BIG DATA APPLICATIONS AND PRINCIPLES

FIRST INTERNATIONAL WORKSHOP, BIGDAP 2014
MADRID, SPAIN, SEPTEMBER 11-12 2014
PROCEEDINGS

ALBERTO MOZO VELASCO
ANTONIO HERNANDO ESTEBAN
SANDRA GÓMEZ CANAVAL (Eds).
Preface

The International Workshop on Big Data Applications and Principles is a meeting where scientists and industrial experts can share their knowledge, experiences and insights on all topics related to the use of Big Data. **BigDap’14** - the first edition of this workshop- was organised in two separate blocks:

- An *academic track*, where scientists presented their latest theoretical or experimental results.

- An *industrial track*, where development and innovation professionals working on the field of Big Data had the opportunity to share their views on the application of this discipline to real-world scenarios.

This disciplinary counterpoint allowed for an enriching exchange between participants and attendees. The wide variety of topics discussed at the workshop proved that this field of research is fertile for experts coming from a large number of domains. Among the addressed topics were aeronautical engineering, medicine, telecommunications, social networks, the automotive industry, banking, law and various areas of computer science and engineering.

**BigDap’14** was held at the Escuela Técnica Superior de Ingeniería de Sistemas Informáticos of the Universidad Politécnica de Madrid, in the beautiful Campus Sur on September 11 and 12, 2014. The event was hosted by the FP7 project ONTIC (Online Network Traffic Characterization, http://ict-ontic.eu) that is funded by European Commission under the Seventh Framework Programme and welcomed a wide number of researchers, professionals, graduate and undergraduate students.

We would like to thank all the researchers that submitted papers as well as all invited speakers for their invaluable part in making this workshop a success.
Organization

Program Committee

Alberto Mozo          Universidad Politécnica de Madrid, Spain, Chair
Antonio Hernando     Universidad Politécnica de Madrid, Spain, Co-Chair
Daniele Apiletti     Politecnico di Torino, Italy
Fernando Arias       EMC Spain
Alejandro Bascuñana   Ericsson Spain
Elena Baralis        Politecnico di Torino, Italy
Luca Cagliero        Politecnico di Torino, Italy
Sotiria Chatzi       ADAPTIT, Greece
Luigi Grimaudo       Politecnico di Torino, Italy
Miguel Angel López   SATEC Spain
Miguel Angel Monjas  Ericsson Spain
Jose María Ocón      SATEC Spain
Philippe Owezarski   Centre National de la Recherche Scientifique, France
Spyros Sakellariou   ADAPTIT, Greece
Bo Zhu               Universidad Politécnica de Madrid

Organizing Committee

Sandra Gómez Canaval Universidad Politécnica de Madrid
Enrique Fernández Cantón Universidad Politécnica de Madrid
Alex Martínez Bravo   Universidad Politécnica de Madrid
Alvaro Guadamillas Herranz Universidad Politécnica de Madrid
Nuria Manchado Bustelos Universidad Politécnica de Madrid
Sponsors

We deeply thank the sponsors that made BigDap 2014 possible:

This workshop was supported by project ONTIC-619633 funded by the European Commission under the Seventh Framework Programme

Seventh Framework Program

European Commission

Online Network Traffic Characterization Project

BIGDAP-14 Chairs want to thank to E.T.S. Ingeniería de Sistemas Informáticos at Universidad Politécnica de Madrid for providing workshop facilities
Table of Contents

Big Data and Applications iii
Preface iv
Organization v
Table of Contents viii

1 Electronic Health Records Analytics: Natural Language Processing and Image Annotation 1
  1.1 Introduction ......................................................... 1
  1.2 Related ............................................................... 3
    1.2.1 Automatic Image Annotation ................................. 4
  1.3 Proposed Approach .................................................. 5
    1.3.1 Text analysis in TIDA ........................................... 6
    1.3.2 Training the models ............................................. 8
    1.3.3 PET Image Segmentation ....................................... 10
  1.4 Conclusions and future work ....................................... 12

Bibliography 15

2 Anomaly detection in recordings from in-vehicle networks 23
  2.1 Introduction .......................................................... 24
  2.2 Related work .......................................................... 24
# TABLE OF CONTENTS

2.3 Anomaly detection using one-class support vector machines .......................... 25
2.4 Enhancing SVDD to multivariate time series ................................................. 28
2.4.1 Transforming time series to feature vectors ............................................. 29
2.4.2 Working with subsequences ................................................................. 29
2.4.3 Determining the classification threshold ............................................... 30
2.5 Experimental results .................................................................................... 31
2.6 Conclusion ................................................................................................. 34

Bibliography ........................................................................................................ 36

3 Network traffic analysis by means of Misleading Generalized Itemsets* ............ 39
3.1 Introduction ................................................................................................. 40
3.2 Preliminary concepts and problem statement ............................................... 41
3.3 The MGI-Cloud architecture ................................................................. 44
3.3.1 Data retrieval and preparation ................................................................. 44
3.3.2 Taxonomy generation .............................................................................. 46
3.3.3 Level-sharing itemset mining ................................................................. 46
3.3.4 MGI extraction ....................................................................................... 47
3.4 Experiments ............................................................................................... 48
3.4.1 Result validation ...................................................................................... 48
3.4.2 Scalability with the number of cluster nodes ......................................... 49
3.5 Conclusions and future perspectives ............................................................ 49

Bibliography ........................................................................................................ 51

4 Unsupervised Detection of Network Attacks in the dark ................................. 53
4.1 Introduction ................................................................................................. 54
4.2 Related Work & Contributions .................................................................... 56
4.3 Unsupervised Detection of Attacks .............................................................. 56
4.4 Automatic Characterization of Attacks ....................................................... 58
4.5 Experimental Evaluation ............................................................................ 59
### TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.6 Computational Time and Parallelization</td>
<td>62</td>
</tr>
<tr>
<td>4.7 Conclusions</td>
<td>64</td>
</tr>
<tr>
<td><strong>Bibliography</strong></td>
<td>66</td>
</tr>
<tr>
<td>5 A Survey of Feature Selection in Internet Traffic Characterization</td>
<td>69</td>
</tr>
<tr>
<td>5.1 Introduction</td>
<td>70</td>
</tr>
<tr>
<td>5.2 Feature Selection</td>
<td>71</td>
</tr>
<tr>
<td>5.2.1 1.1 Feature Selection Framework</td>
<td>71</td>
</tr>
<tr>
<td>5.2.2 Feature Selection Method Categories</td>
<td>72</td>
</tr>
<tr>
<td>5.2.3 Feature Selection Methods in Internet Traffic Characterization</td>
<td>79</td>
</tr>
<tr>
<td>5.3 Future Practical Application and Conclusions</td>
<td>81</td>
</tr>
<tr>
<td><strong>Bibliography</strong></td>
<td>83</td>
</tr>
<tr>
<td>6 Adaptive Quality of Experience (AQoE) control for Telecom Networks</td>
<td>91</td>
</tr>
<tr>
<td>6.1 Introduction</td>
<td>92</td>
</tr>
<tr>
<td>6.2 Description of the Problem</td>
<td>93</td>
</tr>
<tr>
<td>6.3 State of the Art</td>
<td>94</td>
</tr>
<tr>
<td>6.4 Standardization and related activities</td>
<td>96</td>
</tr>
<tr>
<td>6.5 Use Cases Description</td>
<td>98</td>
</tr>
<tr>
<td>6.6 Conclusions and Future Works</td>
<td>99</td>
</tr>
<tr>
<td><strong>Bibliography</strong></td>
<td>101</td>
</tr>
<tr>
<td>7 A Telecom Analytics Framework for Dynamic Quality of Service Management*</td>
<td>103</td>
</tr>
<tr>
<td>7.1 Introduction</td>
<td>104</td>
</tr>
<tr>
<td>7.2 Related Work</td>
<td>105</td>
</tr>
<tr>
<td>7.2.1 QoS architecture on Internet and UMTS</td>
<td>106</td>
</tr>
<tr>
<td>7.2.2 Traffic characterization</td>
<td>108</td>
</tr>
<tr>
<td>7.2.3 Machine Learning algorithms</td>
<td>109</td>
</tr>
<tr>
<td>7.2.4 Quality of Experience</td>
<td>111</td>
</tr>
</tbody>
</table>
TABLE OF CONTENTS

7.3 A Telecom Analytics Framework .................................................. 115
  7.3.1 Analytics ................................................................................. 117
  7.3.2 Policy Controller (PC) ............................................................... 118
  7.3.3 Enforcement Point (EP) ............................................................. 119
7.4 The Process Architecture for Online and Offline Scalability ............... 121
  7.4.1 Development and Implementation Model .................................... 121
  7.4.2 Implementation of the Scenario ................................................ 122
7.5 Conclusions and Future Work .......................................................... 124
7.6 Acknowledgment ........................................................................... 126

Bibliography ....................................................................................... 127

8 Author Index .................................................................................. 134
1 Electronic Health Records Analytics: Natural Language Processing and Image Annotation

Roberto Costumero, Angel Garcia-Pedrero, Isaac Sánchez, Consuelo Gonzalo and Ernestina Menasalvas

Abstract: Big data applications in the Healthcare sector indicate a high potential for improving the overall efficiency and quality of care delivery. Health data analytics highly relies on the availability of Electronic Health Records (EHRs). The complexity of healthcare information management is not only due to the amount of data generated but also by its diversity and the challenges of extracting knowledge from unstructured data. Solutions have not proposed until now an integrated solution to process, mine and extract knowledge. In this paper we propose an architecture, TIDA, to deal in an integrated way with all the information contained in EHR. TIDA (Text, Image and Data Analytics) makes it possible to address the problem of Spanish text indexing in the healthcare domain by adapting different techniques and technologies, besides components have been included to deal with image segmentation.

1.1 Introduction

Significant advances in health information technology, which allow the secure digital storage and transfer of health information among industry players, will create new sources of
Electronic Health Records Analytics: Natural Language Processing and Image Annotation

healthcare data and can support new ways of using analytics. Health data analytics highly relies on the availability of Electronic Health Records (EHRs). The adoption of hospital EHR technology is significantly growing and is expected to continue growing. While one third of the US hospitals (35%) had already implemented some kind of EHR technology in 2011, it is expected that by 2016 nearly every US hospital (95%) will use EHR technology. According to [42], back in 2009, Spain was the second on EHR adoption, with roughly 60%, meanwhile it was projected to have an adoption of 83% by 2013, with Nordic countries leading and US doubling its adoption in the 2011-2013 period.

However technology for health care data analysis is not mature enough due to, among other reasons, a lack of standards, interoperable data schemas and natural text and image processing tools. In particular in this paper we focus on natural text processing and image annotation.

Patient’s medical records contain valuable clinical information written in narrative form either as part of the clinical history or as a report of a image. Thus, in order to find relevant information it is often necessary to extract it from free-texts in order to support clinical and research processes. NLP (Natural Language Processing) is the set of processes to obtain models for natural language understanding. Despite of Spanish occupying the second place in the ranking of number of speakers with more that 500 million speakers (according to [1]) as far as our knowledge there is no natural language tool for Spanish processing.

On the other hand, the constantly development of new image acquisition devices and the increase of data storage capacity are providing a steady growth of the number of medical images produced. With such an exponential increase of medical data in digital libraries, it is becoming more and more difficult to execute certain analysis on search and information retrieval-related tasks. A possible approach to avoid this difficulty is to develop annotation techniques that allow to storage the information contained in the images in a structured way. Traditionally, this task has been carried out by operators; however, even though the manual annotation can be very precise, it suffers from the subjectivity of the operator and it is highly time consuming. Moreover, the repetition of the delineation of organs or pathologies is not assured even when the same operator performs it at two different times. Therefore, automatic techniques are required.
1.2 Related

In this paper, we aim to deal with integration of image and text information from EHR. Consequently, we present advances on Natural Language Processing of Spanish text as well as previous studies carried out in the field on image annotation. The rest of the paper has been structured as follows: Section 1.2 present advances both in text processing tools and image annotation in the medical domain. Section 1.3 presents our proposed approach to deal with text processing in Spanish and image annotation and some results that have been preliminary obtained from the application of the framework in some real cases. It is important noting that results and approach presented in this paper have already been published in [7], [8]. To end with, in section 1.4 we present the main conclusions extracted so far and outline future lines of research.

1.2 Related

In Natural Language Processing (NLP) the input of natural language is transformed in several steps of a pipeline to get computers to understand it. In the treatment of free-text NLP, the text serves as an input to the system which will lead to several structured components with semantic meaning so the information can be manipulated knowing the importance of the different parts of the speech. To do the proper training of NLP models, a properly annotated corpus is needed. [3]

These models are generated from the training data included in the corpus to analyze future input sentences. Although there are several studies introducing different English corpora to train models, we focus here on the ones which have been used in the healthcare domain. As cited in Savova et al. [32], there are no community resources such as annotated medical corpus in the clinical domain, so in the evaluation on tools like cTAKES, own corpus has been developed. Using the gold standard linguistic annotations of Penn TreeBank (PTB) [51] and GENIA corpus [25] together with their own Mayo Clinic EMR corpus, cTAKES models were trained. The lack of corpus in Spanish language for the healthcare domain makes such training difficult these days. cTAKES processes clinical notes written in English identifying different medical concepts from several dictionaries, included own developed ones, but also Unified Medical Language System (UMLS)[3], which gathers medications,
1 Electronic Health Records Analytics: Natural Language Processing and Image Annotation

diseases, symptoms and procedures.

There are different corpora in other languages different than English, but it is very difficult to find annotated corpus in the healthcare domain, as Natural Language Processing has not yet been extensively used in this domain. A multilevel annotated corpus for Catalan and Spanish known as AnCora (ANnotated CORporA) [35] is available (http://clic.ub.edu/corpus/). It has almost half a million words ready to test NLP systems.

1.2.1 Automatic Image Annotation

The annotation techniques consist of two different steps. The first is the segmentation of the image with the aim of identifying regions of interest that can correspond with organs or pathologies to be studied. Since there are very different types of medical images (PET, CT, RX, ...), different algorithms should be used for each of them. The second step is the classification of these regions with semantic labels. In this paper, we are going to refer only to the first one for the particular case of PET images.

Given the high contrast presented by PET images, it is considered that thresholding methods can be a good approach to segment this kind of images or in other word to identify the susceptible regions to be considered as pathologies. Several thresholding methods applied to PET images can be found in the literature. Here, for space reasons, we will only mention few works that allow justify the selection of the methods used in this study.

As far as our knowledge, only the method proposed by Paulino et al [30] uses a fixed threshold of the SUV \( SUV_{th} = 2.5 \) value. This value was determined by a comparative study for different malignant tumors in head and neck for distinct patients. However, some studies [10],[52] show that the ideal threshold depends on the size of the sphere (phantom) used to carry out the statistical studies, on the background activity and on other factors, and in this sense it has been proposed the use of relatives thresholds. These studies have proved that relative threshold are always between 36% and 44% of the maximum SUV. In the study conducted by Prieto et al. [47], 21 dynamic thresholding techniques were analyzed and the results compared with the obtained applying a relative threshold of 42%. The results on
phantoms showed that the dynamics thresholding techniques of Ridler [50], Ramesh [49], Yanni [48] and Otsu [46] provide the best results.

This first step of identifying tumoral regions in an automatic way in a PET image is the base to the whole automatic annotation process. Next the information about the position and extension of the identified regions should be stored as structured information associated to this image. The next step will be the characterization of these regions with low level features to finally associate them some semantical labels. Even though there are a lot of work abording this challenge with different approaches [53, 54], the general conclusion is that there are many research issues that should be solved to bridge the semantic gap between low level image features and high level semantics.

1.3 Proposed Approach

Our proposed approach is based on a integrated architecture that we have called TIDA (Text, Image and Data Analytics).

TIDA is designed to build flexible applications over a typical data storage system relying on the information obtained from healthcare databases, which has previously been gathered together into a common storage, so different components get the information from a common warehouse. In order to fulfill this requirements, the architecture (see Figure 1.1) presents the following components:

- **DB** A common data storage system with all hospital’s data, from reports to images and patient’s structured information which will serve information to the immediate upper level of components.

- **Mayo/Apache cTAKES** A free-text analysis system built upon Mayo’s cTAKES framework. This framework relies on information gathered through UMLS from different dictionaries for diseases, drugs or laboratory test classifications such as ICD, RxNorm, LOINC or SNOMED CT.

- **Image transformation framework** A set of own developed applications to determine
anomalies and automatically annotate medical images using the IBM UIMA architecture which cTAKES is built upon.

- **IBM/Apache UIMA** The two previous components’ output is built using IBM UIMA, to get a structured view of the unstructured data.

- **Patient’s structured data** A component to deal with structured data.

- **Structured data, images and text annotator** In charge of annotating text and images supported by UIMA and the structured information.

- **Apache Lucene** This component indexes all the previously annotated data to serve different kinds of applications.

- **Apache Solr** A semantic search engine to bring quick, reliable semantic search into the picture.

- **API** An API powered by Solr’s output, to bring more functionality and link end-user web applications to the whole set of the architecture.

- **End-user web application** Main goal of the architecture is to serve end-user applications to give different functionalities on the same data structures.

TIDA’s architecture is designed to be language-independent, so the platform principles stay the same no matter which language texts are written in. Language-dependent components UMLS and cTAKES can be adapted so the architecture works as expected.

### 1.3.1 Text analysis in TIDA

Due to the lack of health related corpora in Spanish, we decided to use AnCora [26] as a general domain annotated corpus.

There should be considered two main processes in the text analysis:
1.3 Proposed Approach

Figure 1.1: Principal components of TIDA’s architecture

1. **Indexation**, which is a heavy process due to a complete text analysis and which runs in a pipeline the different processes presented in Figure 1.2, starting with Natural Language Processing and following with the Negation detection algorithms.

2. **Searching**, which thanks to indexation is a very lightweight process.

TIDA’s text analysis relies on cTAKES [32], which has been briefly introduced in Sec-
cTAKES is an English-centric development, which means that, although it has a very good design and is very modular, it has been developed towards an English comprehensive system. This introduces a challenge to make it suitable to use in other languages. It is not on the scope of this paper to introduce cTAKES architecture, but we will introduce key components to develop our work.

cTAKES uses a common NLP pipeline including a Sentence Detector, a Tokenizer, a Part of Speech Tagger, a Chunker and a Dictionary Lookup for Named Entity Recognition. We focus on this paper in the analysis of the training of Spanish models for the Sentence Detector, Part of Speech Tagger and the Chunker with both the AnCora corpus and the in-house one, that we have developed.

### 1.3.2 Training the models

The process is composed of the following phases:

1. **AnCora training models construction.** In a first step models for the Sentence Detector, PoS and Chunker are obtained using the AnCora corpus. Note that the tokenizer is not trained as AnCora already provides tokens.

2. **Manual annotation of the in-house corpus.** A program for sentence detection is used to split the corpus into sentences. Starting from there, by means of a tools developed for this task three independent persons annotate the medical corpus. The tool makes it possible for annotators to correct the sentence detection, tag the PoS and perform the chunker annotation.
3. In-House medical corpus training models (from now on called in-house corpus). The set of documents annotated are split into training and evaluation datasets. The training set will be used to train models with the same 10-fold cross-validation method used for the AnCora corpus.

Note that OpenNLP’s Tokenizer is not trained. Instead, we have built a tokenizer in order to extract the tokens in the medical texts to annotate, in which we have taken into account punctuation symbols, multiple spaces and line breaks to determine the tokens.

In what follows we detail how the Sentence Detector, the PoS, and the Chunker have been trained.

**Sentence boundary detector**
Experiments were conducted splitting the training dataset into ten folds of an equivalent number of documents inside each fold.

We experimented with iteration values ranging from 100 to 600 and cut-off values from 1 to 16 for each corpus. The values that resulted in the maximal F-Measures for the average values of the ten folds are 600 iterations and a cut-off of 16 for the AnCora corpus, and 475 iterations and a cut-off of 7 for the medical texts corpus.

AnCora corpus performed with a precision of 0.948, a recall of 0.963 and a F-Measure of 0.955. In-house corpus performed with a precision of 0.827, a recall of 0.714 and a F-Measure of 0.766.

The main source of errors in this particular component were due to headings or sentences that did not use proper punctuation symbols, so the models were not able to reproduce the sentences correctly.

**Part-of-speech tagger**
As in the Sentence Detector, the experiments were conducted by splitting the training dataset into ten folds of an equivalent number of documents inside each fold. Both corpora are tagged with EAGLES [41] tags.
We experimented with iteration values ranging from 100 to 400 and cut-off values from 1 to 10 for each corpus. The maximal accuracies of the average values of accuracies for the ten folds occurred for 100 iterations and a cut-off of 2 for the AnCora corpus, and 100 iterations and a cut-off of 1 for the medical texts corpus. The medical documents include great amounts of abbreviations and short sentences, so the cut-off value is intended to be low. Ancora corpus showed an accuracy of 0.921 while in-house corpus got an accuracy of 0.774.

**Chunker**

For this component, the training dataset was split into ten folds of an equivalent number of sentences in each fold. Chunker tags used are: Nominal Phrases (NP), Adjetival Phrases (AP), Verbal Phrases (VP), Adverbial Phrases (RP) and Others (O).

We determined the optimal iterations and cut-off values by experimenting with iteration values ranging from 100 to 400 and cut-off values from 1 to 10. The average values for the ten folds showed to be maximal for 100 iterations and a cut-off of 3 for the AnCora corpus, and 150 iterations and a cut-off of 3 for the medical texts corpus.

The precision, recall an F-Measure for the Ancora corpus were 0.908, 0.907 and 0.907, respectively. In-house corpus performed with a precision of 0.852, a recall of 0.860, and a F-Measure of 0.856.

### 1.3.3 PET Image Segmentation

During a PET scan, a radioactive metabolic tracer like 2-deoxy-2-(18F)fluoro-D-glucose (18F-FDG) is injected into the patient’s body. It takes 30 minutes to an hour for the tracer to get completely absorbed by the organs or tissues under study. After the mentioned time, injected tracer can reach in a position to emit gamma rays. The PET scanner detects these gamma rays and thereby determines the movement of the tracer in the body over time. This type of images are also known as metabolic images. These regions are recognized by the higher intensity in the image, due to a high metabolism, or in the other words a high response of the tracer. However, some particular organs as brain, kidneys and bladder also present a high metabolism.
1.3 Proposed Approach

We have applied 4 different thresholding segmentation techniques with the aim of automatically identifying the tumoral regions in the PET images of six different patients in DICOM format. All these images have been previously analyzed and informed by an expert (physician). Therefore the position and extension of each tumoral region is known. The selection of thresholding methods, over all the segmentation techniques mentioned above, relies on two criteria: minimizing the number of parameters to use, and their simplicity from both conceptual and computational point of view.

Therefore the methods selected for this study are the proposed by: Paulino et al. [30], Erdi [10], Otsu [46] and Ridler and Calvard [50]. As it has been mentioned before, most studies regarding thresholding techniques applied to PET image segmentation has been proved on phantoms. One of the main contributions of this work is that the results have been obtained from real patients.

Previously to the segmentation of the images, the voxel intensity values have been transformed to SUV (standard uptake value) values. The SUV is a metabolic measure that depends on several factors: the metabolic activity of each voxel, the time, the tracer dose and the patients’ weight [55].

The four segmentation techniques have been applied to the whole set of images. And the results have been evaluated using three different error indices proposed in [47] and described in Table 1.1.

In Figure 1.3, a set box-and-whisker diagrams representing graphically the basic statistical values of each analyzed index is displayed. The different methods have been represented in the horizontal axis.

The $CE$ index advantage relative to the $VE$ is that it is not only able to compare volumes, but it also considers if the voxels positioning is correct or incorrect, since you could get an identical volume in quantity but the actual volume may be displaced. In a complementary way the DSI index measures the similarity both in volume and position of the two compared volumes.
### Table 1.1: Definition of indices used for evaluation of segmentation results

<table>
<thead>
<tr>
<th>Index</th>
<th>Equation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$VE(%)$</td>
<td>$100 \cdot \frac{\mid</td>
<td>F_0</td>
</tr>
<tr>
<td>$CE(%)$</td>
<td>$100 \cdot \frac{</td>
<td>F_0 \cap B_t</td>
</tr>
<tr>
<td>$DSI(%)$</td>
<td>$100 \cdot \frac{2 \cdot</td>
<td>F_0 \cap F_t</td>
</tr>
</tbody>
</table>

Lower values of $VE$ and $CE$ indices indicate a better segmentation.

A high value of $DSI$ index points out a better performance.

### 1.4 Conclusions and future work

The increase of data generated and the adoption of IT in healthcare have motivated the development of systems such as IBM Watson and frameworks like cTAKES (integrated exclusively in an English speaking environment). The huge amount of Spanish speaking people leads to the development of new solutions adapted to Spanish. Thus, in this paper we have presented TIDA, an approach which focuses into the generation of a complete EHR semantic search engine which brings an integrated solution to medical experts to analyze and identify markers in a system that brings together text, images and structured data analysis.

Results presented demonstrate that the adaptation of existing algorithms and technologies to the Spanish environment is currently working and that cognitive systems can be built to work in the health domain.

In the case of the PET image segmentation, the obtained values for the three indices allow us to estimate a trend in the behavior of the threshold algorithms studied. Erdi algorithm, which offers a fixed and relative threshold, has provided the worst results, therefore it can
1.4 Conclusions and future work

Figure 1.3: Boxplots of the thresholding methods for each error measure: a) VE, b) CE, and c) DSI indexes.

be discarded as a possible technique for automatic delineation of tumor regions. This poor performance is mainly due to the fact already mentioned that there are some organs in the human body that capture similar or higher tracer concentrations than the tumors. Otsu and Ridler methods have provided similar results and quite encouraging in most patients, but for some patients those methods have made a very high error on the estimation of tumor regions, which is reflected in the value of the indices. Firstly, it would be necessary to extend the study
to a more significant number of patients and secondly, it can be possible that another multi-
thresholding version of these algorithms would provide better results. The method proposed
by Paulino, an absolute fixed threshold, has provided better results. In addition to the fact that
it is a very simple method to implement, and not dependent parameters, it will be the subject
of further studies.
Bibliography


BIBLIOGRAPHY


[23] Hripcsak, G. and Austin, J. HM. and Alderson, P. O. and Friedman, C., Use of Natural Language Processing to Translate Clinical Information from a Database of 889,921
BIBLIOGRAPHY


BIBLIOGRAPHY


[38] Zeng, Q. T. and Goryachev, S. and Weiss, S. and Sordo, M. and Murphy, S. N. and Lazarus, R., *Extracting principal diagnosis, co-morbidity and smoking status for
BIBLIOGRAPHY

asthma research: evaluation of a natural language processing system, BMC medical informatics and decision making, BioMed Central Ltd, volume 6, number 1, page 30, 2006.


2 Anomaly detection in recordings from in-vehicle networks

Andreas Theissler
IT-Designers GmbH, Esslingen, Germany
{andreas.theissler@it-designers.de}

Abstract: In the automotive industry test drives are being conducted during the development of new vehicle models or as a part of quality assurance of series-production vehicles. Modern vehicles have 40 to 80 electronic control units interconnected via the so-called in-vehicle network. The communication on this in-vehicle network is recorded during test drives for the use of fault analysis, which results in big data. This paper proposes to use machine learning to support domain-experts by preventing them from contemplating irrelevant data and rather pointing them to the relevant parts in the recordings. The underlying idea is to learn the normal behaviour from the available multivariate time series and then to autonomously detect unexpected deviations and report them as anomalies. The one-class support vector machine “support vector data description” is enhanced to work on multivariate time series. The approach allows to detect unexpected faults without modelling effort as is shown on recordings from test drives. The proposed methodology could be applicable to multivariate time series from other sources, e.g. industrial plants or network traffic with fixed communication patterns.

Keywords: anomaly detection, data mining, fault detection, machine learning
2 Anomaly detection in recordings from in-vehicle networks

2.1 Introduction

This paper proposes an approach to detect anomalies in multivariate times series in recordings from in-vehicle networks. The effectiveness of the approach is shown by applying it to big data recorded during vehicle tests.

Modern vehicles have 40 to 80 electronic control units (ECUs) interconnected via the so-called in-vehicle network. Those ECUs read data measured by sensors, calculate values and control actuators. The sensors’ and actuators’ values are being transmitted over the in-vehicle network to other ECUs. This results in a highly complex network of software and hardware subsystems.

In order to be able to locate faults or to evaluate the behaviour of vehicle subsystems, the communication on the in-vehicle network is being recorded during test drives. This kind of recordings are conducted by manufacturers with prototype vehicles, before start of production, or with series-production vehicles as part of the end of line tests.

The big data resulting from recording test drives is in some cases searched for known fault patterns. Additionally, suspicious behaviour is reported by the test drivers. However, there are no systematic measures to detect unexpected faults. To address this shortcoming, this paper contributes by proposing an approach that

- uses available multivariate time series from in-vehicle networks and extracts the relevant knowledge
- autonomously points the expert to anomalies in the time series

From the reported anomalies, the expert can start investigating the data base of recordings in a goal-oriented way.

2.2 Related work

In [12], the authors propose to use visual analytics to explore data from automotive systems. In [8] a data-driven approach to classify the health state of an in-vehicle network based on
the occurrences of error frames on the CAN bus is proposed. In contrast to the underlying paper, [8] bases on a training set of recordings from fault-free and faulty mode.

In [2] anomaly detection is used on vehicle data in the field of road condition monitoring. Based on a training set of recordings from drives in normal operation mode, potholes are identified as anomalies.

In [6] intrusion detection based on recordings from in-vehicle network communication is presented. The underlying assumption is that the communication on the in-vehicle network has a certain degree of randomness, i.e. entropy. From data recorded in normal operation mode, the normal value of entropy is learnt. An attack, like increasing the frequency of specific messages or message flooding, appears less random and is thereby detected as an anomaly.

In [3] classification between sober and drunk drivers based on ARMA models is proposed. The determination of the order and the coefficients of the models is a great challenge with that approach.

### 2.3 Anomaly detection using one-class support vector machines

Detecting anomalies can be automated by teaching an anomaly detection system normal and abnormal behaviour by the means of a labelled training set and have the system classify unseen data. This corresponds to a two-class classification problem. The task is to assign an unclassified instance to either the normal class $\omega_n$ or the abnormal class $\omega_a$ based on a set of features $f$. For fault-detection two major drawbacks of such a traditional classification approach were identified:

1. Often no abnormal data sets exist beforehand. On the other hand normal data can be obtained by recording data from a system in normal operation mode.

2. Even if abnormal data exists, it is highly likely that it is not representative, because many faults in a system are not known. Using a non-representative training data set of
Anomaly detection in recordings from in-vehicle networks

anomalies, an incorrect decision function is learned.

An alternative is to only learn the normal behaviour and classify deviations as abnormal referred to as one-class classification. Support vector machines (SVM) [11, 1] have shown to yield good results on classification tasks and have been widely used. In [9] the one-class SVM “support vector data description” (SVDD) was introduced to cope with the problem of one-class classification. SVDD finds a closed decision boundary, a hypersphere, around the normal instances in the training data set using a so-called kernel function. It is therefore ideal for anomaly detection.

The hypersphere is determined by the radius $R$ and the center $a$, as illustrated in 2.1, and is found by solving the optimisation problem of minimising the error on the normal class and the chance of misclassifying data from the abnormal class.

The error on the normal class is minimised by adjusting $R$ and $a$ in a way that all instances of the training data set are contained in the hypersphere. Minimising the chance of misclassifying data from the abnormal class is done by minimising the hypersphere’s volume. The trade-off $F$ between the number of misclassified normal instances and the volume of the normal region is optimised by minimising

$$F(R, a) = R^2$$

subject to

$$\|x_i - a\|^2 \leq R^2 \quad \forall i = 1, \ldots, M$$

where $x_i$ denotes the instances and $M$ the number of instances in the training data set, $a$ is the hypersphere’s center, and $\|x_i - a\|$ is the distance between $x_i$ and $a$.

The hypersphere is described by selected instances from the training data set, so-called support vectors. The center $a$ is implicitly described by a linear combination of the support vectors. The remaining instances are discarded.

If all instances are contained in the hypersphere, outliers contained in the training data set massively influence the decision boundary, which is not desired. Slack variables $\xi_i$ are intro-
2.3 Anomaly detection using one-class support vector machines

Figure 2.1: A hypersphere in a 2-dimensional feature space with radius \( R \) and center \( a \) is described by the three support vectors \( SV_1 \ldots SV_3 \).

duced, which allow for some instances \( x_i \) in the training data set to be outside the hypersphere. The parameter \( C \) is introduced controlling the influence of the slack variables and thereby the error on the normal class and the hypersphere’s volume. So the optimisation problem of 2.1 and 2.2 changes into minimising

\[
F(R, a, \xi_i) = R^2 + C \sum_{i=1}^{M} \xi_i
\]

subject to

\[
\|x_i - a\|^2 \leq R^2 + \xi_i \quad \forall i \quad \text{and} \quad \xi_i \geq 0 \quad \forall i
\]

As described in [10], the constrained optimisation problem is transformed into an unconstrained one by integrating the constraints into the equation using the method of Lagrange [4]. The partial derivatives w.r.t. \( R, a, \xi \) are set to 0 and the resulting equations are resubstituted, yielding the following optimisation problem to be maximised:

\[
L(\alpha) = \sum_{i=1}^{M} \alpha_i (x_i \cdot x_i) - \sum_{i,j=1}^{M} \alpha_i \alpha_j (x_i \cdot x_j)
\]
2 Anomaly detection in recordings from in-vehicle networks

subject to

$$0 \leq \alpha_i \leq C \quad \forall i$$  \hspace{1cm} (2.6)

Finally, since strictly spherical-shaped decision boundaries are not appropriate for most data sets, non-spherical decision boundaries are introduced by mapping the data into a higher-dimensional space by the so-called kernel trick [11].

As indicated by 2.5, $x_i$ and $x_j$ are solely incorporated as the inner products $(x_i \cdot x_i)$ and $(x_i \cdot x_j)$ respectively. Instead of actually mapping each instance to a higher-dimensional space using a mapping function $\phi()$, the so-called kernel trick is used to replace the inner products $(\phi(x_i) \cdot \phi(x_j))$ by a kernel function $K(x_i, x_j)$. The radial basis function (RBF) kernel is used, because it is reported to be most suitable to be used with SVDD in [10]. The RBF kernel is given by

$$K(x_i, x_j) = e^{-\frac{\|x_i - x_j\|^2}{\sigma^2}}$$  \hspace{1cm} (2.7)

Incorporating the RBF kernel 2.5 becomes:

$$L(\alpha) = 1 - \sum_{i,j=1}^{M} \alpha_i \alpha_j K(x_i, x_j)$$  \hspace{1cm} (2.8)

A major challenge in one-class classification problems is having to adjust parameters, in this case the parameters $C$ and $\sigma$. The approach proposed in [13] was used to solve this problem.

2.4 Enhancing SVDD to multivariate time series

Based on SVDD, in this section an enhancement is shown, that makes SVDD applicable to multivariate time series. The approach was proposed by this paper’s author in [14].
2.4 Enhancing SVDD to multivariate time series

2.4.1 Transforming time series to feature vectors

A recording contains multiple time-stamped signals, i.e. it corresponds to multivariate time series data [5]. Transforming the multivariate time series to feature vectors is done by transforming the values at each time point $T_i$ to one feature vector $x_i$. Thereby, a $N \times M$ multivariate time series is transformed to $N$ feature vectors of length $M$.

2.4.2 Working with subsequences

Time series data from technical systems can be considered noisy. As a consequence, it is very likely that a fraction of individual data points of previously unseen data lies outside the decision boundary without actually being abnormal, which is confirmed by experiments on the recordings from vehicles. Instead of classifying feature vectors, subsequences in the original time series are formed using a fixed-width non-overlapping window of length $W$. While this approach ignores the order of the subsequences, it takes into account the local neighbourhood of the data points.

In order to classify subsequences, a distance measure for the subsequences has to be defined. Informally spoken, the distance measure should yield a big distance for a subsequence if many data points lie outside the decision boundary or if few data points lie far outside the decision boundary.

As a first step, for every feature vector $x_{tk}$, the distance to the center is calculated by

$$
\text{dist}_{x_{tk}} = \| x_{tk} - a \| \quad (2.9)
$$

which is squared to be able to apply the RBF kernel

$$
\text{dist}^2_{x_{tk}} = \| x_{tk} - a \|^2 \quad (2.10)
$$

Solving the binomial, replacing $a$ by its linear combination of support vectors, and replacing the inner products by the RBF kernel function yields:

$$
\text{dist}^2_{x_{tk}} = 1 - 2 \sum_{i=1}^{M} \alpha_i K(x_{tk}, x_i) + \sum_{i,j=1}^{M} \alpha_i \alpha_j K(x_i, x_j) \quad (2.11)
$$

29
2 Anomaly detection in recordings from in-vehicle networks

Figure 2.2: Subsequence with window length 5 formed from a multivariate time series with $M = 2$. The highlighted feature vectors $x_{t_1}, \ldots, x_{t_5}$ belong to one subsequence.

The distance of a subsequence is now calculated by averaging the distances of the window’s feature vectors.

$$\text{dist}_{\text{subseq}} = \frac{1}{W} \sum_{k=1}^{W} \text{dist}_{x_k}$$

(2.12)

The proposed measure does not indicate the distance between two arbitrary subsequences, but indicates how abnormal a subsequence is. The formation of a subsequence is illustrated in 2.2 for a contrived multivariate time series containing two univariate time series.

2.4.3 Determining the classification threshold

Being able to calculate distances for subsequences allows to classify them. The procedure during training is as follows:

1. train SVDD with feature vectors in training set
2. calculate the distances $\text{dist}_{x_k}$ of the feature vectors
3. form subsequences of length $W$
2.5 Experimental results

4. calculate $dist_{subseq}$ for all subsequences

5. from all $dist_{subseq}$ determine a threshold $thr_{subseq}$

A first approach to determine the threshold $thr_{subseq}$ could be to use the maximum distance in the training set as the threshold for classifying subsequences. However, this is highly sensitive to outliers in the training set since the threshold would be determined solely by the most distant subsequence.

It is proposed to not necessarily include all subsequences in the determination of the threshold, and thereby be robust against outliers. The threshold is determined using box plots known from statistics (see e.g. [7]). For a box plot the first and the third quartile ($Q_1$ and $Q_3$) of the data are calculated. The margin between $Q_1$ and $Q_3$ is referred to as the inter-quartile range, which holds 50% of the data. Based on the inter-quartile range, the so-called whiskers are calculated by $Q_3 + 1.5(Q_3 - Q_1)$ and $Q_1 - 1.5(Q_3 - Q_1)$. The data points outside the whiskers are regarded as outliers.

In this work, outlier distances are the ones that are greater than the upper whisker. Those distances are discarded according to

$$dist_{outlier} > 1.5(Q_3 - Q_1) + Q_3$$

The maximum of the remaining distances is used as the threshold for classification.

2.5 Experimental results

The approach was validated on data sets from a real vehicle. Test drives were conducted in different traffic situations ranging from urban traffic to motorways over a time span of one year to capture recordings from different weather conditions. Different types of faults were injected into the vehicle. Preliminary results were previously presented in [14].

It is recommended to partition the data according to the vehicle’s subsystems like e.g. the engine control. In a first step, the relevant signals were selected using visual analytics as proposed by this paper’s author in [12]. Eight signals were taken into account, e.g. engine
Figure 2.3: False negatives for varied size of training set and fixed size of test set (2423 seconds) with normal data only.

rpm, vehicle speed, ignition timing advance, and the throttle position. The majority of signals on an in-vehicle network are transmitted in a cyclic manner with typical cycle times in the range of 20ms to 1s. In a pre-processing step, the data was resampled to a sample rate of 1s. In the absence of abnormal data it is recommended to start by testing with normal data. If the number of false negatives (FN), i.e. falsely detected anomalies, is too high, the detection system will not be useful.

The size of the training set was varied with a fixed test set size as shown in 2.3. While for very small training sets the number of false negatives acts non-deterministically between very low and very high values, for larger training sets, the training set becomes more representative and the number of false negatives stabilises at low values. This type of experiment can be used as an indicator of how representative the training set is.

The approach is now tested with recordings from different driving conditions. The size of the test sets used for the experiments is relatively small due to the availability of labelled data. Since classification involves basic vector algebra in combination with the RBF kernel
2.5 Experimental results

it is rather fast. Even for large vehicle fleets, classification is orders of magnitude faster than the production of the data. So test data of the size that one would refer to as “big data” can easily be classified sufficiently fast.

As a first step, the system was trained and tested with recordings from one driving condition, i.e. motorway, overland, or urban traffic. The results are given in the first three rows in 2.1.

As can be seen, between 42.9% and 76.9% of the faults were detected (TNR). For the subsequent experiments the system was trained on the recordings from all driving conditions. The last four rows in 2.1 show the results for these experiments. With the combined training set, the number of falsely reported anomalies per hour has significantly decreased compared to the experiments with individual training sets shown in the first three rows, since the training set has become more representative.

<table>
<thead>
<tr>
<th>training.test</th>
<th>training set</th>
<th>test set</th>
<th>FN/h</th>
<th>TN</th>
<th>TNR</th>
<th>precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>motorway,motorway</td>
<td>20843s</td>
<td>4845s</td>
<td>6.7</td>
<td>9</td>
<td>42.9%</td>
<td>50.0%</td>
</tr>
<tr>
<td>overland,overland</td>
<td>24604s</td>
<td>12076s</td>
<td>4.8</td>
<td>31</td>
<td>73.8%</td>
<td>66.0%</td>
</tr>
<tr>
<td>urban,urban</td>
<td>21336s</td>
<td>7224s</td>
<td>10.5</td>
<td>10</td>
<td>76.9%</td>
<td>32.3%</td>
</tr>
<tr>
<td>all,motorway</td>
<td>63631s</td>
<td>4845s</td>
<td>0.0</td>
<td>10</td>
<td>47.6%</td>
<td>100%</td>
</tr>
<tr>
<td>all,overland</td>
<td>63631s</td>
<td>12076s</td>
<td>3.0</td>
<td>27</td>
<td>64.3%</td>
<td>73.0%</td>
</tr>
<tr>
<td>all,urban</td>
<td>63631s</td>
<td>7224s</td>
<td>2.0</td>
<td>9</td>
<td>69.2%</td>
<td>69.2%</td>
</tr>
<tr>
<td>all,all</td>
<td>63631s</td>
<td>24145s</td>
<td>2.1</td>
<td>45</td>
<td>59.2%</td>
<td>76.3%</td>
</tr>
</tbody>
</table>

Table 2.1: Results for motorway, overland, and urban test drives. (FN/h: falsely reported anomalies per hour test drive, TN: correctly detected anomalies, TNR: true negative rate, precision: percentage of correctly detected anomalies in the set of reported anomalies)

In 2.4 the results on a one hour test drive are shown, where the spark plug lead was temporarily disconnected while the vehicle was standing still. From the 10 injected faults, 8 were detected. None of the signals is out of the valid value range, the faults were detected solely due to violations of learnt relationships between signals. No anomalies were falsely reported for this recording.

The percentage of detected anomalies is reasonably high, taking into account that classifica-
2 Anomaly detection in recordings from in-vehicle networks

Figure 2.4: Classification results for an overland drive of one hour where 10 faults were injected by temporarily interrupting the spark plug lead. The results are marked with frames (TN: green, FP: yellow) in the “class” row.

Detection is done solely on the information of the normal class. The percentage of correctly detected anomalies in the set of reported anomalies is high as well, which means that a domain-expert analysing the output of the classification system will not have to spend a large amount of time for fault analysis of reported anomalies that turn out to be normal occurrences.

2.6 Conclusion

This paper addressed the problem of having to cope with big data resulting from vehicle tests. The aim was to report potential errors in the recordings. The key point was to be able to detect unexpected faults without modelling effort. This was achieved by learning from a training set of error-free recordings, and then autonomously reporting deviations in the test
2.6 Conclusion

set as anomalies.
The one-class support vector machine SVDD was enhanced to work on multivariate time series data and the effectiveness of the approach was shown on real data sets.
The proposed methodology could be applicable to multivariate time series from other domains as well, e.g. industrial plants or network traffic with fixed communication patterns.
Bibliography


3 Network traffic analysis by means of Misleading Generalized Itemsets*

Daniele Apiletti, Elena Baralis, Luca Caglier, Tania Cerquitelli, Silvia Chiusano, Paolo Garza, Luigi Grimaudo and Fabio Pulvirenti
Dipartimento di Automatica e Informatica, Politecnico di Torino
{ daniele.apiletti, elena.baralis, luca.caglier, tania.cerquitelli, silvia.chiusano, paolo.garza, luigi.grimaudo, fabio.pulvirenti }@polito.it

Abstract: In the last ten years, with the explosion of the usage of Internet, network traffic analytics and data mining issues have taken primary importance. Generalized itemset mining is an established data mining technique which allows us to discover multiple-level correlations among data equipped with analyst-provided taxonomies. In this work, we address the discovery of a specific type of generalized itemsets, named misleading generalized itemsets (MGIs), which can be used to highlight anomalous situations in potentially large datasets. More specifically, MGIs are high-level patterns with a contrasting correlation type with respect to those of many of their descendant patterns according to the input taxonomy. This work proposes a new framework, named MGI-Cloud, which is able to efficiently extract misleading generalized itemsets. The framework is characterized by a distributed architecture and it is composed by a set of MapReduce jobs. As reference case study, MGI-Cloud has been applied to real network datasets, captured in different stages from a backbone link of an Italian ISP. The experiments demonstrate the effectiveness of our approach in a real-life scenario.

(*)This research was supported in part by project ONTIC-619633 funded by the European Commission under the Seventh Framework Programme.
3 Network traffic analysis by means of Misleading Generalized Itemsets

Keywords: Generalized itemset mining, cloud-based service; network traffic analysis

3.1 Introduction

In the last years we have witnessed an increase of the capability of modern applications (e.g. computer networks, social networks, wireless sensor applications) of generating and collecting information and data that rapidly scale towards "Big Data". Data mining aims at studying effective and efficient algorithms to transform huge amounts of data into interesting and useful knowledge. The interest in data mining is continuously growing both in manufacturing and in research domains: while researchers are mainly interested in proposing new data mining algorithms and tools, companies aim at exploiting the extracted knowledge for creating new business opportunities.

Data mining from Big Data is considered as a challenging task because of its inherent computational cost, which is prohibitive on non-distributed systems. For example, itemset and association rule mining algorithms find application in a wide range of different domains, including medical images [1], biological data [6], and network traffic data [3]. In many research contexts algorithm scalability issues are crucial and become challenging when the considered datasets become larger and larger. Hence, large-scale itemset and association rule mining on MapReduce [7] paradigm have become appealing research topics. In this context, some remarkable attempts to propose itemset and association rule mining algorithms which are able to successfully cope with Big Data have recently been made (e.g., [11, 13, 2]).

This paper focuses on generalized itemset mining [15]. Such a problem is an extension of the traditional itemset mining task, which specifically addresses the analysis of data equipped with analyst-provided taxonomies (i.e., is-a hierarchies). The aim of this work is to discover multiple-level and actionable patterns, called Misleading Generalized Itemsets (MGIs) [5], which represent misleading and thus interesting situations. MGIs are extracted on top of frequent generalized itemsets, which represent co-occurrences among data items at different abstraction levels whose frequency of occurrence in the source dataset is above a given (user-provided) threshold. Generalized itemsets can be characterized by a set of descendant itemsets, which consist of sets of items at lower granularity levels according to the input tax-
3.2 Preliminary concepts and problem statement

A relational dataset \( \mathcal{D} \) consists of a set of records, where each record is a set of items [14]. Each item is a pair (attribute, value). A taxonomy \( \Gamma \) built over the source dataset \( \mathcal{D} \) aggregates the data items into higher-level concepts (i.e., the generalized items). Table 3.1 and Figure 3.2 report two representative examples of relational dataset and taxonomy, respec-
### 3 Network traffic analysis by means of Misleading Generalized Itemsets

<table>
<thead>
<tr>
<th>Rid</th>
<th>Port Number</th>
<th>RTT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Port 80</td>
<td>[0 - 50 ms]</td>
</tr>
<tr>
<td>2</td>
<td>Port 80</td>
<td>[0 - 50 ms]</td>
</tr>
<tr>
<td>3</td>
<td>Port 443</td>
<td>[0 - 50 ms]</td>
</tr>
<tr>
<td>4</td>
<td>Port 2009</td>
<td>[50 - 100 ms]</td>
</tr>
<tr>
<td>5</td>
<td>Port 2009</td>
<td>[50 - 100 ms]</td>
</tr>
<tr>
<td>6</td>
<td>Port 53066</td>
<td>[150 - 200 ms]</td>
</tr>
<tr>
<td>7</td>
<td>Port 80</td>
<td>[0 - 50 ms]</td>
</tr>
</tbody>
</table>

**Table 3.1:** Example dataset $\mathcal{D}$ after discretization.

Table 3.2: Example taxonomy built over items in $\mathcal{D}$

<table>
<thead>
<tr>
<th>RTT 0 - 100 ms</th>
<th>RTT 100 - 200 ms</th>
<th>RTT &gt; 200 ms</th>
</tr>
</thead>
<tbody>
<tr>
<td>RTT 0 - 50 ms</td>
<td>RTT 50 - 100 ms</td>
<td>RTT 150 - 200 ms</td>
</tr>
</tbody>
</table>

A $k$-itemset is a set of $k$ (generalized) items. For example, \{(RTT, 0-50 ms), (Port 80)\} is a 2-itemset, which indicates that the two items co-occur (possibly at different abstraction levels) in the source data. Items/itemsets are characterized by many notable properties [9], such as support, coverage, descent and level of abstraction according to an input taxonomy.

For their definitions please refer to [15, 10, 5]. Similar to [10, 4], we target the correlations among items at same abstraction level, i.e. the itemsets that exclusively contain items with the same level. Such patterns are denoted by *level-sharing itemsets* [10].

The itemset correlation measures the strength of the correlation between its items. Similar to [4], in this paper we evaluate the correlation of a $k$-itemset $I$ by means of the Kulczynsky
3.2 Preliminary concepts and problem statement

<table>
<thead>
<tr>
<th>Frequent itemset (level≥2) [correlation type (Kulc value)]</th>
<th>Frequent descendants [correlation type (Kulc value)]</th>
<th>NOD (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>{ (RTT, 0 - 100 ms), (PortNumber, Registered) } [positive (5/6=0.83)]</td>
<td>{ (RTT, [0 - 50 ms], (Port Number, Port 80)) [positive (7/8=0.88)] }</td>
<td>75</td>
</tr>
<tr>
<td>{ (RTT, 0 - 100 ms), (Port Number, Unknown) } [negative (1/2=0.50)]</td>
<td>{ (RTT, [50 - 100 ms], (Port Number, Port 2009)) [positive (1)] }</td>
<td>0</td>
</tr>
<tr>
<td>{ (RTT, 100 - 200 ms), (Port Number, Unknown) } [negative (2/3=0.66)]</td>
<td>{ (RTT, [150 - 200 ms], (Port Number, Port 53066)) [positive (1)] }</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 3.3: Misleading Generalized Itemsets mined from $D$. $\min \text{sup} = 10\%$, $\max \text{neg} \text{cor}=0.70$, $\min \text{pos} \text{cor}=0.80$, and $\max \text{NOD} = 80\%$.

(Kulc) correlation measure [16] Kulc values range from 0 to 1. By properly setting maximum negative and minimum positive Kulc thresholds, hereafter denoted by $\max \text{neg} \text{cor}$ and $\min \text{pos} \text{cor}$, the itemsets may be classified as negatively correlated, uncorrelated, or positively correlated itemsets according to their correlation value.

Let $\mathcal{L}\mathcal{S}\mathcal{I}$ be the set of all frequent level-sharing itemsets in $D$ according to a minimum support threshold $\min \text{sup}$. Given a frequent level-sharing itemset $X \in \mathcal{L}\mathcal{S}\mathcal{I}$ of level $l \geq 2$, let $\text{Desc}^* [X, \Gamma]$ be the subset of corresponding level-$(l-1)$ X’s descendants for which the correlation type is in contrast to those of $X$. A Misleading Generalized Itemset (MGI) is a pattern in the form $X \triangleright E$, where $X \in \mathcal{L}\mathcal{S}\mathcal{I}$ and $E=\text{Desc}^* [X, \Gamma]$ [5].

For example, by enforcing $\min \text{sup}=10\%$, $\max \text{neg} \text{cor}=0.70$, and $\min \text{pos} \text{cor}=0.80$, MGI $\{(\text{RTT}, [0 - 100 ms]), (\text{Port Number}, \text{Registered})\} \triangleright \{(\text{RTT}, [0 - 50 ms], (\text{Port Number, Port 80}))\}$ is mined from the dataset in Table 1, because $\{(\text{RTT}, [0 - 100 ms]), (\text{Port Number, Registered})\}$ has a positive correlation (0.83), whereas its descendant itemset $\{(\text{RTT}, [0 - 50 ms]) (\text{Port Number, Port 443})\}$ is negatively correlated (0.63).

To measure the degree of interest of a MGI $X \triangleright E$ with respect to its corresponding traditional itemset version ($X$), the Not Overlapping Degree (NOD) measure has been defined.
in [5]. The NOD of an MGI $X \triangleright E$ is defined as $\frac{\sup(X, D) - \cov(E, D)}{\sup(X, D)}$. It expresses the relative difference between the support of the ancestor itemset $X$ and the coverage of its low-level contrasting correlations in $E$. The NOD values range from 0 to 1. The lower NOD value we achieve, the more significant the degree of overlapping between the contrasting low-level correlations in $E$ and their common ancestor $X$ becomes.

The mining task addressed by this paper entails discovering from $D$ all the MGIs for which the NOD value is less than or equal to a maximum threshold $\text{max}_\text{NOD}$. The subset of Misleading Generalized Itemsets mined from Table 3.1 by setting the maximum NOD threshold to 80% is reported in Table 3.3.

3.3 The MGI-Cloud architecture

The MGI-Cloud architecture provides a cloud-based service for discovering hidden and actionable patterns among potentially Big datasets. We focus our analysis on a specific case study, i.e., the analysis of the Internet traffic generated by an Italian ISP. To efficiently cope with Big Data, the system implementation is distributed and most operations are mapped to the MapReduce programming paradigm [7]. The architecture has been designed as a chain of distributed jobs running on an Hadoop cluster, as described below.

3.3.1 Data retrieval and preparation

The first step to analyse network traffic is is collecting network measurements. To this aim, a passive probe is located on the access link (vantage point) that connects an edge network to the Internet. The passive probe sniffs all incoming and outgoing packets flowing on the link, i.e., packets directed to a node inside the network and generated by a node in the Internet, and vice versa. The probe runs Tstat [8], [12], a passive monitoring tool allowing network and transport layer measurement collection. Tstat rebuilds each TCP connection by matching incoming and outgoing segments. Thus, a flow-level analysis can be performed [12]. A TCP flow is identified by snooping the signaling flags (SYN, FIN, RST). The status of the TCP sender is rebuilt by matching sequence numbers on data segments with the corresponding ac-
3.3 The MGI-Cloud architecture

knowledge (ACK) numbers. To evaluate the MGI-Cloud tool in real-world application, we focus on a subset of measurements describing the traffic flow among the many provided by Tstat. The most meaningful features, selected with the support of domain experts, are detailed in the following:

- the Round-Trip-Time (RTT) observed on a TCP flow, i.e., the minimum time lag between the observation of a TCP segment and the observation of the corresponding ACK. RTT is strongly related to the distance between the two nodes.
- the number of hops (Hop) from the remote node to the vantage point observed on packets belonging to the TCP flow, as computed by reconstructing the IP Time-To-Live
- the flow reordering probability (Preord), which can be useful to distinguish different paths
- the flow duplicate probability (Pdup), that can highlight a destination served by multiple paths
- the total number of packets (NumPkt), the total number of data packets (DataPkt), and the total number of bytes (DataBytes) sent from both the client and the server, separately (the client is the host starting the TCP flow)
- the minimum (WinMin), maximum (WinMax), and scale (WinScale) values of the TCP congestion window for both the client and the server, separately
- the TCP port of the server (Port)
- the class of service (Class), as defined by Tstat, e.g., HTTP, video, VoIP, SMTP, etc.

Based on measurements listed above, an input data record is defined by the following features: RTT, Hop, Preord, Pdup, NumPkt, DataPkt, DataBytes, WinMax, WinMin, WinScale, Port, Class. To obtain reliable estimates on reordering and duplicate probabilities, only TCP flows which last more than $P = 10$ packets are considered. This choice allow focusing the
3 Network traffic analysis by means of Misleading Generalized Itemsets*

analysis on long-lived flows, where the network path has a more relevant impact, thus providing more valuable information.

Since frequent itemset mining requires a transactional dataset of categorical values, data has to be discretized before the mining. The discretization step converts continuously valued measurements into categorical bins. Then, data are converted from the tabular to the transactional format. Both the value discretization and the transactional format conversion are performed by a single map only job. Each record is processed by the map function and, if the number of packets is above the threshold (10 packets), the corresponding discretized transaction is emitted as a result of the mapping. This task entails an inherently parallel elaboration, considering that can be applied independently to each record.

3.3.2 Taxonomy generation

To analyze data from a high-level viewpoint, real infraction datasets are equipped with taxonomies. A taxonomy is a set of is-a hierarchies built over data items in $D$. An example taxonomy built over the dataset in Table 1 is depicted in Table 2. Items whose value is an high-level aggregation belonging to the taxonomy (e.g., (RTT, [0 - 100 ms]) are called generalized items. Analyst-provided taxonomies could be generated either manually or semiautomatically by domain experts. To perform our analyzes, we built hierarchies of discretization steps. There were few cases where it was not possible: for instance, Protocols attributes have been grouped in classes based on use case domains (similar to [9]).

3.3.3 Level-sharing itemset mining

Given a preprocessed infraction dataset and a minimum support threshold $\minsup$, this job accomplishes the first MGI mining step, i.e., the extraction of all frequent level-sharing itemsets [10]. This job performs the following tasks.

Dataset extension. This task entails producing a new dataset version which integrates taxonomy information. To enable frequent level-sharing itemset mining from data containing items at different abstraction levels, it generates multiple copies of each record, one for each taxonomy level. While the original record contains only taxonomy leaves (i.e., the dataset
items), each copy contains the corresponding combination of item generalizations at a different abstraction level. To avoid unnecessary I/O operations, the extended dataset version is not materialized on disk, but it is directly generated in the map function of the itemset extraction task and then immediately stored into a compact FP-tree structure [14].

**Itemset extraction.** To efficiently mine frequent level-sharing itemsets [10] from the extended dataset version, this task exploits a variation of the Hadoop-based itemset mining algorithm proposed in [2].

### 3.3.4 MGI extraction

This job performs MGI mining on top of the frequent level-sharing itemsets. Specifically, it accomplishes the task stated in Section 3.2. This step consists of a MapReduce job, as described in the following. The contribution of this job is new because, to the best of our knowledge, no cloud-based service currently supports MGI mining from Big Data.

To extract MGIs we combine each frequent level-sharing itemset $I$ with its corresponding set of descendant itemsets $\text{Desc}[I, \Gamma]$. More specifically, In the map function for each level-sharing itemset $I$, the following two pairs ($key$, $value$) are emitted: (i) a pair ($key$, $value$), where $key$ is the direct ancestor of itemset $I$ and $value$ the itemset $I$ with its main properties (i.e., support and Kulc values) and (ii) a pair ($key$, $value$), where $key$ is the itemset $I$ is the value: itemset $I$ with its main properties (i.e., support and Kulc values). Two pairs are emitted because each itemset can be a descendant of an itemset and a parent of another one at the same time. The first pair allows us to associate $I$ with the ancestor key, whereas the second pair is used to associate $I$ to itself if MGIs in the form $I \triangleright \mathcal{E}$ are extracted. The generated pairs allow us to map each itemset and its corresponding descendants to the same key. Hence, in the reduce function, each key is associated with a specific itemset $I$ and the corresponding set of values contains both the (ancestor) itemset $I$ and its respective descendants. By iterating on the set of values associated with key $I$, we generate candidate MGIs $I \triangleright \mathcal{E}$, where $\mathcal{E}$ is the set of $I$’s descendants in contrast to $I$ in terms of correlation type, and we compute the corresponding NOD values. Finally, only the MGIs satisfying the max_NOD threshold are stored into the HDFS file system.
3 Network traffic analysis by means of Misleading Generalized Itemsets

3.4 Experiments

We performed experiments on the BigNetData dataset. This relational network traffic dataset was obtained by performing different capture stages on a backbone link of a nation-wide ISP in Italy that offers us three different vantage points. ISP vantage points expose traffic of three different Points-of-Presence (POP) in different cities in Italy; each PoP aggregates traffic from more than 10,000 ISP customers, which range from home users to Small Office Home Office (SOHO) accessing the Internet via ADSL or Fiber-To-The-Home technology. It represents therefore a very heterogeneous scenario. The dataset has size 192.56 GB and it consists of 413,012,989 records, i.e., one record for each bi-directional TCP flow.

The MapReduce jobs of the MGI-Cloud workflow (see Section 3.3) were developed in Java using the new Hadoop Java APIs. The experiments were performed on a cluster of 5 nodes running Cloudera’s Distribution of Apache Hadoop (CDH4.5). Each cluster node is a 2.67 GHz six-core Intel(R) Xeon(R) X5650 machine with 32 Gbyte of main memory running Ubuntu 12.04 server with the 3.5.0-23-generic kernel. All the reported execution times are real times obtained from the Cloudera Manager web control panel.

3.4.1 Result validation

We examined the MGIs extracted from the BigNetData dataset to validate their interestingness and usefulness in a real-life context, i.e., the analysis of the Internet traffic generated by an Italian ISP. We focused our analysis on the pattern related to either protocols or RTT values, because we deemed such patterns as interesting to understand application/service server geography.

As an example let use consider the following MGI extracted by enforcing $\max_{\text{neg\_cor}}=0.2$, $\min_{\text{pos\_cor}}=0.3$, and $\max_{\text{NOD}}=70\%$:

$\{(\text{CLASS}=\text{CHAT}) \text{ (RTT\_MIN}=100-200)\} \Rightarrow \{(\text{CLASS}=32) \text{ (RTT\_MIN}=165-170), \text{ (CLASS}=513) \text{ (RTT\_MIN}=145-150)\}$. The high-level itemset $\{(\text{CLASS}=\text{CHAT}) \text{ (RTT\_MIN}=100-200)\}$ is negatively correlated whereas its frequent descendants $\{(\text{CLASS}=32) \text{ (RTT\_MIN}=165-170), \text{ (CLASS}=513) \text{ (RTT\_MIN}=145-150)\}$ are positively correlated and they cover a significant
portion of data already covered by the high-level itemset (especially CLASS=32, with the 32%). This means that the traffic flows associated with any chat protocol and characterized by RTT between 100 and 200 ms are less likely to occur than expected, whereas the flows associated with two specific chat protocols, i.e., MSN (class 32) and Skype (class 513), are likely to have RTTs in the ranges 165-170 ms and 145-150 ms, respectively. Hence, in this case, analyzing only the high-level itemset instead of the complete MGI could be misleading and the pattern may indicate that only some specific chat protocols (MSN, Skype) often rely on servers physically located relatively far away with each other. In this specific example, MGI analysis proves its effectiveness in the network environment, i.e. helping network administrator to understand and optimize networks and identify anomalous situations. Nevertheless, there are many other possible use cases because of the generality of our approach and its compatibility with huge datasets due to its distributed architecture.

3.4.2 Scalability with the number of cluster nodes

We evaluated the scalability of the proposed architecture by measuring the speedup achieved increasing the number of Hadoop cluster nodes. Specifically, we considered three configurations: 1 node, 3 nodes, and 5 nodes. Figure 3.1 reports the speedup achieved setting min_sup to 1%, max_neg_cor to 0.1, min_pos_cor to 0.3, and max_nod to 60%. The first box in Figure 3.1 (i.e., 1 node) corresponds to a run of MGI-Cloud on a single node. Speedup with increasing nodes is computed against the single-node performance. The achieved results show that our approach scales roughly linearly with the number of nodes and the speedup approximately corresponds to the number of cluster nodes.

3.5 Conclusions and future perspectives

This paper presents a cloud-based service for discovering Misleading Generalized Itemsets from Big Data equipped with taxonomies. To cope with Big Data the architecture runs on an Hadoop architecture. The usefulness of the proposed architecture was conducted on real network environment. However, the offered service could find application in many other
Network traffic analysis by means of Misleading Generalized Itemsets

Figure 3.1: Speedup on the BigNetData dataset.

application contexts. As future work, we aim at extending the current architecture as well as testing its applicability in other contexts (e.g., social network analysis).
Bibliography


BIBLIOGRAPHY


4 Unsupervised Detection of Network Attacks in the dark

Philippe Owezarski\textsuperscript{1,2}, Pedro Casas\textsuperscript{3} and Johan Mazel\textsuperscript{4}

\textsuperscript{1}CNRS, LAAS, 7 avenue du colonel Roche, F-31400 Toulouse, France
\textsuperscript{2}Univ de Toulouse, LAAS; F-31400 Toulouse, France
\textsuperscript{3}FTW-Telecommunication Research Center, Vienna, Austria
\textsuperscript{4}NII-JFLI, Tokyo, Japan

\{owe@laas.fr\}, \{casas@ftw.at\}, \{johanmazel@nii.ac.jp\}

Abstract: The unsupervised detection of network attacks represents an extremely challenging goal. Current methods rely on either very specialized signatures of previously seen attacks, or on expensive and difficult to produce labeled traffic data-sets for profiling and training. In this paper we present a completely unsupervised approach to detect attacks, without relying on signatures, labeled traffic, or training. The method uses robust clustering techniques to detect anomalous traffic flows. The structure of the anomaly identified by the clustering algorithms is used to automatically construct specific filtering rules that characterize its nature, providing easy-to-interpret information to the network operator. In addition, these rules are combined to create an anomaly signature, which can be directly exported towards standard security devices like IDSs, IPSs, and/or Firewalls. The clustering algorithms are highly adapted for parallel computation, which permits to perform the unsupervised detection and construction of signatures in an on-line basis. We evaluate the performance of this new approach to discover and to build signatures for different network attacks without any previous knowledge, using real traffic traces.

Keywords: Anomaly Detection & Characterization, Robust Clustering, Automatic Generation of Signatures, Autonomous Security.
4 Unsupervised Detection of Network Attacks in the dark

4.1 Introduction

The detection of network attacks is a paramount task for network operators in today’s Internet. Denial of Service attacks (DoS), Distributed DoS (DDoS), network/host scans, and spreading worms or viruses are examples of the different attacks that daily threaten the integrity and normal operation of the network. The principal challenge in automatically detecting and analyzing network attacks is that these are a moving and ever-growing target.

Two different approaches are by far dominant in the literature and commercial security devices: signature-based detection and anomaly detection. Signature-based detection systems are highly effective to detect those attacks which they are programmed to alert on. However, they cannot defend the network against unknown attacks. Even more, building new signatures is expensive and time-consuming, as it involves manual inspection by human experts. Anomaly detection uses labeled data to build normal-operation-traffic profiles, detecting anomalies as activities that deviate from this baseline. Such methods can detect new kinds of network attacks not seen before. Nevertheless, anomaly detection requires training to construct normal-operation profiles, which is time-consuming and depends on the availability of purely anomaly-free traffic data-sets. In addition, it is not easy to maintain an accurate and up-to-date normal-operation profile.

In this paper we present a completely unsupervised method to detect and characterize network attacks, without relying on signatures, training, or labeled traffic of any kind. Our approach relies on robust clustering algorithms to detect both well-known as well as completely unknown attacks, and to automatically produce easy-to-interpret signatures to characterize them, both in an on-line basis. The analysis is performed on packet-level traffic, captured in consecutive time slots of fixed length $\Delta T$ and aggregated in IP flows (standard 5-tuples). IP flows are additionally aggregated at 9 different flow levels $l_i$. These include (from finer to coarser-grained resolution): \textit{source IPs} ($l_1$: IPsrc), \textit{destination IPs} ($l_2$: IPdst), \textit{source Network Prefixes} ($l_{3,4,5}$: IPsrc/24, /16, /8), \textit{destination Network Prefixes} ($l_{6,7,8}$: IPdst/24, /16, /8), and \textit{traffic per Time Slot} ($l_9$: tpTS).

The complete detection and characterization algorithm runs in three successive stages. The first step consists in detecting an anomalous time slot where an attack might be hidden. For
doing so, time series $Z_t^l$ are built for basic traffic metrics such as number of bytes, packets, and IP flows per time slot, using the 9 flow resolutions $l_{1..9}$. Any generic anomaly-detection algorithm $F(.)$ based on time-series analysis [1, 2, 3, 4, 5] is then used on $Z_t^l$ to identify an anomalous slot. Time slot $t_0$ is flagged as anomalous if $F(Z_{t_0}^l)$ triggers an alarm for any of the $l_t$ flow aggregation levels. Tracking anomalies at multiple aggregation levels provides additional reliability to the anomaly detector, and permits to detect both single source-destination and distributed attacks of very different intensities.

The unsupervised detection and characterization algorithm begins in the second stage, using as input the set of IP flows captured in the flagged time slot. The method uses robust clustering techniques based on Sub-Space Clustering (SSC) [11], Density-based Clustering [10], and Evidence Accumulation (EA) [12] to blindly extract the suspicious flows that compose the attack. In the third stage, the evidence of traffic structure provided by the clustering algorithms is used to produce filtering rules that characterize the detected attack and simplify its analysis. The characterization of an attack can be a hard and time-consuming task, particularly when dealing with unknown attacks. Even expert operators can be quickly overwhelmed if simple and easy-to-interpret information is not provided to prioritize the time spent in the analysis. To alleviate this issue, the most relevant filtering rules are combined into a new traffic signature that characterizes the attack in simple terms. This signature can ultimately be integrated to any standard security device to detect the attack in the future, which constitutes a major step towards autonomous security.

The remainder of the paper is organized as follows. Section 4.2 presents a short state of the art in the unsupervised anomaly detection field and describes our main contributions. Section 4.3 briefly describes the unsupervised detection algorithm that we have developed. Section 4.4 presents the automatic characterization algorithm, which builds easy-to-interpret signatures for the detected attacks. Section 4.5 presents the validation of our proposals, discovering and characterizing single source/destination and distributed network attacks in traffic traces from an operational backbone network. Section 4.6 evaluates the computational time of the unsupervised detection approach, considering the parallelization of the clustering algorithms. Finally, section 4.7 concludes the paper.
4 Unsupervised Detection of Network Attacks in the dark

4.2 Related Work & Contributions

The problem of network attacks and anomaly detection has been extensively studied in the last decade. Most approaches analyze statistical variations of traffic volume-metrics (e.g., number of bytes, packets, or flows) and/or other traffic features (e.g., distribution of IP addresses and ports), using either single-link measurements or network-wide data. A non-exhaustive list of methods includes the use of signal processing techniques (e.g., ARIMA, wavelets) on single-link traffic measurements [1, 2], PCA [7, 8] and Kalman filters [4] for network-wide anomaly detection, and sketches applied to IP-flows [3, 6].

Our approach falls within the unsupervised anomaly detection domain. Most work has been devoted to the Intrusion Detection field, targeting the well known KDD’99 data-set. The vast majority of the unsupervised detection schemes proposed in the literature are based on clustering and outliers detection, being [15, 16, 17] some relevant examples.

Our unsupervised algorithm has several advantages w.r.t. the state of the art: (i) first and most important, it works in a completely unsupervised fashion, which means that it can be directly plugged-in to any monitoring system and start to work from scratch, without any kind of calibration or previous knowledge. (ii) It combines robust clustering techniques to avoid general clustering problems such as sensitivity to initialization, specification of number of clusters, or structure-masking by irrelevant features. (iii) It automatically builds compact and easy-to-interpret signatures to characterize attacks, which can be directly integrated into any traditional security device. (iv) It is designed to work on-line, using the parallel structure of the proposed clustering approach.

4.3 Unsupervised Detection of Attacks

The unsupervised detection stage takes as input all the IP flows in the anomalous time slot, aggregated according to one of the different aggregation levels used in the first stage. Let \( Y = \{y_1, ..., y_n\} \) be the set of \( n \) flows in the flagged time slot. Each flow \( y_i \in Y \) is described by a set of \( m \) traffic attributes or features on which the analysis is performed. The selection of these features is a key issue to any anomaly detection algorithm, and it becomes critical.
4.3 Unsupervised Detection of Attacks

in the case of unsupervised detection, because there is no additional information to select the most relevant set. In this paper we shall limit our study to detect and characterize well-known attacks, using a set of standard traffic features widely used in the literature. However, the reader should note that the approach can be easily extended to detect other types of attacks, considering different sets of traffic features. In fact, more features can be added to any standard list to improve detection and characterization results.

The set that we shall use here includes the following $m = 9$ traffic features: number of source/destination IP addresses and ports, ratio of number of sources to number of destinations, packet rate, ratio of packets to number of destinations, and fraction of ICMP and SYN packets. According to previous work on signature-based anomaly characterization [9], such simple traffic descriptors permit to describe standard network attacks such as DoS, DDoS, scans, and spreading worms/virus. Let $x_i = (x_i(1),...,x_i(m)) \in \mathbb{R}^m$ be the corresponding vector of traffic features describing flow $y_i$, and $X = \{x_1,...,x_n\}$ the complete matrix of features, refereed to as the feature space.

The algorithm is based on clustering techniques applied to $X$. The objective of clustering is to partition a set of unlabeled elements into homogeneous groups of similar characteristics, based on some measure of similarity. Our goal is to identify in $Y$ the different aggregated flows that may compose the attack. For doing so, the reader should note that an attack may consist of either outliers (i.e., single isolated flows) or compact small-size clusters, depending on the aggregation level of flows in $Y$. For example, a DDoS attack is represented as an outlier flow if the aggregation is done for IPdst, consisting of all the attacking IP flows sent towards the same victim. On the contrary, the attack is represented as a cluster if we use IPSrc flow-resolution. To avoid the lack of robustness of general clustering techniques, we have developed a parallel-multi-clustering approach, combining the notions of Density-based Clustering [10], Sub-Space Clustering [11], and Evidence Accumulation [12]. In what follows, we shall present the general idea behind the approach.

Instead of directly partitioning the complete feature space $X$ using a traditional inter-flow similarity measure (i.e., the Euclidean distance), we do parallel clustering in $N$ different sub-spaces $X_i \subset X$ of smaller dimensions, obtaining $N$ different partitions $P_i$ of the flows in $Y$. Each sub-space $X_i$ is constructed using only $r < m$ traffic features; this permits to analyze
the structure of $\mathbf{X}$ from $N(m, r)$ different perspectives, using a finer-grained resolution. In particular, we do clustering in very-low dimensional sub-spaces, using $r = 2$. To deeply explore the complete feature space, we analyze all the $r$-combinations-obtained-from-$m$ sub-spaces; hence, $N(m) = m(m - 1)/2$. The information provided by the multiple partitions $P_i$ is then combined to produce a new similarity measure between the flows in $\mathbf{Y}$, which has the paramount advantage of clearly highlighting both those outliers and small-size clusters that were simultaneously identified in different sub-spaces. This new similarity measure is finally used to easily extract the anomalous flows from the rest of the traffic.

4.4 Automatic Characterization of Attacks

The following task after the detection of a group of anomalous flows is to automatically produce a set of $K$ filtering rules $f_k(\mathbf{Y})$, $k = 1, \ldots, K$ to characterize them. In the one hand, such filtering rules provide useful insights on the nature of the anomaly, easing the analysis task of the network operator. On the other hand, different rules can be combined to construct a signature of the anomaly, which can be used to easily detect its occurrence in the future. To produce filtering rules $f_k(\mathbf{Y})$, the algorithm selects those sub-spaces $\mathbf{X}_i$ where the separation between the anomalous flows and the rest of the traffic is the biggest. We define two different classes of filtering rule: absolute rules $f_A(\mathbf{Y})$ and relative rules $f_R(\mathbf{Y})$. Absolute rules are only used in the characterization of small-size clusters, and correspond to the presence of dominant features in the flows of the anomalous cluster. An absolute rule for feature $j$ has the form $f_A(\mathbf{Y}) = \{y_i \in \mathbf{Y} : x_i(j) = \lambda\}$. For example, in the case of an ICMP flooding attack, the vast majority of the associated flows use only ICMP packets, hence the absolute filtering rule $\{nICMP/nPkts = 1\}$ makes sense ($nICMP/nPkts$ corresponds to the fraction of ICMP packets).

On the other hand, relative filtering rules depend on the relative separation between anomalous and normal-operation flows. Basically, if the anomalous flows are well separated from the rest of the traffic in a certain partition $P_i$, then the features of the corresponding sub-space $\mathbf{X}_i$ are good candidates to define a relative filtering rule. A relative rule defined for feature $j$ has the form $f_R(\mathbf{Y}) = \{y_i \in \mathbf{Y} : x_i(j) < \lambda$ or $x_i(j) > \lambda\}$. We shall also define a covering
4.5 Experimental Evaluation

relation between filtering rules: we say that rule \( f_1 \) covers rule \( f_2 \) \( \leftrightarrow f_2(Y) \subset f_1(Y) \). If two or more rules overlap (i.e., they are associated to the same feature), the algorithm keeps the one that covers the rest.

In order to construct a compact signature of the anomaly, we have to devise a procedure to select the most discriminant filtering rules. Absolute rules are important, because they define inherent characteristics of the anomaly. Regarding relatives rules, their relevance is directly tied to the degree of separation between flows. In the case of outliers, we select the \( K \) features for which the normalized distance to the normal-operation traffic (statistically represented by the biggest cluster in each sub-space) is among the top-\( K \) biggest distances. In the case of small-size clusters, we rank the degree of separation to the rest of the clusters using the well-known Fisher Score (FS) [14], and select the top-\( K \) ranked rules. The FS basically measures the separation between clusters, relative to the total variance within each cluster. To finally construct the signature, the absolute rules and the top-\( K \) relative rules are combined into a single inclusive predicate, using the covering relation in case of overlapping rules.

4.5 Experimental Evaluation

We evaluate the ability of the unsupervised algorithm to detect and to automatically construct a signature for different attacks in real traffic from the WIDE project data repository [18]. The WIDE network provides interconnection between different research institutions in Japan, as well as connection to different commercial ISPs and universities in the U.S.. Traffic consists of 15 minutes-long raw packet traces; the traces we shall work with consist of packets captured at one of the trans-pacific links between Japan and the U.S.. Traces are not labeled, thus our analysis will be limited to show how the unsupervised approach can detect and characterize different network attacks without using signatures, labels, or learning.

We shall begin by detecting and characterizing a distributed SYN network scan directed to many victim hosts under the same /16 destination network. Packets in \( Y \) are aggregated using IPdst/24 flow resolution, thus the attack is detected as a small-size cluster. The length of each time slot is \( \Delta T = 20 \) seconds. As we explained in section 4.3, the SSC-EA-based clustering algorithm constructs a new similarity measure between flows in \( Y \), using the multiple
clustering results obtained from the different sub-spaces. Let us express this new similarity measure as a $n \times n$ matrix $S$, in which element $S(i,j)$ represents the degree of similarity between flows $i$ and $j$. Figure 4.1.(a) depicts a histogram on the distribution of inter-flows similarity, according to $S$. The structure of flows in $Y$ provided by $S$ evidences the presence of a small isolated cluster in multiple sub-spaces. Selecting this cluster results in 53 anomalous IPdst/24 flows; a further analysis of the packets in these flows reveals multiple IP flows of SYN packets with the same IPsrc address and sequential IPdst addresses, scanning primary the same TCP port. Such a behavior is characteristic of a worm in the spreading phase.

Regarding filtering rules, figures 4.1.(b,c) depict some of the partitions $P_i$ where both absolute and top-$K$ relative rules were produced. These involve the number of sources and destinations, and the fraction of SYN packets. Combining them produces a signature that can be expressed as $(n_{Srcs} = 1) \land (n_{Dsts} > \lambda_1) \land (n_{SYN} / n_{Pkts} > \lambda_2)$, where both $\lambda_1$ and $\lambda_2$ are obtained by separating clusters at half distance. Surprisingly enough, the extracted signature matches quite closely the standard signature used to detect such an attack in current signature-based systems [9]. The beauty and main advantage of our unsupervised approach relies on the fact that this new signature has been produced without any previous information about the attack or baseline traffic, and now it can be directly exported towards any security device to rapidly detect the same attack in the future.

Figures 4.1.(d,e) depict different rules obtained in the detection of a SYN DDoS attack. IP flows are now aggregated according to IPsrc resolution. The distribution analysis of inter-flows similarity w.r.t. $S$ selects a compact cluster with the most similar flows, corresponding to the set of attacking hosts. The obtained signature can be expressed as $(n_{Dsts} = 1) \land (n_{SYN} / n_{Pkts} > \lambda_3) \land (n_{Pkts} / \text{sec} > \lambda_4)$, which combined with the large number of identified sources $(n_{Srcs} > \lambda_5)$ confirms the nature of a SYN DDoS attack. This signature is able to correctly isolate the most aggressive hosts of the DDoS attack, i.e., those with highest packet rate.

Figures 4.1.(f,g) depict the detection of an ICMP flooding DoS attack. Traffic is aggregated in IPdst flows, thus the attack is now detected as an outlier rather than as a small-size cluster. Absolute rules are not applicable in the case of outliers detection. Relative rules correspond to the separation of the outlier from the biggest cluster in each sub-space,
4.5 Experimental Evaluation

(a) Detecting a distributed SYN network scan using $S$.

(b) SYN Network Scan (1/2)

(c) SYN Network Scan (2/2)

(d) SYN DDoS (1/2)

(e) SYN DDoS (2/2)
4 Unsupervised Detection of Network Attacks in the dark

Figure 4.1: Filtering rules for characterization of attacks in WIDE.

which statistically represents normal-operation traffic. Besides showing typical characteristics of this attack, such as a high packet rate of exclusively ICMP packets from the same source host, both partitions show that the detected attack does not involve the largest elephant flows in the time slot. This emphasizes the ability of the algorithm to detect attacks that are not necessarily different from normal-operation traffic in terms of volume, but that they differ in other, less evident characteristics. The obtained signature can be expressed as
\[(\frac{nICMP}{nPkt} > \lambda_6) \land (nPkt/sec > \lambda_7)\].

4.6 Computational Time and Parallelization

The last issue that we analyze is the Computational Time (CT) of the algorithm. The SSC-EA-based algorithm performs multiple clusterings in \(N(m)\) low-dimensional sub-spaces \(X_i \subset X\). This multiple computation imposes scalability issues for on-line detection of attacks in very-high-speed networks. Two key features of the algorithm are exploited to reduce scalability problems in number of features \(m\) and the number of aggregated flows \(n\) to analyze. Firstly, clustering is performed in very-low-dimensional sub-spaces, \(X_i \in \mathbb{R}^2\), which is faster...
4.6 Computational Time and Parallelization

than clustering in high-dimensional spaces [13]. Secondly, each sub-space can be clustered independently of the other sub-spaces, which is perfectly adapted for parallel computing architectures. Parallelization can be achieved in different ways: We shall use the term ”slice” as a reference to a single computational entity.

Figure 4.2 depicts the CT of the SSC-EA-based algorithm, both (a) as a function of the number of features $m$ used to describe traffic flows and (b) as a function of the number of flows $n$ to analyze. Figure 4.2.(a) compares the CT obtained when clustering the complete feature space $X$, referred to as $\text{CT}(X)$, against the CT obtained with SSC, varying $m$ from 2 to 29 features. We analyze a large number of aggregated flows, $n = 10^4$, and use two different number of slices, $M = 40$ and $M = 100$. The analysis is done with traffic from the WIDE network, combining different traces to attain the desired number of flows. To estimate the CT of SSC for a given value of $m$ and $M$, we proceed as follows: first, we separately cluster each of the $N = m(m-1)/2$ sub-spaces $X_i$, and take the worst-case of the obtained clustering time as a representative measure of the CT in a single sub-space, i.e., $\text{CT}(X_{\text{SSCwc}}) = \max_i \text{CT}(X_i)$. Then, if $N \leq M$, we have enough slices to completely parallelize the SSC algorithm, and the total CT corresponds to the worst-case, $\text{CT}(X_{\text{SSCwc}})$. On the contrary, if $N > M$, some slices have to cluster various sub-spaces, one after the other, and the total CT becomes $(N\%M + 1)$ times the worst-case $\text{CT}(X_{\text{SSCwc}})$, where $\%$ represents integer division. The first interesting observation from figure 4.2.(a) regards the increase of $\text{CT}(X)$ when $m$ increases, going from about 8 seconds for $m = 2$ to more than 200 seconds for $m = 29$. As we said before, clustering in low-dimensional spaces is faster, which reduces the overhead of multiple clusterings computation. The second paramount observation is about parallelization: if the algorithm is implemented in a parallel computing architecture, it can be used to analyze large volumes of traffic using many traffic descriptors in an on-line basis; for example, if we use 20 traffic features and a parallel architecture with 100 slices, we can analyze 10000 aggregated flows in less than 20 seconds.

Figure 4.2.(b) compares $\text{CT}(X)$ against $\text{CT}(X_{\text{SSCwc}})$ for an increasing number of flows $n$ to analyze, using $m = 20$ traffic features and $M = N = 190$ slices (i.e., a completely parallelized implementation of the SSC-EA-based algorithm). As before, we can appreciate the difference in CT when clustering the complete feature space vs. using low-dimensional sub-
4 Unsupervised Detection of Network Attacks in the dark

Figure 4.2: Computational Time as a function of \( n^o \) of features and \( n^o \) of flows to analyze. The number of aggregated flows in (a) is \( n = 10000 \). The number of features and slices in (b) is \( m = 20 \) and \( M = 190 \) respectively.

spaces: the difference is more than one order of magnitude, independently of the number of flows to analyze. Regarding the volume of traffic that can be analyzed with this 100% parallel configuration, the SSC-EA-based algorithm can analyze up to 50000 flows with a reasonable CT, about 4 minutes in this experience. In the presented evaluations, the number of aggregated flows in a time slot of \( \Delta T = 20 \) seconds rounds the 2500 flows, which represents a value of \( \text{CT}(X_{\text{SSCwc}}) \approx 0.4 \) seconds. For the \( m = 9 \) features that we have used \( (N = 36) \), and even without doing parallelization, the total CT is \( N \times \text{CT}(X_{\text{SSCwc}}) \approx 14.4 \) seconds.

4.7 Conclusions

The completely unsupervised algorithm for detection of network attacks that we have presented has many interesting advantages w.r.t. previous proposals. It uses exclusively unlabeled data to detect and characterize network attacks, without assuming any kind of signature, particular model, or canonical data distribution. This allows to detect new previously unseen network attacks, even without using statistical-learning. By combining the notions of
Sub-Space Clustering and multiple Evidence Accumulation, the algorithm avoids the lack of robustness of general clustering approaches, improving the power of discrimination between normal-operation and anomalous traffic. We have shown how to use the algorithm to automatically construct signatures of network attacks without relying on any kind of previous information.

Finally, we have evaluated the computational time of our algorithm. Results confirm that the use of the algorithm for on-line unsupervised detection and automatic generation of signatures is possible and easy to achieve for the volumes of traffic that we have analyzed. Even more, they show that if run in a parallel architecture, the algorithm can reasonably scale-up to run in high-speed networks.
Bibliography


5 A Survey of Feature Selection in Internet Traffic Characterization

Bo Zhu and Alberto Mozo
Escuela Técnica Superior de Ingeniería de Sistemas Informáticos
Carretera de Valencia Km. 7 Madrid (28051)
Universidad Politécnica de Madrid
{bozhumatias@ict-ontic.eu, a.mozo@upm.es}

Abstract: In the last decade, the research community has focused on new classification methods that rely on statistical characteristics of Internet traffic, instead of previously popular port-number-based or payload-based methods, which are under even bigger constrictions. Some research works based on statistical characteristics generated large feature sets of Internet traffic; however, nowadays it’s impossible to handle hundreds of features in big data scenarios, only leading to unacceptable processing time and misleading classification results due to redundant and correlative data. As a consequence, a feature selection procedure is essential in the process of Internet traffic characterization. In this paper a survey of feature selection methods is presented: feature selection frameworks are introduced, and different categories of methods are briefly explained and compared; several proposals on feature selection in Internet traffic characterization are shown; finally, future application of feature selection to a concrete project is proposed.

Keywords: feature selection, Internet traffic characterization, big data

(*)This research was supported in part by project ONTIC-619633 funded by the European Commission under the Seventh Framework Programme.
5.1 Introduction

In recent years, the world has witnessed an explosion of available information in almost every domain. The sharp increase of the scale of data sets poses a great challenge for scientific researchers when they try to characterize data under research and extract useful knowledge at an acceptable cost. Features are used to convey information as measurable properties to characterize certain aspects of objects under observation in data analysis, machine learning etc. Due to the fast progress of hardware and storage technologies, the scale of feature sets has raised from tens to thousands or even more. In the case of Internet traffic characterization, besides old sample features like protocol category, complex features like Fourier transform of the inter-arrival time of packets [1] are also considered in the latest research works. Handling such a big feature set could be computationally expensive; furthermore, irrelevant and redundant features may also decrease the accuracy of characterization results; finally too many features can severely jeopardize the interpretability of results. As a consequence, feature selection serves as a fundamental procedure of preprocessing in big data scenarios before stepping forward to further application of statistical or machine learning techniques. The main objective of feature selection is to select a subset of features as simplified as possible without suffering a significant decline of accuracy for classification or forecasting, i.e. experimenting with a subset in place of full feature set results in equal or better classification accuracy. The process of feature selection can be totally supervised. In many existing research initiatives domain experts were required to provide a candidate subset of features considering possible domain relevance. However, nowadays due to the large size of feature sets, purely manual feature selection becomes infeasible. Consequently, various feature selection methods have been proposed to generate core feature subsets utilizing different theories and techniques. Feature selection can provide several advantages: It significantly reduces the computational burden, which in turn increases the performance (running time, precision etc.) of classification or prediction models; By getting rid of redundant, irrelevant features or even noise, further processing results don’t suffer from bad impacts brought by these interfering
5.2 Feature Selection

factors, hence their simplicity and interpretability is improved; It avoids the overfitting problem by building models with only core features, which increases the quality of classification models [2]; As a preprocessing step, feature selection can also help to deeply understand target data sets. Together with domain knowledge, the potential candidate features can be double-validated. The rest of the article is organized as follows: in section 2 we provide an introduction to feature selection frameworks and different categories of feature selection methods as well as a thorough comparison between them; section 3 briefly introduces several feature selection research works in the domain of Internet traffic characterization. Finally in section 4 we propose a future application of feature selection to the ONTIC project and draw conclusions.

5.2 Feature Selection

5.2.1 1.1 Feature Selection Framework

In [3],[4] a basic framework of feature selection was proposed by H. Liu et.al, which is shown in 5.1.

From the figure above, we can see that the feature selection process contains four key procedures: feature subset generation, candidate subset evaluation, stopping criteria judgment and result validation. Previous feature selection research proposals distinguish themselves from each other mainly based on the first two procedures. Since different searching strategies are applied during the generation process, various candidate feature subsets will be available utilizing different methods, and therefore the subset quality and running time performance also vary. Similarly, the optimal feature subset under one evaluation measurement may not be the best option when considering another measurement. This will be discussed in detail in the following section. As mentioned in [5], some previous work focused on the removal of irrelevant feature, but failed to handle redundancy problems. Another framework, shown in 5.2, was proposed in [6] to address the negative effect of redundancy on the speed and accuracy of learning algorithms. First they defined a synthetic function f based on a series of metrics such as irrelevance, redundancy, relevance, sample size, etc. A smaller data set was
5. A Survey of Feature Selection in Internet Traffic Characterization

Figure 5.1: Basic framework of feature selection [3]

obtained taking into consideration function f, on which different feature selection algorithms were conducted, resulting in a hypothesis H. Finally a score was generated comparing the defined function and the obtained hypothesis using several predefined scoring criteria.

Since searching through possible candidate feature subsets in the generation process is computationally costly, in [7] H. Liu et.al proposed an improved framework of feature selection to avoid initial subset search step and explicitly eliminate redundant features. This framework is shown in 5.3. In [5], a similar framework was also proposed, which first removes irrelevant features using T-Relevance as a relevance measure quantified by symmetric uncertainty, and then eliminates redundant features by means of a novel feature selection algorithm named FAST.

5.2.2 Feature Selection Method Categories

Previous works on feature selection categories distinguish themselves mainly based on the different search strategies used in the subset generation process [8] or evaluation measure-
5.2 Feature Selection

Figure 5.2: Framework of feature selection based on irrelevance and redundancy [6]

Figure 5.3: Improved framework of feature selection [7]

ments considered in the subset evaluation process [9], or both of them at the same time [3].

Search Strategy In terms of search strategy, feature selection methods can be generally cat-
A Survey of Feature Selection in Internet Traffic Characterization

ategorized into 3 types:

Complete Search: This kind of search strategy carries out a complete traversal within the whole feature space, which can be exhaustive or non-exhaustive.

Exhaustive search is simple and intuitive, but suffers from a heavy computational cost. Search methods like breadth first search belong to this type. The time complexity is $O(2^n)$, which makes it infeasible to apply to feature sets of large scale.

Non-exhaustive complete search includes a heuristic process with a backtracking step; once certain judgment criteria are satisfied, the backtracking will be triggered. This ensures that the global optimal subset will be obtained while reducing search space. Examples are:

- Brand and Bound Search: Subproblems are further divided into subproblems until feasible solutions better than the current solution are found; otherwise, this subproblem is pruned.

- Beam Search: An improved variant of B & B Search [10]. At first, the K best features are estimated using scoring criteria. Downstream search starts from these K features by adding one more feature, and K feature subsets with highest score are stored in the queue. In this way, only the most promising K features are considered at each depth level, which significantly reduces search space.

- Best First Search: Similar to Beam search, at each depth level only the feature with the lowest value of the evaluation function is selected for further expansion.

Heuristic Search:

Heuristic Search carries out the search process within the state space. It assesses every candidate and obtains the best one. Then the search process starts over from this temporary point.
5.2 Feature Selection

until the optimal result is finally achieved. Thus many search paths can be discarded directly, which improves its performance greatly. The goodness of each candidate is evaluated by an assessment function established with the help of domain-specific information or heuristic information.

- Sequential Forward Selection: Forward methods start with empty feature subset \( F \). Each iteration, the feature \( f \) that makes the value of feature objective function \( F(f) \) optimal is added to \( F \). This process is repeated until the number of features of the expected subset is reached or the predefined classification accuracy threshold is achieved. The main drawback of naive sequential selection methods is the inference of nested subsets. Since the correlations between features are not taken into consideration, if feature \( A \) is entirely dependent on feature \( B \) and both features are added into the same subset, then a nested subset with redundancy is generated.

- Sequential Backward Selection: As opposed to SFS, backward selection starts from the full feature set, and each iteration it discards the feature that makes the feature objective function optimal after elimination. The process goes on until stopping criteria are satisfied. Plus-L Minus-R Selection: This is an improvement based on naive sequential selection. Starting from the empty set, each iteration \( L \) features are added, then \( R \) (\( L \geq R \)) features are eliminated to make the feature objective function optimal. The backward LRS method also exists. LRS methods provide a way to consider correlation between features, and consequently avoid nested subsets. The choice of parameters \( L \) and \( R \) has a crucial impact on algorithm performance and should be selected with caution.

- Sequential Floating Forward Selection: SFFS improves forward LRS, but in contrast, the \( L \) and \( R \) in SFFS are not constant, i.e. SFFS starts from the empty set, and each iteration it chooses the subset \( f \) from unselected features that makes the value of the feature objective function optimal after adding \( f \). Then from chosen subset \( f \), it elim-
inates a subset of \( f \) (namely \( g \)) that makes the value of the feature objective function optimal after eliminating \( g \).

**Random Search:**
Random search was introduced into feature selection because the scale of feature sets keeps increasing so sharply that the computational cost using previous search strategies becomes unacceptable. Random search is a simple improvement of greedy search that obtains an optimal feature subset by random sampling. This can greatly reduce computational complexity, which consequently increases efficiency and precision. However, one defect of random search is that most methods get trapped easily in local optimal solutions, but multi-repetition could help to overcome this problem. The initialization of pre-established parameters with proper values is another crucial task for random search methods.

- **Simulated Annealing:** As an improvement of classic hill-climbing search, it introduces a probability function which can give a probability value to determine whether to choose a generated subset as current best subset. This provides the advantage to jump out of local optimal subset at certain probability.

- **Genetic Algorithm:** First several feature subsets are generated randomly, for which fitness values are calculated [11]. Then operations of crossover and mutation are conducted with a certain probability, producing the next generation of subsets. The reproduction process follows rules of nature, i.e. subsets with higher fitness are more likely to be selected to generate offspring subsets. The optimal feature subset could be obtained after \( N \) (sufficiently large) generations.

- **Particle Swarm Optimization:** PSO method is very similar to GA method, but it replaces crossover and mutation operators with the control of the accelerating speed of particles (feature subsets in this case) [12]. Two optimal values are tracked by each particle: \( pbest \) (optimal value found by the particle itself) and \( lbest \) (current optimal value found by any particle among the neighbors of the particle).
5.2 Feature Selection

Some concrete search methods can adopt two search strategies at the same time. For example, the ant colony optimization algorithm is a random search method as well as a heuristic search method.

**Evaluation Measurement** Evaluation measurement is used to evaluate the goodness of feature subsets provided in a subset generation process. In terms of operating functions, feature selection methods can also be categorized into 3 types:

*Filter Method:*
Filter methods judge the goodness of subsets via analysis of internal characteristics of the selected features. Various ranking criteria assess different characteristics among features by generating a score for each feature. A corresponding ranking list illustrates the fitness of every feature regarding certain criteria. A threshold can be pre-established to filter out features with low scores. This type of feature selection methods is independent from training process and specific inductive algorithms [13]. As a result, filter methods can be promoted among different algorithms with low computational requirements. Thus filter methods can be utilized to pre-reduce plenty of obviously irrelevant features, but useful features might also be filtered. It can be further classified into 4 types [4].

1. **Correlation Measurement** Under the assumption that a good feature subset contains high feature-class relevance and low inter-feature relevance, some previous research works focused on measuring relevance using a linear correlation coefficient. The Pearson correlation coefficient is one of those simple and popular measurements, as shown below:

\[ R(i) = \frac{cov(X_i, Y)}{\sqrt{var(X_i)var(Y)}} \]

2. **Distance Measurement** Like unsupervised learning algorithms, distance-based feature selection is based on the assumption that good feature subset can make the distance between samples belonging to the same class as small as possible, and samples that
belong to different classes as far as possible. Some common distance measurements include Euclidean distance, Mahalanobis distance, Minkowski distance etc.

3. Information Measurement Measurements such as mutual information [14],[15] and information gain are based on the simple assumption that the more information we gain, the better a feature subset is. Information gain is a concept of information theory, which originates from the concept of entropy (also a measurement of uncertainty):

$$H(Y) = \sum_{i=1}^{m} P_i \log_2 P_i$$

After adding condition variable $X (X = x_i)$, the conditional entropy of $Y$ is:

$$H(Y) = \sum_{i=1}^{m} P_{x=x_i} H(Y|X = x_i)$$

After adding conditional $X$, the uncertainty of $Y$ has decreased. The decrease in uncertainty is an increase in certainty, called information gain, calculated as follows:

$$IG(Y|X) = H(Y) - H(Y|X)$$

4. Consistency Measurement Consistency-based feature selection methods follow the simple assumption that good feature subsets should consist of as few features as possible while maintaining high consistency at the same time. i.e. if two samples contain the same values of a certain feature set, but belong to different classes, then this feature set should not be considered as an optimal subset. Regarding the interference of noise, it’s more reasonable to set an inconsistency rate to avoid ignoring good features. One example is rough set theory [14, 16, 17, 18].

**Wrapper Method:**
For wrapper methods, feature selection is integrated into training process [19]. Acting as classifiers, wrappers classify feature sets using the selected subset. Measurements of model
5.2 Feature Selection

forecasting capability, such as classification accuracy, serve as evaluation criteria of the quality of selected subsets. The evaluation process of wrapper methods depends on specific inductive algorithm, which in return limits its portability to other algorithms. Although the quality of feature subsets generated by wrapper methods generally outperforms that of filter methods, the much heavier computational cost should also be carefully bargained.

Hybrid Method:
With the objective of reducing the computational cost of wrapper methods while maintaining their outstanding classification accuracy, several research works were carried out to seek the possibility to speed up the convergence of wrapper methods by combining filter methods. Several hybrid methods were proposed in [13, 20, 21, 22] to absorb the advantages of both types.

Label-based category
Based on whether data are labeled, feature selection methods can also be generally divided into supervised feature selection and unsupervised feature selection. Since the quality of the original data set has great impacts on future performance of machine learning algorithms, the lack of class labels brings more difficulties for feature selection since less natural grouping information can be acquired from the original data. Especially in nowadays’ big data environment, Internet traffic researchers face bigger challenges, because Internet traffic data tend to be more structurally complex, vague, label-missing and of greater scale. Several supervised feature selection methods are listed [23, 24, 25, 26, 27], and a series of valuable works on unsupervised feature selection methods can be found in [28, 29, 30, 31, 32, 33, 34].

5.2.3 Feature Selection Methods in Internet Traffic Characterization

In the research domain of Internet traffic characterization, feature selection keeps attracting special attention because the scale of Internet traffic feature sets has experimented a rapid explosion during the last decade. Work in [35] has shown the effectiveness of feature se-
lection to improve computational performance without severely reducing classification accuracy when conducting traffic flow identification. They undertook two different evaluation measurements - consistency and correlation, and further compared greedy and best first (both forward and backward directions) search strategies. [1] generated an original traffic set that contains 248 features, on which direct importation to classification models is infeasible due to the existence of high irrelevance and redundancy, and a great computational cost. Fast Correlation-Based Filter, proposed in [36], together with a novel wrapper-based method to determine threshold, were used to select useful features. The same data sets were reused in [37], which proposed a new feature selection method called BFS. BFS is more competitive in maintaining the balance of multi-class classification results in comparison with FCBF regarding metrics like g-mean and classification accuracy. [38] proposed an application-based feature subset selection using parameter estimation for each logistic regression model established for the corresponding application class. This can effectively resolve the imbalance problem caused by elephant flows. [39] discussed real-time Internet traffic identification that has special requirements of simplicity and effectiveness on feature subsets. [40] performed Correlation based, Consistency based, and PCA [41] feature selection on a real-time Internet traffic dataset obtained by a packet capture tool. [42] proposed a mutual-information-based feature selection and automatic determiner of the number of relevant features. An outlier detection method which improved PCA was also brought out to avoid the influence of outliers in real traffic. In [43] new evaluation metrics named goodness, stability and similarity were used to assess the advantages and defects of existing feature selection methods. Then they integrated six relatively competitive FS methods (Information Gain, Gain Ratio, PCA, Correlation-Based Feature Selection, Chi-square, and Consistency-Based Feature Selection) to combine their strengths. This new integrated technique consists of two procedures: first, a consistent subset generated from the results of different feature selection methods was discovered; then based on this candidate subset, a proposed measurement of support was used to obtain the final feature subset. [44] first proposed a novel feature selection metric named Weighted Symmetrical Uncertainty (WSU), then brought about a hybrid FS method that pre-filters features using WSU. Later it adopted a wrapper method that selected features regarding the Area Under roc Curve (AUC) metric. An additional algorithm SRSF was also proposed to
5.3 Future Practical Application and Conclusions

get rid of the effect of dynamic traffic flows. [45] proposed a novel feature selection method that made use of both linear correlation coefficient measurement and non-linear mutual information measurement. Zulaiha et. Al first proposed a wrapper method in [46] using Bees Algorithm as search strategy and Support Vector Machine as the classifier. After comparison, it turned out that BA yielded better results than other FS methods such as Rough-DPSO, Rough Set, Linear Genetic Programming, MARS and Support Vector Decision Function Ranking. Then in [47] they proposed a hybrid FS method called LGP_BA which is a combination of Linear Genetic Programming and Bee Algorithm that achieved better accuracy and efficiency.

5.3 Future Practical Application and Conclusions

The ONTIC (Online Network Traffic Characterization) project, funded by the seventh Framework Programme for Research and Technological Development of European Commission, aims for accurate identification and characterization of network traffic regarding different application types, which will significantly contribute to problems such as dynamic QoS management, network intrusion detection and fast detection of network congestion [48]. To achieve this objective, a Peta Byte size data set composed of real network traffic summaries will be generated from several data flows. These flows will be captured in the core network of an ISP for several months. Making use of such an adequate raw data set, more than two hundred features are generated to form the feature space with the help of a TCP sTatistic and Analysis Tool - Tstat [49]. These features are used to characterize Internet traffic flows and inherently classify them into corresponding Internet traffic classes, which are often of different granularities: it can be simply divided into normal and abnormal classes, or various specific application categories in terms of detailed circumstances. There will be more than 240 features in our raw feature set; therefore, it’s necessary to conduct a feature selection procedure to reduce to expected scale (in our case 5-8 features are sufficient). Since different feature selection methods adopt specific and independent criteria to perform the feature pruning step, generated subsets are probably discriminative and of different sizes, which makes direct comparison meaningless. Several feature selection methods will be selected and implemented towards the original full feature set, and a thorough comparison of experiment
results regarding some generic metrics like classification accuracy and running speed will be conducted making use of different methods. In this paper we provide an introduction to feature selection frameworks and different categories of methods. Several algorithms are briefly explained, including both classic and newly-proposed methods. Then feature selection in the domain of Internet traffic characterization is especially discussed. Considering the variety of existing methods and rapid proposal of new methods, this survey is far from complete. Future application of feature selection to ONTIC project is also proposed to verify the practical significance of feature selection.
Bibliography


BIBLIOGRAPHY


84


BIBLIOGRAPHY


BIBLIOGRAPHY


6 Adaptive Quality of Experience (AQoE) control for Telecom Networks

Alejandro Bascuñana, Fabian Castro, David Espadas, Patricia Sánchez and Miguel Ángel Monjas
Ericsson España, Madrid R&D Center, Madrid, Spain
{alejandro.bascunana, fabian.c.castro, david.espadas, patricia.sanchez.canton, miguel-angel.monjas}@ericsson.com

Abstract: Early detection of quality of experience (QoE) degradation patterns which could end up in congestion situations is currently at the top of the agenda for the communication service providers (CSP), especially those providing mobile services. Mobile Operators already have the possibility of classifying users and applying congestion mitigation policies depending on the customer segment they belong to. Classification can be done in different and even highly flexible ways, but no operator is currently giving their users automatic adaptation procedures so that their QoS meet their different expectations, profiles and device usage when QoE degradation scenarios occur. This position paper aims at describing this problem, analyzing, among other issues, the need to have a larger amount of processed information about how and when users consume services, along with a complete profiling of said users, and the need to build a new automated priority group distribution that enables the network to provide the best QoE while optimizing bandwidth usage and other relevant parameters.

(*)This research was supported in part by project ONTIC-619633 funded by the European Commission under the Seventh Framework Programme.
6 Adaptive Quality of Experience (AQoE) control for Telecom Networks

**Keywords:** QoE, Heterogeneous users, Classification, Automatic adaptation, QoE degradation scenarios, automated group distribution, crowded events, quality of experience (QoE) degradation patterns

### 6.1 Introduction

Early detection of potential quality of experience (QoE) degradation patterns which could end up in congestion situations is currently a hot topic for the communication service providers (CSP), especially for Mobile Operators. However, detection of said patterns is not a trivial task, as Mobile Networks are evolving towards scenarios of exponential mobile data traffic growth, where heterogeneous users, with different needs and profiles, demand the best QoE according to their expectations and personal preferences. Although it could be argued that Mobile Operators could cope with the growth of congestions situations simply with higher investments in capacity, a more intelligent and adaptive approach is required, as network resources are always limited and finite and there is a tremendous pressure over the CSP to improve their OPEX and CAPEX [2]. Today Mobile Operators are able to classify users and apply congestion mitigation policies depending on the customer segment they belong to. Segmentation can be done in different and even highly flexible ways (i.e. there are operators which build their segments based on user’s own tailored service offering, while others have a more schematic segmentation based on generic rules for all the users with the same subscription profile âˆ”typically Gold, Silver and Bronze segmentsâˆ”’, etc.) with different bandwidth limits, different congestion mitigation measures and even different customer care strategies depending on the customer segment. However, no Mobile Operator is going a step further by giving their users the necessary procedures so that their QoS meet their different expectations, profiles, and use of mobile devices (professional, leisure time, etc.) in QoE degradation scenarios. Fine tuning of the networks is done currently by human experts that typically set up static optimization rules which get applied in case of QoE degradation scenarios, in a reactive, or at least planned, manner. Telco operators are looking for a simplification in their

---

This research was supported in part by project ONTIC-619633 funded by the European Commission under the Seventh Framework Programme.
6.2 Description of the Problem

Data traffic is foreseen to grow about 50 percent each year until 2018 [1]. This implies a very high growth rate of the data going through telecom networks. In addition to that, from an economics point of view, there is an increasing CAPEX and OPEX pressure in the communication service providers (CSP) operation activities [2]. Among all the options that could be taken to reduce such pressure, the one we are focusing on in our current work is operation management optimization, by proposing investments in network optimization tools to decrease the CAPEX, making a better use of current network resources; OPEX will be reduced by automating these activities. All these actions will lead to a better use of the resources and therefore to reach the goals of CAPEX and OPEX optimization. Another key trend is the adoption of Big Data technologies [3] in the telecom industry with the introduction of new products and tools in the market such as Ericsson CEA [4], which allows telecom companies to make complex analysis about their customer needs, problems, etc. by means of analytic tools. In this high data growth scenario CSPs are looking for automatic procedures to improve the provided Quality of Experience levels (QoE) for each of the applications and services used by the users. This can be done by making an optimized use of available network resources. QoE [6] provides a subjective measure about the experience of the user with an application or service and therefore is different from the current way of managing the user experience via QoS levels. On the other hand the automatic procedures will help CSPs to reduce their churn rates and to improve their customer satisfaction index differentiating their offering from others. The optimization of the perceived quality of experience (QoE)
when using different applications, services, etc. is becoming more and more a cornerstone to CSPs. The QoE optimization is later on done via the configuration of QoS parameters. Therefore there is a higher pressure on the CSPs to give their end users the best QoE even in potential QoE degradation scenarios. Different network situations can lead to such scenarios. Planned or unplanned crowded events with thousands of people attending them on the same location are, among others, the typical scenarios that can trigger a "QoE degradation" pattern in the network. Once these QoE degradation scenarios are detected, the main goal of mobile operators should be to enable the best use of their resources, assuring that network resources are distributed properly among their customers -i.e. matching available bandwidth with expected QoE, minimizing denial of service, and accommodating the different priorities dynamically, etc. 6.1 summarizes the evolution from the current QoE control scenario to one based on analytics and on user personalization. We foresee new ways of optimize networks in QoE degradation scenarios. First of all, network QoS control will evolve from manual to automatic. Policies will be automatically defined taking into account user preferences and network parameters. One advantage of this QoE automation scenario is the transition from planned-in-advance QoE degradation mitigation actions to scenarios where no actions have to be explicitly planned or deployed, from the manual set-up of optimization rules to the automatic generation of rules carried out by analytics systems; and last but not least, from a scenario where network optimization is carried out in an ad-hoc way in critical situation to scenarios where network optimization is done continuously.

6.3 State of the Art

In the typical mobile scenario, when a user terminal (i.e. a smartphone) starts a PDN connection, a default bearer is established. Said bearer is characterized by the UE1 IP address and certain Quality of Service (QoS), meaning that all the traffic running over the same bearer will obtain the same treatment in the Radio Access Network (RAN) and in the transport network in terms of QoS and priority. More than one bearer can be established in order to give
different treatment in the radio network to different services. Upon a congestion situation, the RAN applies admission control and may tear down established bearers based on the bearers’ priority.

This solution based on dedicated bearers provides service and subscriber differentiation. However the majority of mobile data traffic (e.g. Internet or over-the-top services traffic) is currently delivered over default bearers.

Certain alternatives have been explored. For example, congestion-awareness, based on statistics regarding what locations are prone to be congested at certain periods of time, is a more advanced solution provided by vendors like Ericsson. With such an approach a Policy and Charging Rules Function (PCRF) can make policy decisions based on these statistics and also considering the current UE location information and time. In order to prevent congestion the PCRF can decide to limit the bandwidth assigned to certain users for the total traffic or for specific services. Statistic data is populated in a database accessible by the PCRF. This data is mainly a table containing locations and congested time periods.

The main handicaps with this type of solution are that operator must maintain this information as much updated as possible and that location information received in the PCRF is not always accurate due to the signaling penalty that may cause to propagate to the PCRF all the

![Figure 6.1: New scenarios for enhancing user’s QoE](image)

<table>
<thead>
<tr>
<th>As-is</th>
<th>Proposed</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Manual scenario</td>
<td>• Automatic scenario</td>
</tr>
<tr>
<td>• Planning in advance</td>
<td>• No need of planning</td>
</tr>
<tr>
<td>• Only solve scheduled scenarios</td>
<td>• Can solve unscheduled scenarios</td>
</tr>
<tr>
<td>• Very basic set of rules provided by the PCRF operator</td>
<td>• Advanced and automatic generated set of rules</td>
</tr>
<tr>
<td>• Ad-hoc optimization of the network resources</td>
<td>• General and continuous network optimization</td>
</tr>
</tbody>
</table>
6 Adaptive Quality of Experience (AQoE) control for Telecom Networks

location changes of all the users. Besides, the congestion decisions are based on historical preconfigured data, and this means that the PCRF decision is purely a prediction. Depending on accuracy of this prediction the goodness of this solution can vary, meaning that it can be decided to throttle traffic unnecessarily.

6.4 Standardization and related activities

Congestion mitigation can be achieved at various levels. As mentioned in the section above, upon a congestion situation, the RAN can apply admission control measures and even tear down established bearers based on the bearers’ priority. In addition to that, 3GPP is defining solutions for so-called user plane congestion management (UPCON) [5] that leverages the PCRF capability to make policy decisions. The proposal enhances current 3GPP packet core architecture by introducing a new entity named RCAF (RAN Congestion Awareness Function), which collects information related to user plane congestion from the RAN OAM, determines the list of impacted UE and sends the congestion report to the PCRF. 6.2 depicts the mentioned architecture:

![Figure 6.2: 3GPP UPCON Architecture](image)

The information sent from the RCAF to the PCRF contains the UEs impacted by congestion identified by the UE IMSI, an identifier of the area that is undergoing a congestion situation, and the severity of the congestion, even a no-congestion state. In this architecture the PCRF provides policies for congestion mitigation to one or more of the following network
entities:

1. to the Policy and Charging Enforcement Function (PCEF), over the Gx interface;

2. to the Traffic Detection Function (TDF), over the Sd interface;

3. to any Application Function (AF), over the Rx interface;

4. to any other network entity that could play an enforcement role within the packet core (in a proprietary way);

Congestion mitigation actions are enforced in the aforementioned network entities according to the policy decision by the PCRF:

- The PCEF/TDF can perform per-user bandwidth limitation, prioritization and traffic gating according to the policies provided by the PCRF.

- The AF (e.g. an application server or proxy) can directly or indirectly support the congestion mitigation, e.g. by adapting the sending rate, through media transcoding or compression, or by delaying push services.

- Based on policies provided by the PCRF, the PCEF/TDF may also perform actions to support congestion mitigation measures in the RAN, e.g. the policy can control when packet marking (such as e.g. proposed by RAN-based solutions for RAN user plane congestion management solutions) should be performed.

- The PCRF may limit/reject the authorization of new requests for application flows, based on current procedures. For deferred delivery of service the PCRF may send a re-try interval to the (operator’s or third-party’s) AF, which indicates when service delivery may be retried. The value of the re-try interval depends on operator policies (e.g. it may vary depending on the congestion level but may also be set taking other criteria into account). The PCRF may send updated re-try intervals, e.g. if the congestion level changes.
The PCRF can ask the UE to offload to alternative access networks, provided there are other networks (i.e. Wi-Fi) available in the congested area.

This solution is still under definition but keeps a clear principle: congestion mitigation actions in the packet core are orchestrated by the PCRF. It sends appropriate policies to the available enforcement points for them to execute the mitigation actions. The success or not of this type of architecture depends a lot on the intelligence put in the PCRF and the freshness and accuracy of the congestion information reported to the PCRF or even on the capacity of early detection of congestion patterns which could anticipate actual congestion situations. Fine-grained management of users and the capacity to determine congestion mitigation actions according not only to the customer segment the user belongs to, but also to their service usage patterns or to their type of terminal. How the PCRF can adapt its decisions based on the users and network situation with the minimum human intervention is crucial for the viability of this solution and the successful deployment in real operators’ networks.

6.5 Use Cases Description

The following is a description of two use cases where adaptive QoE management can provide value both to end-users and to operators:

1. On unplanned (e.g. random problems in urban traffic, spontaneous demonstrations, etc.) or planned but unknown to the operator situations (i.e. the last Star Wars movie premiere in a downtown theater), the network can become overloaded up to the point of congestion. Minimizing the possibility of these congestions while meeting as much as possible the requirements from the users is requested by operators globally. Dynamic bearer assignment per user in real-time may help meet the changing requirements, e.g. assigning enough bearer capacity for completing a satisfactory video streaming session, while reducing bearer capacity for users who are only texting, even if his/her customer segment may imply high capacity bearer in all cases.

2. The operators usually need to plan for seasonal people agglomerations in specific areas (e.g. vacation periods) and put in place the extra capacity to cope with this demand.
6.6 Conclusions and Future Works

This forecasting is based mainly on calendar, demographics and previous experiences. This implies a considerable and mostly manual planning effort, i.e. OPEX, plus the cost of the allocated network resources specifically to these events, which cannot be easily modified on-the-fly, in case the forecast proves incorrect as the season progresses. With a more dynamic approach to capacity assignment, the operator will obtain the best possible performance from the network, in terms of matching the requirements of the end users in real-time, going beyond the static planning methodology, towards a dynamic adaptation which will maximize the indicators of choice, e.g. individual QoE which takes into account the expectations per service delivered.

6.6 Conclusions and Future Works

Adaptive QoE within packet networks serving end-users (as those of the Mobile Operators) is in itself a challenging area of study, especially because it is not a theoretical experiment but a real problem that operators have to cope with every day. Therefore, any solution that could be devised has to be subjected to a reality check. It poses a lot of pressure on researchers as they need to access relevant operators’ data sources in order to be able to make predictions, determine end-users’ service usage patterns and get them classified. But not only real and varied data for early detection, prediction and user profiling is needed (depending on the operator, usage patterns are different, depending on the size of its user base, the type of customers or the very socioeconomic variables of the market where the operator carries out its operations). Closest to real test-beds are needed in order to validate the solutions. Only if the new solutions are ”good-enough” to provide a measurable advantage over existing solutions, these new approaches will be adopted. An additional challenge is the compliancy with current standards such as those ruling the internal operations of telecom operator networks. Any solution that could be envisaged to complement, for instant, the existing UPCON architecture, must be compliant with it. If the organizational issues presents a real challenge on any solution, the most compelling one relates to how to manage the huge amounts of information that are needed to identify congestion patterns or to create user profiles based on service consumption. This problem combines a huge amount of data (Big Data), which in some situations
have to be processed in real time (Stream Processing) and analyzed (Stream Analytics). Furthermore, all of these tasks have to be addressed with a specific domain knowledge that is specific to the telecom scenario. Ericsson is working in the lines described in the previous paragraphs. First, we are envisaging an UPCON-compliant architecture that can provide enhanced congestion mitigation procedures. Secondly, we are proposing procedures to cope with congestion situations by means of a wide range of Big Data Analytics techniques. We need to choose the most appropriate algorithms and technologies for each of the problems being addressed. Finally, we aim to be able to detect congestion patterns and to fine-grained classify users according to their service usage patterns so that each of them are applied a personalized congestion mitigation policy. Finally, we aim to make the network self-aware and be able to determine whether the policies being applied are actually enhancing the behavior of the network. And last but not least, we are sharing our proposals with some of our customers in order to set up joint projects that allow us to test our solutions in real scenarios and come out with effective solutions.
Bibliography

http://www.ericsson.com/thinkingahead/networked_society/networked_society_essentials


http://www.ericsson.com/ourportfolio/products/customer-experience-assurance

http://www.3gpp.org/ftp/Specs/archive/23_series/23.705/23705-0a0.zip

Retrieved 13:27, September 1, 2014, from
7 A Telecom Analytics Framework for Dynamic Quality of Service Management*

Álvaro Guadamillas¹, Miguel Ángel López², Nikolaos Maravitsas³, Alberto Mozo¹,
Fabio Pulvirenti⁴

¹Escuela Técnica Superior de Ingeniería de Sistemas Informáticos, Universidad Politécnica
de Madrid, ²SATEC, Madrid, Spain, ³ADAPTIT SA, Athens, Greece, ⁴Dipartimento di
Automatica e Informatica Politecnico di Torino

{a.guadamillas@ict-ontic.eu}, {miguelangel.lopez@satec.es}, {nmar@adaptit.gr},
{a.mozo@upm.es},{fabio.pulvirenti@polito.it}

Abstract: Since the beginning of Internet, Internet Service Providers (ISP) have seen the
need of giving to users’ traffic different treatments defined by agree- ments between ISP
and customers. This procedure, known as Quality of Service Management, has not much
changed in the last years (DiffServ and Deep Packet Inspection have been the most chosen
mechanisms). However, the incremental growth of Internet users and services jointly with
the application of recent Machine Learning techniques, open up the possibility of going one
step forward in the smart management of network traffic. In this paper, we first make a
survey of current tools and techniques for QoS Management. Then we introduce clustering
and classifying Machine Learning techniques for traffic characterization and the concept of
Quality of Experience. Finally, with all these components, we present a brand new framework

(*)This research was supported in part by project ONTIC-619633 funded by the European Commission under
the Seventh Framework Programme.
that will manage in a smart way Quality of Service in a telecom Big Data based scenario, both for mobile and fixed communications.

**Keywords:** Machine Learning, Big Data, Traffic Characterization, Quality of Service, Quality of Experience

### 7.1 Introduction

Accurate identification and categorization of network traffic, according to application type, is a major task for large scale network monitoring and management aspects, with the most crucial of them being QoS evaluation, capacity planning and intrusion detection. An enormous amount, in the order of Terabytes, of data may be transferred through the core network of an ISP on a daily basis, and an exponential growth in traffic is expected, as more than 50 billion devices will be connected to the Internet in the forthcoming future. This prediction imposes a tough challenge for network data capture and analysis.

Current approaches that try to tackle the problem lack scalability and accuracy and are mostly based in empirical formulas. So this leave an open opportunity to develop an accurate and massively scalable platform for both online and offline characterization of network traffic pattern evolution is missing that should be a key element in facing the challenges. Top applications potentially benefitting from such a platform include proactive congestion control mechanisms and intrusion detection systems.

To this end, the ONTIC project proposes to design, develop and evaluate:

1. A novel architecture of massively-scalable online techniques able to, at one hand, characterize network traffic data streams, identifying traffic pattern evolutions, and on the other hand proactively detect anomalies in real time at very high network speeds, i.e., hundreds of thousands of packets per second.

2. An innovative set of massively-scalable offline data mining techniques to characterize network traffic, exploiting big-data analytic approaches and cloud-based distributed computation paradigms on extremely large network traffic datasets.
The current project will attempt to integrate the above online and offline techniques into an autonomous unsupervised network traffic classification platform. Classification of traffic is essential for a proper QoS management. In addition to this classification, it is important to exactly find out which features define each class so that it is possible to know how best assign network resources and achieve a good QoS management. If more the classification of traffic does not remain static, the network will better know how to treat the traffic in different situations and both the user experience and the network management will improve. This is the target of the proposed framework in this paper, which will use QoS principles to improve the traditional QoS management. Our proposed framework solution will combine traditional techniques and has to keep in mind that (1) traffic pattern will keep in constant evolution and (2) the amount of traffic will be raised in time. Latest studies on traffic characterization are focusing on ML algorithms to face traffic pattern evolution [1], and Quality of Service Management needs to move to these ML algorithms to evolve its traditional out of date techniques. In addition, traffic needs to be handled as Big Data, and the new architecture shown in this paper must meet scalability and parallelization requisites, as we are going to see in the proposed framework.

### 7.2 Related Work

Traffic characterization in QoS/QoE environment is increasingly gaining importance, as noticeable from the massive number of works on these topics. Traffic characterization is already the core of many fundamental network operation and maintenance activities, such as the enforcement of Quality of Service guarantees or Traffic Engineering, and it has the potential to solve difficult network management problems and security issues, such as intrusion detection and prevention. In addition, the explosion of the IP-based applications and services, and, on the other hand, the multitude of access networks from which to choose, made the experience quality perceived by the users a primary subject of interest for the ISPs. Facing this kind of quality competition, the concept of Quality of Experience (QoE) emerged, combining user perception, experience, and expectations, with non-technical and technical parameters such as application and network level quality of service (QoS).
7 A Telecom Analytics Framework for Dynamic Quality of Service Management

Furthermore, the boom of mobile devices, content, server virtualization, and advent of cloud services drove the networking industry to re-examine traditional network architectures and made Big Data an emerging hot topic. Existing network architectures were not designed to meet current requirements, hence, Software Defined Networking (SDN) is spreading, promising to be dynamic, manageable, cost-effective, adaptable, seeking to be suitable for today’s and future applications. One of the main reasons, and, at the same time, effects of SDN diffusion is Big Data explosion which made conventional analytics, such as traditional Machine Learning and Data Mining techniques, unsuitable.

7.2.1 QoS architecture on Internet and UMTS

Quality of Service has traditionally focused its efforts in giving a differentiated treatment to the traffic that is traversing the network. In order to determine which kind of traffic deserves better or worst treatment, the QoS management strategy has evolved according to time and needs of both users and Internet Service Providers (ISPs) [2].

Since the beginning of Internet, Quality of Service management has been based on static classification of the traffic with no assumption that traffic may change. One of the first most popular architectures defining a QoS field was the IP architecture with the inclusion of the field Type of Service, originally defined in RFC 791 [3]. This RFC uses 3 bits (defining "precedence type") to divide traffic in 8 predefined classes with different discard priority each in case of congestion.

However, the 8 possible IP precedence types were aimed to ARPAnet needs, which originated the protocol under the Department of Defense, so the use if this field was not intended to be commercial. The Integrated Services architecture (IntServ), defined in RFC 1633 [4], allows the flows of an end-to-end service making a reservation of resources in each router of the path. This reservation (made through RSVP protocol) guarantees the QoS needed for the service, but every device along the path needs to maintain state information for each reservation, what makes scalability with hundreds of thousands of flows become an issue.

Due to the disuse of the Type of Service field, in RFC 2474 [5] the definition of this field was entirely changed in order to define the new Differentiated Services architecture (Diff-
In DiffServ, the differentiation of the traffic became customer-oriented. Although this architecture keeps backward compatibility with IP type of service, DiffServ uses the IP Type of Service header (6 bits) as Differentiated Service (DS) where the upper 6 bits contain a value called the Differentiated Service Code Point (DSCP). This DSCP is used to differentiate traffic in 4 classes with 3 different drop priorities each. The classes are used in the core of the network to divide the available link bandwidth so, for instance, a class with assigned 40% of available bandwidth of the link can be defined by an ISP for customers that pay more for their services. DiffServ also defines the possibility of pre-processing the packets at the edge of the network by controlling customers’ traffic through a Service Level Agreement (SLA) document. This SLA will contain user contracted service rates and will let ISP, for example, drop packets in case that traffic form a specific user is exceeding the contracted wideband. Lastly, DiffServ also defines routing policies for each packet in the different nodes (routers) that implement DiffServ architecture. This is known as Per Hop Behavior (PHB) and the RFC distinguishes 4 PHBs: the Default PHB specifies that a packet marked with a DSCP value (recommended) of '000000' gets the traditional best effort service; Class-Selector PHB use DSCP values of the form 'xxx000' (with x as 0 or 1) to preserve backward compatibility with the IP-precedence scheme mentioned above; Expedited Forwarding (EF) PHB is the key ingredient in DiffServ for providing a low-loss, low-latency, low-jitter, and assured bandwidth service, giving to the packets a guaranteed bandwidth service (the recommended DSCP value for EF is '101110'); Assured Forwarding (AF) PHB uses the defined DiffServ classes to give different forwarding assurances as explained before. DiffServ has been widely accepted by ISPs and latest researches of this architecture are focusing in the implementation of DiffServ into virtualized network architectures [6].

In mobile communications, QoS management plays also an important role. UMTS is the standardization of the access to the network for mobile devices produced by the 3rd Generation Partnership Project (3GPP). As its study scope goes from the mobile device until the edge of the core, the QoS differs from the mentioned so far: it cares more about the quality of the radio communication. Nevertheless, the same principles of DiffServ can be found in UMTS Management of QoS: architecture capable of providing different levels of QoS (by using UMTS specific control mechanisms); mechanisms to allow efficient use of radio
7 A Telecom Analytics Framework for Dynamic Quality of Service Management

capacity (in DiffServ case, bandwidth capacity); and UMTS also meets the QoS requirements contracted by the user. UMTS ETSI TS 127 107 [7] defines 4 classes with different QoS requirements, where the main distinguishing factor is how delay sensitive the traffic is:

- Conversational class, meant for traffic which is very delay sensitive and is intended to carry real-time traffic flows, like video telephony.

- Streaming class, a bit less delay sensitive than conversational, but it is used too for real-time traffic flows.

- Interactive class is intended to a user (either human or machine) who is on line requesting data from remote equipment (such as web browsing, data base retrieval or server access).

- Background class is aimed to traffic where users and receivers exchange traffic in the background.

7.2.2 Traffic characterization

With growth of Internet users, Internet Service Providers have seen increased their need of a deeper understanding of the composition of traffic [2]. By having a wide knowledge of the traffic, it is possible for ISPs to achieve important tasks as re-allocation of network resources, improve capacity planning and provisioning or improve fault diagnosis. These network activities will obtain best results if traffic is properly associated to source applications or services, and a complete understanding of these sources has been achieved. This is the reason why traffic characterization is important for a good management of Internet and, particularly for us, a best optimization of Quality of Service Management.

Traditionally traffic characterization techniques have been based on the assumption of a specific traffic pattern. First uses of TCP/IP Internet (after DARPA’s initial experiments) were intended to communicate companies and universities with simple services as mailing, remote access or document sharing. Later on, with the expansion of Internet for domestic
7.2 Related Work

use and web 2.0, HTTP traffic became popular [8]. The availability of broadband user con-
nexions spread the use of P2P, multimedia traffic and cloud services [9] [10]. Lately, the
popularization of wireless access technologies is increasing the use of services such as Voice
over IP or Skype. In other words, the continuous evolution of Internet has led to a complex
existence of traffic patterns in the networks that still needs to be completely understood [2].

The constant change in traffic has resulted in different traffic characterization techniques.
IANA registered ports has for long been used as traffic classification technique for application
recognition. However, many applications use unpredictable or unregistered port numbers and
the inevitability of IPv4 address exhaustion has motivated port address translation used by
Network Address Translators (NATs) [2][11].

When port detection for traffic classification became inaccurate, Deep Packet Inspection
(DPI) techniques were popularized. This technique uses application payload to associate
flows with application, what means that payload is visible (there is no encryption of the
packet) and the application protocol can be interpreted. It is easy to detect the problems of
DPI: customers may use encryption or tunnelling for their communications, and application
protocol interpretation has to be updated with every new version of each application proto-
col. In addition, some countries may impose privacy regulations to ISPs, and the costs of

7.2.3 Machine Learning algorithms

QoS and, more generally, Network traffic classification issues have been object of attention
of many researchers in the last years. A lot of them, for many reasons, developed and adapted
machine learning and data mining techniques to this environment.

Machine learning algorithms aims at extracting previously unknown interesting information,
such as dependencies (frequent itemset mining and association rules), groups of data objects
(clustering algorithms) and categories of new observations (classification and regression).

Association rule mining is a data analysis method able to discover interesting and hidden
correlations among data. It consists of two steps: (i) Frequent Itemset Extraction, in which all
the frequent patterns are mined from the input dataset and (ii) Association Rules generation.
An itemset is frequent if the number of transactions in which it appears is over a minimum support threshold. Association rules are extracted from frequent itemsets and highlight correlations among items.

Unsupervised learning, instead, aims at discovering hidden structures in unlabeled data. The most popular algorithms are the Clustering ones: their target is to group sets of objects in such a way that objects in the same groups (clusters) are more similar to each other than to those in other groups (clusters). Finally, Supervised Learning is the learning task of extracting a function (or a model) that best approximates the distribution of the input dataset. This kind of algorithms usually work on a labeled sample of the input data (training set) and produce an inferred function which can be used for mapping (label) new examples.

These techniques have been strongly involved and adopted in QoS analysis and Network traffic classification. The two topics are correlated because all QoS schemes have some degree of IP traffic classification implicit in their design. In fact, both Diffserv and Intserv expect that routers are able to recognise and differentiate fine-grained classes of traffic. Moreover, real-time traffic classification is the core-component of automated QoS architectures [12].

Hence, traffic classification importance has increased in the last years and, because of the key role of information mining in this scenario, many researchers have adopted machine learning techniques. Also, both the growing number of services of their classes and the size of the network has pushed towards the spread of data mining approaches.

In [13], classification algorithms as Nearest Neighbours have been used to build a signature-based framework for mapping QoS Classes, while [14] and [15] have exploited Genetic algorithms and Naive Bayesian classifier for the same issue. Some Call Admission Control (CAC) schemes, mechanisms providing QoS by limiting the entry of traffic at the edges of the network, have been implemented by means of Neural and Bayesian Networks [16]. Finally, [17] have exploited Support Vector Machines and Decision Trees to evaluate and build Quality of Experience prediction models of Multimedia Streams. In a less specific field of research, like general traffic classification, we have witnessed a huge number of works using the most various Machine Learning techniques. Clustering algorithms have been exploited in [18],[19],[20] and [21], in order to group unlabeled data in traffic classes, often connected
7.2 Related Work

to applications. Classification algorithms, instead, have been adopted in [22], [23] and [24]. Precisely [25] focuses on real-time classification. Finally, also Association Rules and Frequent Itemset Mining have been employed as an effective tool to highlight and summarize the most important features of large datasets ([26], [27]).

In the last years, applying machine learning techniques in the network context has often entailed working on huge amount of data like network traffic datasets (e.g. Tstat [28], NetFlow [29], etc.). These types of databases are often so large that they are a typical example of Big Data. In these cases, traditional approaches are starting to show their limitations. Furthermore, the shift towards horizontal scaling in hardware has highlighted the need of parallelization in Machine Learning techniques. Hence, recently, an increasing number of researchers have adopted the MapReduce programming paradigm. Designed to simplify the processing of large databases, MapReduce main idea is to share the processing of the data in independent parallel tasks; Hadoop [30] is one of the most widely diffused MapReduce framework. We can find an increasing number of MapReduce implementations of the most important Machine Learning algorithms.

The Frequent Itemset Mining problem has been addressed by [31] and [32]: these works present distributed implementation of the most popular Frequent Itemset Mining algorithms. Even Clustering problems have been addressed by many researchers in the last years. [33] and [34], have focused on K-means-based [35] distributed implementations while [36] takes particularly into account data exchanged between nodes.

The most important milestone among the distributed classification algorithms, instead, is certainly Planet [37]: they developed a MapReduce implementation of a Regression Tree in the context of predictions over ads.

7.2.4 Quality of Experience

In the recent past, big networks have been tested in an objective and tangible way by taking into account a number of variables that determine the network. As we’ve seen in the previous sections, such measures compose the Quality of Service (QoS) of the network. The term QoS refers to the ability of the network to achieve a more deterministic and is crucial
for the minimum-required quality delivery in many ways, including data transported with a
minimum packet loss, delay, maximum bandwidth, minimum and stable jitter and so on. The
aggressive adoption of QoS-ensuring formulas in network monitoring and management, oc-
curred simply because it was cost-effective and was enough for the Internet services of that
time.

With the explosion of information and real time Internet application, QoS management
is not enough to ensure end-user satisfaction. One could argue that QoS does not consider
the user’s perception. This important gap, is filled with techniques that take into account the
user’s opinion and compose Quality of Experience (QoE).

The QoE is an end-to-end subjective metric that involves human influence factors. It com-
bines and bonds together user perception, user expectations, and QoS measurements that
result in the experience of the end application and network performance. It essentially de-
scribes the satisfaction of the user in respect with the service. To that extent ISPs and service
providers need to know whether their network is delivering the right QoE to the end users.
They need to integrate QoE paradigms into their traffic management system. This is why
the empowerment of more thorough understanding of quality as perceived by end-users is
receiving ever increasing attention from network operators and large organizations.

Developing and implementing a QoE-aware system requires passing over several chal-
lenges, as QoE relies on both complex non-deterministic human metrics and technical factors
such as:

- Context, which can include access type, movement (mobile/stationary), location, envi-
  ronment, social and cultural background, purpose of usage

- User factors such as expectations, requirements, perception, psychological factors etc.

- Application factors such as Application type, for example Web access, VoIP, streaming
  stored video and media, live video streaming

- Application QoS measures such as initial delays, video (stalls), video buffering strate-
gies, transport characteristics such as UDP, HTTP/TCP
7.2 Related Work

- Network-level QoS such as delay, and jitter, bitrate, packet loss, etc.
- Content, such as video codec, format, resolution, duration, content and type of video.
- End-device characteristics such as , device performance, user interface
- Service characteristics such as reliability, availability, etc.

Interestingly, QoE can be approximated under both subjective and objective approaches. Heavy user involvement is substantial for subjective measurements as required by the ITU-T standard Mean Opinion Score (MOS), where the users are asked to quantify the quality using a 5-point scale scoring system in the form of 5-Excellent, 4-Good, 3-Fair, 2-Poor, 1-Bad. Realistically subjective tests have to go further than that. To increase accuracy server other techniques are uses such as crowdsourcing. Crowdsourcing requires the installation of special software on end-devices that measure several aspects of user behaviour during the usage of the service. In the case of video streaming it is interesting to investigate how many times and at what point the user has stopped the video due to bad quality, has restarted the video due to jitter etc. After that studies can be conducted on laboratory environment where the validation of the crowdsourcing test results and the filtering of unreliable user ratings takes place.

Objective QoE measurements estimates the QoE with parametric models, with little to no user involvement. In that case the parametric model is usually a function of the network-level QoS . It is obvious that the complexity of this task rises as a QoE-aware system has to define (a)which factors influence the QoE and (b)accurate mapping between QoE measurements and QoS metrics. The functions that describe the mapping of Network-Level QoS to QoE can behave differently for different types of applications. Such functions can be [38]: linear, exponential, logarithmic, power.

There are several QoE/QoS correlation models for Internet Services in recent studies. One of the most used ones is the IQX hypothesis In [39], the authors assume a linear dependence on the QoE level following the differential equation:

\[ \frac{\partial QoE}{\partial QoS} \sim -(QoE - \gamma) \]
The above equation is solved by a generic exponential function called IQX hypothesis, which suggest that QoE and QoS are related in this manner:

\[ QoE = a \ast (\exp(-b \ast QoS) + c) \]

where a,b,c are positive parameters and have to do with several application settings. In this study, the QoE has been considered as the MOS, while the QoS has been represented from variables like packet loss, jitter, response and download times. In [40] there is a significant example that focuses on YouTube QoE and utilizes the IQX hypothesis with results shown in 7.1.

\[ f(x) = \alpha e^{-\beta x} + \gamma \text{ or } QoE = \alpha e^{-\beta QoS} + \gamma \]

In [41] a more holistic objective approach is proposed that considers a multidimensional QoS/QoE mapping for video applications, where the QoE is measured by the Video Quality Metric (VQM) [42]. Thus, the VQM is of n-dimensional a function of several QoS parameters.

\[ VQM = f(x_1, x_2, x_3, x_4, \ldots, x_n), x_i \text{ is a QoS parameter} \]
In [43] a method that only relies on limited user involvement is considered. Viewers were presented with the a video sample where the testers alter the video quality. The users where asked to point the moment where the change of quality became noticeable by using the method of limits [44]. After that, several statistical analysis methods where used such as Discriminate Analysis, to produce a prediction model that maps QoS changes to QoE changes. The aforementioned methods were evolved in more recent studies, such as in [45] in which QoE models were rebuilt to more complete prediction systems, utilizing Machine Learning methods, such as of Decision Trees [46] and Support Vector Machines (SVM) [47]. A very appealing work is done in [48] where SVMs and DTs are used to create a QoE model and are compared against other Machine Learning methods like Naive Bayes, kNearest Neighbours, Random Forest and Neural Networks.

7.3 A Telecom Analytics Framework

We propose a framework that will make use of ML clustering and classifying algorithms in order to characterize traffic in real time. The characterization of traffic will evolve in time according to traffic patterns and, with the information acquired per class of traffic, the treatment in every packet entering in real time to the core of the network will fit the needs of both users (e.g. give preference to the traffic most important at that moment for a customer) and network (e.g. if congestion is detected, incoming traffic will have different dropping precedence in order to less contribute to a worse congestion). ONTIC framework will be focused on the core of the ISP network, so we will consider that traffic arriving is originated both from Mobile and Fixed networks. 7.2 shows a high level description of the main modules that will contain ML algorithms for QoS/QoE dynamic management and traffic characterization mechanisms.

According to this figure, the first module receiving traffic is called Enforcement Point (EP), a node that will do real-time processing of incoming traffic. As we may assume, this module will be in charge of classifying the traffic for QoS/QoE management and make a first treatment of the traffic depending on (1) the characteristic of the traffic, (2) the state of the network (i.e., if it is congested at that moment or not many network resources are available) and (3) the profile of the users that are using the network.
For EP to gather enough information to make decisions on the traffic, it will make use of Policy Controller. This PC node will contain the results of the Machine Learning algorithms contained in the third node, the Analytics node. PC will hold (1) the information that classifies the packets in flow aggregates with similar characteristics and (2) the policies to apply in each flow aggregate in the different situations of the network that EP module can find.

Finally, Analytics module is the node in charge of performing ML algorithms on the traffic. This will be the novelty in the framework: Analytics module will generate knowledge by studying (1) the traffic that has traversed the network (both stored in a Database or in real time) and (2) studying how users of the network use their traffic. The conclusions of these studies will be delivered to PC module so that actions performed by EP are continuously up to date.

In the next sub sections we are going to see in deeper detail each one of the modules.
7.3 A Telecom Analytics Framework

7.3.1 Analytics

As we have already seen, this module will use ML to generate for the framework updated knowledge about the traffic and the users of the network. 7.3 shows the main modules to achieve this knowledge.

![Analytics module architecture](image)

Figure 7.3: Analytics module architecture

The source point of the module is the traffic that is going through the network if the algorithms are doing online analysis or a database with the records of the packets if it is offline analysis. In both cases, they are represented by Traffic Storage. With this network traffic, the node follows two different paths: one for flow clustering and another for user profile clustering.

The Flow Clustering aims to achieve a clusterization of traffic so that flows remain grouped by flow parameters with direct implications on QoS such as minimum, mean, maximum or standard deviation of packets length, inter-arrival times or duration of the flows. Tstat tool [28] can extract a big amount of flow features that could be included for flow clustering and the survey made in 2008 [1] makes a selection of the most useful features used for this task.
However, we will need to study which features best group the flows for QoS management purpose. The other use of traffic will serve to detect user profiles. A first step performed by User Detection module will determine which packets belong to which user of the net; once the different users are grouped with their traffic, Usage Detection module will create relations between flow clusters and user flows. The result of these relationships will be a vector per user containing weights of usage of traffic clusters called “Traffic usage” in the figure. In this way:

\[
\text{User } j \text{ traffic usage } = [w_{c1}w_{c2}w_{c3} \ldots w_{cn}]
\]

Where \(w_{ci}\) is the weight of the usage of the cluster \(i\) and \(\sum_{i=1}^{n} w_{ci} = 1\).

Finally, once it is known how users make use of the clusters, we can use ML algorithms in the User Profile clustering to create user profiles.

### 7.3.2 Policy Controller (PC)

Policy Controller will create the instructions for EP to best treat traffic in every situation. It will use both flow and user profile clusters from Analytics module and generate classification and shaping rules for EP node.

When both flow and user profile clusters have been obtained, it is important to create associations between both sets of clusters. Initially, two sets of rules have been defined: (1) classification rules, the will help EP to group flows with similar characteristics; and (2) shaping rules, that will help EP to decide how to best let traffic enter in case of different situations in the network (e.g. congestion detection or major popularity of certain users in the network). PC Rules Generator module will be in charge of this task, and it is intended to be a manual process at the beginning of the development and turn automatic when enough experience is acquired.

The most important aspect to determine in this module is how often the rules should be updated. To decide this, a study on how often patterns change both in traffic and in user profiles can help to obtain conclusions.
7.3 A Telecom Analytics Framework

7.3.3 Enforcement Point (EP)

Finally, in this architecture Enforcement Point will be the element that will take real-time decisions on the traffic, with help of all the information gathered in both PC and Analytics modules. 7.5 shows the main modules that are part of the EP architecture.

As we can see in the architecture, the packets will first arrive to flow detection module. Here, there will be an association between the packets and the current flows they are part of. From this point, the system will treat each packet as a part of a certain flow.

Once the packets are related to flows, the classifier and marker module will make use of classification rules from PC to determine to which cluster that packet will belong. We have seen that each cluster will be characterized by features directly related to QoS, so this classification is important to properly treat the packet in the next step. With the classification, the DSCP of the packet will be changed to indicate which class the packet is part of. This way, in the core of the network the packet will be able to receive appropriated treatment too.

Finally, shaper module will make use of meter and shaping rules to decide which treatment the packet must receive (e.g. if the packet must enter the network, must be buffered or must be dropped). As the decisions must be taken per flow, the meter will take measures of every
packet and associate these measures to the flow, so that it can have a control, for instance, of how many packets of one flow have passed through, what max, min or mean inter arrival time currently exists in that flow or the wideband associated.

By combining both meter measures and shaping rules, the packet enters to the network (or not) by accomplishing a balance between user satisfaction and network capacities in every moment.
7.4 The Process Architecture for Online and Offline Scalability

The proposed implementation model is based on three main criteria: the type of processing to be performed, the nature of the input data and the need for system scalability.

Taking into account the description of the scenario, detailed in Section 3, the above criteria are particularized as follows:

- The “Enforcement Point” process is executed for each packet coming from the network, should be run online and has to be scalable in terms of maximum network packet rate (this rate is expected to grow in the future). Therefore this process requires a real-time implementation with an execution cycle equal or less than the minimum time between consecutive packet arrival \( T_{\text{exec}} \leq \frac{1}{\nu} \).

- Although the ”Analytics” module also processes packets arriving from the network, this may need to consult old packages, so it will take an offline model in which scalability should be achieved in terms of volume of input data (data which will be stored in a repository).

- In both cases the processing of the packets has a sequential structure or pipeline of tasks.

7.4.1 Development and Implementation Model

In order to carry out the implementation of the scenario, according to their characteristics (sequential input data and pipelines of tasks) and needs (scalability and real-time execution), the use of a distributed real-time computing environment is necessary. This environment will have the following features:

- Capacity to continuous execution of sequential input data (input data streaming).

- Capacity of parallel execution of both processes (for example independent JVMs) and tasks (Threads).
• Must ensure the processing of every input data.

• It will provide the opportunity to design each program as a set of tasks connected with each other through input and output data connectors. This topology of tasks, at the same time, will be deployed automatically as a hierarchical structure formed by: processes that can be duplicated and can be executed in parallel on different machines of a cluster of servers, threads that allow for parallel sequences which share the execution space and finally tasks the same type that runs within each thread. The number of each of these three components (processes, threads and tasks) may be configured to increase the parallelism of the system designed according to their needs in terms of both number of processes and threads and the distribution of data between the parallelized tasks.

7.4.2 Implementation of the Scenario

The design of the process architecture for the "Dynamic QoS" scenario, as shown in 7.6, presents an structure of two main processing modules (Enforcement Point (EP) and Analytics) and an additional module called "Policy Controller" considered as a repository of the classification rules and which must be synchronized with the other two modules because the classification rules are produced in the Analytics module and are used in the EP module.

This architecture is presented in an integrated way in order to facilitate the execution of the two processing modules simultaneously, although the Analytics module runs in Offline mode taking data from a traffic repository while the EP module is running in online mode taking the packets directly from the network. The advantage of this parallel execution is that it gets the update of the classification rules in the EP module as frequently as possible.

Regarding the internal design of the modules, it is important the decision to split the Analytics module into three processes: Rules Generator Process, Flow Characterization Process and User Characterization Process. These processes are executed in parallel and synchronously by updating of the flow clusters.

The management of the input data is performed through the components called "injectors". These components include the required logic to maintain the sequential model of data arrival (streaming) and the logic necessary to group these data and distribute them to the tasks.
7.4 The Process Architecture for Online and Offline Scalability

The execution components are each of the tasks defined within a process (Flow Clustering, User Detection, Classifier + Marker, etc.) and it work synchronously driven by input data coming either from an injector or from another task.

Finally, once the structure of processes and tasks has been designed, the development environment will allow a deployment configuration which will provide the scalability that
the system needs. This configuration will allow the deployment of the same process on different nodes of the cluster available and, within each process, will allow the specification of a structure of threads which execute the tasks in parallel following different models which may vary from a simple load balancing (using round robin) up to the generation of data sets and the distribution of them to the different threads.

As an example for the deployment of the "Rules Generator" process (as shown in 7.7) we can define two different threads for the two data injectors (Flows injector and User Profiles injector) followed for sets of new parallel threads that process the input data from its injector (with an scalability according to the arrival rate of the data). Thus, if the system knew the number of clusters in each time, it could create the same number of threads for "rule generation” task, or otherwise, performing a static configuration of threads in accordance with the characteristics of the execution nodes. Moreover, taking into account the possibilities of data grouping provided by the environment, the entire process could be replicated in a dynamic cluster in which new process nodes could be added by implementing of a new performance monitoring process.

7.5 Conclusions and Future Work

In this paper we have seen that, along the evolution of Internet, different QoS management techniques have been used to face the demand of network resources. Since the beginning of Internet, a classification of traffic has always been considered, as IP Type of Service field shows and, later on, IntServ and DiffServ architectures have shown. In mobile communications it is also important to differentiate among the services that infrastructures are holding. However, the classification of traffic has never been dynamic, and nowadays it is difficult to place the growing number of services in the current classes for current QoS management techniques.

Machine Learning brings the chance to create dynamic classification and characterization of traffic in substitution of traditional techniques such as Deep Packet Inspection or port-based classification. In addition, Quality of Experience gives a new understanding of customers' needs and satisfaction by giving preference to what is important to user, not what user has
7.5 Conclusions and Future Work

Figure 7.7: Example of scalability for the module “Rules Generator”.

contracted with the ISP.

The proposed framework in this paper aims to handle Big Data ISP traffic and apply Ma-
chine Learning techniques in order to smartly manage Quality of Service. This framework
will contain an Analytics framework that will study traffic in order to achieve the optimal
classification both of the profile of the customers and the charac-
teristics of the flows. With
this knowledge, the Policy Controller module will create rules that may help Enforcement
Point to make intelligent decisions about how traffic should enter to the network in different
situations. Finally, this Enforcement Point will handle with Big Data traffic making use of
parallelization to best process the huge amount of data (about 200000 packets per second)
that will need to be treated.

In order to achieve the proposed goals for this framework, ONTIC project will focus his
efforts in: (1) making a selection of which ML algorithms best fit the needs of classification
and characterization of the traffic, as well as the input parameters for these algorithms (they
must be related to flow features); (2) study how to best classify users per profile and per the usage they do of each cluster generated; and (3) create policies that will help to generate both shaping and classifying rules. A further step in case 3 would be to generate an automation process that would generate these rules without human supervision.

In addition, ONTIC project wills to contribute in the development of the areas covering Machine Learning and Big Data, so the results of the investigation will be incorporated to tools such as Traffic Identification Engine (TIE) [38] in order to join efforts with other communities.

### 7.6 Acknowledgment

The authors would like to thank ERICSSON Spain R&I department for their collaboration in the definition of the high level architecture and the work done related to Quality of Experience and Business Support System.
Bibliography


BIBLIOGRAPHY


8 Author Index

Apiletti, Daniele 39
Baralis, Elena 39
Bascuñana, Alejandro 91
Cagliero, Luca 39
Casas, Pedro 53
Castro, Fabián 91
Cerquitelli, Tania 39
Costumero, Roberto 1
Chiusano, Silvia 39
Espadas, David 91
García-Pedrero, Angel 1
Garza, Paolo 39
Gonzalo, Consuelo 1
Grimaudo, Luigi 39
Guadamillas, Alvaro 103
López, Miguel Ángel 103
Maravitsas, Nikolaos 103
Mazel, Johan 53
Menasalvas, Ernestina 1
Monjas, Miguel Ángel 91
Mozo, Alberto 69, 103
Owezarski, Philippe 53
Pulvirenti, Fabio 39, 103
Sánchez, Isaac 1
Sánchez, Patricia 91
Theissler, Andreas 23
Zhu, Bo 69
8 Author Index