Lecture Notes in Computer Science 9951

Commenced Publication in 1973
Founding and Former Series Editors:
Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

More information about this series at http://www.springer.com/series/7408

Dieter Fiems · Marco Paolieri
Agapios N. Platis (Eds.)

Computer Performance Engineering

13th European Workshop, EPEW 2016
Chios, Greece, October 5–7, 2016
Proceedings

Springer

Editors
Dieter Fiems
Ghent University
Gent
Belgium

Agapios N. Platis
University of the Aegean
Chios
Greece

Marco Paolieri
University of Florence
Firenze
Italy

ISSN 0302-9743 ISSN 1611-3349 (electronic)
Lecture Notes in Computer Science
ISBN 978-3-319-46432-9 ISBN 978-3-319-46433-6 (eBook)
DOI 10.1007/978-3-319-46433-6

Library of Congress Control Number: 2015946767

LNCS Sublibrary: SL2 – Programming and Software Engineering

Printed on acid-free paper

This Springer imprint is published by Springer Nature
The registered company is Springer International Publishing AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

It is our pleasure to present the proceedings of EPEW 2016, the 13th European Performance Engineering Workshop, held October 5–7, 2016 in Chios, Greece.

The goal of this annual workshop series is to gather academic and industrial researchers working on all aspects of performance engineering. The papers presented at the workshop reflect the diversity of modern performance engineering, with topics ranging from the analysis of queueing networks and stochastic processes, to performance analysis of computer systems and networks, and even modeling of human behavior.

The call for papers gathered 25 submissions by authors from 13 countries. Each paper was peer reviewed by an average of three reviewers from the Program Committee (PC) on the basis of its relevance, novelty, and technical quality. After the collection of reviews, PC members discussed the quality of the submissions for one week before deciding to accept 14 papers.

This year, we were honored to have two keynote speakers. Prof. Kishor S. Trivedi from Duke University (USA) addressed current research on the quantitative analysis of network survivability. Prof. Nicholas Ampazis from the University of the Aegean (Greece) explored the use of deep learning approaches for performance analysis.

We thank our keynote speakers, as well as all PC members and external reviewers, who returned their reviews on time despite the tight reviewing deadline, and provided constructive and insightful comments. We also express our gratitude to the Organizing Committee at the University of the Aegean for their continuous and valuable help, the EasyChair team for their conference system, and Springer for their continued editorial support. Above all, we would like to thank the authors of the papers for their contribution to this volume, which we hope that you, the reader, will find useful and inspiring.

August 2016

Dieter Fiems
Marco Paolieri
Agapios N. Platis

Organization

Program Committee

Simonetta Balsamo	University of Venice Ca' Foscari, Italy
Marta Beltran	King Juan Carlos University, Spain
Marco Bernardo	University of Urbino, Italy
Laura Carnevali	University of Florence, Italy
Giuliano Casale	Imperial College London, UK
Vittorio Cortellessa	University of L'Aquila, Italy
Tadeusz Czachorski	Polish Academy of Sciences, Poland
Dieter Fiems	Ghent University, Belgium
Jean-Michel Fourneau	University of Versailles, France
Stephen Gilmore	University of Edinburgh, UK
András Horváth	University of Turin, Italy
Gábor Horváth	Budapest University of Technology and Economics, Hungary
Alain Jean-Marie	CNRS University of Montpellier, France
Helen Karatza	Aristotle University of Thessaloniki, Greece
William Knottenbelt	Imperial College London, UK
Spyros Kokolakis	University of the Aegean, Greece
Samuel Kounev	University of Würzburg, Germany
Vasilis Koutras	University of the Aegean, Greece
Lasse Leskelä	Aalto University, Finland
Catalina M. Lladó	University of the Balearic Islands, Spain
Andrea Marin	University of Venice Ca' Foscari, Italy
Marco Paolieri	University of Florence, Italy
Roberto Pietrantuono	University of Naples Federico II, Italy
Agapios Platis	University of the Aegean, Greece
Philipp Reinecke	HP Labs Bristol, UK
Anne Remke	WWU Münster, Germany
Sabina Rossi	University of Venice Ca' Foscari, Italy
Markus Siegle	Bundeswehr University Munich, Germany
Miklos Telek	Budapest University of Technology and Economics, Hungary
Nigel Thomas	Newcastle University, UK
Petr Tuma	Charles University, Czech Republic
Maaike Verloop	CNRS University of Toulouse, France
Jean-Marc Vincent	Joseph Fourier University, France
Demosthenes Vouyioukas	University of the Aegean, Greece

Joris Walraevens	Ghent University, Belgium
Katinka Wolter	Freie Universität Berlin, Germany
Armin Zimmermann	Technische Universität Ilmenau, Germany

Additional Reviewers

Nikolas Roman Herbst	University of Würzburg, Germany
Jóakim von Kistowski	University of Würzburg, Germany
Jürgen Walter	University of Würzburg, Germany

Invited Talks

Survivability Quantification for Networks

Kishor S. Trivedi

Department of Electrical and Computer Engineering,
Duke University, Durham, NC 27708, USA
ktrivedi@duke.edu

Abstract. Survivability is a critical attribute of modern computer and communication systems. The assessment of survivability is mostly performed in a qualitative manner and thus cannot meet the need for more precise and solid evaluation of service loss or degradation in presence of failure/attack/disaster. This talk addresses the current research status of quantification of survivability. First, we carefully define survivability and contrast it with traditional measures such as reliability, availability and performability [2, 8, 7]. We use "survivability" as defined by the ANSI T1A1.2 committee – that is, the transient performance from the instant an undesirable event occurs until steady state with an acceptable performance level is attained [1]. Thus survivability can be seen as a generalization of recovery after a failure or any undesired event [3]. We then discuss probabilistic models for the quantification of survivability based on our chosen definition. Next, three case studies are presented to illustrate our approach. One case study is about the quantitative evaluation of several survivable architectures for the plain old telephone system (POTS) [5]. The second case study deals with the survivability quantification of communication networks [4] while the third is that of smart grid distribution automation networks [6]. In each case hierarchical models are developed to derive various survivability measures. Numerical results are provided to show how a comprehensive understanding of the system behavior after failure can be achieved through such models.

References

1. ANSI T1A1.2 Working Group on Network Survivability Performance: Technical report on enhanced network survivability performance. ANSI, Tech. Rep. TR No. 68 (2001)
2. Avizienis, A., Laprie, J., Randell, B., Landwehr, C.E.: Basic concepts and taxonomy of dependable and secure computing. IEEE Trans. Dependable Sec. Comput. **1**(1), 11–33 (2004)
3. Heegaard, P.E., Helvik, B.E., Trivedi, K.S., Machida, F.: Survivability as a generalization of recovery. In: 11th International Conference on the Design of Reliable Communication Networks. DRCN 2015, pp. 133–140 (2015)
4. Heegaard, P.E., Trivedi, K.S.: Network survivability modeling. Computer Netw. **53**(8), 1215–1234 (2009)
5. Liu, Y., Mendiratta, V.B., Trivedi, K.S.: Survivability analysis of telephone access network. In: 15th International Symposium on Software Reliability Engineering. ISSRE 2004, pp. 367–378 (2004)

6. Menasché, D.S., Avritzer, A., Suresh, S., Leão, R.M.M., de Souza e Silva, E., Diniz, M.C., Trivedi, K.S., Happe, L., Koziolek, A.: Assessing survivability of smart grid distribution network designs accounting for multiple failures. Concurrency and Comput. Pract. Exp. **26** (12), 1949–1974 (2014)
7. Meyer, J.F.: On evaluating the performability of degradable computing systems. IEEE Trans. Comput. **29**(8), 720–731 (1980)
8. Trivedi, K.S.: Probability and Statistics with Reliability, Queuing, and Computer Science Applications. Wiley (2001)

Deep Learning Models for Performance Modelling

Nicholas Ampazis

Department of Financial and Management Engineering,
University of the Aegean, Chios 82100, Greece
n.ampazis@fme.aegean.gr

Abstract. Deep learning approaches to performance modelling and prediction of computer systems can be considered as a "black-box" approach, where many layers of information processing stages in hierarchical neural networks architectures are exploited for feature learning in prediction or classification tasks. Examples of deep learning applications to performance modelling span from anomaly detection to optimization, to capacity planning, and, with the advent of cloud computing, to automatic resource provisioning.

Keywords: Deep learning · Machine learning · Neural networks · Performance modelling

1 Introduction

Deep Learning (DL) [1] is a rapidly growing discipline that, during the last few years, has revolutionalised machine learning and artificial intelligence research due to the availability of "big data", new algorithms for neural networks training, and extremely fast dedicated hardware. Companies like Google, Microsoft, Amazon, Facebook and Apple use deep learning to solve difficult problems in areas such as speech and image recognition, machine translation, natural language processing, resource planning or even to reduce power consumption by manipulating computer servers and related equipment like cooling systems [2].

The essence of DL is to compute hierarchical features or representations of observational data, where the higher-level features or factors are defined from primary lower-level measurements. Based on the features extracted from the data in the training set, the calculations within the model are adjusted so that known inputs produce desired outputs. The theory then extends to the fact that, similarly to classical machine learning, a trained deep learning system will correctly recognize the patterns when presented with new examples [7].

Deep learning can be seen as a more complete, hierarchical and a "bottom up" way for feature extraction without human intervention. In the past manually designed features were used in demanding tasks such as, for example, image and video processing. These rely on human domain knowledge and it is hard to manually tune them. Thus, developing effective features for new applications was a slow process. Deep learning overcomes this problem of feature extraction by adaptively determining operator coefficients, like for example in convolutional layers which are exceptionally good at discovering and extracting features from data. These features are propagated to the next layer to form a hierarchy of nonlinear features that grow in complexity (e.g. in an image processing task, from blobs/edges → noses/eyes/cheeks → faces). The final layer uses all these generated features for classification or regression. Deep learning can be thought of as "feature engineering" done automatically by algorithms [3, 6].

2 Applications

In applications of classical Machine Learning (ML) methods to performance modeling or prediction, it was sufficient to identify the core inputs (features) of the performance functions, and the ML algorithm would take care of inferring how they map to target Key Performance Indicators (KPI). Such models are built on the basis of a so called training phase, during which the application is tested with different workloads and is parameterized with different configurations, with the purpose of observing the corresponding achieved performance. Thus their advantage is that the task is reduced to fitting the input data to their desired output values without exploiting any additional knowledge about the application. However input features have to be manually crafted, e.g. small versus large jobs to encode workload intensity, number and types of servers to encode infrastructure, etc. Similarly, KPI outputs like throughput (e.g. max jobs/sec), response time (e.g. execution time of a job) or consumed energy (e.g. Joules/job) would have to carefully defined in order to discriminate the task as being a regression, a classification or a clustering problem.

Relative to other machine learning techniques, DL has four key advantages:

- It can detect complex relationships among features
- It can extract new low-level features from minimally processed raw data
- It can handle multiclass problems with high-cardinality
- It can produce results with unlabeled data

These four strengths suggest that deep learning can produce useful results where other methods may fail. It may also build more accurate models than other methods, and it can reduce the time needed to build a useful model.

Already DL is utilized in order to solve highly practical problems in all aspects of business. For example:

- Payment systems providers use DL to identify suspicious transactions in real time [5].
- Organizations with large data centers and computer networks use DL to mine log files and detect threats [8].
- Vehicle manufacturers and fleet operators use DL to mine sensor data to predict part and vehicle failure [9].
- Deep learning helps companies with large and complex supply chains predict delays and bottlenecks in production [4].

With the increased availability of deep learning software and the skills to use it effectively, we expect the list of commercial applications to grow rapidly in the next several years.

References

1. Bengio, I.G.Y., Courville, A.: Deep Learning. MIT Press (2016, in press)
2. Clark, J.: Google cuts its giant electricity bill with DeepMind-powered AI (2016). http://www.bloomberg.com/news/articles/2016-07-19/google-cuts-its-giant-electricity-bill-with-deepmind-powered-ai. Accessed 19 July 2016
3. Dettmers, T.: Deep learning in a nutshell: core concepts (2015). https://devblogs.nvidia.com/parallelforall/deep-learning-nutshell-core-concepts/. Accessed 18 May 2016
4. Ge, L.: Diving into deep learning: what deep learning could mean for the industrial economy (2015). http://gelookahead.economist.com/deep-learning. Accessed 27 May 2015
5. Harris, D.: How PayPal uses deep learning and detective work to fight fraud (2015). https://gigaom.com/2015/03/06/how-paypal-uses-deep-learning-and-detective-work-to-fight-fraud. Accessed 3 March 2015
6. Jaokar, A.: Evolution of deep learning models (2015). http://www.opengardensblog.futuretext.com/archives/2015/07/evolution-of-deep-learning-models.html?attest=truewpmp_tp=1. Accessed 17 May 2016
7. Kamadolli, S.: Getting real with deep learning (2015). https://medium.com/@kamadoll/getting-real-with-deep-learning-3b7ef698766d#.muip7g5o3. Accessed 18 May 2016
8. Najafabadi, M.M., Villanustre, F., Khoshgoftaar, T.M., Seliya, N., Wald, R., Muharemagic, E.: Deep learning applications and challenges in big data analytics. J. Big Data **2**(1), 1–21 (2015)
9. Okanohara, D.: Deep learning in real world: automobile, robotics, bio science (2015). http://hiroshi1.hongo.wide.ad.jp/hiroshi/files/internet/okanohara_2016.pdf. Accessed 27 June 2016

Contents

Modeling and Analysis of Human Behavior

Modeling and Simulation Tools