Vempala, and Adrian Vetta, "On clusterings: Good, bad and spectral," *J. ACM*, vol. 51, pp. 497–515, May 2004.

[15] Ulrike von Luxburg, "A tutorial on spectral clustering," *Statistics and Computing*, vol. 17, pp. 395–416, 2007.

[16] Jianbo Shi and J. Malik, "Normalized cuts and image segmentation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 22, no. 8, pp. 888 –905, 2000.

[17] Marina Meila and Jianbo Shi, "Learning segmentation by random walks," in *NIPS*, 2000, pp. 873–879.

[18] Vinoo Alluri, Petri Toiviainen, Iiro P Jääskeläinen, Enrico Glerean, Mikko Sams, and Elvira Brattico, "Large-scale brain networks emerge from dynamic processing of musical timbre, key and rhythm," *Neuroimage*, vol. 59, no. 4, pp. 3677–3689, 2012.

[19] T. Puoliväli, F. Cong, V. Alluri, Q. Lin, P. Toiviainen, A. K. Nandi, E. Brattico, and T. Ristaniemi, "Semi-blind independent component analysis of functional MRI elicited by continuous listening to music," in *International Conference on Acoustics, Speech, and Signal Processing 2013 (ICASSP2013)*, Vancouver, Canada, May 2013.

[20] Valeri Tsatsishvili, Fengyu Cong, Tuomas Puoliväli, Vinoo Alluri, Petri Toiviainen, Asoke K Nandi, Elvira Brattico, and Tapani Ristaniemi, "Dimension reduction for individual ICA to decompose fMRI during real-world experiences: Principal component analysis vs. canonical correlation analysis," in *European Symposium on Artificial Neural Networks 2013*, Bruges, Belgium, April 2013.

[21] Rui Xu, Donald Wunsch, et al., "Survey of clustering algorithms," *Neural Networks, IEEE Transactions on*, vol. 16, no. 3, pp. 645–678, 2005.

[22] K-R Müller, Sebastian Mika, Gunnar Rätsch, Koji Tsuda, and Bernhard Schölkopf, "An introduction to kernel-based learning algorithms," *Neural Networks, IEEE Transactions on*, vol. 12, no. 2, pp. 181–201, 2001.

[23] Quan Wang, "Kernel principal component analysis and its applications in face recognition and active shape models," *arXiv preprint arXiv:1207.3538*, 2012.