UNIVERSIDAD DE MÁLAGA

DOCTORAL THESIS

Deep Neuroevolution: Smart City Applications

Author:
M.Sc. Andrés CAMERO

Supervisor:
Prof. Dr. Enrique ALBA

A thesis submitted in fulfillment of the requirements for the degree of Doctor of Philosophy

in the

Programa de Doctorado en Tecnologías Informáticas
Departmento de Lenguajes y Ciencias de la Compuación
E.T.S.I. Informática

February 12, 2021

UNIVERSIDAD DE MÁLAGA

UNIVERSIDAD
DE MÁLAGA

AUTOR: Andrés Camero Unzueta

http://orcid.org/0000-0002-8152-9381

EDITA: Publicaciones y Divulgación Científica. Universidad de Málaga

Esta obra está bajo una licencia de Creative Commons Reconocimiento-NoComercial-SinObraDerivada 4.0 Internacional:
http://creativecommons.org/licenses/by-nc-nd/4.0/legalcode
Cualquier parte de esta obra se puede reproducir sin autorización
pero con el reconocimiento y atribución de los autores.
No se puede hacer uso comercial de la obra y no se puede alterar, transformar o hacer obras derivadas.

Esta Tesis Doctoral está depositada en el Repositorio Institucional de la Universidad de Málaga (RIUMA): riuma.uma.es

UNIVERSIDAD
DE MÁLAGA

UNIVERSIDAD DE MÁLAGA

M.Sc. Andrés Camero Unzueta, estudiante del programa de doctorado Tecnologías Informáticas de la Universidad de Málaga, autor de la tesis, presentada para la obtención del título de doctor por la Universidad de Málaga, titulada:

Deep Neuroevolution: Smart City Applications

Realizada bajo la tutorización y dirección del Prof. Dr. Enrique Alba Torres.

Declaro que

La tesis presentada es una obra original que no infringe los derechos de propiedad intelectual ni los derechos de propiedad industrial u otros, conforme al ordenamiento jurídico vigente (Real Decreto Legislativo 1/1996, de 12 de abril, por el que se aprueba el texto refundido de la Ley de Propiedad Intelectual, regularizando, aclarando y armonizando las disposiciones legales vigentes sobre la materia), modificado por la Ley 2/2019, de 1 de marzo.

Igualmente asumo, ante a la Universidad de Málaga y ante cualquier otra instancia, la responsabilidad que pudiera derivarse en caso de plagio de contenidos en la tesis presentada, conforme al ordenamiento jurídico vigente.

Fecha y firma:

UNIVERSIDAD DE MÁLAGA

This page intentionally left blank.

UNIVERSIDAD DE MÁLAGA

UNIVERSIDAD DE MÁLAGA

Prof. Dr. Enrique Alba Torres, perteneciente al Departamento de Lenguajes y Ciencias de la Computación de la Universidad de Málaga,

Certifica

que, M.Sc. Andrés Camero Unzueta, Magíster en Ingeniería del Software e Inteligencia Artificial por la Universidad de Málaga, ha realizado en el Departamento de Lenguajes y Ciencias de la Computación de la Universidad de Málaga, bajo su dirección, el trabajo de investigación correspondiente a su Tesis Doctoral (por compendio de artículos) titulada:

Deep Neuroevolution: Smart City Applications

Revisado el presente trabajo, estimo que puede ser presentado al tribunal que ha de juzgarlo. Y para que conste a efectos de lo establecido en la legislación vigente, autorizo la presentación de esta Tesis Doctoral en la Universidad de Málaga.

Fecha y firma:

UNIVERSIDAD DE MÁLAGA

This page intentionally left blank.

UNIVERSIDAD DE MÁLAGA

"*For I know the plans I have for you"*, declares the Lord, "*plans to prosper you and not to harm you, plans to give you hope and a future."*

Jeremiah 29: 11

UNIVERSIDAD DE MÁLAGA

This page intentionally left blank.

UNIVERSIDAD DE MÁLAGA

UNIVERSIDAD DE MÁLAGA

Preface

E.T.S.I. Informática
Departmento de Lenguajes y Ciencias de la Compuación

Doctor of Philosophy

Deep Neuroevolution:
Smart City Applications

by M.Sc. Andrés CAMERO

In the last decade the emergence of Deep Learning and deep neural networks has revolutionized society. More and more applications are making use of it, from autonomous cars to robotic assistants, and our cities have started to adopt this new technology to become smart. However, as these networks grow, we face new challenges. One particular problem is the search for an optimal network design, i.e., finding the network that best suits a given problem. This includes the architecture, the weights, the neuron types, and even the hyper-parameters of the algorithm used to train the network. Various approaches have been proposed to address this problem, but in practice few have been adopted because of the excessive computational time they require. In recent years a new approach has emerged: deep neuroevolution. Based on the use of advanced metaheuristics, it offers a new way of solving the neural network optimization problem as a whole. These techniques have allowed finding unprecedented solutions and in increasingly reasonable times, opening the way to its practical adoption. In this doctoral thesis, we present a novel approach to optimizing a recurrent neural network, a type of deep neural network especially suited for solving problems that require learning temporal dependencies, to a given problem. Specifically, we propose a solution based on deep neuroevolution and on the mean absolute error random sampling, a training-free technique to characterize and compare neural networks, and we validate our proposal on several Smart City related problems.

UNIVERSIDAD DE MÁLAGA

This page intentionally left blank.

UNIVERSIDAD DE MÁLAGA

Acknowledgements

This thesis summarizes the four years of hard work spent to become a *Doctor*. Nevertheless, this could have not be possible without the invaluable help I have received during this time.

First, I would like to thank Prof. Dr. Enrique Alba for being my supervisor and friend during this journey. He has provided me with his technical advice and comments, but most important, he has been a good companion, always open to discussing *the good, the bad, and the ugly* of science and life. Also, I am grateful for having shared time, ideas, discussions, and coffees with my NEO Lab fellows. In particular, I would mention: Christian Cintrano, Javier Ferrer, Jamal Toutouh, José Ángel Morell, Amr Abdelhafez, and Rubén Saborido.

Second, I wish to express my gratitude to Universidad de Málaga, Andalucía Tech, Consejería de Economía y Conocimiento de la Junta de Andaluía, Ministerio de Economía, Industria y Competitividad, Gobierno de España, and European Regional Development Fund grant numbers TIN2017-88213-R (6City), and UMA18-FEDERJA-003 (Precog), and to the Helmholtz Association's Initiative and Networking Fund (INF) under the Helmholtz AI platform grant agreement (ID ZT-I-PF-5-1), for supporting my research.

Third, I would like to thank my family, especially to my mother, Maria Teresa, and my father, Ramiro, who I am sure is rejoicing in Heaven. To Carmen, who has been a second mother, and to my beloved children, Ignacio and Antonio (my personal AI), who delight me every day with their smiles.

And last, but not least, I would like to thank God for crossing my path with Catherine, my wife, the best partner ever...

UNIVERSIDAD DE MÁLAGA

This page intentionally left blank.

UNIVERSIDAD DE MÁLAGA

Contents

UNIVERSIDAD DE MÁLAGA

UNIVERSIDAD DE MÁLAGA

List of Figures

UNIVERSIDAD DE MÁLAGA

UNIVERSIDAD DE MÁLAGA

List of Tables

UNIVERSIDAD DE MÁLAGA

This page intentionally left blank.

UNIVERSIDAD DE MÁLAGA

List of Abbreviations

ACO	**A**nt **C**olony **O**ptimization
ANN	**A**rtificial **N**eural **N**etwork
BO	**B**ayesian **O**ptimization
BP	**B**ack**p**ropagation
CMA-ES	Standard **C**ovariance **M**atrix **A**daptation **E**volution **S**trategy
CNN	**C**onvolutional **N**eural **N**etwork
CS	**C**omputer **S**cience
DB	**D**ata **B**ase
DE	**D**ifferential **E**volution
DL	**D**eep **L**earning
DLOPT	**D**eep **L**earning **OPT**imization
DN	**D**eep **N**euroevolution
DNN	**D**eep **N**eural **N**etwork
EA	**E**volutionary **A**lgorithm
EC	**E**volutionary **C**omputation
ES	**E**volutionary **S**trategy
EXALT	**E**volutionary e**X**ploration of **A**ugmenting **L**STM **T**opologies
EXAMM	**E**volutionary e**X**ploration of **A**ugmenting **M**emory **M**odels
FFN	**F**eed**f**orward **N**etwork
GA	**G**enetic **A**lgorithm
GAN	**G**enerative **A**dversarial **N**etworks
GRU	**G**ated **R**ecurren **U**nit
HS	**H**armonic **S**earch
IoT	**I**nternet **o**f **T**hings
IT	**I**nformation **T**echnology
JCR	**J**ournal **C**itation **R**eports
LB	**L**ook **B**ack
LSTM	**L**ong **S**hort **T**erm **M**emory
MAE	**M**ean **A**bsolute **E**rror
MAPE	**M**ean **A**absolute **P**ercentage **E**rror
ML	**M**achine **L**earning
MRS	**M**ean absolute error **R**andom **S**ampling
MSE	**M**ean **S**quared **E**rror
mSSEA	**m**icro **S**teady **S**tate **E**volutionary **A**lgorithm
NA	**N**ot **A**vailable
NAS	**N**eural **A**rchitecture **S**earch
NEAT	**N**euro **E**volution of **A**ugmenting **T**opologies
PDF	**P**robability **D**ensity **F**unction
PSO	**P**article **S**warm **O**ptimization
RNN	**R**ecurrent **N**eural **N**etwork
RESN	**R**andom **E**rror **S**ampling-based **N**euroevolution
RMSE	**R**oot **M**ean **S**quared **E**rror
SA	**S**imulated **A**nnealing

UNIVERSIDAD DE MÁLAGA

SC	**S**mart **C**ity
SD	**S**tandard **D**eviation
SGD	**S**tochastic **G**radient **D**escent
SJR	**S**CImago **J**ournal **R**ank
SMReg	**S**upport vector **M**achine for **Reg**ression
ssGA	**S**teady **S**tate **G**enetic **A**lgorithm
SVM	**S**upport **V**vector **M**achine
WP	**W**ork **P**ackage

UNIVERSIDAD DE MÁLAGA

To my beloved wife and children

UNIVERSIDAD DE MÁLAGA

This page intentionally left blank.

UNIVERSIDAD DE MÁLAGA

Chapter 1

Introduction

In recent decades, thanks to the rapid increase in computing power and data storage, there has been a resurgence of interest in developing artificial neural networks (ANN). The "old" techniques, combined with this new computational power have brought new applications with amazing results, ranging from autonomous cars to robotic personal assistants [59, 72, 76].

To a large extent, this resurgence has been driven by the *Deep Learning* (DL) [76]. In plain English, the DL consists of designing and training an ANN ("the old") of greater complexity and size, with an immense amount of data ("the new"). However, as these networks grow, they become more complex. So, a big open question is: How do we design a *deep neural network* (DNN)?

Traditionally, designing ANN has been the task of human experts. Thanks to their problem domain and technical expertise, they can accomplish this task. Nonetheless, this optimization is rather a trial and error work. Thus, as the complexity of a DNN increases, this way of designing becomes inefficient (and impractical). The situation gets even worse as the amount of data used to train the network grows. For instance, just training a single DNN can take days. Therefore, the whole optimization process could take months. Then, it is *mandatory* to improve the design process [48, 76, 96].

Given the special importance of finding an optimal design in the overall performance, including both the result and the learning process, numerous proposals have been made to automate the search for a solution [48, 96]. However, most proposals are (still) based on training trial and error. In other words, a design is chosen, the network is trained (generally using a gradient descent-based technique), and the result is evaluated. While this approach is capable of delivering good solutions, it is very costly, and therefore, underutilized in practice.

An intuitive solution to reduce the optimization cost is to limit the training effort (e.g., to train for a shorter time or to stop the training earlier if a condition is met). This approach allows reducing the time significantly. However, given the high complexity of the DNNs, it is possible that a good design is discarded because it is *slow to learn*, i.e., the model needs a longer training time to achieve its optimal performance.

Looking for alternatives to avoid this costly optimization process, some authors [25, 89, 116] have proposed to extend an idea originally introduced in the 1990, the *Neuroevolution* [127]. Neuroevolution is in the crossroads of machine learning (ML) and evolutionary computation (EC), and consists of evolving the design of an ANN from scratch to a fully usable one. Thus, this extension, known as *Deep Neuroevolution* (DN), consists of the adaptation of the "old" neuroevolution [127] to the challenges of the DL [76].

Using metaheuristics (e.g., genetic algorithms, ant colonies, simulated annealing) it is possible to navigate through the design space, thus finding the network that best fits a given problem [97]. While at first glance this is also a trial and error approach, there is a big difference: DN is a targeted search that exploits the topological

UNIVERSIDAD DE MÁLAGA

characteristics of the problem, and it is not (necessarily) based on traditional network training (i.e., gradient descent-based learning). Therefore, the DN opens a path that is expected to improve the current results.

In this doctoral thesis, we present our DN approach to optimize the design of an DNN, and by using real applications in the domain of the *Smart City* (SC), we show its performance. The remainder of this section introduces the plan, goals, and methodology of this work. Section 2 briefly reviews related work. Sections 3, 4, 5, and 6 summarize the contributions of this work. Finally, Section 7 outlines the conclusions.

1.1 Work Proposal

In this doctoral thesis, we propose to address the problem of DNN design by DN, paying special attention to *recurrent neural networks* (RNN), because RNNs are intrinsically the deepest networks. Specifically, the objective of this thesis is the development of a *hybrid* DN technique, that is, one that uses both *traditional* ML methods, and techniques based on metaheuristics. In this way, we seek to take advantage of the benefits of both strategies, in order to obtain the best possible result.

In particular, we focus our work on the study of the intrinsic properties of RNNs, that is, we look for a simple and efficient way to characterize and compare RNN architectures. Then, we propose to use this information as input (the "heuristics") to define a metaheuristic algorithm for RNN optimization. Finally, once the best network for the problem has been chosen, we propose to train this network with state-of-the-art methods. It is important to note that in a tentative way we do not limit the training to BP or to the use of metaheuristics, but we will look for the best way to solve the problem.

The following list enumerates the main goals of this doctoral thesis:

G1 Define a method for characterizing RNNs that allows for comparison of its expected performance without the need for training.

G2 Design and implement a technique to optimize an RNN based on metaheuristics, and the previously mentioned characterization.

G3 Apply the RNN optimization technique proposed to problem-solving in the context of the SC.

G4 Disseminate the results by developing (and making publicly available) a software library with the mentioned methods and applications.

It should be noted that as part of the guidelines of this work, we propose to use the tool *TensorFlow* [1], an open source library for machine learning used (and initially developed) by Google. The background idea is, besides reusing the DNN implementation, to maximize the impact of the G4 goal.

1.2 Thesis Methodology

To accomplish the defined goals, we propose to carry out a theoretical and empirical study of the optimal design of an RNN. Particularly, we planned a 36-month work plan. The activities were grouped into 6 work packages (WP), aiming to fulfill the defined goals. The WP of the *project* are described below:

UNIVERSIDAD DE MÁLAGA

WP0 *Project Management*: Activities related to the control and planning of the doctoral thesis, as well as the bureaucratic activities required for its presentation.

WP1 *Documentation and Dissemination*: All the activities related to publishing, collaborating, and advertising the results of this thesis. Also, the preparation of the doctoral thesis compendium.

WP2 *Characterization*: Activities leading to the development of an RNN architecture characterization (goal **G1**).

WP3 *Optimization*: Development of a technique for optimizing the design of an RNN based on metaheuristics and the defined characterization (goal **G2**).

WP4 *Smart City*: Grouping of activities related to the study of three real problems in the context of the SC, and their resolution through the use of an optimized RNN with the defined technique (goal **G3**). As part of this activity, the specific problems to be addressed will be defined.

WP5 *DLOPT*: Activities related to the development and maintenance of an open source software library incorporating the results of this work (goal **G4**).

The packages **WP0** and **WP1** are support functions for the doctoral thesis. Thus, their life cycle is linked to the duration of the whole thesis. Figure 1.1 shows the WP, their main activities, and the relationship between WP. **WP2** and **WP3** may be considered as the technical basis of the thesis, while **WP4** is the application. Also, is is important to remark that the output of **WP5** will encapsulate all the technical and application developments of this thesis. The activities that appear with dotted lines are considered desirable for the achievement of this doctoral thesis.

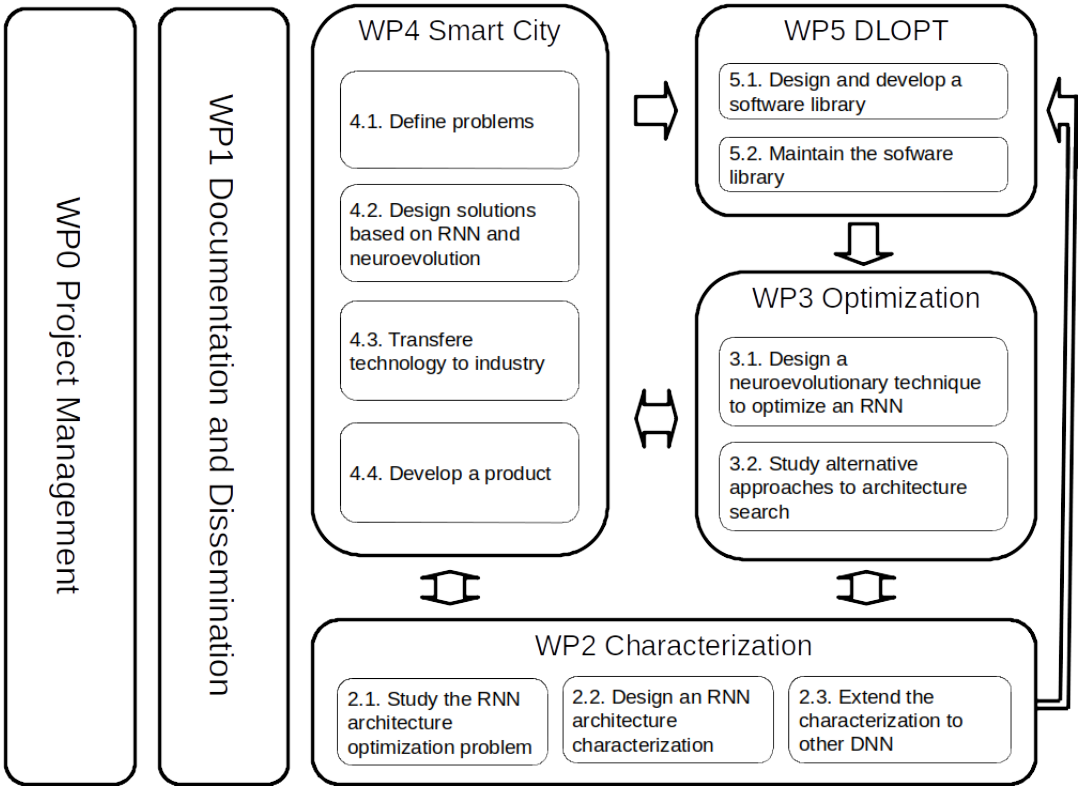


FIGURE 1.1: Work-packages, main activities, and their relations

UNIVERSIDAD DE MÁLAGA

Figure 1.2 shows the high-level planning, including the most relevant milestones of each WP (highlighted in orange on the calendar).

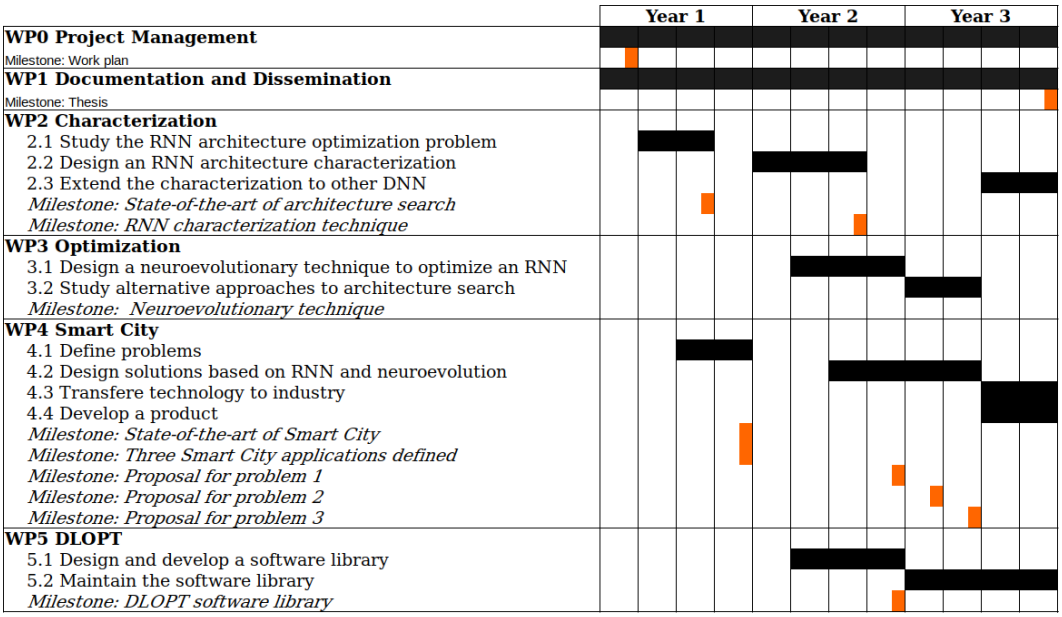


FIGURE 1.2: High-level plan. The milestones are highlighted in orange

1.3 Contribution

This thesis is presented as a *compendium* of publications. All the contributions of this work are in line with the goals of this doctoral thesis (Section 1.1), and pursue a common objective: improve the state-of-the art of science. Given the nature and relevance of the specific problems tackled here, we expect that its contribution has a positive impact on the research community and society.

Particularly, we divided the contribution of this thesis into four key aspects:

MRS We have proposed the Mean absolute error Random Sampling (MRS) technique to estimate the performance of an RNN (on a given problem) based on the distribution of the error observed on a random sampling (i.e., a training-free approach). We have empirically validated our proposal, showing that the MRS is a reliable low-cost estimate of the performance of an RNN [25, 31].

RESN We have introduced the Random Error Sampling-based Neuroevolution (RESN), an EA-based algorithm that exploits the MRS to optimize the architecture of an RNN. The results of our experiments show that we achieve a state-of-the-art performance, and that we reduce significantly the time needed to perform the optimization task [26].

UNIVERSIDAD DE MÁLAGA

DN and SC We have contributed to enlarge the scientific corpus of DN, SC, and AI applied to SC. Particularly, we have proposed alternatives to tackle smart mobility [20, 24, 29], smart electricity [27, 28], and smart waste management [21, 27, 30] problems. Also, we have reviewed the state-of-the-art of SC [23], and we have disseminated the work of this thesis in CAEPIA 2018 Doctoral Consortium competition [22].

DLOPT We have released the Deep Learning OPTimization (DLOPT) software library, our open source (GNU GPL v3 license) contribution to the DN community [19]. It is important to remark that most of the work related to this thesis is available in the DLOPT, thus we are also fostering the replication of our studies.

Figure 1.3 presents a general overview of this thesis, i.e., a *graphical abstract*. The key aspects of our contribution, the SC applications, and the techniques are put into perspective, and the most important relations are shown with different colors.

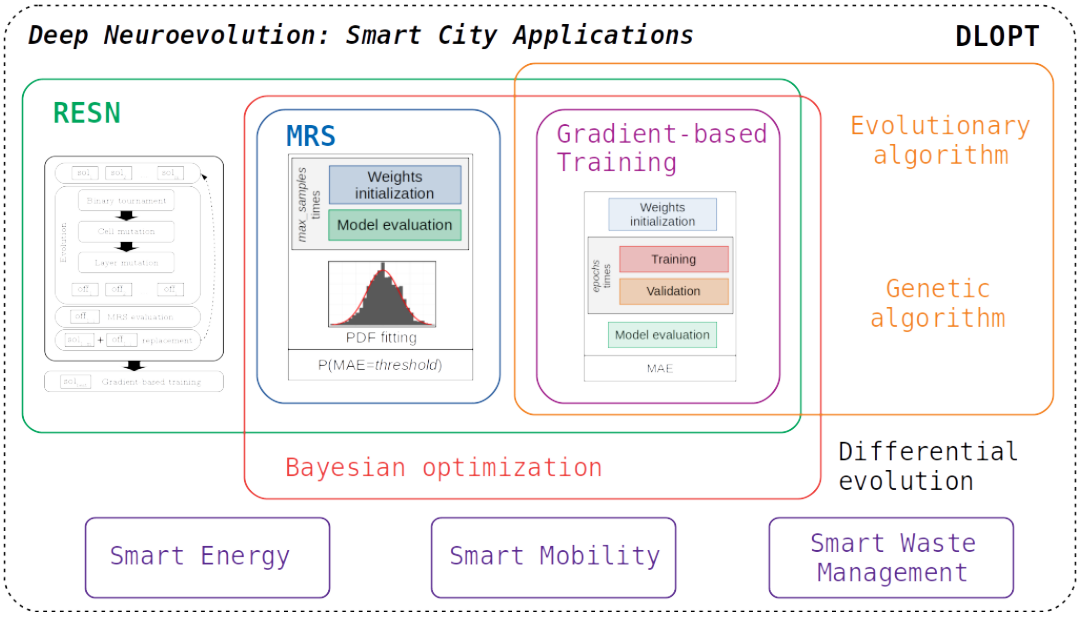


FIGURE 1.3: Deep neuroevolution: Smart city applications thesis overview

The following list enumerates the publications related to this thesis in chronological order. It is worth noticing that **four** of these works have been published (and one is in second round revision) in a **JCR-indexed journals**, **one** has been published in an **SJR-indexed journal**, **three** in the proceedings of an **international conference**, one was awarded the **Second Prize** for the best student work in Doctoral Consortium CAEPIA 2018, one has been published in the proceedings of a national conference, and two are available in arXiv. Moreover, up to this date, according to Google Scholar[1], these works have received **more than 130 citations**.

Along with the publication details, we have included the available metrics (2019) for each publication, i.e., the Journal Impact Factor (JIF), SCImago Journal Rank (SJR) indicator, and H5-index.

[1] https://scholar.google.de/citations?user=D5xGiRAAAAAJ&hl=en [Accessed: 1-Oct-2020]

UNIVERSIDAD DE MÁLAGA

1. A. Camero et al. "Tile Map Size Optimization for Real World Routing by Using Differential Evolution". In: *2017 IEEE Congress on Evolutionary Computation, CEC 2017 - Proceedings*. 2017. ISBN: 9781509046010. DOI: 10.1109/CEC.2017.7969478
   • H5-index = 68

2. Andrés Camero, Javier Arellano-Verdejo, and Enrique Alba. "Road Map Partitioning for Routing by Using a Micro Steady State Evolutionary Algorithm". In: *Engineering Applications of Artificial Intelligence* 71 (2018), pp. 155–165. ISSN: 09521976. DOI: 10.1016/j.engappai.2018.02.016
   • JCR COMPUTER SCIENCE, ARTIFICIAL INTELLIGENCE – SCIE, **Q1**, ranking 33/136, 2019 JIF = 4.201

3. Andrés Camero et al. "Customer Segmentation Based on the Electricity Demand Signature: The Andalusian Case". In: *Energies* 11.7 (2018), p. 1788. DOI: 10.3390/en11071788
   • JCR ENERGY & FUELS – SCIE, **Q3**, ranking 63/112, 2019 JIF = 2.702

4. Andrés Camero, Jamal Toutouh, and Enrique Alba. "Low-Cost Recurrent Neural Network Expected Performance Evaluation". In: *arXiv preprint arXiv:1805.07159* (2018)
   • Second round revision in Machine Learning, JCR COMPUTER SCIENCE, ARTIFICIAL INTELLIGENCE – SCIE, **Q2**, ranking 57/136, 2019 JIF = 2.672

5. Andrés Camero et al. "Evolutionary Deep Learning for Car Park Occupancy Prediction in Smart Cities". In: *Learning and Intelligent OptimizatioN Conference LION*. 2018. DOI: 10.1007/978-3-030-05348-2_32
   • H5-index = 12, and it has been **cited more than 40 times**

6. A. Camero et al. "Waste Generation Prediction in Smart Cities through Deep Neuroevolution". In: *Congreso Iberoamericano de Ciudades Inteligentes (ICSC-CITIES 2018)*. 2018. DOI: 10.1007/978-3-030-12804-3_15
   • First Ibero-American Congress on Information Management and Big Data

7. Andrés Camero, Jamal Toutouh, and Enrique Alba. "Comparing Deep Recurrent Networks Based on the MAE Random Sampling, a First Approach". In: *Advances in Artificial Intelligence, Conference of the Spanish Association for Artificial Intelligence (CAEPIA)*. ed. by Francisco Herrera et al. Cham: Springer International Publishing, 2018, pp. 24–33. DOI: 10.1007/978-3-030-00374-6_3
   • Lecture Notes in Artificial Intelligence, 18th Conference of the Spanish Association for Artificial Intelligence

8. Andrés Camero and Enrique Alba. "Neuroevolucion Profunda: Aplicaciones en Ciudades Inteligentes". In: *XVIII Conferencia de la Asociacion Española para la Inteligencia Artificial*. 2018, pp. 1387–1392 Conference of the Spanish Association for Artificial Intelligence
   • **Second Prize** for the best student work in Doctoral Consortium CAEPIA 2018

9. A. Camero, J. Toutouh, and E. alba. "DLOPT: Deep Learning Optimization Library". In: *arXiv preprint arXiv:1807.03523* (2018)
   • Software library available on GitHub (https://github.com/acamero/dlopt)

10. Andrés Camero and Enrique Alba. "Smart City and Information Technology: A Review". In: *cities* 93 (2019), pp. 84–94. DOI: 10.1016/j.cities.2019.04.014

UNIVERSIDAD DE MÁLAGA

• JCR URBAN STUDIES, **Q1**, ranking 2/42, 2019 JIF = 4.802, more than **35 citations**

11. Andrés Camero et al. "Waste Generation Prediction Under Uncertainty in Smart Cities Through Deep Neuroevolution". In: *Revista Facultad de Ingeniería Universidad de Antioquia* 93 (2019), pp. 128–138. DOI: 10.17533/udea.redin.20190736
    • SJR Engineering (miscellaneous), **Q3**, 2019 SJR = 0.14

12. Andrés Camero, Jamal Toutouh, and Enrique Alba. "Random Error Sampling-based Recurrent Neural Network Architecture Optimization". In: *Engineering Applications of Artificial Intelligence* 96 (2020), p. 103946. ISSN: 0952-1976. DOI: 10.1016/j.engappai.2020.103946
    • JCR COMPUTER SCIENCE, ARTIFICIAL INTELLIGENCE – SCIE, **Q1**, ranking 33/136, 2019 JIF = 4.201

13. Andrés Camero et al. "Bayesian Neural Architecture Search Using A Training-Free Performance Metric". In: *arXiv preprint arXiv:2001.10726* (2020)
    • Under review

UNIVERSIDAD DE MÁLAGA

This page intentionally left blank.

UNIVERSIDAD DE MÁLAGA

Chapter 2

State-of-the-art

In this chapter, we briefly present the state-of-the-art of DN and its applications to SC. First, we review the history of ANNs, from their beginning to DL. Second, we introduce the ANN design optimization problem. Third, we outline the DN approach to ANN optimization. And finally, we review the applications of DN to SC.

2.1 Artificial Neural Networks

The human brain is a *machine* capable of performing very complex tasks, such as pattern recognition, motor control of a limb or perception of sensory stimuli, much faster than any machine invented by man [57]. For this reason, great efforts have been (and continue to be) made to understand how they work [9, 44].

This *machine* is made up of about 100 trillion neurons (cells of the nervous system specialized in the reception and conduction of stimuli), which communicate with each other, forming complex circuits, which are capable of carrying out brain function [44]. This biological model was the inspiration for McCulloch and Pitts [87] to propose a new computer model in 1943: the ANN.

An ANN is a network of computer units (neurons) connected to each other, where each unit performs a calculation from an input and communicates its result (output) to the connected units. Typically, the connections are weighted, which indicates how strong the connection between two units is. Also, the calculation units are usually grouped into modules or layers. ANNs have properties and capabilities that are very useful for solving complex problems, highlighting its *nonlinearity*, adaptability, fault tolerance, among others [57]. Thanks to these properties it has been possible to overcome the results obtained in multiple problems.

The ANN aroused great interest and developed rapidly, but the first decades of its evolution were marked by techniques that required a great effort to design it, and a broad knowledge of the problem domain to transform it into something that the ANN could process [8, 76]. It was not until the mid-1980s, when based on the *stochastic gradient descent* method, it was possible to pose (and understand) how to reduce human intervention: the *backpropagation* method (BP) [75, 108].

BP is a procedure to compute the gradient of an objective function (e.g., the *loss function*) with respect to the weights of an ANN [108]. This method is a practical application of the chain rule. The idea is that the gradient of a target function with respect to the input of a layer (of neurons) can be calculated backwards and forwards from the gradient of the output of the same layer. This allows the error to be propagated from the output to the input, thus, adjusting the weights of the network.

After a decade of study (in the late 1990s), the BP method and ANN (in general) were dropped, because it was thought that in practice it was not possible to *learn* complex concepts from scarce previous information [76]. In other words, it was unclear how to train a multi-layered hidden ANN with a highly variable non-linear

UNIVERSIDAD DE MÁLAGA

function as a target [15]. However, a decade later (in the mid-2000s), thanks to Hinton et al. [59], the interest on ANN revived on a new concept: the DL.

DL is a type of ML that allows training ab ANN composed of multiple hidden layers, known as DNN [76]. Circuit theory suggests that the deeper the architecture of an ANN, the more efficient it will be in terms of the number of computational elements required to represent highly non-linear and variant functions [15].

There are multiple types of ANNs: *feedforward* (FNN) [13], *convolutional* (CNN) [52, 76], *radial-basis* [120], *recurrent* (RNN) [57], and *modular* [54, 120], among others [57]. In particular, RNNs are characterized by forward propagation of information (like FNNs) and by the inclusion of at least one backward connection (feedback). This *feedback* can be from a neuron to itself (*self-feedback*) or to other neurons. It can also originate from the hidden layers or from the output of the RNN. Because of their architecture, RNNs are essentially the deepest DNNs. Figure 2.1 shows an RNN. Thanks to the backward connections, RNNs are capable of capturing the non-linear dynamic behavior of a system [57].

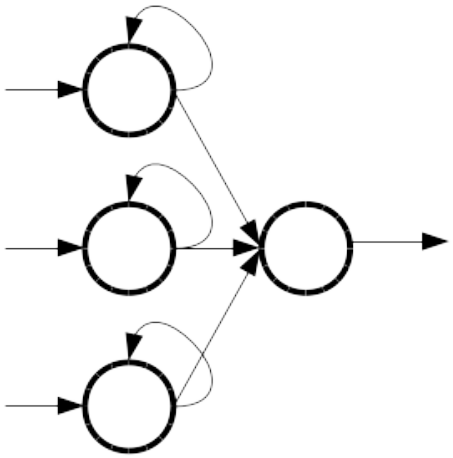


FIGURE 2.1: A simple recurrent neural network

As the DNN becomes more complex, the number of parameters increases. More neurons mean more weights, and more hidden layers mean more non-linearity (more difficulty in propagating an error, e.g., BP). In the case of RNNs, feedback adds more weights and depth, i.e., it makes the situation even worse. Then, considering the importance of selecting an appropriate DNN design for an optimal performance [96], it becomes mandatory to approach the design in an intelligent way.

2.2 Artificial Neural Network Design

The problem of designing an ANN consists of finding a network structure, which includes the selection of the appropriate node or neuron type (i.e., activation function, memory, etc.), the number of neurons and their arrangement (i.e., the architecture), and a set of weights to minimize a target function [48, 96].

There are multiple algorithms, techniques, and procedures that allow us to address this problem, either by considering one of these *dimensions*, a combination or all at the same time [57]. Likewise, it is important to note that there is a big overlap between the ANN design and *hyperparameter optimization* [48], i.e., the setting of the parameters of the algorithm used to optimize the design (and even the algorithm itself).

UNIVERSIDAD DE MÁLAGA

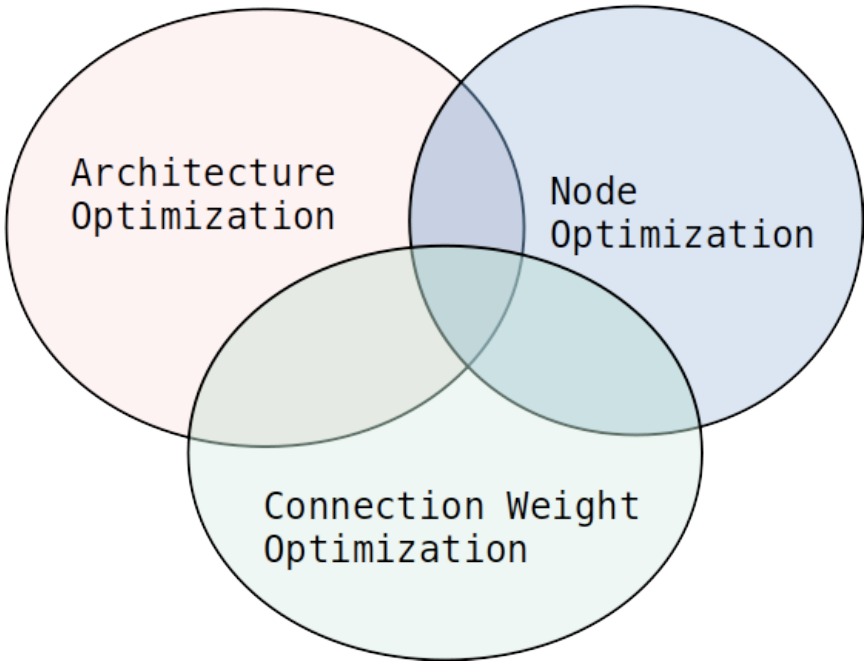


FIGURE 2.2: Components of the neural network design

Figure 2.2 depicts the spectrum of ANN design optimization: architecture, node (or neuron), and weight optimization [96]. Architecture optimization, a.k.a. *neural architecture search* (NAS), has started to gain popularity in the past few years [48]. There have been proposed several strategies to automatically cope with this task, ranging from random search-based [16, 77], to reinforcement learning [130, 131], Bayesian optimization (BO) [110, 125], and EC-based strategies [34, 127]. Currently, state-of-the-art NAS techniques have outperformed human expert designs, but due to the vast amount of time required by these methods to optimize the architecture, they are still not massively adopted [48, 96].

On the other hand, node optimization has been tackled from three main perspectives [96, 122]: choosing the activation function [81], optimizing the arguments of the activation function [97], and by placing a complete model at the node [35, 60]. It has been shown that the node has an important effect on the performance of the network [60, 85]. However, most ANN design strategies do not consider node optimization in conjunction with architecture or weight optimization [48, 96]. Contrary, most approaches use a *fixed* node design (with some exceptions [79, 98]).

In spite of the great importance and success of architecture and node optimization, so far, most effort on ANN design has been made on optimizing the weights. Weight optimization consists of optimization the connection weights of an ANN. The most popular methods are BP [108] and *Conjugate Gradient* [58] (and its variants [96]). Both methods achieve very good results. But, in spite of their great performance (and popularity), in high dimension problems, which is the case of DNNs, both methods have a tendency to stagnate at local minimums and (especially) on plateaus [40]. Moreover, they require several tuning to make their parameters optimal.

Particularly, in the case of RNNs, due to their recurrent edges (feedback), most gradient-based optimization procedures fail to train them [14, 70, 80, 99]. This issue is mainly explained by the fact that gradient-based approaches keep an activation vector. Then, the feedback makes this vector extremely deep, and thus the exploding and the vanishing gradient problems is aggravated [14].

UNIVERSIDAD DE MÁLAGA

Several strategies have been proposed to handle these gradient-related issues. Among them, and directly related to RNNs, is the Long Short-Term Memory (LSTM) cell approach proposed by Hochhreiter and Schmidhuber [60]. The LSTM is a *neuron design* consisting of units (called *memory blocks*) in the recurrent layer. In spite of the great success of LSTM to mitigate these gradient-related problems (LSTM-based networks are easier to train than standard RNNs), they still do not solve the whole optimization problem. In other words, not only the neuron design affects the learning process but also the architecture (i.e., how these neurons are arranged), the weight initialization [101], and all the parameters of the optimization (training/learning) algorithm [57].

In the search for alternatives to BP, in the 1980s, Engel [49] proposed optimizing the weights of an FNN by applying the technique of simulated annealing (SA). Almost simultaneously, Montana and Davis [92] proposed to use a genetic algorithm (GA) to optimize the weights of an FNN. The good results obtained encouraged research in this area (metaheuristics applied to the optimization of ANNs) [5, 96], including both solutions for the optimization of the weights [94, 126], and for the optimization of the architecture and weights simultaneously [3, 4, 82, 83]. A new paradigm was also developed called *neuroevolution* [127], which considers the evolution of the components of the ANN as a fundamental part of it, in addition to learning, giving rise to the *evolutionary ANN*.

2.3 Deep Neuroevolution

In the last years, due to the boom of DL, combined with the constantly growing volume of data, and the heterogeneity and sophistication of technological devices that the *fourth industrial revolution* has brought, new challenges related to the design of DNNs has arisen. There is a global need to have *simple* and *adaptive* DNN, capable of adapting to the fast changing data, as well as optimization methods that are *efficient* in terms of their capacity to act with the least amount of data and in the shortest time. For these reasons, and in view of BP limitations (i.e., it only optimizes the weights of the network), some authors have recalled to this *old idea* of evolving the components of the ANN, a.k.a. neuroevolution [94].

Metaheuristics are well-known optimization algorithms especially suited to address complex, non-linear, and non-differentiable problems [11]. This means that they efficiently combine exploration and exploitation strategies to provide *good* solutions, with bounded computational resources. Thus, some authors have proposed to take advantage of metaheuristics to tackle the DNN design problem.

Coming back from the late approaches proposed in the 1980 and 1990's [3, 4, 49, 92, 127], some authors have proposed to adapt these ideas to the current needs. For example, Zhining and Yunming [129] used a genetic algorithm (GA) to optimize a CNN. Also related to CNN evolution, Rosa et al. [106] proposed to use harmony search (HS). Rere et al. [104] studied simulated annealing (SA), and later on [103] studied both SA, differential evolution (DE), and HS.

Within the DN approaches, the NeuroEvolution of Augmenting Topologies (NEAT) method [74, 113] is (maybe) the most popular. In NEAT, a GA is used to increasingly evolving complex neural network topologies (structures) and weights. Despite being of great success, and of proposing an all-in approach (i.e., NEAT designs and optimizes the structure and the weights), NEAT has some limitations, e.g., the fitness landscape is misleading and many parameters need to be optimized. To cope with these limitations, some authors have proposed to extend the original

UNIVERSIDAD DE MÁLAGA

NEAT algorithm [90, 96]. Some of them are especially tailored to RNNs, for example, NEAT-LSTM [102] and CoDeepNeat [78].

Also related to RNNs, Ibrahim and El-Amary [63] proposed to optimize the weights using particle swarm optimization (PSO). ElSaid et al. [47] proposed an ant colony optimization (ACO) algorithm to refining the cellular structure of an LSTM-based RNN. Ororbia et al. [98] proposed the Evolutionary eXploration of Augmenting Memory Models (EXAMM), a technique capable of evolving an RNN using a wide variety of memory structures. ElSaid et al. [46] proposed the Evolutionary eXploration of Augmenting LSTM Topologies (EXALT), an epigenetic-based weight initialization and node-level mutation technique.

Lately, generative adversarial networks (GAN) have popped-up in the scene. For example, Wang et al. [124] proposed Evolutionary GAN (E-GAN), an evolutionary strategy (ES)-based technique to evolving a population of networks (i.e., generators) that are mutated by applying stochastic gradient descent (SGD) according to different loss functions. Then, the generators are evaluated by a single discriminative network that returns the fitness value. On this line, Lipizzaner [109] and Mustangs [118] present a coevolutionary approach for evolving two populations, i.e., the generators and the discriminators, to improve diversity while training.

Even though intelligent methods pose competitive solutions, they have not been massively adopted yet because they are computationally intensive [48, 96]. Whether they are a neuroevolutionary or gradient-based approach, they still require evolving/training a model, and evaluating its performance on test data (i.e., they are data-driven). Thus, they require a lot of time and computational resources to optimize the design [6, 17, 48, 112].

So far, few methods have been proposed to address this data-driven dependency problem [48]. For example, Domhan et al. [43] proposed to rapidly asses a candidate architecture by predicting its performance based on a statistical modelling of the learning curve. Klein et al. [69] used Bayesian neural networks to estimate the performance of a DNN based on the learning curve, reducing the searching time by up to 50%. Zela et al. [128] proposed to use shorter training times, in a combination of BO and Hyperband, to estimate the performance of candidate designs.

However, despite the great advances made so far to speed-up the evaluation of a candidate design, all methods rely on optimizing the weights (training) of the network. Thus, we are still *locked* to the pros and cons of weight optimization (especially its high computational cost). Therefore, we pose the question: is it possible to predict the performance of a network (on a give data set) without training it? This would dramatically reduce the cost of its design. As a sneak peek of this doctoral thesis, the answer is yes [25–27, 31].

2.4 Smart City

The SC concept emerged a few years ago as a combination of "ideas about how information and communication technologies might improve the functioning of cities" [12]. In spite of its increasing popularity, the concept is still *work-in-progress*, meaning that there is no clear consensus on its definition and implications [23, 37, 39, 84, 86, 88]. Moreover, there are several overlapping concepts, e.g., *digital city* [64], *intelligent city* [71], *green city* [95]. However, independent of the *flavor*, all SC approaches are directly (explicitly or implicitly) related to technology (i.e., computer science or information technology) [23].

UNIVERSIDAD DE MÁLAGA

The SC initiatives are often classified into six domains: *Smart Economy, Environment, Governance, Living, Mobility*, and *People* [86]. *Smart Economy* encompasses e-business and e-commerce initiatives, as well as the new economic opportunities that are enabled by technology (e.g., innovation, manufacturing and service delivery, flexibility of the labor market).

Smart Environment refers to the initiatives that are related to green buildings and urban planning, smart urban services (e.g., water, waste, public lighting management), smart energy (e.g., smart energy grids, controlling and monitoring aided by intelligent systems), among others.

The group of initiatives that are concerned with improving democratic processes, e-government (i.e., public services), and public planning are gathered in the *Smart Governance* domain.

Under *Smart Living* are jointed the initiatives that use technology to improve and enable new lifestyles, fostering healthy and safest cities. While *Smart Mobility* encompasses all the transportation and logistics initiatives that use technology as a key to improve their performance.

Finally, *Smart People* gathered the actions that seek to improve creativity and foster innovation (e.g., work-at-home, education, continuous learning).

Figure 2.3 (adapted from [23]) presents the six SC domains, and a short list of initiatives, as an example.

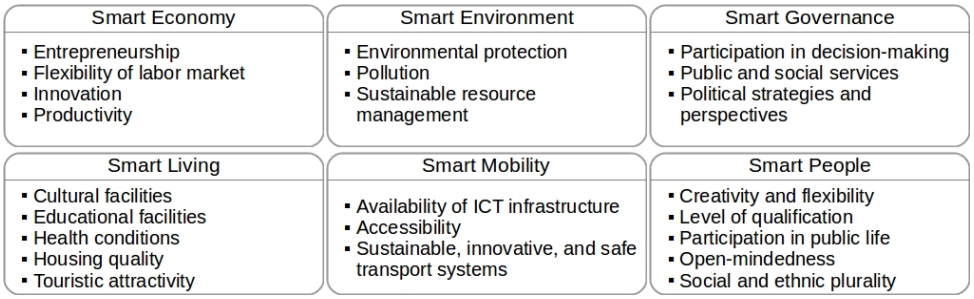


FIGURE 2.3: Smart city domains and a set of example initiatives

So far, the SC literature has focused mainly on enabling technologies, e.g., *Internet of Things* (IoT) and Big Data [23, 37]. However, there is a *shared vision* that intelligent applications will give a meaning to the smart part of SC [23, 37, 39, 84, 86, 88].

Therefore, considering the positive impact to society of the SC initiatives, and the current state of the matter, we envision a great opportunity in applying artificial intelligence, and in particular deep neuroevolution, to the SC problems.

UNIVERSIDAD DE MÁLAGA

Chapter 3

Deep Neuroevolution

In this chapter, we summarize the main contributions of this thesis related to DN. These results are in line with the goals **G1**, **G2**, and **G4** of the plan (Section 1.1).

Particularly, we have contributed to DN corpus with five manuscripts, one [26] has been published (and one is in second round revision [31]) in a JCR-indexed journal, one has been published in a Lecture Notes in Artificial Intelligence volume [25], one is still under review [27], and one is available as a preprint [19]. Also, we have developed a software library (DLOPT), which is freely available on GitHub.

The list of the publications and their detailed description is available in Appendix C. Also, Appendix A briefly introduces the problems and data sets used to test our proposals.

The remainder of this chapter introduces the studies mentioned before. First, we introduce the MRS [31]. Section 3.2 introduces a study on the relation of the MRS and the architecture [25]. Section 3.3 presents RESN [26]. Section 3.4 overviews our BO-based approach to NAS [27]. Finally, Section 3.5 introduces DLOPT [19].

3.1 Low-Cost RNN Expected Performance Evaluation

RNNs are a powerful tool, especially good for tackling time-dependent (or sequential) problems. However, due to recurrence, they are hard to optimize [14]. Small changes in any of the optimization components (Section 2.2) may produce a big deviation in the performance, i.e., they are very sensitive to their configuration [70, 99].

Many strategies have been proposed to deal with the sensitivity of RNN to their configuration. Ranging from specific node architectures (e.g., LSTM [60] or GRU [35]), that aim to alleviate the problems that arise when using gradient-based learning [14], to approaches like NEAT-LSTM [102] or CoDeep-NEAT [78], that try to optimize the whole RNN at once.

Despite the great improvements made up to this date, there are still many open challenges. Particularly, optimizing an RNN (and a DNN in general) is a demanding task (i.e., time and computational resources). Moreover, the number of parameters involved in the optimization, i.e., the weights, architecture, node, and learning rule parameters, is extremely large. Therefore, finding a way to search for an optimal set of parameters (configuration) is a must.

Regarding this challenge, some authors have proposed intelligent automatic hyper-parameter methods [6, 17, 66, 112]. Nonetheless, these methods have not yet been massively adopted because they require high computational resources and time. In general, this may be explained because these methods are data driven, i.e., they require training and evaluating every candidate solution (which is a very computationally intensive task).

UNIVERSIDAD DE MÁLAGA

Thus, in this study [31], we proposed to address two research questions related to this challenge: Is it possible to characterize the performance of an ANN on a given data set without training it? And, can we rank a set of ANNs using this characterization?

To answer these questions, we started from fairly simple premise: Given an ANN architecture, and a problem, it is easy to show that the output of the network changes as the weights change [57]. Then, we propose a *naive* approach (inspired in signal analysis [7]) to characterize an ANN based on "the variation of the observed output in regard to the expected one (i.e., the error) as the weights change": the MRS.

Note that these questions are directly related to the **G1** goal of this thesis (Section 1.1). Therefore, we may consider this work to be the *backbone* of the doctoral thesis. Said so, the main contribution (Section 1.3) of this work [31] is:

MRS We proposed the MRS, a low-cost (time and computational resources) technique to predict the performance of an ANN based on a random sampling.

Roughly speaking, the MRS predicts how *easy* it would be to train a given ANN (i.e., how likely is to have a good performance) by randomly sampling the output (on a given input). To take each sample, the ANN is initialized with a set of random weights (e.g., normally distributed). Then, an error metric is calculated (e.g., the mean absolute error, MAE). Later, a truncated normal distribution is fitted to the distribution of the measured MAE values, and a p_t probability of finding a set of weights whose error is below a user-defined threshold is estimated. Finally, the p_t probability is used as a proxy (predictor) of the performance. Therefore, we can use the p_t to rank (and sort) a set of architectures.

Figure 3.1 (taken from [31]) presents a high level view of the MRS, for the particular case of an RNN. Given an RNN and an input data set (e.g., time series), multiple random samples are taken. The performance metrics are calculated (MAE) and a probability density function (PDF, truncated normal distribution) is fitted to them.

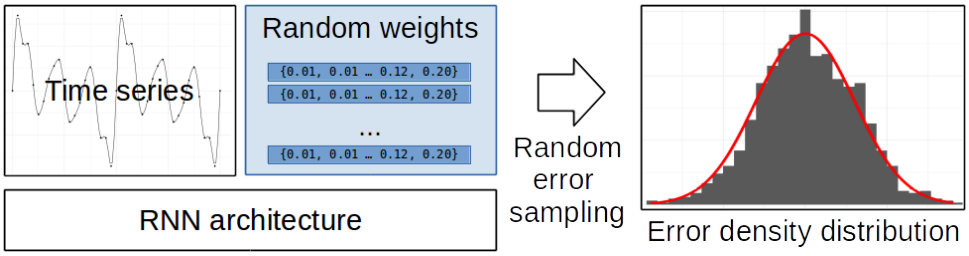


FIGURE 3.1: An overview of the MRS

Algorithm 1, taken from [31], presents the pseudo-code of the MRS. Given an RNN architecture (*ARQ*), a number of time steps or look back (*LB*), and a user-defined data set (*data*), the algorithm takes *MAX_SAMPLES* random samples of the MAE, using normally distributed sets of weights. Then, a truncated normal distribution is fitted to the MAE values, and p_t is estimated for the defined *THRESHOLD*.

To answer the research questions posed in this study, we focused on RNNs, and we chose three prediction problems: the *sine wave*, the *car parks occupancy rate* [114], and the *appliance energy consumption forecast* [32]. Appendix A introduces the details of these data sets.

Then we defined two sets of experiments, (*i*) a correlation study between the MRS and the MAE after training the networks using a gradient-based algorithm, and (*ii*) an analysis of the performance of MRS to rank multiple-hidden-layer RNNs.

UNIVERSIDAD DE MÁLAGA

Algorithm 1 MRS pseudo-code

Given: an architecture (*ARQ*), a number of time steps or look back (*LB*), a user-defined time series (*data*), a number of samples (*MAX_SAMPLES*), and a *THRESHOLD*.
rnn $\leftarrow$ InitializeRNN(ARQ, LB)
mae $\leftarrow \emptyset$
while sample $\leq$ MAX_SAMPLES **do**
    weights $\leftarrow$ GenerateNormalWeights(0, 1)
    UpdateWeights(rnn, weights)
    mae[sample] $\leftarrow$ MAE(rnn, data)
    sample $++$
end while
mean, sd $\leftarrow$ FitTruncatedNormal(mae)
$p_t \leftarrow$ PTruncatedNormal(mean, sd, THRESHOLD)

As for the (*i*) correlation study, we studied the relationship between the MRS and the MAE after training a network using a gradient-based algorithm, more precisely we used Adam [68]. First, we defined a search space (Table 3.1). Then, for each candidate architecture we calculated p_t (with t (THRESHOLD) equal to 0.01), and MAX_SAMPLES equal to 100. Later, we trained all the architectures for up to 100 epochs, adding a 0.5 dropout and an early stopping criterion when the validation loss was below 0.00001. Finally, we computed the correlation of Pearson.

Table 3.1 introduces the search space for the experimental study. HL stands for the number of hidden layers, and NPL stands for the number of neurons per layer.

TABLE 3.1: RNN architectures search space

Problem	HL	NPL	Stride	Look back
Sine wave	1	[1, 100]	1	[2, 30]
	2	[1, 100]	5	{2, 5, 10, 20, 30}
	3	[1, 100]	10	{2, 5, 10, 20, 30}
Car parks occupancy rates, Appliances energy consumption	1	[10, 100]	1	[2, 30]
	2	[10, 100]	5	{2, 5, 10, 20, 30}
	3	[10, 100]	10	{2, 5, 10, 20, 30}

Tables 3.2, 3.3, and 3.4 present the correlation between the MRS (log $p_{0.01}$) and the MAE after training. As part of the study, we also analyzed the correlation between the MAE after training and the total number of LSTM cells (NC), the timesteps (LB), the *mean* MAE sampled, and the standard deviation (*SD*) of the MAE sampled.

TABLE 3.2: Correlation between the MAE of an RNN trained with Adam and the MRS (sine wave)

HL	NC	LB	Mean	SD	log $p_{0.01}$
1	-0.511	-0.057	-0.639	-0.601	-0.628
2	-0.416	-0.150	-0.478	-0.515	-0.651
3	-0.422	-0.101	-0.571	-0.554	-0.615
All	-0.323	-0.091	-0.572	-0.541	-0.586

In all studied problems there is a moderate-to-strong negative correlation [107] between the MRS ($\texttt{log}\ p_t$) and the MAE observed after *training*. This means that a

UNIVERSIDAD DE MÁLAGA

TABLE 3.3: Correlation between the MAE of an RNN trained with Adam and the MRS (car parks)

HL	NC	LB	Mean	SD	log $p_{0.01}$
1	-0.152	0.484	-0.182	0.032	0.136
2	-0.102	0.795	0.130	-0.447	-0.400
3	-0.090	0.838	0.149	-0.550	-0.504
All	0.056	0.734	0.056	-0.440	-0.460

TABLE 3.4: Correlation between the MAE of an RNN trained with Adam and the MRS (appliances energy consumption)

HL	NC	LB	Mean	SD	log $p_{0.01}$
1	0.195	0.315	0.053	-0.288	-0.279
2	0.202	0.547	0.087	-0.525	-0.543
3	0.187	0.572	0.073	-0.524	-0.524
All	0.132	0.503	0.067	-0.384	-0.404

good training result (small value) is correlated to a higher value of p_t. It is interesting to note that the `log` p_t is the only variable that consistently shows a moderate-to-strong negative correlation. While the rest (i.e., NC, LB, mean, and SD) vary from case to case.

As for the (*ii*) analysis of the performance of the MRS to rank multiple-hidden-layer RNNs, we defined two tests: A pairwise performance comparison, and a linear fitting to predict the performance. The results of the pairwise comparison are introduced on Table 3.5. Particularly, the table presents the rate of positive comparisons, i.e., given two RNNs A and B, we checked if $p_t^A \geq p_t^B$ implies $MAE^A \leq MAE^B$, meaning that p_t is a good proxy for comparing the expected performance.

TABLE 3.5: Pairwise performance comparison ratio of the MRS

Problem	Ratio
Sine wave	0.636
Car parks occupancy rate	0.651
Appliances energy consumption	0.615

Moreover, we fit a linear model, using the `log` $p_0.01$ and the MAE after training, to the 80% of the RNNs. Then, using the p_t, we predicted the performance (MAE) of the remainder 20%. Table 3.6 presents the linear fitting results. RSE stands for the residual standard error of the predicted values, and SCC stands for the Spearman correlation coefficient between the predicted and observed decile. Overall, the results show that the MRS is useful to predict the *performance* of an RNN architecture.

TABLE 3.6: Linear fitting of the MRS and the after training MAE

Problem	RSE	SCC
Sine wave	0.079	0.399
Car parks occupancy rate	0.016	0.344
Appliances energy consumption	0.019	0.437

UNIVERSIDAD DE MÁLAGA

3.2 Comparing RNN Based on the MRS, a First Approach

In this work, we extended our analysis of the MRS [31]. Particularly, we dived deeper into the sine wave problem (Appendix A introduces the details of this problem) to get insights of the technique. The main contributions of this work [25] are:

MRS We studied the relations between the number of LSTM cells, the look back or time steps, the distribution of the cells (i.e., the number of hidden layers), and the MRS.

MRS We analyzed the MRS memory and time consumption.

First, using the MRS Algorithm 1 we sampled a set of RNN architectures with up to three hidden layers, with up to 100 neurons per layer, and considering a look back from one to 30, i.e., more than sixteen thousands architectures. Then, we study the relation between the above mentioned characteristics.

Figure 3.2 shows the relationship between the number of hidden cells and the estimated probability $p_{0.01}$. The different values of the look back are represented with colors. There is a clear trend in p_t, it rapidly increases from one to 25 cells (independently of the look back). Then, from 25 cells, the value of $p_{0.01}$ tends to *converge*, i.e., according to the MRS the probability of finding a set of weights whose error is below the defined threshold tends to stabilize (in this problem).

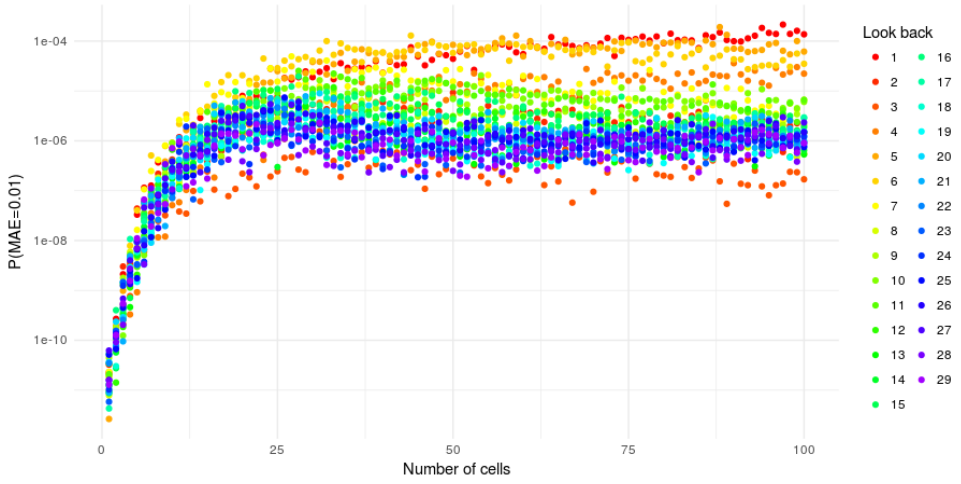


FIGURE 3.2: Relationship between the MRS and the architecture for one-hidden-layer RNNs

Second, we extracted the memory and time consumption from the execution logs of all the RNNs processed. Table 3.7 summarizes the time and memory usage. In spite of the simplicity of the problem (i.e., the sine wave), on average, there is clear difference between the time needed to train an RNN using Adam and the MRS. Even if we divide by ten the results of Adam (i.e., to get a rough idea of the time needed to train an RNN for 100 epochs), the difference is huge in favor of the MRS.

TABLE 3.7: Time and memory usage comparison: MRS vs Adam

	Mean time [s]	SD time	Mean mem [MB]	SD mem
Adam (1000 epochs)	996	0.006	127	6.338
MAE random sampling	6	0.001	150	98.264

UNIVERSIDAD DE MÁLAGA

3.3 MRS-based RNN Architecture Optimization

In line with our previous experimental findings [25, 31], we decided to go further in the quest for a low-cost alternative for RNN optimization. Therefore, in this work, we addressed the architecture optimization problem aided by the MRS (Section 3.1).

Particularly, in this study [26], we posed four research questions, all of them aimed to address the **G2** (Section 1.1): (*RQ1*) Can a *hybrid* (MRS and gradient-based) architecture optimization technique get the same error performance of a solely gradient-based one? (*RQ2*) Can a *hybrid* architecture optimization technique get the same error performance of a non-gradient-based approach (e.g., DN algorithm)? (*RQ3*) Can we reduce the architecture optimization time by using this *hybrid* approach? (*RQ4*) Can we improve the performance of an expert designed architecture/solution?

To tackle this challenges, we introduced RESN, an EA that uses the MRS to guide the search. We empirically validated our proposal, and compared our technique to training-based architecture optimization techniques, DN approaches, and human expert designed solutions. The results of our experiments show RESN achieve a state-of-the-art optimization performance, while reducing by half the overall time.

Therefore, the main contribution (refer to Section 1.3) of this work is:

> **RESN** We proposed RESN, a $(\mu + \lambda)$EA-based algorithm that uses the MRS to speed-up the RNN architecture optimization process.

Particularly, we designed RESN to maximizing p_t (from the MRS [31]). Given an input X, an output Y, a set of RNNs architecture (ARQ), and look back or time steps (LB) constraints, the problem is stated as:

$$\begin{aligned} &\text{maximize fitness} = p_t(X, Y) \\ &\text{subject to min\_ARQ} \leq ARQ \leq \text{max\_ARQ} \\ &\text{min\_LB} \leq LB \leq \text{max\_LB} \end{aligned} \tag{3.1}$$

Then, we proposed a DN algorithm based on the $(\mu + \lambda)$ EA [11] to solve that problem (Algorithm 2). At a glance, the *population* is a group of solutions, where each *solution* represents an RNN architecture. The initial population is randomly set by the **Initialize** function. Then, this population is assessed by the **Evaluate** function, computing p_t (i.e., the MRS) for each solution. Once the initial population is evaluated, the evolutionary process begins.

The evolution process of the population is divided into: selection, mutation, evaluation, replacement, and self-adjustment. Once the termination criterion is met, i.e., the number of *evaluations* is greater than the budget (*max_evaluations*, the **Best** (i.e., the one with the highest p_t) solution is selected and trained using Adam [68] for a predefined number of *epochs*.

Figure 3.3 depicts a high-level view of RESN. The proposed approach combines EC and ML techniques to optimize the architecture and the weights of an RNN. It is worth noticing that the **Evaluate** function can be changed seamlessly by any other fitness function, e.g., the full training metrics, or another performance proxy.

We tested RESN on four problems: the sine wave, the waste generation prediction problem [51], the coal-fired power plant flame intensity prediction problem [98], and the EUNITE load forecast problem [33]. Refer to Appendix A for more details.

We proposed four experiments to address the research questions. **E1**: RESN vs. Gradient-based Architecture Optimization, we compared RESN against (i) a

UNIVERSIDAD DE MÁLAGA

Algorithm 2 RESN: Random Error Sampling-based Neuroevolution

```
population ← Initialize(population_size)
Evaluate(population)
evaluations ← population_size
while evaluations ≤ max_evaluations do
    offspring ← BinaryTournament(population, offspring_size)
    offspring ← CellMutation(offspring, cell_mut_p, max_step)
    offspring ← LayerMutation(offspring, layer_mut_p)
    Evaluate(offspring)
    population ← Best(population + offspring, population_size)
    evaluations ← evaluations + offspring_size
    SelfAdapting(layer_mut_p, max_step, cell_mut_p)
end while
solution ← Best(population, 1)
rnn_trained ← Train(solution, epochs)
return rnn_trained
```

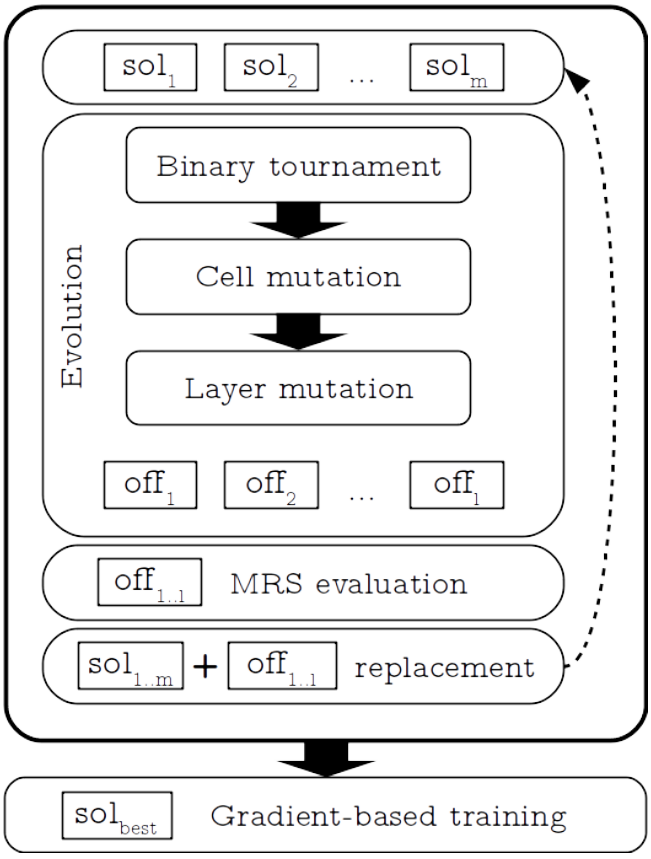


FIGURE 3.3: The global scheme of RESN

modified version of the same algorithm (i.e., a version that replaced the MRS by the results of Adam training), (ii) the *Short training* [21] algorithm, and (iii) a *Random Search* algorithm. **E2**: RESN vs. Neuroevolution, we benchmarked RESN against EXALT [46] and EXAMM [98], state-of-the-art DN techniques. **E3**: Optimization Time, we logged the execution times. And **E4**: RESN vs. Expert Design, we compared our results against the winner [33] of the EUNITE load forecast competition and to recent solutions to the same challenge [73].

UNIVERSIDAD DE MÁLAGA

Summarizing **E1**, (E1.i) the results of RESN exceed GDET (i.e., the modified version of RESN that used the Adam training results to evaluate the performance of a solution). On average, RESN obtained a MAE of 0.105 while GDET got a 0.142 (Wilcoxon rank-sum test p-value equal to 0.001).

When (E1.ii) comparing RESN against *Short Training* [21], there is no significant improvement in the mean MAE (i.e., Wilcoxon rank-sum test p-value equal to 0.665). However, RESN almost cut by half the time needed to optimize the RNN (test **E3**). Table 3.8 presents the results, where *MAE* is MAE of the final solution, *No. LSTM* is the total number of LSTM in the RNN, *LB* is to the look back, *No. HL* represents the number of hidden layers, and *Time* is the total time (i.e., the optimization process and the training of the final solution) in minutes.

TABLE 3.8: E1.ii results. RNN optimization in the Waste generation prediction problem, a comparison between Short training and RESN

		MAE	No. LSTM	LB	No. HL	Time [min]
Short Training	Mean	0.073	451	6	5	97
	Median	0.073	420	5	5	70
	Max	0.076	1252	16	8	405
	Min	0.071	127	2	1	33
	SD	0.001	228	2	2	75
RESN	Mean	0.079	793	17	3	51
	Median	0.073	513	16	2	45
	Max	0.138	2038	30	8	103
	Min	0.069	444	2	1	40
	SD	0.017	493	11	3	13

To conclude our E1 experiment, we compared RESN against random search (E1.iii). Despite the relative good error performance of random search, with a MAE equal to 0.091 on average, the Wilcoxon rank-sum test revealed that RESN (and *Short training*) beat random search (p-values are 0.017 and 0.002 respectively). Moreover, random search took nearly 50x the time of RESN. Thus, RESN is a fast and reliable approach (test **E3**).

Later, we compared RESN against the state-of-the-art of neuroevolution (test **E2**). As a summary, RESN improved EXALT by ten times (Wilcoxon rank-sum test p-value is 2.958e-06). Table 3.9 depicts the mean square error (MSE) of the solution obtained by RESN and EXALT [46] in the coal-fire power plan problem.

Then, we processed the results of EXAMM [98]. Particularly, we computed the average MSE for RNNs Evolved With Individual Memory Cells and RNNs Evolved With All Memory Type (Table 1 in the cited article) and for RNNs Evolved With Simple Neurons and Memory Cells (Table 2 in the cited article). The values are 0.001690 and 0.001601, respectively. Although the experiments are not comparable (i.e., EXAMM uses a different experimental design, but the same data set), it is quite interesting to notice that RESN achieves a result (0.005208) that is in the same order of magnitude that EXAMM, but with a much simpler approach, i.e., we only use fully connected stacked LSTM layers, instead of multiple types of cells.

Finally, we compared RESN against human expert designed solutions (test **E4**) in the EUNITE load forecast problem. The results are summarized in Table 3.10. The column SVM corresponds to the results presented by Chen et al. [33], the winner of the "Electricity Load Forecast using Intelligent Adaptive Technologies" competition organized by EUNITE, and the other columns, i.e., BP, RBF, SVR, NNRW, KNNRW, and WKNNRW, correspond to the results presented by Lang et al. [73]. NA stands

UNIVERSIDAD DE MÁLAGA

TABLE 3.9: E2 results (MSE). RESN vs EXALT in the coal-fire power plant problem

Fold	EXALT	RESN
0	0.028749	0.001541
1	0.031769	0.006536
2	0.023095	0.003821
3	0.019229	0.000570
4	0.023170	0.003336
5	0.036091	0.000617
6	0.012879	0.017061
7	0.019358	0.004032
8	0.018151	0.001912
9	0.019475	0.013996
10	0.030016	0.006120
11	0.031207	0.002942
Average	0.024432	0.005208

for *Not Available*. From the results, we concluded that the performance of RESN is comparable to a human expert, i.e., "the error of the best solution found is as good as the best solution proposed by the experts".

TABLE 3.10: E4 results (MAPE). RESN vs Expert design

	SVM	BP	RBF	SVR	NNRW	KNNRW	WKNNRW	RESN
Mean	2.879	NA	NA	NA	NA	NA	NA	2.281
Median	2.945	NA	NA	NA	NA	NA	NA	2.242
Max	3.480	NA	NA	NA	NA	NA	NA	3.273
Min	1.950	1.451	1.481	1.446	1.438	1.348	1.323	1.370
SD	0.004	NA	NA	NA	NA	NA	NA	0.414

As a summary of this study, we concluded that RESN is a competitive approach to RNN design. Particularly, it achieves a comparable error performance of training-based RNN techniques and neuroevolutionary approaches but considerably reduces the computational time, and it is as good as human expert designed solutions.

3.4 Bayesian NAS Using a Training-Free Performance Metric

According to the plan defined to accomplish this doctoral thesis, more precisely to the activity "WP3 3.2 Study alternative approaches to architecture search", and aiming to collaborate with an international research group, we decided to explore the joint use of the MRS and BO. Particularly, this work is the output of a three-month research stay with the Natural Computing Research Group (University of Leiden, The Netherlands), directed by Prof. Dr. Thomas Bäck.

In this study [27], we tackled the architecture optimization problem with a hybrid approach based on the combination of BO [65, 91] for optimizing the architecture, the MRS [31] for predicting the performance of candidate solutions, and Adam [68] for training the solution (i.e., the *optimized* architecture).

Therefore, the main contributions (refer to Section 1.3) of this study are:

UNIVERSIDAD DE MÁLAGA

MRS We proposed an RNN architecture optimization technique based on BO and the MRS.

DN and SC We introduced multiple alternatives to cope with variable-length solutions on BO.

DN and SC We proposed a strategy to improve the performance of the surrogate model of BO for variable-length solutions based on the augmentation of the initial set of solutions, i.e., the *warm-start*.

To face the RNN architecture optimization, we formulated the following problem:

$$\arg\max_{\mathbf{h}\in\mathcal{H}} \text{MRS}(\mathcal{D}, \mathbf{h}, p_t, Q),$$

where $\mathcal{D}$ is a data set, p_t is the estimated probability of the MRS, $\mathbf{h}$ is a solution, and Q is the number of random samples. It is worth noticing that BO does not reconcile well with variable-length solutions [110], while the architecture search space is naturally not of fixed-length [48]. Therefore, in this study we propose several strategies to handle this issue.

To cope with the variable-length problem, first, we proposed three different encoding schemes. Assuming that the number of neurons per layer is restricted to $[\underline{N}..\bar{N}]$, the number of layers is $m \in [\underline{M}..\bar{M}]$, and T denotes the maximum look back.

- **Plain**: the total length of this encoding is $m + 1$.

  $$\mathbf{h} = [h_1, h_2, \ldots, h_m, l] \in (\{0\} \cup [\underline{N}..\bar{N}])^m \times [1..T],$$

  where h_i is the number of neurons per each layer and l is the number of time steps. Note that h_i can take value zero, meaning there is no neuron in this layer and hence it is effectively dropped in the decoding procedure.

- **Flag**: the total length of this encoding is $2m + 1$.

  $$\mathbf{h} = [h_1, b_1, h_2, b_2, \ldots, h_m, b_m, l] \in [\underline{N}..\bar{N}]^m \times \{0, 1\}^m \times [1..T],$$

  where $b_i \in \{0, 1\}$ is the so-called "flag" that disables layer h_i if $b_i = 0$ when decoding such a representation to compute the actual architecture.

- **Size**: the total length of this encoding is $m + 2$.

  $$\mathbf{h} = [h_1, h_2, \ldots, h_m, s, l] \in [\underline{N}..\bar{N}]^m \times [1..m] \times [1..T],$$

  where $s \leq m$ is the number of layers from the start of the representation that are considered in decoding, namely only $h_1, h_2, \ldots, h_s$ are used to generate the actual architecture.

Then, considering that all three representations is a *many-to-one mapping*, i.e., multiple encoded solutions represent a single architecture, we proposed two strategies to deal with this problem: the infeasible representation and the constraint handling.

The *infeasible representation* strategy consists of selecting a *canonical* encoding, i.e., one of the many-to-one solutions, and penalizing the fitness of all others non-canonical ones. For example, a representation $[h_1, \ldots h_q, 0, \ldots, 0, l]$ (where $h_i > 0, 1 \leq i \leq q$) shall be called the *feasible* (i.e., the canonical or preferred representation), then all representation variations of the same architecture are to be considered infeasible.

UNIVERSIDAD DE MÁLAGA

On the other hand, we proposed the *constraint handling* strategy, aiming to avoid generating infeasible representations by constraining the method that proposes new candidate representations in BO.

Also in this study, we proposed to improve the performance of the surrogate model of BO with a "warm-start" strategy. The basic idea is to augment the set of initial solutions with a set of infeasible solutions. This set of solutions are to be generated before the optimization starts, and they have to be assigned with a penalized fitness value, i.e., zero, the minimum fitness value in our case. Thus, the optimization process is started with a priori information.

Algorithm 3 introduces our proposal for RNN architecture optimization based on BO, the MRS, and Adam. Please, refer to the original paper for the details [27].

Algorithm 3 Efficient Architecture Optimization of Recurrent Neural Networks

given: A data set $\mathcal{D}$, the objective function MRS, an encoding scheme $\texttt{code} \in \{\text{plain}, \text{flag}, \text{size}\}$, and the random forests algorithm RF.
$C \leftarrow 0.5, t \leftarrow 0, p_{\text{m}} \leftarrow 0.01, Q \leftarrow 100$
Determine the search space $\mathcal{H}$ according to code
Generate $X \subseteq \mathcal{H}$ using Latin Hypercube Sampling
$Y \leftarrow \{\text{MRS}(\mathcal{D}, \mathbf{h}, t, p_{\text{m}}, Q) \colon \mathbf{h} \in X\}$
if "warm-start" is enabled **then**
    $(X_{\text{warm}}, Y_{\text{warm}}) \leftarrow \text{WARM}(\mathcal{H})$
    $X \leftarrow X \cup X_{\text{warm}}$
    $Y \leftarrow Y \cup Y_{\text{warm}}$
end if
$X' \leftarrow \{D_{\texttt{code}}(\mathbf{h}) \colon \mathbf{h} \in X\}$ ▷ decoded solutions
$\mathcal{M} \leftarrow \text{RF}(X, Y)$ ▷ surrogate model training
while the stop criterion is not fulfilled **do**
    $t \leftarrow t + 1$
    if "constraint-handling" is enabled **then**
        $\mathbf{h}^* \leftarrow \arg\max_{\mathbf{h} \in \mathcal{H}} \text{EI}(\mathbf{h}; \mathcal{M}) - Ct \cdot \text{PENALTY}(\mathbf{h}, X)$
    else
        $\mathbf{h}^* \leftarrow \arg\max_{\mathbf{h} \in \mathcal{H}} \text{EI}(\mathbf{h}; \mathcal{M})$
    end if
    $\mathbf{h}^{*\prime} \leftarrow D_{\texttt{code}}(\mathbf{h}^*)$ ▷ decoding
    if $\mathbf{h}^{*\prime} \notin X'$ **then** ▷ for unseen architectures
        if "infeasible-solution" is enabled **and**
          $\texttt{code} \neq$ size **and** $\mathbf{h}^*$ is infeasible **then**
            $y^* \leftarrow 0$
        else
            $y^* \leftarrow \text{MRS}(\mathcal{D}, \mathbf{h}^{*\prime}, t, p_{\text{m}}, Q)$
        end if
        $X' \leftarrow X' \cup \{\mathbf{h}^{*\prime}\}$
    else
        $S \leftarrow \{y \colon (\mathbf{h}, y) \in (X, Y) \wedge D_{\texttt{code}}(\mathbf{h}) = \mathbf{h}^{*\prime}\}$
        $y^* \leftarrow \text{RANDOM}(S)$
    end if
    $X \leftarrow X \cup \{\mathbf{h}^*\}$
    $Y \leftarrow Y \cup \{y^*\}$
    $\mathcal{M} \leftarrow \text{RF}(X, Y)$
end while
$y_{\text{best}} \leftarrow \max\{Y\}$
$\mathbf{h}_{\text{best}}$ is the corresponding solution to y_{best}
$\mathbf{h}_{\text{trained}} \leftarrow \text{SGD}(\mathcal{D}, \mathbf{h}_{\text{best}})$
return $\mathbf{h}_{\text{trained}}$

UNIVERSIDAD DE MÁLAGA

We tested the approach on three prediction problems: *sine wave*, *waste generation prediction* problem [50], and *EUNITE load forecast* [33]. Appendix A introduces the details of these data sets.

Table 3.11 summarizes the results on the sine wave problem. MLES and GDET are the results presented in [26]. The rest of the columns correspond to the combination of the strategies introduced in this study. Particularly, the combinations are presented in the following order: `[constraint][warm start][infeasible][encoding]`. Where `C` stands for constraint handling, `W` for warm start, and `I` for infeasible representation. The different encodings are represented with: `F` (flag), `S` (size), and `P` (plain). A dash (-) means that the corresponding alternative was not used. Thus, `C-F` means that the test consisted of the combination of the constraint handling and the flag encoding.

Also, we performed a pairwise comparison using the Conover test for a two-way balanced complete block design, and the Holm p-value adjustment method. The results are presented in the row label Conover. Groups sharing a letter are not significantly different ($\alpha = 0.01$).

TABLE 3.11: Sine optimization results (MAE of the best solution)

	GDET	MLES	—F	–IF	-W-F	-WIF	C–F	CWIF	—S	C—S	—P	–IP
Mean	0.1419	0.1047	0.0785	0.0882	0.0816	0.1119	0.0839	0.1452	0.0857	0.0745	0.1198	0.1363
Median	0.1489	0.0996	0.0738	0.0882	0.0772	0.0861	0.0789	0.0935	0.0748	0.0721	0.1170	0.1244
Max	0.2695	0.2466	0.1172	0.1266	0.1185	0.3677	0.1276	0.5723	0.1794	0.0962	0.1700	0.3290
Min	0.0540	0.0631	0.0449	0.0505	0.0518	0.0492	0.0631	0.0577	0.0584	0.0525	0.0922	0.0665
SD	0.0513	0.0350	0.0194	0.0182	0.0161	0.0695	0.0154	0.1367	0.0274	0.0109	0.0177	0.0558
Conover	a	bc	**d**	ef	**de**	bf	**def**	c	**def**	**d**	a	a

Table 3.12 presents the results for the waste generation prediction problem. *Cities* corresponds to the results presented in [21], and MLES to the results presented in [26]. According to the Conover test, there is no significant difference between our results and the competitors. However, *Cities* results are obtained by training every candidate solution using Adam, thus it is more time-consuming than RESN.

TABLE 3.12: Waste optimization results (MAE of the final solution)

	Cities	MLES	—F	–IF	-W-F	-WIF	C–F	CWIF	—S	C—S	—P	–IP
Mean	0.0728	0.0790	0.0722	0.0821	0.0730	0.0812	0.0728	0.0728	0.0732	0.0725	0.0744	0.0735
Median	0.0731	0.0728	0.0723	0.0735	0.0725	0.0730	0.0725	0.0725	0.0736	0.0723	0.0737	0.0731
Max	0.0757	0.1377	0.0791	0.1227	0.0806	0.1231	0.0767	0.0767	0.0756	0.0760	0.0920	0.0883
Min	0.0709	0.0691	0.0695	0.0691	0.0703	0.0698	0.0692	0.0701	0.0691	0.0688	0.0717	0.0701
SD	0.0012	0.0172	0.0019	0.0156	0.0020	0.0177	0.0018	0.0014	0.0015	0.0016	0.0041	0.0027
Conover	**abc**	**abc**	**a**	bcd	**ab**	d	**ab**	**ab**	bcd	**ab**	cd	bcd

Table 3.13 summarizes the results on the EUNITE load forecast problem. SVM corresponds to the results presented by Chen et al. [33], the winner of EUNITE competition, and RBF and WK+, WKNNRW are taken from [73].

TABLE 3.13: Optimization results (MAPE of the final solution)

	SVM	RBF	WK+	—F	–IF	-W-F	-WIF	C–F	CWIF	—S	C—S	—P	–IP
Mean	2.879	NA	NA	2.726	3.148	2.595	3.066	2.158	2.844	2.321	2.235	4.823	5.287
Median	2.945	NA	NA	2.466	2.933	2.368	2.814	2.099	2.846	2.125	2.050	5.040	5.213
Max	3.480	NA	NA	6.271	5.207	4.594	6.031	3.364	3.901	6.271	4.605	6.999	11.004
Min	1.950	1.481	1.323	1.840	1.759	1.593	1.919	1.452	2.033	1.654	1.657	3.142	3.415
SD	0.004	NA	NA	0.888	1.013	0.765	1.000	0.440	0.515	0.774	0.564	0.884	1.727
Conover	NA	NA	NA	abc	a	bc	a	**d**	ab	ce	**de**	f	f

To continue with our study, we compared the run time of the MRS against Adam (one of the most popular gradient-based training methods). Particularly, we selected 16 runs from the previous experiments (i.e., 100 architectures evaluated in 16 runs, summing up to 1600 RNNs). Then, for each RNN we performed the MRS (with 100 samples), and we trained for 10 epochs using Adam.

Figure 3.4 depicts the run time measured for both approaches (in seconds). The results show that, on average, the MRS is 2.6 times faster than Adam.

UNIVERSIDAD DE MÁLAGA

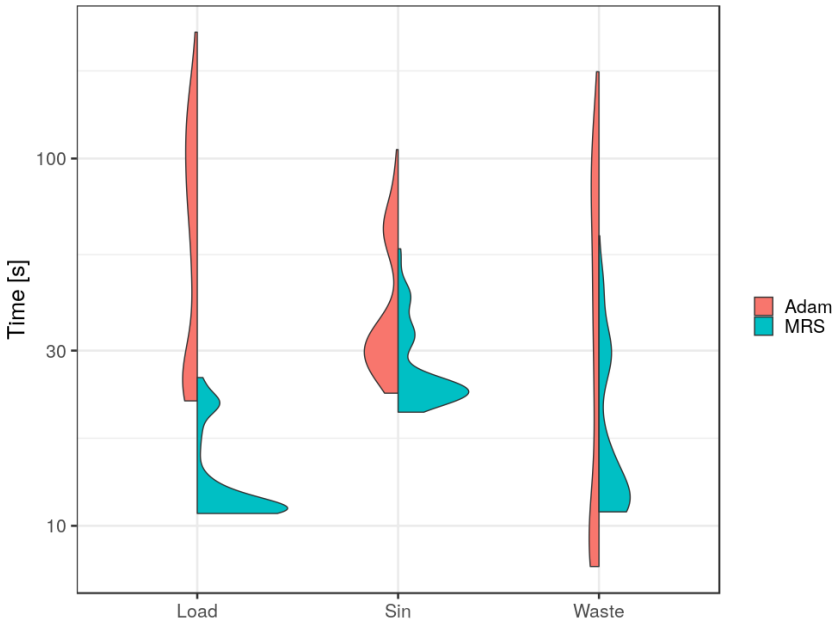


FIGURE 3.4: Time comparison: Adam (10 epochs) vs MRS (100 samples)

Finally, we studied the effect of the number of samples on the outcome of the MRS is affected. We repeated the *waste prediction* and *EUNITE load forecast* experiments using the `C-S` configuration, and 30, 50, and 200 samples per each solution evaluated.

Table 3.14 presents the error trade-off results. The Conover test for a two-way balanced complete block design, and the Holm p-value adjustment method results are presented in the row *Conover*. Note that groups sharing a letter are not significantly different ($\alpha = 0.01$).

An interesting insight of the results is that doubling the number of samples (used in this study) did not reduce the error. In other words, 100 samples seems to be a good approximation of the performance. Also, it is worth noticing that with as few as 30 samples, it is possible to estimate the performance of a network.

TABLE 3.14: Waste and Load trade-off results

	Samples	30	50	100	200
Waste (MAE)	Mean	0.0734	0.0734	0.0725	0.0723
	Median	0.0732	0.0740	0.0723	0.0726
	Max	0.0778	0.0780	0.0760	0.0757
	Min	0.0694	0.0690	0.0688	0.0685
	SD	0.0017	0.0020	0.0016	0.0018
Load (MAPE)	Mean	2.664	2.616	2.235	2.137
	Median	2.510	2.555	2.050	2.073
	Max	4.436	3.750	4.605	3.146
	Min	1.930	1.884	1.657	1.521
	SD	0.597	0.492	0.564	0.405
Conover		a	a	b	b

UNIVERSIDAD DE MÁLAGA

3.5 DLOPT: Deep Learning Optimization Library

DNN optimization is a tough task, and a proper design is a key to success. Therefore, we *compiled* all DN algorithms presented in this thesis as a software library, and made them publicly and freely available for the community on *GitHub*. Thus, the output of this work (i.e., the DLOPT code) is directly related to **G4** 1.1.

In this work [19], we introduced DLOPT, an open source software library developed on Python 3 language, and using Keras [36] and Tensorflow [1]. The source code is available on `https://github.com/acamero/dlopt`.

Therefore, the main contribution of this work is:

DLOPT We introduced DLOPT, a software library for DNN optimization based on the DN techniques presented in this doctoral thesis.

At a glance, this library is composed of three core classes: `ModelOptimization`, `Problem`, and `Solution`. This design aims to decouple the problem being solved (e.g., optimizing the architecture of an RNN), the technique used to solve the problem (e.g., using a $(\mu + \lambda)$EA-based algorithm), and the solution representation (i.e., the way the RNN is encoded).

DLOPT offers multiple implementations of these core classes, as well as a broad set of tools (e.g., data loaders, command-line applications, among others), and the data and scripts used in several works presented in this thesis [21, 22, 25–27, 29–31]. Also, The code adheres to the PEP-8[1] style guide, and it is available under the GNU GPL v3 license.

[1] `https://www.python.org/dev/peps/pep-0008/`

UNIVERSIDAD DE MÁLAGA

Chapter 4

Smart City

In this chapter, we introduce our main contribution to the SC domain. Particularly, we published a survey on SC and its relation to IT [23]. The publication is summarized in the remainder of this chapter. Refer to Appendix C for the details of this publication

4.1 Smart City and Information Technology: A Review

SC is attracting the attention of public opinion and researchers all over the world. Since it is a relatively new concept, there is no consensus on its definition nor scope. Therefore, in this paper, we surveyed the computer science (CS) and information technology (IT) literature about the SC, using data analysis techniques to present the topic from a data-based point of view, aiming to discover its major trends, and (hoping) to provide a single entry point for newcomers.

Therefore, the main contributions of this work [23] are:

DN and SC Using data analysis techniques, we automatically processed all CS and IT publications indexed by the JCR on SC. Thus, we offered a *broad and objective* review of the topic.

DN and SC We presented a *trend analysis* of the distribution and growth of CS and IT research in SC.

DN and SC We discussed the applications, techniques, and the relations between research fields. Also, we provided a network of interactions, and pointed out the *hot spots* in the SC literature. Hence, we provided a *single entry point* for SC from a CS and IT point of view.

To conduct our survey, we followed the guidelines proposed by Vom Brocke et al. [123]. Particularly, we defined the scope, conceptualized the topic, performed a literature search, analyzed and synthesized the literature, and propose a new research agenda. By doing so, we aimed to grant the reproducibility of this work.

We searched on the Web of Science for articles and proceeding papers indexed by the JCR related to SC. We repeated this search on two consecutive years (October 3, 2016, and October 9, 2017), thus we were able to compare the evolution of the field in time. We will refer as this two result sets as *snapshots* Figure 4.1 depicts the number of publications per year (up to October 9, 2017).

To gain insights on the *origin* of SC research, we studied the affiliation country of the authors. Figure 4.2 summarizes the contribution of each country (*count*) to SC from the fields CS and IT (2017 snapshot). The number of contributions increases as the color goes from green to red. The countries shown in gray do not have any publication in the snapshot.

UNIVERSIDAD DE MÁLAGA

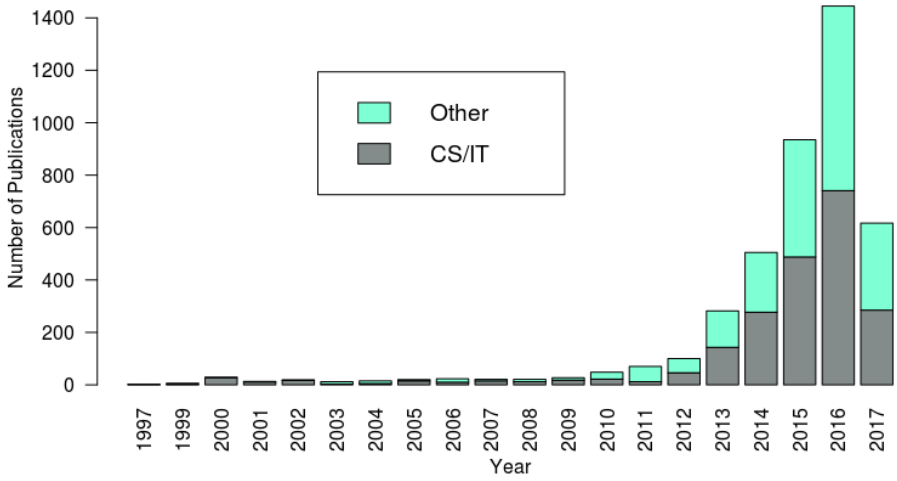


FIGURE 4.1: SC publications indexed by JCR up to Oct 9, 2017

Our study revealed that roughly 30% of the publications were done in collaboration between researchers of different affiliation countries. The top ten countries are: China (316 publications), Italy (272), United States (217), Spain (187), India (119), Germany (118), United Kingdom (117), France (104), Japan (97), and Australia (76). It is worth noticing that the top ten countries contributed with nearly 60% of the total research on SC. Moreover, half of them belong to Europe.

To continue with our study, we built the citation graph for both snapshots. Each publication is represented as a node in the graph, and the citations are represented as edges. The size of each node is set accordingly to the number of citations informed by the Web of Science. It is important to note that the citation graph only include the links between publications in the set. Therefore, citations to or from publications (i.e., edges in the graph) that are not in the collection will not be considered by the JCR). The citation graphs and density maps were built using VOS Viewer [45].

Figure 4.3 presents the citation density map for publications classified as CS and IT related to SC. On the left, it is presented the 2016 snapshot, while the 2017 snapshot is available on the right. The density increases from blue to red, i.e., as the density of citations increases, the map turns to red. The font size of the publications (name and year in the plot) gets bigger as the number of citations reported by the JCR increases.

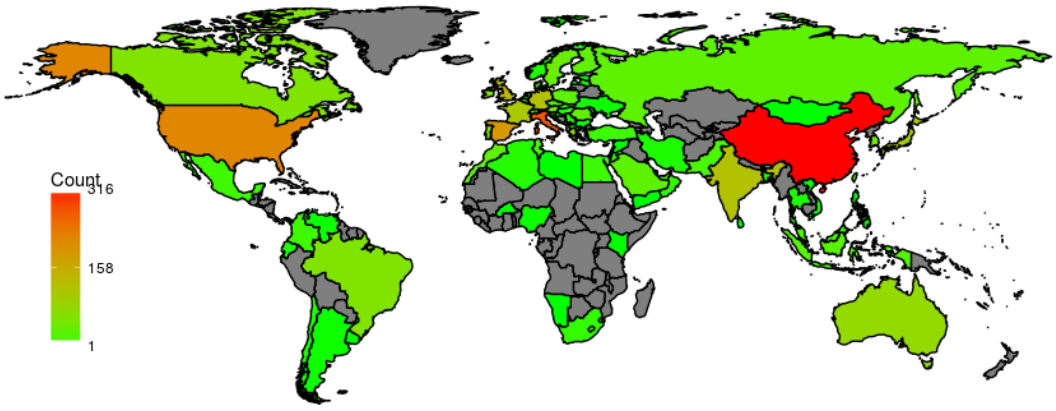


FIGURE 4.2: CS/IT publications related to Smart City per country of affiliation of the authors

UNIVERSIDAD DE MÁLAGA

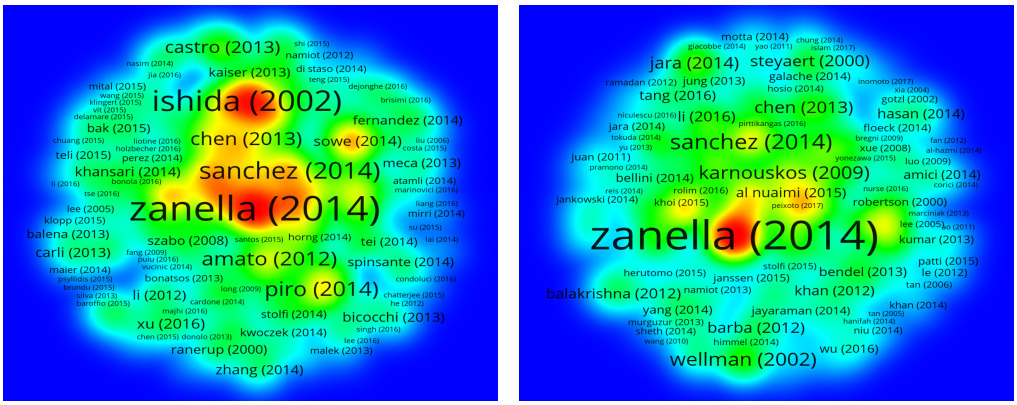


FIGURE 4.3: Density map of JCR publications classified as CS/IT related to SC

Figure 4.5 shows the citation graph of the publications that are strongly related to the densest part (i.e., to the *hot spot*) of Figure 4.3 for the year 2017. More details and insights, e.g., description of the works, their relation to SC domains (Figure 2.3), *distance* between the most important works, trends, and evolution, are available in the original manuscript [23].

In this work, we also analyzed the *author keywords*. Particularly, we counted the number of times each keyword was used *as is*, meaning that we did not perform any preprocessing (e.g., normalization). We found that 3,560 keywords were used, but only 658 keywords were used more than once. Figure 4.4 presents the word cloud[1]. From this data, we inferred that the domain has started to converge, and that there is a strong relation between SC and IoT.

As a summary of the insights presented in this work, we reveal that the majority of the publications in the snapshots are not directly related among them (i.e., there are no citations within the snapshot). Therefore, we considered SC itself to be still work-in-progress. Also, we conclude that there is no relation between the wealthiness of a country and its SC development. Furthermore, most CS and IT literature on SC do not include policy and urban planning in their scope. The results also showed that most works focus on enabling technologies (e.g., IoT, Big Data).

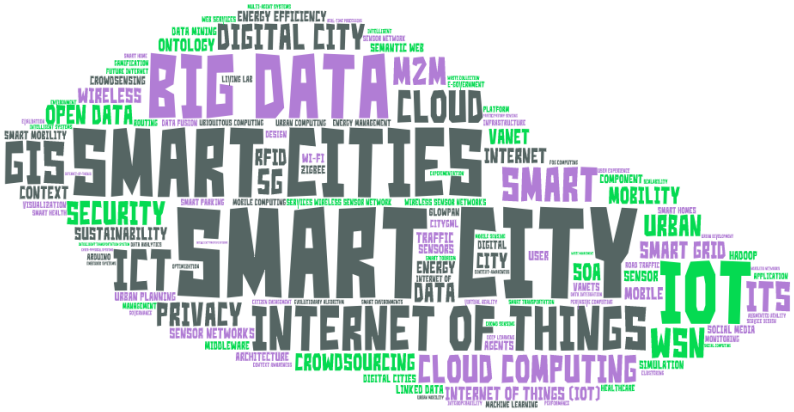


FIGURE 4.4: SC word cloud based on author keywords extracted from CS and IT publications indexed by JCR

[1]Created using `https://wordart.com` [Accessed: Aug, 2020]

UNIVERSIDAD DE MÁLAGA

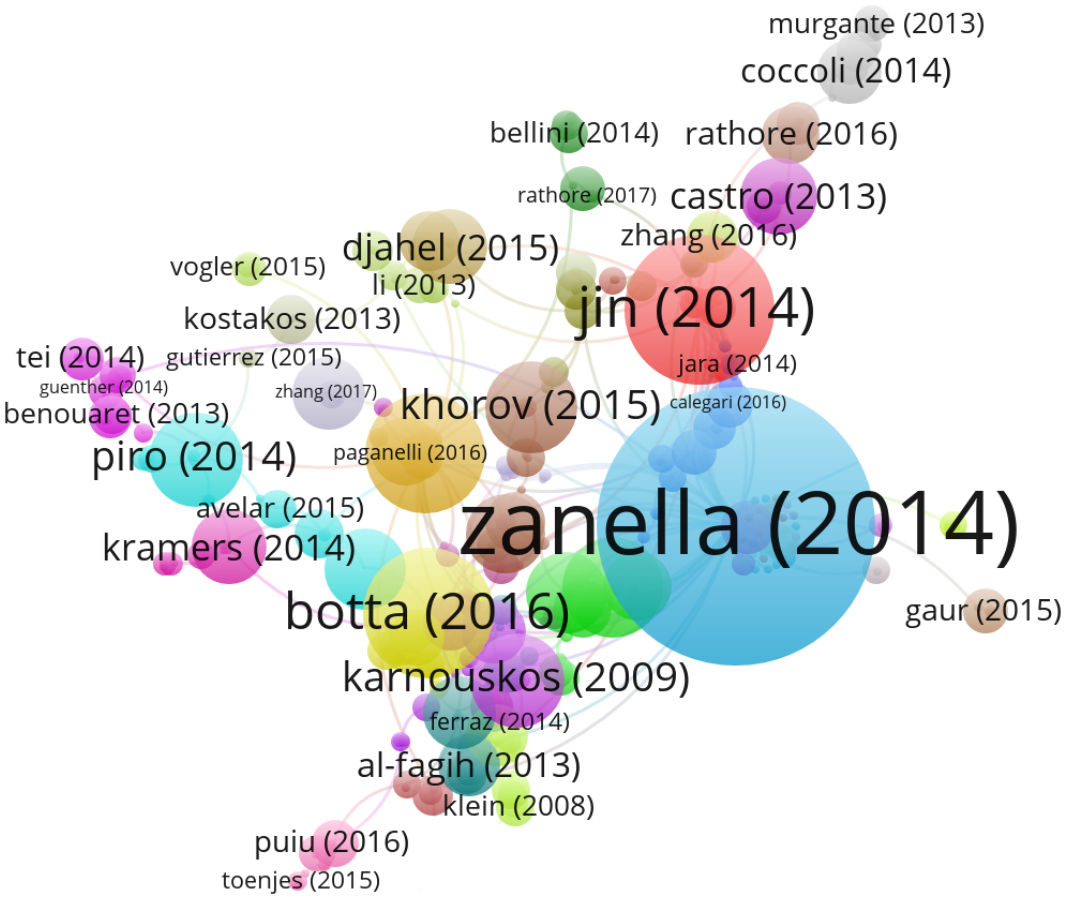


FIGURE 4.5: CS and IT publications classified as SC that are strongly related to the main hot spot shown in Figure 4.3

UNIVERSIDAD DE MÁLAGA

Chapter 5

Towards Intelligent Smart City Applications

In this chapter, we summarize our initial approaches to the SC. Particularly, in these studies, besides from having got in touch with the domain, they helped us to clarify potential interesting problems to apply DN. It is important to highlight that the contributions of these works are related to the goal **G3** of the plan of the thesis. Although these papers are not DN proposals/applications, the scope of the problems (i.e., the SC) and the techniques used to tackle those problems (e.g., DE, EA, ...) are strongly related to the goals of this thesis.

We have published three papers, two of them in JCR indexed journals [24, 28], and one in the proceedings of an international conference [20]. Appendix C presents the details of these publications, and Appendix A introduces the problems and data sets covered by them.

The remainder of this chapter introduces the referred articles. The first two sections summarize our proposals to partition real-world maps to improve the shortest path computation [20, 24]. Then, we present our study to segment electricity customers [28].

5.1 Tile Map Size Optimization for Real World Routing by Using DE

A common problem in road travelling is finding the shortest path between two places. This problem has been studied for decades, and many solutions (i.e., algorithms) have been proposed to cope with it. Also, this problem is very important in the context of smart mobility [23]. However, in (real-world) practice, the immense size of maps poses a challenge for this algorithms due to the impact over the temporal complexity of the routing algorithms.

In this work [20], we propose to tackle the shortest path problem in the context of real (large) maps. We proposed a data partitioning strategy, and we tested it in a real-world scenario. Therefore, the main contribution (Section 1.3) of this work is:

> **DN and SC** We proposed a strategy to automatically partitioning a real-world road map into *tiles*, and to selecting the minimum set of tiles needed to compute the shortest path between two places using DE, that improves the time performance of the shortest path computation.

Particularly, to address the mentioned problem, we proposed to follow a *divide et impera* strategy. More precisely, given a road map G_m, we proposed to partition the map into a *tile map* $M = \{G_1, ..., G_n\}$, where $G_i \subseteq G_m$ is a tile of size $s_i = t_{lat}, t_{lon}$, that

UNIVERSIDAD DE MÁLAGA

covers the region R_i of the map. We restricted that there is no overlap between tiles, i.e., $R_i \bigcap R_j = \emptyset$, $\forall i \neq j$, and that $\bigcup_i^n G_i = G_m$. Figure 5.1 shows a map partitioned into a set of *tiles*.

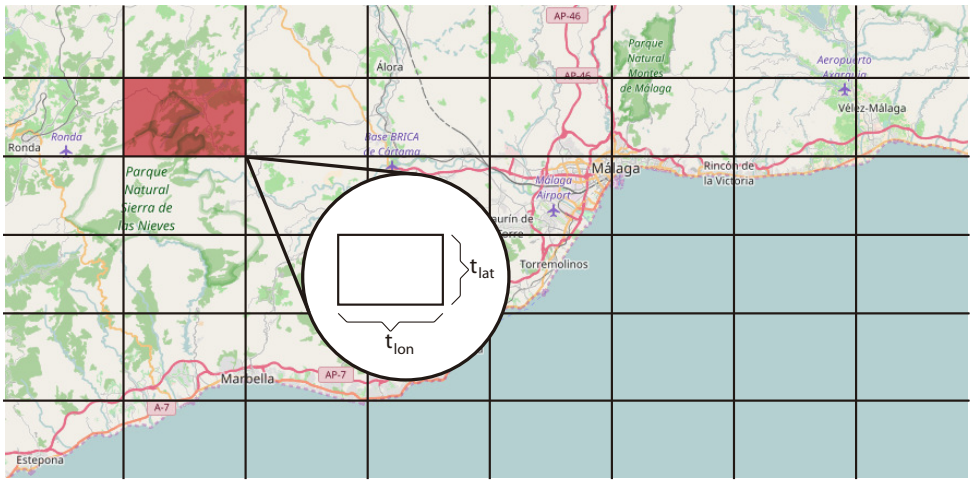


FIGURE 5.1: Tested region partitioned into tiles (Malaga, Spain)

Then, given a number N of *origin-destination places* P_k, we proposed to minimize the shortest path computation time, considering the time needed to compute the path ρ, and the time needed for preparing the data ϕ, i.e., loading the map from a data base (DB) into memory. Equation 5.1 states the problem.

$$\text{minimize } \frac{1}{N}\sum_{k}^{N}(\phi(P_k) + \rho(P_k)) \tag{5.1}$$

$$\text{subject to } s = s_i, 1 \leq i \leq n \tag{5.2}$$

$$R_i \bigcap R_j = \emptyset, 1 \leq i, j \leq n, i \neq j \tag{5.3}$$

$$\bigcup_{i}^{n} G_i = G_m \tag{5.4}$$

We designed a DE-based [115] algorithm to solve Equation 5.1, where each solution is encoded as a two dimensions vector (i.e., the lat and long size).

We implemented a full stack solution using Java, Python, and MySQL. The data is extracted, transformed and loaded into a custom DB (*step 0*). Then, the data is partitioned into tiles using the DE algorithm (*step 1*). Finally, every time a route is requested, a subset of the tile map is loaded into main memory from the DB, and the shortest path is computed using the Dijkstra [41] algorithm (*step 2*). Figure 5.2 depicts the implemented system at a high level. We tested our proposal using the map of the Province of Malaga (extracted from Open Street Map, latitudes 36.47 and 36.86, and longitudes -4.99 and -4.04), Spain, to test. The map includes approximately 46,000 intersections and 64,000 streets, distributed in a 4500 Km^2 area, and encompassing a dozen cities, including: Malaga, Marbella, Torremolinos, Mijas, Fuengirola, among others. Figure 5.1 shows the selected region.

To set a baseline, we computed the itineraries without partitioning. Particularly, we implemented two variations of this approach: *Dij-DB* and *Dij-Mem*. Dij-DB stores the map in a DB, and loads the data into main memory for computing SP. Dij-Mem

UNIVERSIDAD DE MÁLAGA

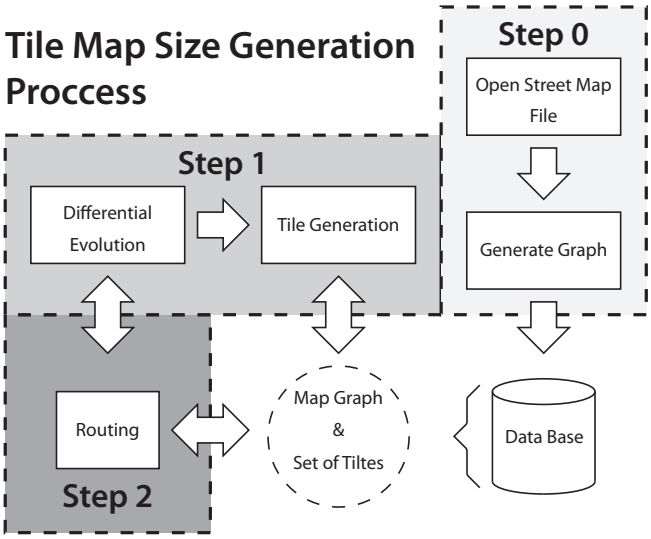


FIGURE 5.2: Tile map routing system overview

retains the map into main memory. Notice that the last approach will not scale up to larger maps, i.e., a larger map may overflow memory.

As to test the partitioning strategy, we implemented three search strategies: DE, Manual, and Random. The Manual search consisted of the combinations of the following values: 0.01, 0.1, 0.2, 0.5, 1, and 10 (i.e., 36 tile size pairs), and we selected the 30 best solutions.

We repeated the experiments 30 times, considering 182 pairs of points, and compared the results. Figure 5.3 summarizes the performance (fitness) of the strategies tested. The results show that an efficient data managing strategy results into great time improvements. Our proposed strategy (tile map and DE) even improved Dij-Mem! Meaning that it is not necessary to have all data loaded in memory to achieve the best performance.

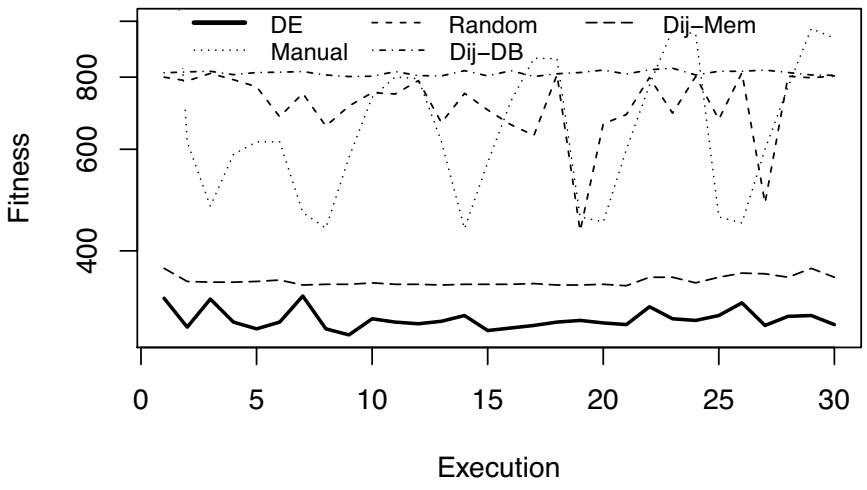


FIGURE 5.3: Fitness comparison of the tested approaches

UNIVERSIDAD DE MÁLAGA

5.2 Road Map Partitioning for Routing by Using an mSSEA

In this work [24], we continued exploring the *tile map* partitioning strategy. Particularly, we extended the proposed approach (Equation 5.1) by enabling heterogeneous tile sizes, and we proposed a micro Steady State Evolutionary Algoritm (mSSEA) to search for the optimal partitioning scheme. We tested our proposal in two different conditions, a sparsely populated big area (Province of Malaga, Spain), and a dense populated region (Mexico City, Mexico).

The main contribution of this work is:

DN and SC We proposed an improvement to the tile map partitioning strategy, which is suitable for dense populated areas, and we introduced a $(\mu + \lambda)$-microSSEA to optimize the partitioning.

First, we proposed to relax the tiling scheme proposed in [20]. Particular, we proposed to divide the map into heterogeneous tiles, i.e., tiles of different size.

Equation 5.5 depicts the defined problem, where G_m is a given a road map, $G_i \subseteq G_m$ is a tile, R_i is the region covered by the tile G_i, N is the number of *origin-destination places* P_k, ρ is the time needed to compute a path, and ϕ is the time needed for preparing the data.

$$\text{minimize } \frac{1}{N}\sum_{k}^{N}(\phi(P_k) + \rho(P_k)) \tag{5.5}$$

$$\text{subject to } R_i \bigcap R_j = \emptyset, 1 \leq i, j \leq n, i \neq j \tag{5.6}$$

$$\bigcup_{i}^{n} G_i = G_m \tag{5.7}$$

Then, we proposed an mSSEA to solve Equation 5.5. This algorithm is based on the steady state genetic algorithm (ssGA) [117] and the *microGA* [38]. Particularly, we designed the algorithm with the intention to reduce the number of fitness evaluations (each evaluation is costly), while maintaining the diversity of the population through the restarts. Algorithm 4 introduces the mSSEA at a high level. A small population of individuals is evolved. When the nominal convergence is reached, the best λ individuals from the population are selected. Then, the algorithm is restarted, using the selected individuals and $\mu - \lambda$ new individuals as the initial population.

We benchmarked our proposal against eight Dijkstra-based [41] competitors, i.e., different data managing strategies that use the same point-to-point shortest path algorithm (Dijkstra), and against a Map-Reduce approach [10]. Table 5.1 presents the list of competitors. *Tile* stands for the tile map partitioning [20], and *Tile+* stands for the heterogeneous (unconstrained) tile map partitioning introduced in this study.

First, we benchmarked the mSSEA against the state-of-the-art in the task of finding the best heterogeneous tile partitioning scheme. We repeated five independent times the optimization using the Province of Malaga, Spain, and 182 pairs of points. The results are summarized in Table 5.2.

Then, we compared the heterogeneous partitioning scheme against the original tile map partitioning. Figure 5.4 presents the results (fitness, in milliseconds) of the two best solutions reported in the tile map study [20] (*DE*, and *D-Mem*), along with the mSSEA results for the Province of Malaga, Spain. Overall, the heterogeneous tile partitioning and mSSEA outperform all its competitors.

UNIVERSIDAD DE MÁLAGA

Algorithm 4 Pseudo-code of $(\mu + \lambda)$-microSSEA

```
1: population ← initializePopulation(μ)
2: while not GlobalConvergence() do
3:     while not NominalConvergence() do
4:         Evaluate(population)
5:         parents ← Selection(population)
6:         offspring ← Recombination(parents)
7:         mutated ← Mutation(offspring)
8:         Evaluate(mutated)
9:         Replace mutated in population
10:    end while
11:    population ← getBest(population, λ)
12:    Insert(population, initializePopulation(μ − λ))
13: end while
```

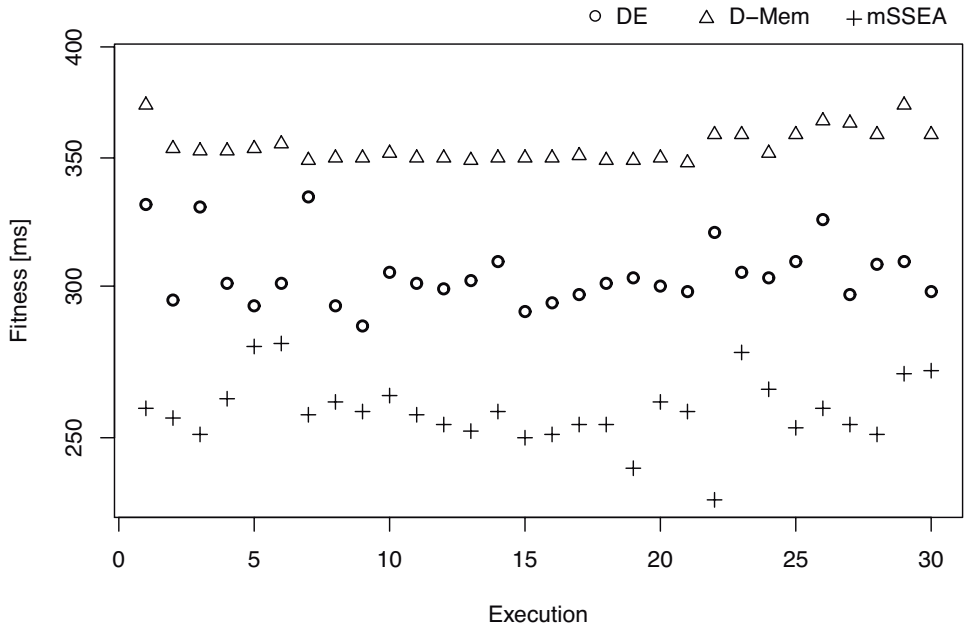


FIGURE 5.4: Performance comparison in Malaga, Spain (the lower the better)

Figure 5.5 shows a heterogeneous tile map partitioning scheme for the Province of Malaga. The solution representation suggests that there is a relation between the rod map density and the size of the tile.

Then, to study the scalability of our proposal, we repeated our experiments using Mexico City (latitudes 18.99 and 19.92, and longitudes -99.60 and -98.61), and considering 2450 pairs of points. This data set includes more than 250,000 intersections and nearly 360,000 streets, i.e., it is 6 times larger than the Province of Malaga. The results are summarized in Table 5.3. In this case, the tile map partitioning showed to be even more important to speed up the shortest path computation.

Finally, we outlined a qualitative comparison between our approach and MR [10]. Particularly, we compared the ratio between the time needed to compute the shortest

UNIVERSIDAD DE MÁLAGA

Strategy	Partitioning	Description
D-DB	No	The map is stored in a DB, the map is retrieved and loaded in main memory every time a shortest path is computed
D-Mem	No	The map is retained in main memory all time
Rand	Tile	The tile partitioning scheme is selected randomly, and it is stored in a DB. Then, every time a shortest path is computed, a subset of the tiles is loaded into main memory
Manual	Tile	Idem, but the partitioning scheme is set manually
DE	Tile	Idem, but the partitioning scheme is set using DE [20]
CMA-ES	Tile+	Idem, but the standard Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [56] is used to set the heterogeneous partitioning scheme
$(\mu + \lambda)$ES	Tile+	Idem, but the heterogeneous partitioning scheme is set using a mu plus lambda ES [11]
PSO	Tile+	Idem, but using a PSO [67] to set the heterogeneous partitioning scheme
MR	Other	A MapReduce-based approach for computing the shortest path over large-scale road networks [10]

TABLE 5.1: Shortest path data managing competitors

TABLE 5.2: The mSSEA against the state-of-the-art

	CMA-ES	$(\mu + \lambda)$ES	PSO	mSSEA
Median	187	205	202	191
Mean	194.0	203.0	209.8	188.4
SD	16.2	13.8	18.9	5.0
Min	177	181	192	182
Max	215	215	233	193
Avg Partition Size	100	100	100	53.2

TABLE 5.3: Overall comparison of the fitness (Mexico City) in [ms]

	D-Mem	D-DB	mSSEA
Median	12771	16997	147
Mean	12807	19459	238
Max	46734	5915208	42671
Min	11607	14648	119
SD	1473	119173	1688

path using the proposed strategy and the time reported without using the data management strategy. In other words, the improvement made by the data managing technique. On average, MR reported a ratio of 6.7. In our case, the ratio for Malaga was equal to 1.4, while for Mexico City was equal to 53.8. Thus, we concluded that our tile map partitioning scheme presents a competitive alternative to MR in terms of performance.

UNIVERSIDAD DE MÁLAGA

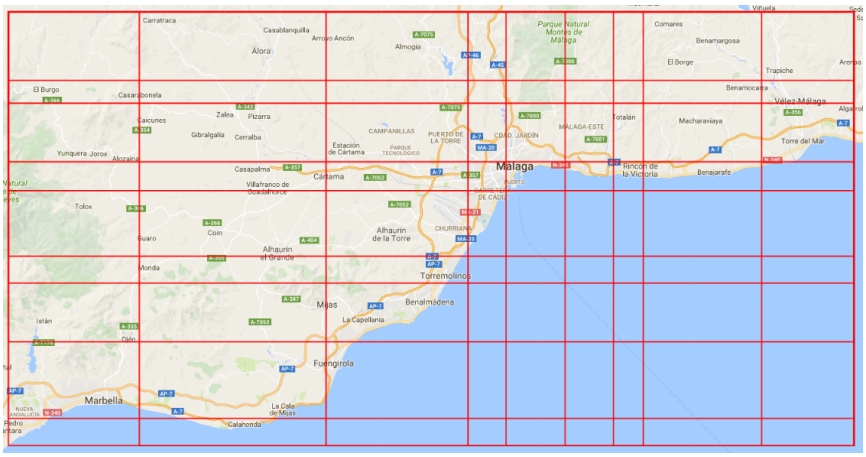


FIGURE 5.5: Heterogeneous tile map partitioning for Malaga, Spain

5.3 Customer Segmentation Based on the Electricity Demand Signature

A smart meter allows electric utilities to gain an in-depth understanding of the consumption habits of their customers. Thanks to this, electric utilities can improve their own processes (e.g., load forecast, capacity planning, etc.) and tailor the services they offer to their customer needs, an important step toward smart electricity [23].

Ideally, companies treat each single customer as an individual. However, in practice, it is very complicated. Instead, companies target their services to (small) groups of customers. Therefore, the challenge is to combine this detailed individual-level data in a meaningful way that is still insightful for a set of numerous customers.

In this work [28], we propose the *demand signature*, an innovative approach to characterize and group electricity consumers based on their demand habits. We tested our approach in a real-world scenario, and in collaboration with an electricity managing consultant company (Bettergy S.L.[1]). Thus, the main contributions (Section 1.3) of this work are:

DN and SC We introduced the *electricity demand signature*, an innovative approach to characterize the behavior of an electricity customer based on the relative importance of the measurements of the daily load curve, which is estimated using a ssGA.

DN and SC We compared the electricity demand signature against the *characteristic typical load curve* [119] in a real world scenario. The results show that our proposal is capable of managing a large and varied number of customers.

Figure 5.6 outlines the electricity demand signature computation process at a high-level. The most relevant features are extracted from the daily load curves using an ssGA. Then, these features are combined into the electricity demand signature.

Intuitively, the most relevant information of a data set is the subset of the data that allows the characterization of the original data set without losing information (a.k.a. feature selection). Then, given a daily load curve L, sampled at an arbitrary daily rate R (i.e., the number of samples per day), Y the original set of samples or features of L (i.e., $|Y| = R$), $S \subseteq Y$, and $B = \{L_1, ..., L_n\}$ a set of daily load curves.

[1] https://www.bettergy.es/

UNIVERSIDAD DE MÁLAGA

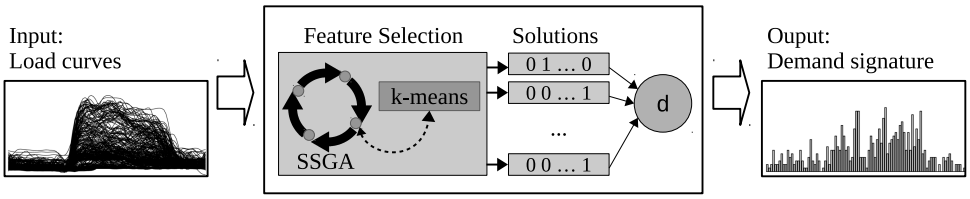


FIGURE 5.6: Electricity demand signature computation

We defined the problem of selecting the most relevant information of a data set as a maximization problem (Equation 5.8):

$$\begin{aligned} &\text{maximize} \;\; \alpha \times \rho(\kappa(B,Y,G),\kappa(B,S,G))^2 + (1-\alpha) \times (1 - \frac{|S|}{|Y|}) \\ &\text{subject to} \;\; S \subseteq Y \end{aligned} \tag{5.8}$$

Where $\kappa(B,S,G)$ is the function that partitions the set B into G groups using the features selected by S, and $\rho(P_1,P_2)$ is the *Rand index* [62] (a measurement of partitions similarity). Note that we added the ratio of selected futures to balance the quality of the clustering and the number of features, and a parameter (α) to set the importance of each term.

Then, since two different subsets (e.g., S_i and S_j) may have the same *fitness* value (i.e., the target function in Equation 5.8), we defined the electricity demand signature as the vector $\phi \in \mathbb{N}^R$ ($R = |Y|$), that sums the subsets found over 30 independent runs of the optimization process [111]). In other words, the electricity demand signature represents the frequency or density distribution of the features selected. Finally, to maximize Equation 5.8 we designed an ssGA.

Figure 5.7 depicts a year of daily load curves (stacked) on the left, and the computed electricity demand signature on the left (shown as a histogram). A higher bar in the histogram implies that a feature is more relevant.

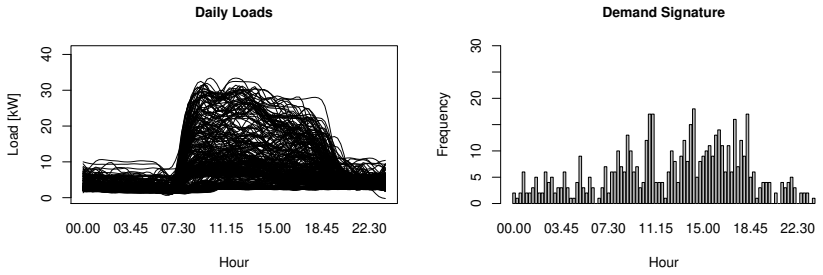


FIGURE 5.7: Daily load curves and the electricity demand signature

To evaluate the usefulness of the electricity demand signature we (*i*) benchmarked it against the state-of-the-art of feature selection, and (*ii*) compared it to the load demand characterization [119]. All tests were executed using the load demand of 64 buildings located in Andalusia, Spain, recorded every 15 minutes for a year. Refer to the Appendix A for a detailed overview of the data set.

First, we compared our ssGA against *Information Gain* (IG) [105] and *Correlation-based Feature Selection* (CFS) [55], and *Random Search* as a sanity check. Table 5.4 summarizes the results in terms of fitness (target function in Equation 5.8) for all

UNIVERSIDAD DE MÁLAGA

buildings. A value closer to one is desirable. The results show that the ssGA is the most suitable approach to tackle our problem.

TABLE 5.4: Fitness results summary for the feature selection problem

	ssGA	Random	IG	CFS
Mean	0.96	0.84	0.83	0.73
Median	0.97	0.88	0.83	0.85
Max	0.99	0.92	0.94	0.99
Min	0.91	0.34	0.33	0.38
SD	0.02	0.11	0.08	0.24

Second, we compared the electricity demand signature against the characteristic typical load curve [119]. Particularly, we computed both characterizations for all customers (independently), and then we performed an internal cluster validation study [53], i.e., we grouped the customers using *k-means*, *PAM* and *Hierarchical* techniques [2, 53] (for sizes ranging from two to ten), and we evaluate the quality of the grouping using the Connectivity, Dunn, and Silhouette indexes. At a glance, the Connectivity index analyses the number of elements that are placed in the same cluster as their nearest neighbors, it ranges from zero to infinity, and it has to be minimized. The Dunn index seeks for dense and well-separated clusters, thus a higher value is desirable. The Silhouette coefficient contrasts the average distance between the elements of one cluster to the average distance to the elements of another one (a higher value is better). Table 5.5 summarizes the best results for all combinations, where *Cls* stands for the *optimal* number of classes according to the metric.

TABLE 5.5: Optimal grouping based on the internal cluster validation

	Connectivity		Dunn		Silhouette	
	Score	Cls	Score	Cls	Score	Cls
Demand Signature	12.91	2	0.62	8	0.29	2
Typical Load Curve	3.49	2	0.42	4	0.75	2

Overall, the results suggest that the electricity demand signature is a meaningful way for characterizing electricity customers, and that it can be helpful to group customers according to their behavior.

UNIVERSIDAD DE MÁLAGA

This page intentionally left blank.

UNIVERSIDAD DE MÁLAGA

Chapter 6

Deep Neuroevolution: Smart City Applications

In this chapter, we summarize the main contributions of this thesis related to the applications of DN to solve SC problems. These results are encompassed in the goal **G3** of the plan. Particularly, we have delivered four studies. One was published in an SJR journal, two in the proceedings of an international conference, and one in the proceedings of a national conference. The latter was awarded the *Second Prize* for the best student work in *Doctoral Consortium CAEPIA 2018*).

In the remainder of this chapter we summarize our works. First, we present our approach to predicting the car park occupancy rate [29]. Then, we outline our contributions to smart waste management [21, 30]. Finally, the work presented in *Doctoral Consortium CAEPIA 2018* is introduced. More details of the publications and the data sets are available in Appendices C and A, respectively.

6.1 Evolutionary DL for Car Park Occupancy Prediction in SCs

Smart mobility is one of the most *popular* SC domains. Within this domain, predicting the car park occupancy rate poses an interesting problem, not only because it is technically challenging, but also it has a notable impact in daily life of citizens.

In this work [29], we proposed to apply a DN algorithm to tailor an RNN for predicting the occupancy rate of 29 car parks located in Birmingham, UK. Particularly, the main contributions of this study is:

DN and SC We defined two DN algorithms (GA and EA-based) to automatically design efficient RNNs to predict the car park occupancy rate.

Particularly, we proposed two DN techniques to optimize the architecture of an RNN, a GA-based and a EA-based approach. We posed the architecture optimization challenge as a minimization problem, where the target function is the MAE on the test data. In both cases, we trained each solution using Adam [68] for a fixed number of epochs, then, we evaluated its performance on test.

At a glance, in the GA-based [61] approach, we represented each solution as a variable length integer vector. Specifically, we encoded two training features, the *dropout* (to avoid over-fitting) and the look back or time steps, and the architecture (the number of neurons per each hidden layer) of the RNN. Thus, the vector length varies according to the number of hidden layers. On the other hand, we designed a $(1+1)$EA [11], using the same encoding defined for the GA. Note that we evolved a single solution because of the high computational cost of evaluating the fitness (i.e., training and computing the MAE on the test data) of a solution.

UNIVERSIDAD DE MÁLAGA

We tested our proposal using the occupancy rates of 29 car parks operated by the National Car Parks in the city of Birmingham, UK. Please, refer to Appendix A for more details. The code and the data are available in DLOPT [19] GitHub. First, we benchmarked the GA and EA algorithms against *expert defined* architectures, aiming to asses the benefit of automatic DNN design. Second, we compared our results to the state-of-the-art in prediction, to gain insights of the quality of our solutions in the targeted problem.

Table 6.1 summarizes the results of the architecture optimization benchmark (30 independent runs). The expert designed architecture models (i.e., *Small*, *Medium*, and *Large*) were taken from Google Tensorflow sample models[1]. The results show that the automatic design (i.e., GA and EA-based) outperforms the manual design in terms of the error (MAE). Regarding the architecture, the DN approaches produced smaller (i.e., with less trainable parameters) and deeper networks. Thus, we concluded that not the number but the arrangement of the neurons is the key to tailor a DNN to a given problem.

TABLE 6.1: Car park occupancy rate architecture optimization

Architecture	Hidden Layers	Trainable Params	MAE
Small	2	511,228	0.34
Medium	2	5,171,428	0.32
Large	2	27,234,028	0.84
Mean GA-based	6	53,696	0.08
Max GA-based	7	100,964	0.09
Min GA-based	5	7,684	0.08
Mean EA-based	6	7,136	0.08
Max EA-based	7	8,452	0.10
Min EA-based	5	5,820	0.08

Continuing with our study, we analyzed all evaluated architectures (i.e., not only the final solution). We sorted them by their MAE into deciles. Then, we computed the smooth density estimates for each decile. Figure 6.1 presents the density for the architectures grouped by the number of hidden layers (left), and the number of neurons (right). The density estimates show that the best solutions (first decile) are mainly consisting of architectures of eight layers, arranging 250 to 350 neurons.

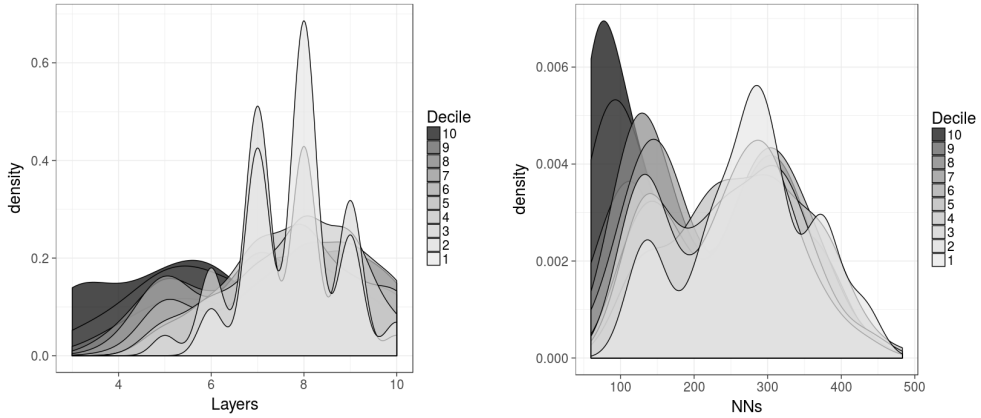
Later, considering that one of the goals of this study is to improve the state-of-the-art in car park occupancy rate prediction, we compared the best solution found to six prediction techniques [114]: polynomials (P), Fourier series (F), k-means clustering (KM), polynomials fitted to the k-means centroids (KP), shift and phase modifications to KP polynomials (SP), and time series (TS). In the referred study, the authors considered each parking lot (out of the 29) to be a single entity. Therefore, for each technique, they trained 29 predictors (one for each parking lot).

Table 6.2 summarizes the prediction results. Our optimized RNN does not exceed all its competitors. Moreover, there is no significant difference between its predictions and the ones made using *polynomials* (Wilcoxon test p-value=0.096) and *time series* (Wilcoxon test p-value=0.099). Nonetheless, it is important to remark that a single RNN predicted the 29 parking lots at the same time, while the other techniques use one predictor for each parking lot (i.e., 29 predictors).

In this study, we concluded that RNN architecture optimization is a promising field of study, because most efforts have been made in regard to FFN architectures.

[1] https://github.com/tensorflow/models

UNIVERSIDAD DE MÁLAGA



(a) (b)

FIGURE 6.1: Density distribution of hidden layers/neurons by decile (performance)

TABLE 6.2: Mean absolute error of the predicted occupancy

Parking lot	P	F	KM	KP	SP	TS	RNN
Median	0.070	0.078	0.087	0.086	0.074	0.069	0.077
Mean	0.067	0.079	0.102	0.101	0.073	0.067	0.079
Max	0.132	0.148	0.177	0.179	0.139	0.129	0.137
Min	0.016	0.024	0.048	0.049	0.025	0.023	0.033
SD	0.029	0.033	0.035	0.035	0.033	0.026	0.028

Also, the results show that an appropriate architecture improves the performance of an RNN, and that adding hidden layers or neurons arbitrarily to an RNN do not guarantee an improvement in its generalization capability. Therefore, the architecture optimization is very important.

6.2 Waste Generation Prediction in SCs Through DN

According to the United Nations, 55% of the World population lives in urban areas, and by 2030, it is expected that the number will rise up to 66% [121]. This fast growing context brings new challenges, ranging from housing to utilities capacity planning. Among these challenges, managing the waste is a very important (and sometimes under-looked) one. Nobody wants to live in a *dirty* city, and a poor waste management introduces a public health risk.

In this work [21], we addressed the waste collection problem. Particularly, we proposed a DN approach to tailor an RNN that predicts the filling level of the waste containers of a city, Then, using these accurate predictions, it is possible to improve the planning for the collection of the containers. Therefore, the main contribution (Section 1.3) of this work is:

DN and SC Using a Self Adapting (1+1)EA-based DN technique, we optimized an RNN that predicts the filling level of the waste containers of a city. We tested our approach on a real use case, and compared our results against the state-of-the-art.

We analyzed a real case study, introduced by Ferrer and Alba [50, 51], the waste collection planning of a city in Andalusia, Spain. Particularly, we faced the *waste*

UNIVERSIDAD DE MÁLAGA

generation prediction problem (Appendix A), that consists of predicting the filling level of 217 paper containers, distributed in the metropolitan area of the referred city.

To solve the problem, we designed a hybrid DN algorithm based on the Adam weights optimizer [68] and the $(1+1)$EA [11].Particularly, it optimizes the architecture of an RNN (i.e., the number of hidden layers and the number of neurons per layer), and tunes the look back or time steps for the truncated in time weights optimization (i.e., the training using Adam). Algorithm 5 presents a high-level view of the proposed DN algorithm.

Algorithm 5 Self Adapting (1+1)EA-based RNN architecture optimizer

```
1: solution ← Initialize()
2: Evaluate(solution, evaluation_epochs)
3: evaluations ← 1
4: while evaluations ≤ max_evaluations do
5:     mutated ← Mutate(solution, mut_element_p, mut_length_p, max_step)
6:     Evaluate(mutated, evaluation_epochs)
7:     if Fitness(mutated) ≤ Fitness(solution) then
8:         solution ← mutated
9:     end if
10:    evaluations ← evaluations + 1
11:    SelfAdapting()
12: end while
13: solution ← Evaluate(solution, final_epochs)
14: return solution
```

At a glance, an RNN architecture and the look back is encoded using a variable-length integer vector (*solution*). The solution is assessed by a *Fitness* function, in this case, it is trained using Adam for a pre-defined number of epochs. Then, the solution is evolved by a *Mutate* function that add/remove layers and neurons to the architecture, and alter the look back with a probability (*mut_element_p*, *mut_length_p*) within a maximum range of modification (*max_step*). After each evolutionary step, the algorithm self-adapts its parameters [42]. Once the termination criteria is met, the solution is trained (*Evaluate*) using Adam for a *final_epochs* number of epochs.

We benchmarked our proposal against the results presented by Ferrer and Alba [50, 51], using the custom metric defined in the referred studies. Specifically, we computed the "mean absolute error in the filling predictions of the next month" (MM) using the solutions given by our algorithm, i.e., we predicted the filling level for the 217 on the test data (a month), and summed up the predictions per container. Then, we computed the mean absolute difference between the predicted values and the ground truth. A smaller MM value is desired.

We run 30 independent times the optimization process, we computed the MM on the test data, and we picked the median solution. Table 6.3 shows the performance of the median solution (RNN), and the results presented in [50, 51], i.e., Linear Regression, Gaussian Processes, and Support Vector Machines for Regression (*SMReg*). The results show that our RNN outperformed all its competitors.

6.3 Waste Generation Prediction Under Uncertainty

Due to the great importance of waste prediction in the context of smart waste management, and considering the great improvements made by DN in this domain [21],

UNIVERSIDAD DE MÁLAGA

TABLE 6.3: MM statistics for the waste generation prediction problem

Method	Error
RNN	0.028
Gaussian Processes	0.038
Linear Regression	0.074
SMReg	0.095

we decided to dove deeper into the problem. Particularly, we tackled the problem of predicting the filling level under uncertainty. This specific issue is very important for a real-world smart waste management application because it is known that the measurements of the filling level of a waste container usually have deviation with respect to reality. For example, in the context of paper containers, an empty cardboard box may be placed just underneath the sensor, thus the filling level could be deceptive.

Therefore, the main contribution (Section 1.3) of this work [30] is:

DN and SC We studied the effect on the error of the predictions made under uncertainty by an RNN on the waste prediction problem.

In this study, we tailored an RNN to the waste prediction problem (refer to Appendix A) using the DN strategy proposed in our previous study on the matter [21]. Then, we synthesized an uncertainty data set based on the original data.

Particularly, we defined three error levels ($p = 5, 10, 20$), that correspond to the percentage of faulty measurements. Also, we defined two error types, the *random* error, i.e., the sensor gives an erroneous measurement, and the *zero* error, i.e., there is no data for a specific sensor and day. To simulate these errors, we replaced a p percent of the valid filling levels in the data set with (*random* error) random numbers in the range [0, 100] (i.e., *random* error), or with a zero (*zero* error). We combined the three error levels with the two error types, and we prepared six synthetic instances.

Then, to study the effect of these errors, we used the DN algorithm to search for an optimal RNN on the original (unmodified) data set 30 independent times. Then, we retrained each RNN on the six synthetic instances. Table 6.4 summarizes the results of this experiment.

TABLE 6.4: Reliability benchmark on waste prediction

	5% Error		10% Error		20% Error	
	Random	Zero	Random	Zero	Random	Zero
Mean	0.084	0.075	0.099	0.076	0.118	0.080
Median	0.081	0.074	0.092	0.075	0.114	0.076
Max	0.113	0.097	0.215	0.098	0.186	0.116
Min	0.075	0.067	0.087	0.070	0.093	0.069
SD	0.008	0.006	0.023	0.007	0.018	0.011

We selected the median solution per percentage and type error, and computed the custom metric defined by Ferrer and Alba [50, 51], i.e., the "mean absolute error in the filling predictions of the next month" (MM). Table 6.5 presents the results calculated for the RNN predicting under uncertainty (i.e., *Random* and *Zero* followed by the percentage), as well as the results introduced by Ferrer and Alba [50, 51] (Gaussian Processes, Linear Regression, and SMReg), and in our previous study [21] (RNN).

As expected, the results show that the performance of the predictions is negatively affected by the uncertainty in the data. Also, no having data (i.e., the error type *zero*)

UNIVERSIDAD DE MÁLAGA

TABLE 6.5: MM statistics for the waste generation prediction problem

Method	MM
RNN	0.028
Gaussian Processes	0.038
Linear Regression	0.074
SMReg	0.095
Random 5%	0.037
Zero 5%	0.029
Random 10%	0.064
Zero 10%	0.030
Random 20%	0.074
Zero 20%	0.042

has less impact than the error of type *random*. In other words, having noisy data is worst than having no data (constraint to the limits of this experiment). Finally, it is worth noticing that even with 5% missing data (i.e., error type zero), the RNN stills outperforms its competitors Moreover, a missing datum (zeros) has less impact than a random noisy datum. On the other hand, if the uncertain data represent less than the 5%, the RNN still beats all its competitors.

6.4 Doctoral Consortium

In the last decade the emergence of Deep Learning and deep neural networks has revolutionized society. More and more applications are making use of it, from autonomous cars to robotic assistants. However, as these networks grow, we face new challenges. One particular problem is the search for an optimal DNN design, in other words, finding the network that best suits a given problem.

Various approaches have been proposed to address the problem, but in practice few have been adopted because of the excessive computing time they require. In recent years, the resurgence of advanced metaheuristics applied to the design of neural networks, known as DN, has made it possible to find unprecedented solutions in increasingly reasonable times, opening the way to their practical adoption.

In this work [22], we present the doctoral research project to which this thesis refers. Therefore, the main contribution of this work is:

MRS, RESN, DN and SC, DLOPT To publicly disseminated the content of the doctoral thesis "Deep Neuroevolution: Smart City Applications" (*Neuroevolucion Profunda: Aplicaciones en Ciudades Inteligentes*).

It is worth noticing that this publication was awarded the Second Prize for the best student work in *Doctoral Consortium CAEPIA 2018*. Also, it is important to highlight that several recommendations were received during the presentation of this work (and they were included in the following studies). Furthermore, the competition was an excellent way of improving the outreach of the thesis (to other CS domains), as well as to enlarging the network of contacts.

Moreover, readers may be interested in comparing the plan presented in the manuscript with the one presented in this thesis. To cite but one example, the plan presented in Doctoral Consortium CAEPIA 2018 is exactly the plan presented in this thesis (Figure B.2). Note that the plan is presented in Spanish, i.e., in the format it

UNIVERSIDAD DE MÁLAGA

was presented during the competition. However, all activities, milestones, and dates are the same as the ones presented in this doctoral thesis (Chapter 1).

Besides presenting the doctoral thesis research project, in this manuscript we discussed the implications of the whole process of becoming a doctor. Particularly, we face this question from three different areas: personal, scientific and industrial.

In the *personal* area, we envisioned that this thesis will allow the doctoral candidate to develop the skills necessary to systematically understand a field of study, as well as to develop the capacity to carry out quality research in that field. Moreover, it is also expected that the candidate will develop the ability to critically analyze, evaluate and synthesize new and complex ideas, as well as the capacity of integrating in multidisciplinary and multicultural teams, through the daily work in the environment of the research group and promoting the stay in external centers.

In the *scientific* area, it is expected that the doctoral candidate will contribute to the state-of-the-art, which will be validated through the achievement of publications. And in the *industrial* field, due to high applicability of this thesis, it is expected that the candidate will contribute to solving real world problems in the context of the SC. Ideally, these contributions will be transferred to the industry, either as a technical collaboration or through the creation of a commercial product. Although this point is extremely complex, managing to contribute to it would have a positive impact on the economy and, more importantly, could mean a contribution to our cities.

Overall, it is expected that the doctoral candidate will contribute to society, using his skills to improve the well-being of humanity.

UNIVERSIDAD DE MÁLAGA

This page intentionally left blank.

UNIVERSIDAD DE MÁLAGA

Chapter 7

Conclusions and Future Work

Writing the conclusion of this doctoral thesis is not an easy task. This work encompasses all the scientific output gathered during this four-year program, but most important, it underlies all the lessons learned during this time.

First, we can discuss the scientific output highlighted in this doctoral thesis. Particularly, we have presented a work plan, with clearly defined objectives (refer to Section 1.1), and the studies carried out to fulfil these goals. All this work led to this compendium and to several publications, including **four JCR-indexed journal** publications (and one in second round revision), **one SJR-indexed journal** publication, **three international conference** papers, **two national conference** publications, where one was awarded the **Second Prize** for the best student work in Doctoral Consortium CAEPIA 2018, and two studies available in arXiv.

Second, we can comment on the contribution of this thesis and the accomplishment of our goals. Particularly, we have split our contribution into four key aspects, namely the **MRS**, **RESN**, **DN and SC** problems, and **DLOPT**, and we have presented the *intermediate* steps toward the accomplishment of our goals made by each study.

More specifically, we introduced the **MRS**, a training-free technique to predict the performance of an ANN, and we highlighted the following *intermediate* contributions.

MRS We proposed the MRS, a low-cost technique (i.e., time and computational resources) to predict the performance of an ANN based on a random sampling [31].

MRS We studied the relations between the number of LSTM cells, the look back or time steps, the distribution of the cells (i.e., the number of hidden layers), and the MRS [25].

MRS We analyzed the MRS memory and time consumption [25].

MRS We proposed an RNN architecture optimization technique based on BO and the MRS [27].

Therefore, we have accomplished our first goal: Define a method for characterizing RNNs that allows for comparison of its expected performance without the need for training.

Also, we have proposed a DN approach to optimize an RNN that relies on the MRS to search for an optimal architecture:

RESN We proposed RESN, a $(\mu + \lambda)$EA-based algorithm that uses the MRS to speed-up the RNN architecture optimization process [26].

UNIVERSIDAD DE MÁLAGA

Thus, our second goal is fulfilled: Design and implement a technique to optimize an RNN based on metaheuristics, and the previously mentioned characterization.

Moreover, we have applied our DN proposals to SC problems, e.g., car park occupancy rate prediction [29], waste generation prediction [21, 30], load forecast prediction [27, 31], among others. Furthermore, we have applied several AI techniques (besides DN) to SC problems, e.g., DE [20], EA [24], GA [28], k-means [28]. Also, we have contributed to enlarge the scientific corpus of SC and its relation to IT [23]. The following list enumerates our contributions in detail:

DN and SC We introduced multiple alternatives to cope with variable-length solutions on BO [27].

DN and SC We proposed a strategy to improve the performance of the surrogate model of BO for variable-length solutions based on the augmentation of the initial set of solutions, i.e., the *warm-start* [27].

DN and SC Using data analysis techniques, we automatically processed all CS and IT publications indexed by the JCR on SC. Thus, we offered a *broad and objective* review of the topic [23].

DN and SC We presented a *trend analysis* of the distribution and growth of CS and IT research in SC [23].

DN and SC We discussed the applications, techniques, and the relations between research fields. Also, we provided a network of interactions, and pointed out the *hot spots* in the SC literature. Hence, we provided a *single entry point* for SC from a CS and IT point of view [23].

DN and SC We proposed a strategy to automatically partitioning a real-world road map into *tiles*, and to selecting the minimum set of tiles needed to compute the shortest path between two places using DE, that improves the time performance of the shortest path computation [20].

DN and SC We proposed an improvement to the tile map partitioning strategy, which is suitable for dense populated areas, and we introduced a $(\mu + \lambda)$-microSSEA to optimize the partitioning [24].

DN and SC We introduced the *electricity demand signature*, an innovative approach to characterize the behavior of an electricity customer based on the relative importance of the measurements of the daily load curve, which is estimated using a ssGA [28].

DN and SC We compared the electricity demand signature against the *characteristic typical load curve*. The results show that our proposal is capable of managing a large and varied number of customers [28].

DN and SC We defined two DN algorithms (GA and EA-based) to automatically design efficient RNNs to predict the car park occupancy rate [29].

UNIVERSIDAD DE MÁLAGA

DN and SC Using a Self Adapting (1+1)EA-based DN technique, we optimized an RNN that predicts the filling level of the waste containers of a city. We tested our approach on a real use case, and compared our results against the state-of-the-art [21].

DN and SC We studied the effect on the error of the predictions made under uncertainty by an RNN on the waste prediction problem [30].

Thus, we have fulfilled the goal number three: Apply the RNN optimization technique proposed to problem-solving in the context of the SC.

Furthermore, we have put together all major contributions of this thesis in the **DLOPT** [19] library, including the MRS and RESN, and many DN applications to SC problems [21, 29, 30]. Also, we have contributed (i.e., coded) to the MIP-EGO (Bayesian Optimization) software library [27].

DLOPT We introduced DLOPT (`https://github.com/acamero/dlopt`), a software library for DNN optimization based on the DN techniques introduced in this doctoral thesis [19].

Therefore, we also have fulfilled our fourth objective: Disseminate the results by developing (and making publicly available) a software library with the mentioned methods and applications.

In line with these key aspects, we can discuss on our contribution to the application domains. Figure 7.1 depicts the relation between the application domains and the techniques of the studies. Each column encompass all related sub-algorithms, thus, EA includes $(\mu + \lambda)$EA, (1+1)EA, and so on. The use of the MRS is highlighted with a green dot.

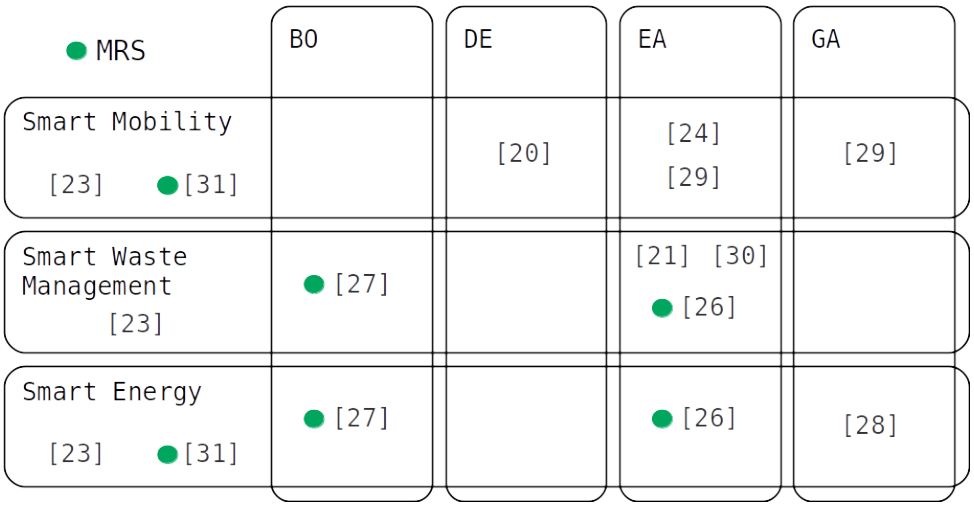


FIGURE 7.1: Studies grouped by application domain and technique

We have contributed extensively to three relevant SC domains, namely Smart Mobility, Smart Waste Management, and Smart Energy, from multiple technical points of view, including BO, DE, EA, and GA. We have contributed to enlarge the scientific corpus, and we have provided new baselines for the problems tackled. Thus, we have translated our technical expertise into real applications. Providing a meaningful output for society.

UNIVERSIDAD DE MÁLAGA

Also, we can highlight some of our major (technical) findings. First, and in our personal opinion the most interesting one, is that our experiments show that there is a moderate-to-strong negative correlation between the MRS and the MAE observed after *training* a network [31]. In plain English, a higher estimated probability estimated by the MRS implies that the studied network is likely to perform well on the given problem. This implies that it is possible to asses the expected performance of an ANN without training it. Second, it is quite remarkable that thanks to the MRS, RESN is capable of cutting by half the time needed to optimize the architecture of an RNN (compared to short training results), without losing error performance [26]. These two major findings (and most of the work done in this thesis) point us out to the question posed earlier: Is it possible to predict the performance of a network (on a give data set) without training it? Now, we state that the answer is yes!

Then, we can discuss on the the impact of this thesis. So far, the *publications* have received more than 130 citations[1], and the DLOPT repository has been forked several times. Furthermore, we have publicly disseminated this thesis in *Doctoral Consortium* CAEPIA 2018 [22], where we obtained the Second Prize for the best student work. Thus, enlarging the outreach of this work. Thus, we concluded that the *community* supports the novelty and usefulness of this doctoral thesis.

However, this is just the beginning. We are aware that there are multiple challenges not yet addressed in this thesis. Ranging from extending the MRS to other DNN types (e.g., CNN), to studying in deep the method proposed (e.g., using a different PDF, weight initialization, among others). Thus, we propose the following future work:

FW1 Extend the MRS to other metrics, i.e., instead of M (from MAE), use a different error metric (e.g., accuracy, MSE, etc.).

FW2 Use a different PDF (other than truncated normal) to fit the sampled errors in the MRS, e.g., Weibull distribution.

FW3 Study the MRS applied to other DNN types (e.g., CNN).

FW4 Use the distribution of the probabilities estimated by the MRS to build a meta-model that reveals insights out of the architecture-performance relation.

FW5 Propose an augmenting/decreasing strategy for BO to cope with variable length solutions search space.

Finally, the most important (personal) conclusion of this thesis is that hard work, discipline, and perseverance are the key to succeed in this *adventure*. As John Milton wrote in his poem *Lost Paradise*, "Long is the way and hard, that out of Hell leads up to light"...

UNIVERSIDAD DE MÁLAGA

[1] https://scholar.google.com/citations?user=D5xGiRAAAAAJ&hl=en [Accessed: 1-Oct-2020]

Bibliography

[1] Martín Abadi et al. "TensorFlow: A System for Large-Scale Machine Learning." In: *OSDI*. Vol. 16. 2016, pp. 265–283.

[2] Charu C Aggarwal and Chandan K Reddy. *Data clustering: algorithms and applications*. CRC press, 2013.

[3] E Alba, JF Aldana, and JM Troya. "Genetic algorithms as heuristics for optimizing ANN design". In: *Artificial Neural Nets and Genetic Algorithms*. Springer. 1993, pp. 683–690.

[4] Enrique Alba, JF Aldana, and José M Troya. "Full automatic ANN design: A genetic approach". In: *International Workshop on Artificial Neural Networks*. Springer. 1993, pp. 399–404.

[5] Enrique Alba and Rafael Martí. *Metaheuristic Procedures for Training Neural Networks*. Vol. 35. Springer Science & Business Media, 2006.

[6] Saleh Albelwi and Ausif Mahmood. "A framework for designing the architectures of deep convolutional neural networks". In: *Entropy* 19.6 (2017), p. 242.

[7] Ronald L Allen and Duncan Mills. *Signal analysis: time, frequency, scale, and structure*. John Wiley & Sons, 2004.

[8] Shun-ichi Amari. "Backpropagation and Stochastic Gradient Descent Method". In: *Neurocomputing* 5.4-5 (1993), pp. 185–196.

[9] Michael A Arbib. *The metaphorical brain 2: Neural networks and beyond*. John Wiley & Sons, Inc., 1989.

[10] Sabeur Aridhi et al. "A MapReduce-based approach for shortest path problem in large-scale networks". In: *Engineering Applications of Artificial Intelligence* 41 (2015), pp. 151–165.

[11] Thomas Back. *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms*. Oxford university press, 1996.

[12] M. Batty et al. "Smart cities of the future". In: *European Physical Journal: Special Topics* 214.1 (2012), pp. 481–518.

[13] G Bebis and M Georgiopoulos. "Feed-Forward Neural Networks: Why Network Size is so Important". In: *IEEE Potentials* (1994), pp. 27–31.

[14] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. "Learning long-term dependencies with gradient descent is difficult". In: *IEEE transactions on neural networks* 5.2 (1994), pp. 157–166.

[15] Yoshua Bengio et al. "Greedy Layer-Wise Training of Deep Networks". In: *Advances in Neural Information Processing Systems* 19.1 (2007), p. 153.

[16] James Bergstra and Yoshua Bengio. "Random search for hyper-parameter optimization". In: *The Journal of Machine Learning Research* 13.1 (2012), pp. 281–305.

UNIVERSIDAD DE MÁLAGA

[17] James S. Bergstra et al. "Algorithms for Hyper-Parameter Optimization". In: *Advances in Neural Information Processing Systems 24*. Ed. by J. Shawe-Taylor et al. Curran Associates, Inc., 2011, pp. 2546–2554.

[18] Ronald Newbold Bracewell and Ronald N Bracewell. *The Fourier transform and its applications*. Vol. 31999. McGraw-Hill New York, 1986.

[19] A. Camero, J. Toutouh, and E. alba. "DLOPT: Deep Learning Optimization Library". In: *arXiv preprint arXiv:1807.03523* (2018).

[20] A. Camero et al. "Tile Map Size Optimization for Real World Routing by Using Differential Evolution". In: *2017 IEEE Congress on Evolutionary Computation, CEC 2017 - Proceedings*. 2017. ISBN: 9781509046010. DOI: 10.1109/CEC.2017.7969478.

[21] A. Camero et al. "Waste Generation Prediction in Smart Cities through Deep Neuroevolution". In: *Congreso Iberoamericano de Ciudades Inteligentes (ICSC-CITIES 2018)*. 2018. DOI: 10.1007/978-3-030-12804-3_15.

[22] Andrés Camero and Enrique Alba. "Neuroevolucion Profunda: Aplicaciones en Ciudades Inteligentes". In: *XVIII Conferencia de la Asociacion Española para la Inteligencia Artificial*. 2018, pp. 1387–1392.

[23] Andrés Camero and Enrique Alba. "Smart City and Information Technology: A Review". In: *cities* 93 (2019), pp. 84–94. DOI: 10.1016/j.cities.2019.04.014.

[24] Andrés Camero, Javier Arellano-Verdejo, and Enrique Alba. "Road Map Partitioning for Routing by Using a Micro Steady State Evolutionary Algorithm". In: *Engineering Applications of Artificial Intelligence* 71 (2018), pp. 155–165. ISSN: 09521976. DOI: 10.1016/j.engappai.2018.02.016.

[25] Andrés Camero, Jamal Toutouh, and Enrique Alba. "Comparing Deep Recurrent Networks Based on the MAE Random Sampling, a First Approach". In: *Advances in Artificial Intelligence, Conference of the Spanish Association for Artificial Intelligence (CAEPIA)*. Ed. by Francisco Herrera et al. Cham: Springer International Publishing, 2018, pp. 24–33. DOI: 10.1007/978-3-030-00374-6_3.

[26] Andrés Camero, Jamal Toutouh, and Enrique Alba. "Random Error Sampling-based Recurrent Neural Network Architecture Optimization". In: *Engineering Applications of Artificial Intelligence* 96 (2020), p. 103946. ISSN: 0952-1976. DOI: 10.1016/j.engappai.2020.103946.

[27] Andrés Camero et al. "Bayesian Neural Architecture Search Using A Training-Free Performance Metric". In: *arXiv preprint arXiv:2001.10726* (2020).

[28] Andrés Camero et al. "Customer Segmentation Based on the Electricity Demand Signature: The Andalusian Case". In: *Energies* 11.7 (2018), p. 1788. DOI: 10.3390/en11071788.

[29] Andrés Camero et al. "Evolutionary Deep Learning for Car Park Occupancy Prediction in Smart Cities". In: *Learning and Intelligent OptimizatioN Conference LION*. 2018. DOI: 10.1007/978-3-030-05348-2_32.

[30] Andrés Camero et al. "Waste Generation Prediction Under Uncertainty in Smart Cities Through Deep Neuroevolution". In: *Revista Facultad de Ingeniería Universidad de Antioquia* 93 (2019), pp. 128–138. DOI: 10.17533/udea.redin.20190736.

UNIVERSIDAD DE MÁLAGA

[31] Andrés Camero, Jamal Toutouh, and Enrique Alba. "Low-Cost Recurrent Neural Network Expected Performance Evaluation". In: *arXiv preprint arXiv:1805.07159* (2018).

[32] Luis M Candanedo, Véronique Feldheim, and Dominique Deramaix. "Data driven prediction models of energy use of appliances in a low-energy house". In: *Energy and buildings* 140 (2017), pp. 81–97.

[33] Bo-Juen Chen, Ming-Wei Chang, et al. "Load forecasting using support vector machines: A study on EUNITE competition 2001". In: *IEEE tran on power systems* 19.4 (2004), pp. 1821–1830.

[34] Yukang Chen et al. "Renas: Reinforced evolutionary neural architecture search". In: *Proceedings of the IEEE Conference on computer vision and pattern recognition*. 2019, pp. 4787–4796.

[35] K Cho et al. "Learning phrase representations using RNN encoder-decoder for statistical machine translatio". In: *arXiv preprint arXiv:1406.1078* (2014).

[36] François Chollet et al. *Keras*. https://keras.io. 2015.

[37] Annalisa Cocchia. "Smart and digital city: A systematic literature review". In: *Smart city*. Springer, 2014, pp. 13–43.

[38] Carlos A Coello Coello Coello and Gregorio Toscano Pulido. "A micro-genetic algorithm for multiobjective optimization". In: *International Conference on Evolutionary Multi-Criterion Optimization*. Ed. by E. Zitzler et al. Springer. 2001, pp. 126–140.

[39] Renata Paola Dameri, Elsa Negre, and Camille Rosenthal-Sabroux. "Triple helix in smart cities: a literature review about the vision of public bodies, universities, and private companies". In: *2016 49th Hawaii Int. Conf. System Sciences (HICSS)*. IEEE. 2016, pp. 2974–2982.

[40] Yann Dauphin et al. "Identifying and Attacking the Saddle Point Problem in High-Dimensional Non-Convex Optimization". In: *Advances in neural information processing systems*. 2014, pp. 2933–2941.

[41] Edsger W Dijkstra. "A note on two problems in connexion with graphs". In: *Numerische mathematik* 1.1 (1959), pp. 269–271.

[42] Carola Doerr. "Non-Static Parameter Choices in Evolutionary Computation". In: *Genetic and Evolutionary Computation Conference, GECCO 2017, Berlin, Germany, July 15-19, 2017, Companion Material Proceedings*. ACM, 2017. DOI: 10.1145/3067695.3067707.

[43] Tobias Domhan, Jost Tobias Springenberg, and Frank Hutter. "Speeding Up Automatic Hyperparameter Optimization of Deep Neural Networks by Extrapolation of Learning Curves". In: *Proceedings of the 24th International Conference on Artificial Intelligence*. IJCAI'15. AAAI Press, 2015, pp. 3460–3468.

[44] Mark Wm Dubin. *How the brain works*. John Wiley & Sons, 2013.

[45] Nees Jan van Eck and Ludo Waltman. "Software survey: VOSviewer, a computer program for bibliometric mapping". In: *Scientometrics* 84.2 (2010), pp. 523–538.

[46] AbdElRahman ElSaid et al. "Evolving Recurrent Neural Networks for Time Series Data Prediction of Coal Plant Parameters". In: *Intl Conf on the Applications of Evolutionary Computation (Part of EvoStar)*. Springer. 2019, pp. 488–503.

UNIVERSIDAD DE MÁLAGA

[47] AbdElRahman ElSaid et al. "Using Ant Colony Optimization to Optimize Long Short-term Memory Recurrent Neural Networks". In: *Proceedings of the Genetic and Evolutionary Computation Conference*. GECCO '18. Kyoto, Japan: ACM, 2018, pp. 13–20. ISBN: 978-1-4503-5618-3. DOI: 10.1145/3205455.3205637. URL: http://doi.acm.org/10.1145/3205455.3205637.

[48] Thomas Elsken, Jan Hendrik Metzen, Frank Hutter, et al. *Neural Architecture Search.* 2019.

[49] Jonathan Engel. "Teaching Feed-Forward Neural Networks by Simulated Annealing". In: *Complex Systems* 2 (1988), pp. 641–648.

[50] Javier Ferrer and Enrique Alba. "BIN-CT: sistema inteligente para la gestión de la recogida de residuos urbanos". In: *International Greencities Congress*. 2018, pp. 117–128.

[51] Javier Ferrer and Enrique Alba. "BIN-CT: Urban Waste Collection based in Predicting the Container Fill Level". In: *arXiv preprint arXiv:1807.01603* (2018).

[52] Kunihiko Fukushima. "Neocognitron: A hierarchical neural network capable of visual pattern recognition". In: *Neural networks* 1.2 (1988), pp. 119–130.

[53] Guojun Gan, Chaoqun Ma, and Jianhong Wu. *Data clustering: theory, algorithms, and applications*. Vol. 20. Siam, 2007.

[54] Frédéric Gruau et al. "Neural Network Synthesis Using Cellular Encoding And The Genetic Algorithm". PhD thesis. 1994.

[55] Mark A. Hall. "Correlation-based feature selection for discrete and numeric class machine learning". In: *Proceedings of the Seventeenth International Conference on Machine Learning*. ICML '00. 2000, pp. 359–366. ISBN: 1-55860-707-2.

[56] Nikolaus Hansen and Andreas Ostermeier. "Adapting Arbitrary Normal Mutation Distributions in Evolution Strategies: The Covariance Matrix Adaptation". In: *Evolutionary Computation, 1996., Proceedings of IEEE International Conference on*. IEEE. 1996, pp. 312–317.

[57] Simon Haykin. *Neural Networks and Learning Machines*. Vol. 3. 2008, p. 906. ISBN: 9780131471399.

[58] M.R. Hestenes and Eduard Stiefel. "Methods of Conjugate Gradients for Solving Linear Systems". In: *Journal of Research of the National Bureau of Standards* 49.6 (1952), p. 409.

[59] GE E Hinton, Simon Osindero, and YW W Teh. "A Fast Learning Algorithm for Deep Belief Nets". In: *Neural Computation* 18.7 (2006), pp. 1527–1554.

[60] Sepp Hochreiter and Jürgen Schmidhuber. "Long short-term memory". In: *Neural computation* 9.8 (1997), pp. 1735–1780.

[61] H Holland John. "Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence". In: *USA: University of Michigan* (1975).

[62] Lawrence Hubert and Phipps Arabie. "Comparing partitions". In: *Journal of classification* 2.1 (1985), pp. 193–218.

[63] Amr M. Ibrahim and Noha H. El-Amary. "Particle Swarm Optimization trained recurrent neural network for voltage instability prediction". In: *Journal of Electrical Systems and Information Technology* 5.2 (2018), pp. 216 –228. ISSN: 2314-7172.

UNIVERSIDAD DE MÁLAGA

[64] Toru Ishida. "Digital city Kyoto". In: *Communications of the ACM* 45.7 (2002), pp. 76–81.

[65] Donald R Jones, Matthias Schonlau, and William J Welch. "Efficient global optimization of expensive black-box functions". In: *Journal of Global optimization* 13.4 (1998), pp. 455–492.

[66] Rafal Jozefowicz, Wojciech Zaremba, and Ilya Sutskever. "An Empirical Exploration of Recurrent Network Architectures". In: *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*. ICML'15. JMLR.org, 2015, pp. 2342–2350.

[67] James Kennedy and Russell C. Eberhart. "Particle Swarm Optimization". In: *Proceedings of the 1995 IEEE International Conference on Neural Networks*. Vol. 4. 1995, pp. 1942–1948.

[68] Diederik P Kingma and Jimmy Ba. "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980* (2014).

[69] Aaron Klein et al. "Learning curve prediction with Bayesian neural networks". In: (2016).

[70] John F. Kolen and Stefan C. Kremer. "Gradient Flow in Recurrent Nets: The Difficulty of Learning LongTerm Dependencies". In: *A Field Guide to Dynamical Recurrent Networks*. Wiley-IEEE Press, 2001, pp. 464–479. ISBN: 9780470544037.

[71] Nicos Komninos. "The architecture of intelligent cities". In: *2nd International Conference on Intelligent Environments* (2006), pp. 13–20.

[72] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "Imagenet classification with deep convolutional neural networks". In: *Advances in neural information processing systems*. 2012, pp. 1097–1105.

[73] Kun Lang et al. "Short-term load forecasting based on multivariate time series prediction and weighted neural network with random weights and kernels". In: *Cluster Computing* (2018), pp. 1–9.

[74] Hugo Larochelle et al. "Exploring strategies for training deep neural networks". In: *Journal of machine learning research* 10.Jan (2009), pp. 1–40.

[75] Yann LeCun. "Une procédure d'apprentissage pour réseau a seuil asymmetrique (a learning scheme for asymmetric threshold networks)". In: *Proceedings of Cognitiva 85, Paris, France*. 1985.

[76] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. "Deep learning". In: *Nature* 521.7553 (2015), pp. 436–444.

[77] Liam Li and Ameet Talwalkar. "Random search and reproducibility for neural architecture search". In: *Uncertainty in Artificial Intelligence*. PMLR. 2020, pp. 367–377.

[78] Jason Liang, Elliot Meyerson, and Risto Miikkulainen. "Evolutionary Architecture Search for Deep Multitask Networks". In: *Proceedings of the Genetic and Evolutionary Computation Conference*. GECCO '18. Kyoto, Japan: ACM, 2018, pp. 466–473. ISBN: 978-1-4503-5618-3. DOI: 10.1145/3205455.3205489. URL: http://doi.acm.org/10.1145/3205455.3205489.

[79] Sai-Ho Ling, Frank HF Leung, and Hak-Keung Lam. "Input-dependent neural network trained by real-coded genetic algorithm and its industrial applications". In: *Soft Computing* 11.11 (2007), pp. 1033–1052.

UNIVERSIDAD DE MÁLAGA

[80] Zachary C Lipton, John Berkowitz, and Charles Elkan. "A critical review of recurrent neural networks for sequence learning". In: *arXiv preprint arXiv:1506.00019* (2015).

[81] Yong Liu and Xin Yao. "Evolutionary design of artificial neural networks with different nodes". In: *Proceedings of IEEE international conference on evolutionary computation*. IEEE. 1996, pp. 670–675.

[82] Teresa B Ludermir, Akio Yamazaki, and Cleber Zanchettin. "An optimization methodology for neural network weights and architectures". In: *IEEE Transactions on Neural Networks* 17.6 (2006), pp. 1452–1459.

[83] Maryam Tayefeh Mahmoudi et al. "Evolving Artificial Neural Network Structure Using Grammar Encoding and Colonial Competitive Algorithm". In: *Neural Computing and Applications* 22.1 (2013), pp. 1–16.

[84] Nilesh Mali and AB Kanwade. "A review on smart city through Internet of things (IOT)". In: *Governance* 2.6 (2016).

[85] Ganesh Mani. "Learning by gradient descent in function space". In: *1990 IEEE International Conference on Systems, Man, and Cybernetics Conference Proceedings*. IEEE. 1990, pp. 242–247.

[86] Catriona Manville et al. "Mapping smart cities in the UE". In: *European Parliament. Directorate-General for Internal Policies. Policy Department: Economic and Scientific Policy A* (2014).

[87] W S McCulloch and W Pitts. "A Logical Calculus of the Idea Immanent in Nervous Activity". In: *Bulletin of Mathematical Biophysics* 5 (1943), pp. 115–133.

[88] Albert Meijer and Manuel Pedro Rodríguez Bolívar. "Governing the smart city: a review of the literature on smart urban governance". In: *International Review of Administrative Sciences* 82.2 (2016), pp. 392–408.

[89] Risto Miikkulainen et al. "Evolving Deep Neural Networks". In: *arXiv preprint arXiv:1703.00548* (2017). eprint: 1703.00548. URL: http://arxiv.org/abs/1703.00548.

[90] Risto Miikkulainen et al. "Evolving deep neural networks". In: *Artificial Intelligence in the Age of Neural Networks and Brain Computing*. Elsevier, 2019, pp. 293–312.

[91] J Močkus. "On Bayesian Methods for Seeking the Extremum". In: *Optimization Techniques IFIP Technical Conference*. Springer. 1975, pp. 400–404.

[92] David J Montana and Lawrence Davis. "Training Feedforward Neural Networks Using Genetic Algorithms". In: *Proceedings of the 11th International Joint Conference on Artificial intelligence - Volume 1* 89 (1989), pp. 762–767.

[93] José Á Morell, Andrés Camero, and Enrique Alba. "JSDoop and TensorFlow.js: Volunteer Distributed Web Browser-Based Neural Network Training". In: *IEEE Access* 7 (2019), pp. 158671–158684. DOI: 10.1109/ACCESS.2019.2950287.

[94] Gregory Morse and Kenneth O Stanley. "Simple Evolutionary Optimization Can Rival Stochastic Gradient Descent in Neural Networks". In: *Proceedings of the 2016 on Genetic and Evolutionary Computation Conference*. ACM. 2016, pp. 477–484.

[95] OECD. *Green growth in cities, OECD green growth studies*. OECD Publishing, 2013. DOI: 10.1787/9789264195325.

UNIVERSIDAD DE MÁLAGA

[96] Varun Kumar Ojha, Ajith Abraham, and Václav Snášel. "Metaheuristic design of feedforward neural networks: A review of two decades of research". In: *Engineering Applications of Artificial Intelligence* 60.January (2017), pp. 97–116.

[97] Varun Kumar Ojha, Ajith Abraham, and Václav Snášel. "Simultaneous optimization of neural network weights and active nodes using metaheuristics". In: *2014 14th International Conference on Hybrid Intelligent Systems*. IEEE. 2014, pp. 248–253.

[98] Alexander Ororbia, AbdElRahman ElSaid, and Travis Desell. "Investigating recurrent neural network memory structures using neuro-evolution". In: *Proceedings of the Genetic and Evolutionary Computation Conference*. ACM. 2019, pp. 446–455.

[99] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. "On the Difficulty of Training Recurrent Neural Networks". In: *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*. ICML 13. Atlanta, GA, USA: JMLR.org, 2013, pp. III–1310–III–1318.

[100] Esther Pearce. "History of the standard industrial classification". In: *Washington, DC Executive Office of the President Office of Statistical Standards, US Bureau of the Budget)(mimeograph)* (1957).

[101] Ernesto Zamora Ramos, Masanori Nakakuni, and Evangelos Yfantis. "Quantitative measures to evaluate neural network weight initialization strategies". In: *2017 IEEE 7th Annual Computing and Communication Workshop and Conference (CCWC)*. 2017, pp. 1–7.

[102] Aditya Rawal and Risto Miikkulainen. "Evolving deep lstm-based memory networks using an information maximization objective". In: *Proceedings of the Genetic and Evolutionary Computation Conference 2016*. ACM. 2016, pp. 501–508.

[103] LM Rere, Mohamad Ivan Fanany, and Aniati Murni Arymurthy. "Metaheuristic algorithms for convolution neural network". In: *Computational intelligence and neuroscience* 2016 (2016).

[104] L.M. Rasdi Rere, Mohamad Ivan Fanany, and Aniati Murni Arymurthy. "Simulated Annealing Algorithm for Deep Learning". In: *Procedia Computer Science* 72 (2015). The Third Information Systems International Conference 2015, pp. 137 –144. ISSN: 1877-0509. DOI: https://doi.org/10.1016/j.procs.2015.12.114. URL: http://www.sciencedirect.com/science/article/pii/S1877050915035759.

[105] Danny Roobaert, Grigoris Karakoulas, and Nitesh V Chawla. "Information gain, correlation and support vector machines". In: *Feature extraction*. Springer, 2006, pp. 463–470.

[106] Gustavo Rosa et al. "Fine-Tuning Convolutional Neural Networks Using Harmony Search". In: *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*. Ed. by Alvaro Pardo and Josef Kittler. Cham: Springer International Publishing, 2015, pp. 683–690.

[107] Allen Rubin. *Statistics for evidence-based practice and evaluation*. Cengage Learning, 2012.

[108] David Rumelhart, Geoffrey E Hinton, and Ronald j. Williams. *Learning Internal Represntations by Error Propagation*. Tech. rep. No. ICS-8506. California Univ San Diego La Jolla Inst for Cognitive Science, 1985.

UNIVERSIDAD DE MÁLAGA

[109] Tom Schmiedlechner et al. "Lipizzaner: A System That Scales Robust Generative Adversarial Network Training". In: *the 32nd Conference on Neural Information Processing Systems (NeurIPS 2018) Workshop on Systems for ML and Open Source Software*. 2018.

[110] Bobak Shahriari et al. "Taking the human out of the loop: A review of Bayesian optimization". In: *Proceedings of the IEEE* 104.1 (2015), pp. 148–175.

[111] David J Sheskin. "Parametric and nonparametric statistical procedures". In: *Boca Raton: CRC* (2000).

[112] Sean C Smithson et al. "Neural networks designing neural networks: multi-objective hyper-parameter optimization". In: *Computer-Aided Design (ICCAD), 2016 IEEE/ACM International Conference on*. IEEE. 2016, pp. 1–8.

[113] Kenneth O Stanley and Risto Miikkulainen. "Evolving neural networks through augmenting topologies". In: *Evolutionary computation* 10.2 (2002), pp. 99–127.

[114] Daniel H Stolfi, Enrique Alba, and Xin Yao. "Predicting Car Park Occupancy Rates in Smart Cities". In: *International Conference on Smart Cities*. Springer. 2017, pp. 107–117.

[115] Rainer Storn and Kenneth Price. "Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces". In: *Journal of global optimization* 11.4 (1997), pp. 341–359.

[116] Felipe Petroski Such et al. "Deep Neuroevolution: Genetic Algorithms Are a Competitive Alternative for Training Deep Neural Networks for Reinforcement Learning". In: *arXiv preprint arXiv:1712.06567* (2017). eprint: 1712.06567. URL: http://arxiv.org/abs/1712.06567.

[117] Gilbert Syswerda. "A study of reproduction in generational and steady state genetic algorithms". In: *Foundations of genetic algorithms* 2 (1991), pp. 94–101.

[118] Jamal Toutouh, Erik Hemberg, and Una-May O'Reilly. "Spatial Evolutionary Generative Adversarial Networks". In: *Proceedings of the Genetic and Evolutionary Computation Conference*. GECCO '19. Association for Computing Machinery, 2019, 472–480. ISBN: 9781450361118. DOI: 10.1145/3321707.3321860. URL: https://doi.org/10.1145/3321707.3321860.

[119] George J Tsekouras, Nikos D Hatziargyriou, and Evangelos N Dialynas. "Two-stage pattern recognition of load curves for classification of electricity customers". In: *IEEE Transactions on Power Systems* 22.3 (2007), pp. 1120–1128.

[120] Andrew M. Ukrainec and Simon Haykin. "A Modular Neural Network for Enhancement of Cross-Polar Radar Targets". In: *Neural Networks* 9.1 (1996), pp. 143–167.

[121] United Nations. *The world's cities in 2016*. http://www.un.org/. Accessed: September 21, 2017. 2016.

[122] Werner Van Geit, Erik De Schutter, and Pablo Achard. "Automated neuron model optimization techniques: a review". In: *Biological cybernetics* 99.4-5 (2008), pp. 241–251.

[123] Jan Vom Brocke et al. "Reconstructing the giant: on the importance of rigour in documenting the literature search process." In: *ECIS*. Vol. 9. 2009, pp. 2206–2217.

[124] Chaoyue Wang et al. "Evolutionary generative adversarial networks". In: *IEEE Transactions on Evolutionary Computation* 23.6 (2019), pp. 921–934.

UNIVERSIDAD DE MÁLAGA

[125] Colin White, Willie Neiswanger, and Yash Savani. "Bananas: Bayesian optimization with neural architectures for neural architecture search". In: *arXiv preprint arXiv:1910.11858* (2019).

[126] D. Whitley, T. Starkweather, and C. Bogart. "Genetic Algorithms and Neural Networks: Optimizing Connections and Connectivity". In: *Parallel Computing* 14.3 (1990), pp. 347–361.

[127] Xin Yao. "A review of evolutionary artificial neural networks". In: *International journal of intelligent systems* 8.4 (1993), pp. 539–567.

[128] Arber Zela et al. "Towards automated deep learning: Efficient joint neural architecture and hyperparameter search". In: *arXiv preprint arXiv:1807.06906* (2018).

[129] You Zhining and Pu Yunming. "The genetic convolutional neural network model based on random sample". In: *International Journal of u-and e-Service, Science and Technology* 8.11 (2015), pp. 317–326.

[130] Zhao Zhong et al. "BlockQNN: Efficient block-wise neural network architecture generation". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2020).

[131] Barret Zoph et al. "Learning transferable architectures for scalable image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 8697–8710.

UNIVERSIDAD DE MÁLAGA

This page intentionally left blank.

UNIVERSIDAD DE MÁLAGA

Appendix A

Problems and Data Sets

In this appendix, we briefly introduce the problems and data sets referred to in this doctoral thesis. All the datasets, except for the *Electricity Demand* (provided by Bettergy), are available on DLOPT GitHub (`https://github.com/acamero/dlopt`).

A.1 Appliance Energy Consumption

The *appliance energy consumption* problem, introduced by Candanedo et al. [32], consists of predicting the next *appliances* and *lights* consumption value using the historical load and the nearby weather of a house in Stambruges, Belgium. The data is recorded every 10 minutes for about 4.5 months (i.e., 19,735 records). Every record includes the date and time, the disaggregated consumption of the appliances and lights, the temperature and humidity of the house (recorded separately for nine rooms), and the weather of the nearest airport weather station (Chievres Airport, Belgium), including the temperature, pressure, humidity, wind speed, visibility, and dew point temperature. Figure A.1 depicts the Min-Max scaled data set distribution.

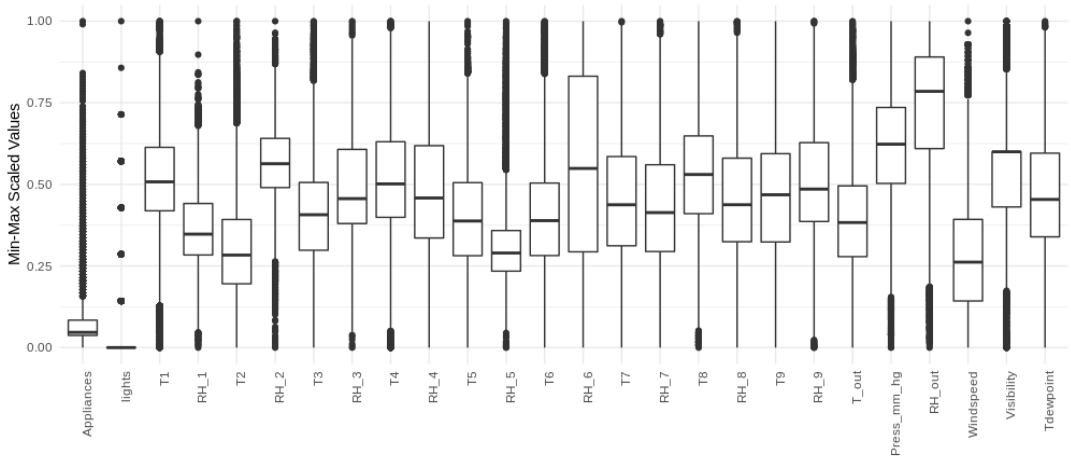


FIGURE A.1: Disaggregated energy consumption and weather data

A.2 Car Parks Occupancy Rate

The *car parks occupancy rate* data set, originally introduce by Stolfi et al. [114], consist of the occupancy rate of 29 car parks operated by the National Car Parks in Birmingham, UK. The occupancy rate of each car park is updated every 30 minutes from 8:00 to 16:30 (i.e., 18 values per car park and day), and the data set comprises the values collected from October 4, 2016, to December 19, 2016. The data set is split into training

UNIVERSIDAD DE MÁLAGA

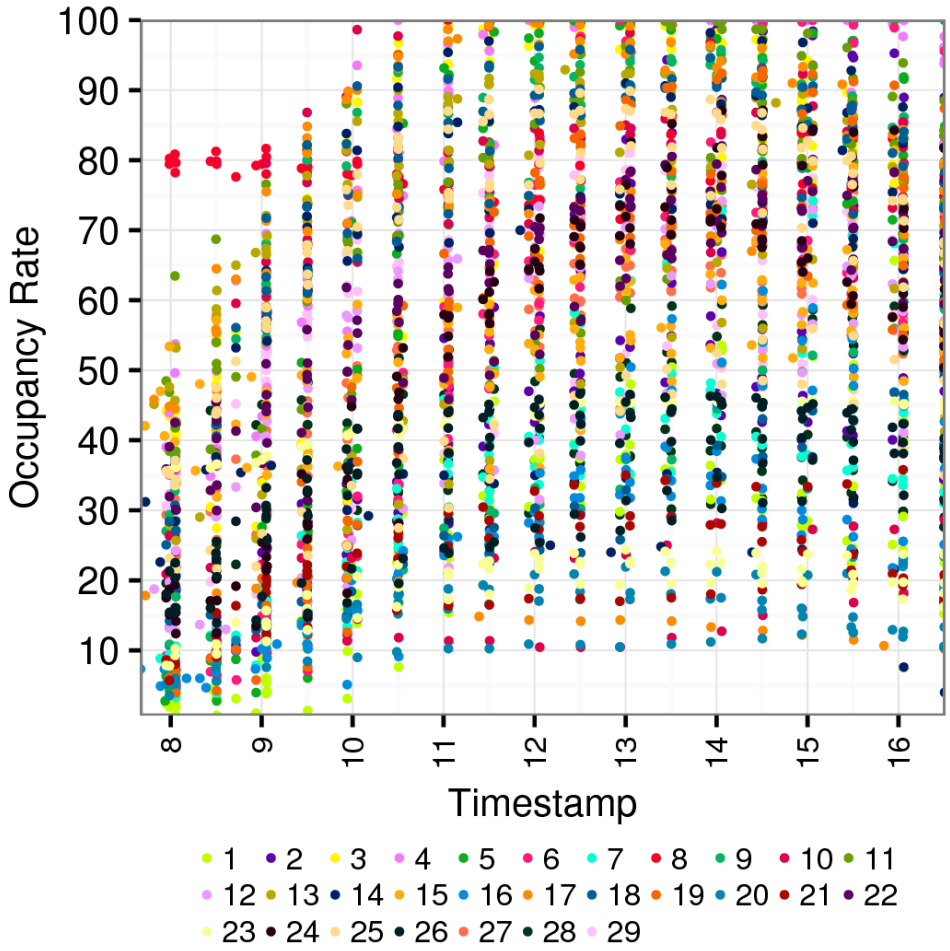


FIGURE A.2: Occupancy rate of 29 car parks in Birmingham

(77 days of occupancy rates per 29 car parks, i.e., 33,292 rates), and testing data (seven days, accounting 3,425 rates). Also, the data set includes the date, time, and week day. The data is licensed under the Open Government License v3.0. The problem is to predict the next occupancy rate of all 29 car parks (at once) given the historical data. Figure A.2 summarizes the occupancy rate of the 29 car parks in Birmingham.

A.3 Coal-Fired Power Plant Flame Intensity

The *coal-fired power plant flame intensity prediction* problem, introduced by Ororbia et al. [98], consists of ten days of values recorded every minute for 12 coal burners. For each coal burner twelve parameters were recorded, including the conditioner inlet temperature, the conditioner outlet temperature, the coal feeder rate, the primary air flow, the primary air split, the system secondary air flow total, the secondary air flow, the secondary air split, the tertiary air split, the total combined air flow, the supplementary fuel flow, and the main flame intensity. The data set has been pre-normalized to the range [0,1]. The problem is to predict the main flame intensity given the historical data.

UNIVERSIDAD DE MÁLAGA

A.4 Electricity Demand

The *electricity demand* data set, provided by BETTERGY, S.L.[1], consists of the load data of 64 buildings located in Andalusia, Spain, recorded every 15 minutes for a year. The data set includes several types of buildings. Table A.1 presents a summary of the customers grouped by their *industrial division* classification, according to the *Standard Industrial Classification* (SIC) [100]. *N* is the number of customers, *MC* is the mean power consumption record (15 minutes), *PC* is the mean of the daily power consumption, and both columns preceded by *Sd* correspond to the standard deviation of the referred value. The power consumption is measured in kilo Watt hour.

TABLE A.1: Electricity demand of 64 buildings in Andalusia, Spain grouped by industria division

Industrial Division	N	MC	PC	Sd MC	Sd PC
Agriculture, Forestry and Fishing	4	46	70	4371	6687
Construction	2	5	2	507	193
Manufacturing	7	140	82	13417	7954
Wholesale Trade	3	70	37	6725	3557
Services					
Hotels, Rooming Houses, Camps, and Other Lodging Services	*1*	*94*	*-*	*9021*	*-*
Business Services	*2*	*16*	*18*	*1582*	*1739*
Amusement and Recreation Services	*2*	*20*	*12*	*1880*	*1141*
Health Services	*11*	*129*	*129*	*12368*	*12351*
Educational Services	*3*	*49*	*62*	*4665*	*5949*
Museums, Art Galleries, and Botanical and Zoological Gardens	*3*	*24*	*39*	*2343*	*3755*
Membership Organizations	*3*	*7*	*5*	*634*	*444*
Engineering, Accounting, Research, Management, and Related Services	*6*	*7*	*7*	*650*	*717*
Miscellaneous Services	*8*	*33*	*74*	*3208*	*7099*
Public Administration	9	41	49	3934	4719
Total	64	61	83	5820	8057

A.5 EUNITE Load Forecast

The *EUNTIE load forecast* problem, originally introduced by EUNITE network in the 2001 competition called "Electricity Load Forecast using Intelligent Adaptive Technologies"[2], consists of a mid-term load forecasting challenge. Particularly, the data set includes the electricity load demand of the Eastern Slovakian Electricity Corporation every half hour, from January 1, 1997, to January 31, 1999. As well as the temperature (daily mean), and the working calendar for the referred period. Then, using the data from 1997 to 1998, the challenge is to predict the maximum daily load of January 1999. The competition was won by Chen et. al [33].

[1] http://www.bettergy.es
[2] http://www.eunite.org/

UNIVERSIDAD DE MÁLAGA

A.6 Sine Wave

A *sine wave* is a mathematical curve that describes a smooth periodic oscillation. Despite its simplicity, by adding sine waves it is possible to approximate any periodic waveform [18]. Equation A.1 expresses a sine wave as a function of time (t), where A is the peak amplitude, f is the frequency, and ϕ is the phase. Then, given the historical data truncated in time (i.e., a limited number of samples), the problem is to predict the next value.

$$y(t) = A \cdot sin(2\pi \cdot f \cdot t + \phi) \tag{A.1}$$

A.7 Waste Generation Prediction

The *waste generation prediction* problem, introduced by Ferrer and Alba [50, 51], consists of the filling level of 217 paper recycling containers from the metropolitan area of a city in Andalusia, Spain, recorded daily for a year. Then, given the historical data, the problem is to predict the filling level of the 217 containers for the next day. It is important to remark that paper containers are more *attractive* than the organic ones, because most paper containers do not need to be collected daily (which is the case of the organic ones in the referred city). Thus, there is room for improvement in the collection planning due to the variability in the filling levels.

UNIVERSIDAD DE MÁLAGA

Appendix B

Summary in Spanish

B.1 Introducción

En las últimas décadas, gracias al rápido aumento de la potencia de cómputo y el almacenamiento de datos, ha habido un resurgimiento del interés en desarrollar redes neuronales artificiales (ANN). Las "viejas" técnicas, combinadas con esta nueva potencia de cálculo han traído nuevas aplicaciones con resultados sorprendentes, que van desde los coches autónomos a los asistentes personales virtuales [59, 72, 76].

En gran medida, este resurgimiento ha sido impulsado por el Aprendizaje Profundo (DL) [76]. En un lenguaje sencillo, el DL consiste en diseñar y entrenar a una ("vieja") ANN de mayor complejidad y tamaño, con una inmensa cantidad de datos ("lo nuevo"). Sin embargo, a medida que estas redes crecen, se vuelven más complejas. Así, esta complejidad trae un nuevo desafío, ¿cómo diseñamos una red neuronal profunda (DNN)?

Tradicionalmente, el diseño de una ANN ha sido tarea de los expertos (humanos). Gracias a su conocimiento del problema y de la técnica, ellos pueden llevar a cabo esta tarea con relativa *facilidad*. No obstante, esta optimización es más bien un trabajo de prueba y error. Luego, conforme la complejidad de un DNN aumenta, esta forma de diseñar se vuelve ineficiente (y poco práctica). La situación empeora aún más a medida que crece la cantidad de datos utilizados para entrenar la red. Por ejemplo, entrenar una sola DNN puede llevar días. Por lo tanto, todo el proceso de optimización puede tomar meses. Podemos decir entonces que es obligatorio mejorar el proceso de diseño [48, 76, 96].

Dada la especial importancia que tiene el diseño de una DNN en su desempeño, incluyendo tanto el tiempo como la dificultad de entrenarla, como su *error* en la tarea asignada, se han formulado numerosas propuestas para automatizar el diseño [48, 96]. Sin embargo, la mayoría de las propuestas se basan (todavía) en el ensayo y error [16, 34, 77, 110, 125, 130, 131]. En otras palabras, se elige un diseño, se entrena la red (generalmente utilizando una técnica basada en el gradiente), y se evalúa el desempeño (error). Si bien este enfoque es capaz de ofrecer buenas soluciones, es muy costoso y, por lo tanto, muy difícil de llevar a la práctica.

Una solución intuitiva para reducir el costo de optimización es limitar el esfuerzo del entrenamiento (e.g., entrenar durante un número acotado de épocas o detener el entrenamiento anticipadamente si se cumple alguna condición). Esto permite reducir significativamente el tiempo. Sin embargo, dada la alta complejidad de los diseños de DNN, es posible que un buen diseño sea descartado de manera temprana porque es *lento para aprender*, es decir, el modelo necesita un tiempo de entrenamiento más largo para lograr su rendimiento óptimo.

Buscando alternativas para evitar este costoso proceso de optimización, algunos autores [25, 89, 116] han propuesto extender una idea originalmente introducida en 1990, la Neuroevolución [127]. La neuroevolución está a mitad de camino entre

UNIVERSIDAD DE MÁLAGA

el aprendizaje de máquina (ML) y la computación evolutiva (EC), y consiste en evolucionar el diseño de una ANN. Luego, la extensión de la técnica original, conocida como Neuroevolución Profunda (DN), consiste en la adaptación de lo "clásico" [127] a los desafíos del DL [76].

Usando metaheurística (e.g., algoritmos genéticos, colonias de hormigas, recocido simulado) es posible navegar a través del espacio de búsqueda (i.e., las DNN diseñadas), encontrando así la red que mejor se ajusta a un problema determinado. Mientras que a primera vista esto también es un enfoque de prueba y error, hay una gran diferencia: la DN es una búsqueda dirigida que explota las características topológicas del problema, y no se basa (necesariamente) en el entrenamiento convencional de ANN (es decir, el aprendizaje basado en el gradiente). Por lo tanto, la DN abre la puerta a nuevos y (esperamos) mejores resultados.

Esta tesis presenta nuestro acercamiento al DN y sus aplicaciones en el dominio de la Ciudad Inteligente (SC). Concretamente, en este apéndice se resume brevemente en español la propuesta y la metodología de trabajo (Sección B.2). La Sección B.3 presentan las principales contribuciones de esta tesis. Finalmente, la Sección B.4 esboza las conclusiones y propone el trabajo futuro.

B.2 Propuesta de Trabajo

En esta tesis doctoral, proponemos abordar el problema del diseño de una DNN a través de la DN. Más concretamente, centramos nuestra atención en las redes neuronales recurrentes (RNN), porque éstas son intrínsecamente las redes más profundas. Así, el objetivo de esta tesis es el el desarrollo de una técnica híbrida de DN, es decir, una que utiliza tanto los métodos *tradicionales* de ML, como las metaheurísticas utilizadas por los acercamientos de la DN. De esta manera, buscamos aprovechar los beneficios de ambas estrategias, para obtener el mejor resultado posible.

En particular, centramos nuestro trabajo en el estudio de las propiedades intrínsecas de las RNNs. Concretamente, buscamos una forma sencilla y de bajo coste computacional que nos permita caracterizar la arquitectura de una red. Esto nos permitirá comparar dos diseños, para así determinar cuál de ellas se adapta mejor a un problema dado. Luego, usando esta información como entrada (es decir, la "heurística"), proponemos definir una metaheurística para la optimización de la arquitectura de una RNN. Finalmente, una vez que hemos encontrado la mejor arquitecture para el problema, proponemos entrenarla con métodos de vanguardia de ML. Es importante señalar que de manera tentativa no limitamos el entrenamiento a ninguna técnica, sino que buscaremos la mejor manera de resolver el problema.

La siguiente lista enumera los principales objetivos de esta tesis doctoral:

G1 Definir un método para caracterizar RNNs que permita la comparación de sus rendimiento esperado sin necesidad de entrenamiento.

G2 Diseñar e implementar una técnica para optimizar una RNN basada en metaheurísticas, y la caracterización antes mencionada.

G3 Aplicar la técnica de optimización de RNNs a la resolución de problemas en el contexto de la SC.

G4 Difundir los resultados mediante el desarrollo (y la puesta a disposición del público) de una biblioteca de software.

UNIVERSIDAD DE MÁLAGA

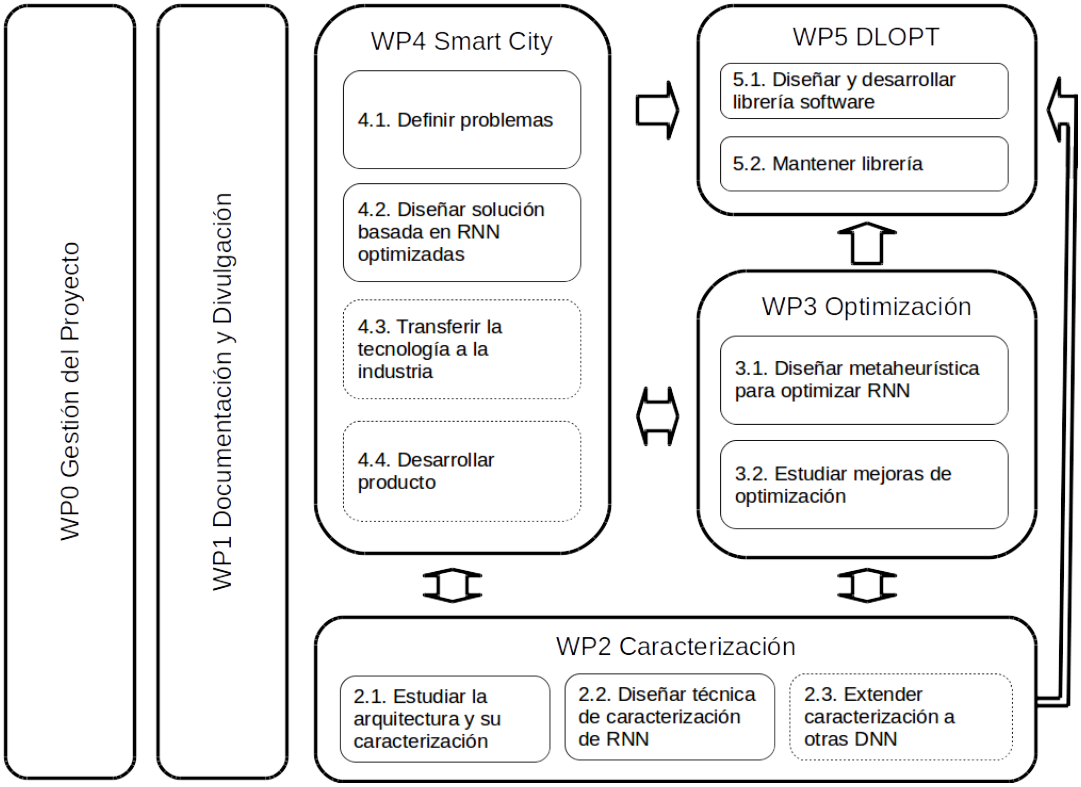


FIGURE B.1: Paquetes de trabajo, actividades principales y relaciones

Cabe señalar que como parte de las directrices de esta tesis, proponemos utilizar la herramienta TensorFlow [1], una biblioteca de código abierto para el aprendizaje automático utilizada (y desarrollada inicialmente) por Google. La idea es, además de reutilizar la implementación de DNN, maximizar el impacto del objetivo G4.

Para cumplir los objetivos propuestos, realizamos un estudio teórico y empírico del diseño óptimo de una RNN. Además de diseñar y construir artefactos de software para implementar los algoritmos definidos, y luego aplicarlos para resolver problemas reales en el dominio de la SC.

El trabajo de tesis fue planificado para su realización en 36 meses. En particular, las actividades se agruparon en 6 paquetes de trabajo (WP), con el fin de cumplir con los objetivos definidos. Los WP del proyecto se describen a continuación:

WP0 *Gestión del Proyecto*: Actividades relacionadas con el control y la planificación de la tesis, así como las actividades burocráticas necesarias para su presentación.

WP1 *Documentación y Divulgación*: Todas las actividades relacionadas con la publicación, la colaboración y la divulgación de los resultados de esta tesis. Además, de la preparación de este compendio.

WP2 *Caracterización*: Actividades conducentes al desarrollo de una caracterización para la arquitectura de RNN (objetivo **G1**).

WP3 *Optimización*: Desarrollo de una técnica para optimizar el diseño de un RNN basado en la metaheurística y la caracterización definida (objetivo **G2**).

WP4 *Ciudad Inteligente*: Agrupación de actividades relacionadas con el estudio de tres problemas reales en el contexto de la SC, y su resolución mediante el uso

UNIVERSIDAD DE MÁLAGA

de una RNN optimizada con la técnica definida en esta tesis (objetivo **G3**). La definición de los problemas a tratar es parte de la misma actividad.

WP5 *DLOPT*: Actividades relacionadas con el desarrollo y el mantenimiento de una biblioteca de software libre que incorpora los resultados (técnicas y datos) de este trabajo (objetivo **G4**).

Los paquetes WP0 y WP1 son funciones de apoyo para la tesis doctoral. Por lo que su ciclo de vida está ligado a la duración de toda la tesis. La Figura B.1 muestra los WPs, sus actividades principales, y sus relaciones. WP2 y WP3 pueden ser considerados como la base técnica de la tesis, mientras que WP4 es la aplicación. Además, es es importante señalar que la salida del WP5 incluirá todas las técnicas y aplicaciones de esta tesis. Las actividades que aparecen con líneas punteadas se consideran deseables para la realización de esta tesis doctoral.

La Figura B.2 muestra el plan de trabajo de alto nivel, incluyendo los hitos más importantes (destacados en naranja).

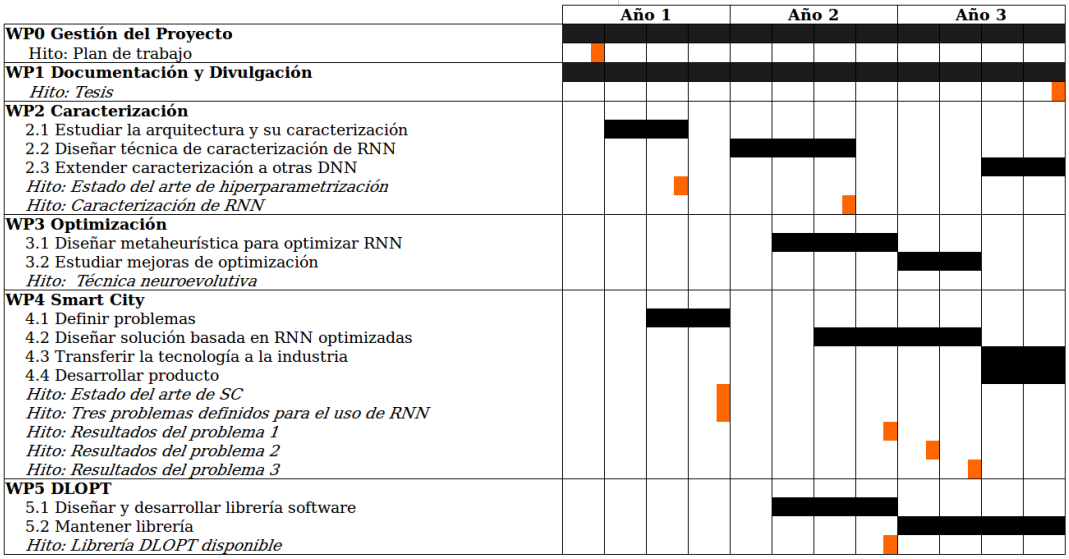


FIGURE B.2: Plan de trabajo. Los hitos se destacan en naranja

B.3 Contribución

Esta tesis se presenta como un compendio de publicaciones. Todas las contribuciones de este trabajo se alinean con los objetivos de esta tesis doctoral (Sección B.2), y persiguen un objetivo común: ampliar el estado del arte. Dada la naturaleza y relevancia de los problemas específicos abordados en esta tesis, esperamos que su contribución tenga un impacto positivo en la comunidad científica y en la sociedad.

En particular, dividimos la contribución de esta tesis en cuatro aspectos clave, cada uno de ellos relacionado directamente con un objetivo:

UNIVERSIDAD DE MÁLAGA

MRS Hemos propuesto la técnica Mean Absolute Error Random Sampling (MRS) para estimar el rendimiento de una RNN (en un problema determinado), la cual se basa en la distribución del error observado en un muestreo aleatorio. Hemos validado empíricamente nuestra propuesta, mostrando que MRS es una estimación fiable y de bajo coste computacional para predecir el rendimiento de una RNN [25, 31].

RESN Hemos introducido una técnica DN basada en MRS. Concretamente, hemos diseñado un algoritmo evolutivo (EA) que explota MRS para optimizar la arquitectura de una RNN. Los resultados de nuestros experimentos muestran que logramos resultados competitivos a la vey que reducimos significativamente el tiempo necesario para realizar optimización [26].

DN and SC : Hemos contribuido a ampliar el corpus científico de DN, SC, y la inteligencia artificial (AI) aplicada a la SC. En particular, en el contexto de la aplicación, hemos propuesto alternativas para abordar problemas de movilidad inteligente [20, 24, 29], electricidad inteligente [27, 28], y gestión de residuos intelgente [21, 27, 30]. Además, hemos revisado el estado del arte de la SC y su relación con la informática [23], y hemos divulgado esta tesis en el marco de la competición Doctoral Consortium [22].

DLOPT Hemos desarrollado la biblioteca de software Deep Learning OPTimization (DLOPT) software library, la cual está disponible bajo la licencia GNU GPL v3 license de código abierto [19]. Es importante recalcar que la mayor parte del trabajo realizado en esta tesis se encuentra disponible ahí, incluyendo los algoritmos, datos y la configuración usada en los experimentos publicados en esta tesis. Con esto, queremos impulsar la reproducción de nuestros estudios.

Durante la realización de esta tesis se han desarollado 14 estudios. De estos, cuatro han sido publicados en revistas indexadas por JCR [23, 24, 26, 28], uno se encuentra en revisión en una revista indexadas por JCR [31], y uno ha sido publicado en una revista indexada por SJR [30]. Además, tres han sido publicados en las actas de conferencias internacionales [20, 21, 29], uno fue premiado con el segundo lugar en Doctoral Consortium CAEPIA 2018 [22], uno se encuentra en un volumen LNAI [25] y dos se encuentra disponibles en arXiv [19, 27]. Por otra parte, según Google Scholar[1], estos trabajos han recibido más de un centenar de citas. Los detalles de estas publicaciones, incluyendo los indicios de calidad de las mismas, se encuentran disponibles en el Apéndice C.

A continuación se presenta un breve resumen de los cuatro aspectos más relevantes de la contribución de esta tesis doctoral.

B.3.1 MRS

A medida que las DNNs se vuelven más complejas y la cantidad de datos aumenta, se hace más necesario contar con DNNs especialmente adaptadas para obtener el mejor desempeño [48, 96]. Sin embargo, optimizar dicho diseño es una tarea muy costosa en términos temporales y de recursos computacionales. Por este motivo, nos

[1] https://scholar.google.de/citations?user=D5xGiRAAAAAJ&hl=en [Accedido: 31-Jul-2020]

UNIVERSIDAD DE MÁLAGA

preguntamos: *Dado un problema, ¿es posible caracterizar el rendimiento de una ANN sin entrenarla?* Y luego, *¿podemos usar dicha caracterización para comparar múltiples ANNs?*

Para responder a estas preguntas, partimos de una premisa bastante simple: Dada una ANN y un problema, es fácil mostrar que la salida de la red cambia cuando los pesos cambian [57]. Entonces, comenzando de esta simple premisa e inspirados en el análisis de señales [7], propusimos caracterizar una ANN mediante "la variación de la salida observada con respecto al esperado (es decir, el error) a medida que los pesos cambian" [31].

A grandes rasgos, MRS predice lo fácil que sería entrenar a una determinada ANN (es decir, la probabilidad de tener un buen rendimiento) mediante el muestreo aleatorio de la salida (en una entrada determinada). Para tomar cada muestra, la ANN se inicializa con un conjunto de pesos (por ejemplo, normalmente distribuidos). Luego, se calcula una métrica de error (por ejemplo, el error medio absoluto, MAE). Más tarde, se ajusta una distribución normal truncada a la distribución de los valores de error medidos, y se estima una probabilidad p_t de encontrar un conjunto de pesos cuyo error está por debajo de un umbral definido por el usuario. Finalmente, p_t se utiliza como un *proxy* (predictor) del rendimiento. Así, podemos utilizar p_t para clasificar (y ordenar) un conjunto de arquitecturas.

La Figura B.3 muestra una vista de alto nivel de MRS. Dado un problema, una arquitectura de RNN y un conjunto de pesos para dicha arquitectura, se mide el error con respecto a la salida esperada. Luego, se ajusta una distribución normal truncada a la densidad de los errores y se estima p_t, nuestro aproximación de lo *fácil* que será obtener un buen desempeño.

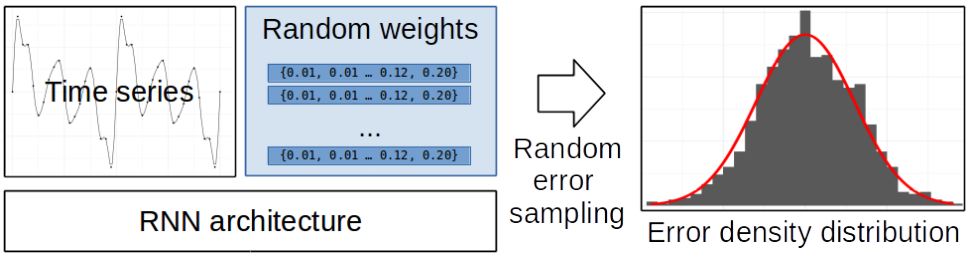


FIGURE B.3: Vista de alto nivel de MRS

El Algoritmo 6, tomado de [31], presenta el pseudo-código de MRS. Dada la arquitectura de una RNN (*ARQ*), un número de pasos en el tiempo o *look back* (*LB*), y un conjunto de datos (*data*), el algoritmo toma *MAX_SAMPLES* muestras aleatorias del error (MAE), usando conjuntos de pesos normalmente distribuidos para inicializar la red cada vez. Luego, se ajusta una distribución normal truncada a los valores de MAE medidos, y se estima p_t para un umbral definido por el usuario *THRESHOLD*.

Para responder a las preguntas planteadas anteriormente, nos centramos en el estudio de RNNs. Elegimos tres problemas de predicción: la onda sinusoidal, la tasa de ocupación de un conjunto de aparcamientos [114], y la previsión de consumo de energía de los aparatos de una casa [32]. En el Apéndice A se presentan los detalles de estos conjuntos de datos. En nuestro primer estudio de MRS [31], definimos dos conjuntos de experimentos, (*i*) un estudio de correlación entre MRS y MAE después de entrenar a las redes mediante un algoritmo basado en gradientes, y (*ii*) un análisis del rendimiento de MRS como mecanismo de comparación de RNNs. En nuestro segundo estudio técnico de la propuesta [25], nos centramos en el estudio de las relaciones entre MRS y los parámetros que definen la arquitectura estudiada, y el uso de recursos (tiempo y computación).

UNIVERSIDAD DE MÁLAGA

Algorithm 6 Pseudo-código de MRS

Given: an architecture (*ARQ*), a number of time steps or look back (*LB*), a user-defined time series (*data*), a number of samples (*MAX_SAMPLES*), and a *THRESHOLD*.
rnn ← InitializeRNN(ARQ, LB)
mae ← ∅
while sample ≤ MAX_SAMPLES **do**
    weights ← GenerateNormalWeights(0, 1)
    UpdateWeights(rnn, weights)
    mae[sample] ← MAE(rnn, data)
    sample + +
end while
mean, sd ← FitTruncatedNormal(mae)
p_t ← PTruncatedNormal(mean, sd, THRESHOLD)

A modo de resumen, la Tabla B.1 muestra el promedio (desde una hasta tres capas ocultas) de los resultados del estudio de correlación [31]. NC representa el número de neuronas, LB el *look back*, Sd la desviación estándar y $p_{0.01}$ es la probabilidad estimada por MRS con un umbral igual a 0.01.

TABLE B.1: Correlación entre MAE post entrenamiento y MRS

Problema	NC	LB	Media	Sd	log $p_{0.01}$
Sine Wave	-0.323	-0.091	-0.572	-0.541	-0.586
Car Parks	0.056	0.734	0.056	-0.440	-0.460
Appliances Energy Consumption	0.132	0.503	0.067	-0.384	-0.404

La Tabla B.2 presenta una comparación del tiempo y la memoria requerida para evaluar una RNN usando MRS (100 muestras) y Adam [68] (1000 épocas). Los resultados muestran que MRS permite evaluar una RNN de forma más eficiente [25].

TABLE B.2: Comparación del tiempo y memoria requerida por MRS y Adam

	Tiempo medio [s]	Sd tiempo	Memoria media [MB]	Sd memoria
Adam	996	0.006	127	6.338
MRS	6	0.001	150	98.264

B.3.2 RESN

Una vez definimos y comprobamos empíricamente la utilidad de MRS para predecir y comparar el desempeño de RNNs, decidimos avanzar en la dirección de la optimización. Particularmente, propusimos RESN, una técnica híbrida basada en $(\mu + \lambda)$EA [11], en MRS [31] y Adam [68], la cual permite optimizar en una primera fase la arquitectura y, en una segunda, los pesos de una RNN (dado un problema).

El Algoritmo 7 presenta una vista de alto nivel de RESN. De un vistazo, la población (*population*) es un grupo de soluciones, donde cada solución representa una arquitectura de RNN. La población inicial se establece de forma aleatoria (*Initialize*). Luego, esta población es evaluada por la función *Evaluate*, la cual calcula p_t (MRS) para cada solución. Una vez que la población inicial ha sido evaluada, comienza el

UNIVERSIDAD DE MÁLAGA

proceso evolutivo, que se divide en: selección (*BinaryTournament*), mutación (*CellMutation* y *LayerMutation*), evaluación (*Evaluate*), reemplazo (*Best*) y auto-ajuste (*SelfAdapting*). Cuando se cumple el criterio de terminación, es decir, que el número de evaluaciones es mayor que el presupuesto (*max_evaluations*), se selecciona y entrena la mejor solución (es decir, la que tiene el p_t más alto) utilizando Adam para un número predefinido de épocas.

Algorithm 7 RESN: Neuroevolución híbrida basada en MRS y Adam

```
population ← Initialize(population_size)
Evaluate(population)
evaluations ← population_size
while evaluations ≤ max_evaluations do
    offspring ← BinaryTournament(population, offspring_size)
    offspring ← CellMutation(offspring, cell_mut_p, max_step)
    offspring ← LayerMutation(offspring, layer_mut_p)
    Evaluate(offspring)
    population ← Best(population + offspring, population_size)
    evaluations ← evaluations + offspring_size
    SelfAdapting(layer_mut_p, max_step, cell_mut_p)
end while
solution ← Best(population, 1)
rnn_trained ← Train(solution, epochs)
return rnn_trained
```

Pusimos a prueba RESN en cuatro problemas: la onda sinusoidal, la predicción de la generación de residuos [51], la predicción de la intensidad de la llama en una planta eléctrica de carbón [98], y el desafío EUNITE para la predicción de la demanda eléctrica [33]. Más detalles sobre los problemas en el Apéndice A.

Concretamente, propusimos cuatro experimentos, E1: RESN vs. Optimización de la Arquitectura basada en el Gradiente, E2: RESN vs. Neuroevolución, E3: Tiempo de Optimización, y E4: RESN vs. Diseño Experto. La Tabla B.3 presenta un resumen de los resultados de los experimentos E1, E2 y E4. Con respecto a E3, nuestros experimentos muestran que RESN acorta el tiempo a la mitad, con respecto a las propuestas basadas en el entrenamiento. Para más detalles sobre las pruebas y los resultados, por favor consulte [26].

TABLE B.3: Resumen de los experimentos E1, E2 y E4 RESN

Experimento	Aproximación	Métrica	Valor
E1	Short Training	Mean MAE	0.073
	RESN	Mean MAE	0.079
E2	EXALT	Mean MSE	0.024
	RESN	Mean MSE	0.005
E4	SVM	Min MAPE	1.950
	BP	Min MAPE	1.451
	RBF	Min MAPE	1.481
	SVR	Min MAPE	1.446
	NNRW	Min MAPE	1.438
	KNNRW	Min MAPE	1.348
	WKNNRW	Min MAPE	1.323
	RESN	Min MAPE	1.370

UNIVERSIDAD DE MÁLAGA

B.3.3 DN and SC

En esta tesis, hemos contribuido a ampliar el estado del arte de AI aplicada a problemas de SC, con especial interés en los métodos de DN. Particular, en el contexto de la aplicación, hemos enfrentado problemas de movilidad inteligente, energía y gestión de residuos. Concretamente, en dos estudios [20, 24] abordamos el problema de optimizar la gestión de los datos (mapa) para minimizar el tiempo de cómputo de la ruta más corta entre dos puntos. Para ello, propusimos una estrategia de particionamiento del mapa, la cual llamamos *Tile Map*, la cual permite minimizar el área del mapa (o subgrafo) utilizada para calcular la ruta más corta. Para optimizar dicho particionamiento propusimos dos métodos, el primero [20] basado en *evolución diferencia* (DE) [115] y el segundo [24] basado en un *algoritmo genético estacionario* (SSGA) [117] y en *microGA* [38]. La Figura B.4 muestra el mapa de la Provincia de Málaga, España, particionado. Nuestros resultados experimentales muestran que el uso del particionamiento reduce el tiempo de cómputo de la ruta más corta entre dos puntos en hasta dos órdenes de magnitud (para mapas de más de 250000 intersecciones).

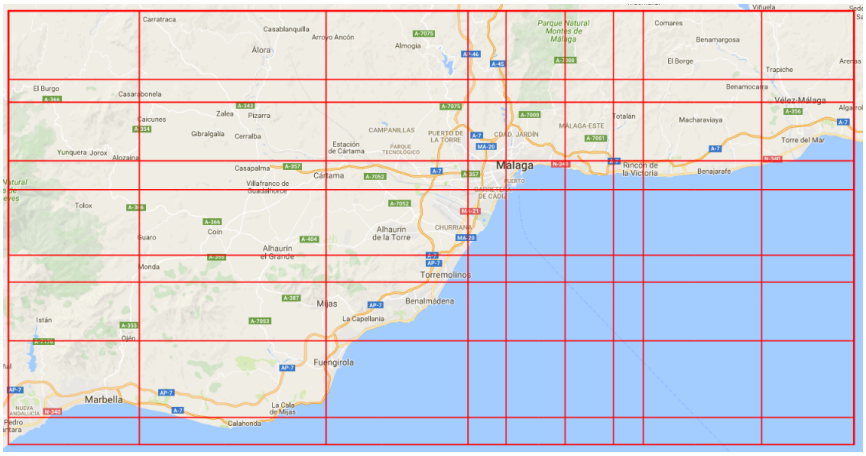


FIGURE B.4: Particionamiento del mapa de la Provincia de Málaga, España

Asimismo, en [29] propusimos utilizar una RNN para predecir la ocupación de 29 edificios de estacionamiento (a la vez) en la ciudad de Birmingham, Inglaterra. Para ajustar la RNN al problema, propusimos dos algoritmos DN, los cuales optimizan la arquitectura, y el número de eventos en el tiempo (*look back*) para truncar los datos al momento de entrenar la red y el *drop out* (para evitar el sobre ajuste). Comparamos nuestros resultados con el estado del arte en predicción y mostramos que una RNN es capaz de igualar el desempeño de 29 predictores individuales. La Tabla B.4 resume los resultados de nuestros experimentos (RNN) y de los competidores [114], donde P corresponde a regresión polinomial, F a series de Fourier, KM a *k*-means clustering, KP a polinomios ajustados a los centroides de *k*-means, SP a cambios de fase aplicados a KP y TS a series temporales. Es importante notar que en [114] se utiliza un predictor para cada edificio, o sea 29 predictores en total.

Con respecto a la energía, en esta tesis hemos estudiado dos problemas. Primero, hemos revisado la caraterización de clientes en base a la demanda de electricidad [28]. Para ellos, hemos propuesto una técnica llamada *electricity demand signature*, la cual recupera desde la curva diaria de demanda los puntos más representativos utilizando un SSGA. Utilizando los datos reales de 64 edificios ubicados en Andalucía, España,

UNIVERSIDAD DE MÁLAGA

TABLE B.4: MAE de la predicción de ocupación

Parking lot	P	F	KM	KP	SP	TS	RNN
Mediana	0.070	0.078	0.087	0.086	0.074	0.069	0.077
Media	0.067	0.079	0.102	0.101	0.073	0.067	0.079
Máx	0.132	0.148	0.177	0.179	0.139	0.129	0.137
Mín	0.016	0.024	0.048	0.049	0.025	0.023	0.033
Sd	0.029	0.033	0.035	0.035	0.033	0.026	0.028

y con la ayuda experta de Bettergy S.L.[2], mostramos la utilidad práctica de nuestra propuesta. La Figura B.5 muestra un año de curvas diarias de demanda eléctrica apiladas (a la izquierda) y el histograma que representa la *electricity demand signature* obtenido de ellas.

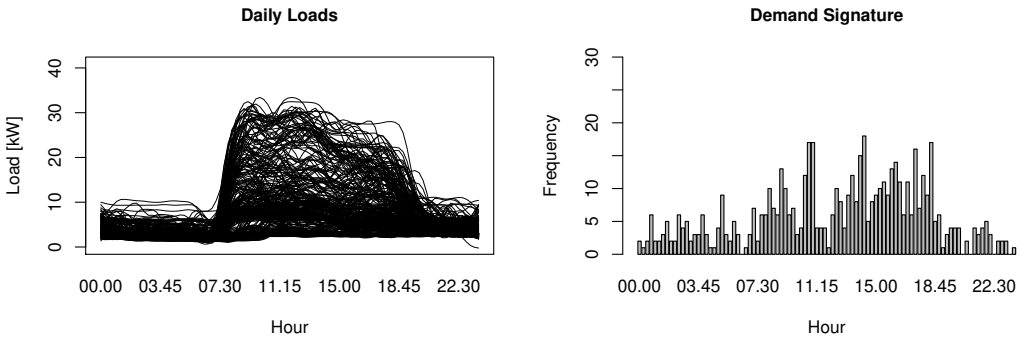


FIGURE B.5: Curva diaria de demanda eléctrica y *electricity demand signature*

Segundo, hemos estudiado la predicción de la demanda eléctrica [27]. Cabe destacar que este trabajo fue realizado en colaboración con el grupo Natural Computing, Universidad de Leiden, dirigido por el Prof. Dr. Thomas Bäck. En él, propusimos un enfoque DN para optimizar la arquitectura de una RNN basado en Optimización Bayesiana (BO) [65, 91] y MRS. Además de contribuir a DN, presenta una serie de alternativas para trabajar con soluciones de largo variable, algo que no es *natural* (i.e., sigue siendo una pregunta abierta) para el estado del arte de BO [110]. La Tabla B.5 resume los resultados obtenidos por nuestro algoritmo (DN) en el problema *EUNITE Load Forecast* para la configuración C–F (más detalles en el trabajo original), y los resultados presentados por Chen et al. [33] (SVM), la propuesta ganadora de la competencia organizada por EUNITE en este problema, y por Lang et al. [73] (RBF y WKNNRW). NA quiere decir que el valor no está disponible.

TABLE B.5: MAPE en la predicción de electricidad EUNITE

	SVM	RBF	WKNNRW	DN
Mediana	2.945	NA	NA	2.099
Media	2.879	NA	NA	2.158
Máx	3.480	NA	NA	3.364
Mín	1.950	1.481	1.323	1.452
Sd	0.004	NA	NA	0.440

[2] https://bettergy.es

UNIVERSIDAD DE MÁLAGA

En lo que respecta a la gestión de residuos inteligente, presentamos dos estudios sobre el uso de RNN para la predicción del nivel de llenado de los contenedores de residuos en una ciudad [21, 30]. En el primero de ellos, propusimos un algoritmo DN híbrido basado en un algoritmo evolutivo (EA) [11] para optimizar la arquitectura y en Adam [68] para optimizar los pesos. Comparamos nuestra propuesta con los resultados publicados en [50, 51], regressión lineal, procesos Gaussianos y máquinas de soporte vectorial para regresión (SMReg), en un caso de estudio real introducido por Ferrer y Alba [50]. Por otra parte, en el segundo estudio de esta serie [30], estudiamos el efecto de la incerteza de los datos, tanto en el caso de no contar con una medida (Zero), como en el caso de que ésta fuese incorrecta (Random), para diferentes proporciones de incertidumbre. La Tabla B.6 resume los resultados de ambos estudios.

TABLE B.6: Error MM de la generación de residuos

Método	Error
RNN	0.028
Procesos Gaussianos	0.038
Regresión lineal	0.074
SMReg	0.095
Random 5%	0.037
Zero 5%	0.029
Random 10%	0.064
Zero 10%	0.030
Random 20%	0.074
Zero 20%	0.042

Además, hemos revisado el estado del arte de la SC y su relación con la informática [23], desvelando desde el análisis de los datos, las tendencias actuales de investigación. Como botón de muestra, la Figura B.6 muestra la evolución de las tendencias de las publicaciones sobre SC clasificadas como tecnologías de la informatición (IT) o ciencias de la computación (CS) en los años 2017 y 2017.

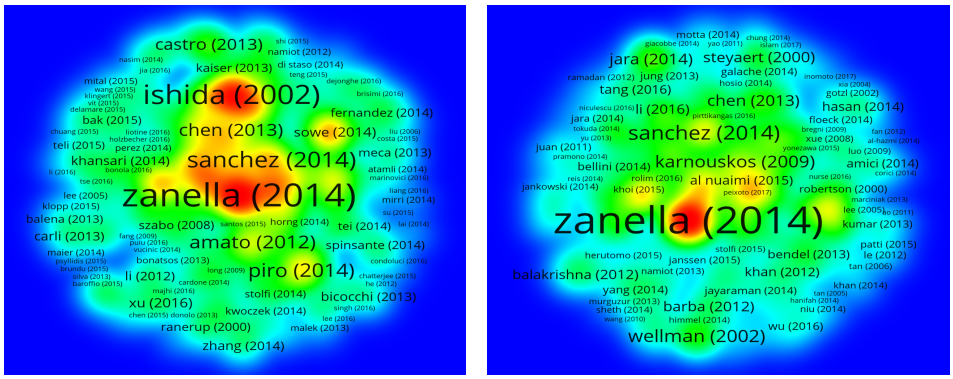


FIGURE B.6: Mapa de densidad de las publicaciones en IT/CS sobre SC

Finalmente, hemos divulgado el contenido de esta tesis en el marco de la competición Doctoral Consortium [22], en la cual hemos obtenido el Segundo Premio al mejor trabajo de doctorandos.

UNIVERSIDAD DE MÁLAGA

B.3.4 DLOPT

Como hemos comprobado a lo largo del trabajo de esta tesis, un *buen* diseño es clave para obtener el mejor resultado de una DNN, no obstante encontrar dicho diseño óptimo es una tarea compleja y que requiere mucho tiempo y recursos computacionales. Por este motivo, decidimos recopilar todos los algoritmos de DN presentados en esta tesis en una biblioteca de software. En este trabajo [19], presentamos DLOPT, nuestra biblioteca de software de código abierto (licencia GNU GPL v3), desarrollada en el lenguaje Python 3, y usando Keras [36] y Tensorflow [1]. El código fuente está disponible en GitHub `https://github.com/acamero/dlopt`.

Es importante recalcar que, además de MRS [25, 31] y RESN [26], en esta biblioteca se encuentran los algoritmos, datos y parámetros utilizados en los estudios [21, 22, 27, 29, 30] incluidos en esta tesis. Así, cualquier interesado puede replicar nuestros estudios, extenderlos y utilizar los métodos propuestos para resolver otros problemas.

B.4 Conclusiones y Trabajo Futuro

Escribir la conclusión de este trabajo no es una tarea fácil. Esta tesis abarca la producción científica de cuatro años, pero más importante aún, reune las lecciones aprendidas durante este tiempo.

Por una parte, podemos discutir los resultados científicos presentados en este compendio. En particular, hemos presentado un plan de trabajo, con objetivos claramente definidos, y los estudios llevados a cabo según dicho plan para cumplir con las metas propuestas. El resultado se materializa en este compendio y las múltiples publicaciones presentadas, incluyendo **cuatro revistas indexadas por JCR** (y una en una segunda fase de revisión), **una revista indexada por SJR**, **tres conferencias internacionales**, **dos conferencias nacionales**, incluyendo un **segundo premio** al mejor trabajo de estudiante en Doctoral Consortium CAEPIA 2018, y dos estudios disponibles en arXiv.

Luego, podemos comentar sobre la contribución de esta tesis y el cumplimiento de los objetivos propuestos. Particularmente, hemos presentado nuestra contribución en cuatro aspectos fundamentales: **MRS**, **RESN**, **DN and SC** y **DLOPT**.

En detalle, hemos introducido **MRS** [25, 31], una técnica que permite predecir el rendimiento de una RNN sin entrenarla. Hemos validado empíricamente nuestra propuesta usando RNNs en varios problemas relacionados con la SC y hemos presentado las siguientes contribuciones intermedias:

MRS Hemos propuesto MRS, una técnica *low-cost* (en términos de tiempo y recursos computacionales) que permite predecir el desempeño de una ANN sin entrenarla [31].

MRS Hemos estudiado la relación entre el número de celdas LSTM, el look back, su distribución (capas ocultas) y MRS [25].

MRS Hemos analizado el uso de recursos computacionales (memoria) y tiempo de MRS [25].

MRS Hemos propuesto un método para optimizar la arquitectura de una RNN basado en MRS y BO [27].

UNIVERSIDAD DE MÁLAGA

Por lo tanto, podemos decir que hemos cumplido nuestro primer objetivo: Definir un método para caracterizar RNNs que permita la comparación de sus rendimiento esperado sin necesidad de entrenamiento.

Además, hemos propuesto **RESN** [31], un enfoque de DN basado en un EA para optimizar una RNN que utiliza MRS. Más concretamente:

RESN Hemos propuesto RESN, un método basado en $(\mu + \lambda)$EA-based para optimizar una RNN que utiliza MRS y Adam [26].

Por lo tanto, cumplimos con nuestro segundo objetivo: Diseñar e implementar una técnica para optimizar una RNN basada en metaheurísticas, y la caracterización antes mencionada.

Además, hemos aplicado nuestras propuestas de DN a los problemas de SC, por ejemplo, la predicción de la tasa de ocupación de un aparcamiento [29], la predicción de la generación de residuos en una ciudad [21, 30], la predicción de la demanda eléctrica [27, 31], entre otros. Además, hemos aplicado varias técnicas de AI (que no son DN) a problemas de la SC, por ejemplo, DE [20], EA [24], GA [28], *k*-means [28]. Asimismo, hemos contribuido a ampliar el corpus científico de la SC y su relación con la informática [23]. En detalle:

DN and SC Introdujimos múltiples alternativas para lidiar con soluciones de largo variable en BO [27].

DN and SC Propusimos una estrategia (*warm-start*) para mejorar el desempeño del modelo surrogado en BO en un escenario de soluciones de largo variable [27].

DN and SC Usando técnicas de análisis de dato, procesamos de manera automática las publicaciones de SC clasificadas como IT por JCR, y presentamos una revisión bibliográfica basada en dicha información [23].

DN and SC Presentamos un análisis de las tendencias en la investigación de SC desde la perspectiva de IT y CS [23].

DN and SC Discutimos las aplicaciones, técnicas y relaciones en la investigación sobre SC. Además, revelamos las redes de interacción, junto con los principales puntos de encuentro (*hot spots*) [23].

DN and SC Propusimos una estrategia para particionar automáticamente mapas de carreteras usando *tiles* y optimizamos el particionamiento usando DE, logrando disminuir el tiempo de cómputo del camino más corto de manera significativa [20].

DN and SC Mejoramos el particionamiento de *tiles*, proponiendo una solución especialmente adaptada para zonas densamente pobladas. Optimizamos el particionamiento con un algoritmo $(\mu + \lambda)$-microSSEA [24].

DN and SC Presentamos la firma de demanda eléctrica (*electricity demand signature*), un acercamiento innovador para caracterizar los hábitos de consumo de los clientes. Esta técnica se basa en la importancia relativa de la demanda diaria y utilizamos un ssGA para construir dicha firma [28].

UNIVERSIDAD DE MÁLAGA

DN and SC Comparamos la firma de demanda eléctrica con la curva típica de consumo (*characteristic typical load curve*), mostrando que nuestro método escala muy bien a grupos grandes y heterogéneos de clientes [28].

DN and SC Definimos dos técnicas de DN (basadas en GA y EA) para optimizar RNN en la predicción de la ocupación de edificios de estacionamiento [29].

DN and SC Usando un algoritmo autoajustable (1+1)EA-based optimizamos una RNN que predice el nivel de llenado de los contenedores de basura de una ciudad [21].

DN and SC Estudiamos el efecto en el error de predicciones realizadas con una RNN en conjuntos de datos con incertidumbre [30].

Así, hemos cumplido con el tercer objetivo: Aplicar la técnica de optimización de RNNs a la resolución de problemas en el contexto de la SC.

También hemos recopilado todas las contribuciones importantes de esta tesis en la biblioteca del DLOPT [19], incluyendo MRS y RESN. Además, hemos contribuido (código) al proyecto MIP-EGO [27] en GitHub.

DLOPT We introduced DLOPT (`https://github.com/acamero/dlopt`), a software library for DNN optimization based on the DN techniques introduced in this doctoral thesis [19].

Por lo tanto, también hemos cumplido con el cuarto objetivo: Difundir los resultados mediante el desarrollo (y la puesta a disposición del público) de una biblioteca de software.

En línea con los aspectos mencionados, podemos analizar nuestra contribución en los diferentes dominios de aplicación y como estos se relacionan con las técnicas empleadas. La Figura B.7 muestra como los estudios relacionan las técnicas con los dominios de aplicación. Cada columna agrupa las técnicas similares, por ejemplo EA incluye $(\mu + \lambda)$EA y (1+1)EA. En verde se resalta el uso de MRS.

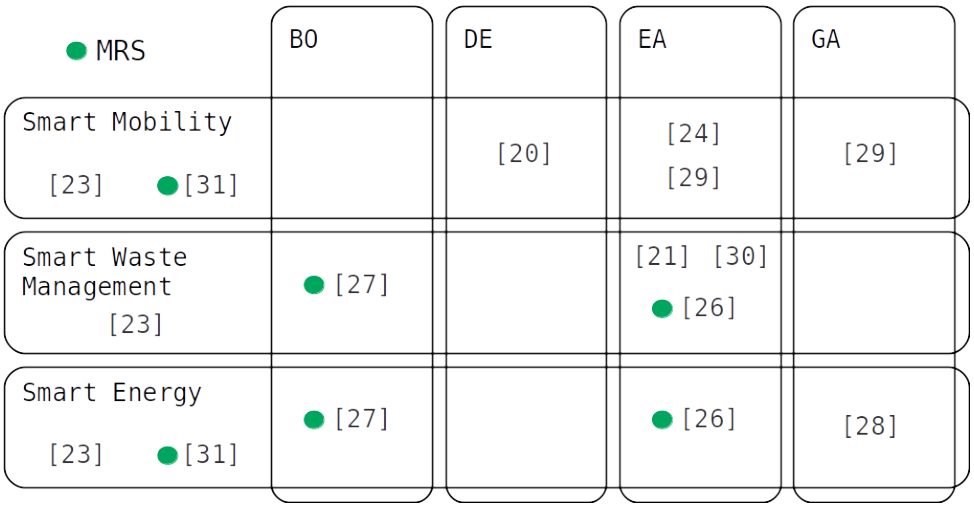


FIGURE B.7: Estudios agrupados por dominio de aplicación y técnica

Podemos concluir entonces que hemos contribuido extensamente a tres dominios de aplicación en el contexto de SC, Smart Mobility, Smart Waste Managemente y Smart Energy, desde múltiples puntos de vista técnicos (BO, DE, EA, GA). Nuestros

UNIVERSIDAD DE MÁLAGA

aportes han contribuido a ampliar la frontera del conocimiento, estableciendo nuevos referentes en algunos casos, y nos han permito trasladar nuestro conocimiento técnico a la realidad. Dando así un mayor sentido a nuestro trabajo, por cuanto éste contribuye al desarrollo de la sociedad.

Cabe destacar también algunos de nuestros resultados (técnicos). Primero, y quizás el más importante, hemos mostrado que existe una correlación moderada-a-fuerte entre la probabilidad estimada por MRS y el error observado luego de entrenar una red neuronal [31]. Esto quiere decir que a medida que aumenta la probabilidad estimada, el error esperado disminuye. Es muy interesante notar que esto permite predecir el desempeño de una NN sin entrenarla. Segundo, gracias al MRS, RESN es capaz de reducir el tiempo requerido para optimizar la arquitectura de una RNN (en contraste con un método *clásico*), sin perjudicar el desempeño [26]. Estos dos hallazgos (junto al resto del trabajo presentado en esta tesis) permiten responder a una pregunta planteada en esta tesis (Capítulo 2), ¿es posible predecir el desempeño de una NN (en un problema dado) sin entrenarla? Ahora podemos responder que sí.

Tambiüen podemos discutir sobre los resultados concretos de este trabajo, es decir, el contenido científico. Hasta ahora, las publicaciones han recibido más de 130 citas[3], y el repositorio de DLOPT se ha bifurcado (*forked*) múltiples veces. Por lo tanto, concluimos que la comunidad apoya la novedad (y utilidad) de MRS y RESN.

Sin embargo, esto es sólo el principio. Somos conscientes de que hay múltiples desafíos que no se han abordado en esta tesis. Que van desde la ampliación de MRS a otros tipos de DNN (por ejemplo, CNN), hasta estudiar en profundidad el método propuesto (por ejemplo, usando un PDF diferente, otra estrategia de inicialización de los pesos, entre otros). Por este motivo planteamos las siguientes líneas de investigación futura:

FW1 Extender MRS a otras métricas, es decir, en reemplazar la M (de MAE) por una función de error diferente (precisión, MSE, etc.).

FW2 Utilizar una PDF diferente a normal truncada para ajustar las muestras en MRS, por ejemplo, la distribución de Weibull.

FW3 Estudiar la aplicación de MRS a otros tipos de DNN types (e.g., CNN).

FW4 Usar la distribución de probabilidades estimada por MRS para construir un meta modelo que permita obtener información sobre la relación entre la arquitectura y el desempeño.

FW5 Proponer una estrategia del estilo aumentar/disminuir el espacio para soportar soluciones de largo variable en BO.

Finalmente, la conclusión (personal) más importante de esta tesis es que el trabajo duro, la disciplina y la perseverancia son la clave para tener éxito en esta aventura. Como John Milton escribió en su poema "El paraíso perdido", "Largo es el camino y difícil, que desde el infierno conduce a a la luz"...

UNIVERSIDAD DE MÁLAGA

[3] https://scholar.google.com/citations?user=D5xGiRAAAAAJ&hl=en [Accessed: 1-Oct-2020]

This page intentionally left blank.

UNIVERSIDAD DE MÁLAGA

Appendix C

Publications of this Thesis

In this Appendix we present the manuscripts delivered during the doctoral research. These studies are the result of the activities planned to achieve the goals of this thesis. First, we introduce the works published in journals that support this thesis. Second, the studies published in the proceeding of international conferences. Third, the works published in other type of venues. Fourth, the works published during the doctoral research time, that do not support this thesis. Finally, we append the original (*as-is*) publications in chronological order.

Along with the publication details, the available metrics (2019) of each publication are presented, i.e., the Journal Impact Factor (JIF), SCImago Journal Rank (SJR) indicator, and H5-index.

C.1 Publications in Journals

1. Andrés Camero, Javier Arellano-Verdejo, and Enrique Alba. "Road Map Partitioning for Routing by Using a Micro Steady State Evolutionary Algorithm". In: *Engineering Applications of Artificial Intelligence* 71 (2018), pp. 155–165. ISSN: 09521976. DOI: 10.1016/j.engappai.2018.02.016
   • JCR COMPUTER SCIENCE, ARTIFICIAL INTELLIGENCE – SCIE, **Q1**, ranking 33/136, 2019 JIF = 4.201

2. Andrés Camero et al. "Customer Segmentation Based on the Electricity Demand Signature: The Andalusian Case". In: *Energies* 11.7 (2018), p. 1788. DOI: 10.3390/en11071788
   • JCR ENERGY & FUELS – SCIE, **Q3**, ranking 63/112, 2019 JIF = 2.702

3. Andrés Camero and Enrique Alba. "Smart City and Information Technology: A Review". In: *cities* 93 (2019), pp. 84–94. DOI: 10.1016/j.cities.2019.04.014
   • JCR URBAN STUDIES, **Q1**, ranking 2/42, 2019 JIF = 4.802, more than **35 citations**

4. Andrés Camero et al. "Waste Generation Prediction Under Uncertainty in Smart Cities Through Deep Neuroevolution". In: *Revista Facultad de Ingeniería Universidad de Antioquia* 93 (2019), pp. 128–138. DOI: 10.17533/udea.redin.20190736
   • SJR Engineering (miscellaneous), **Q3**, 2019 SJR = 0.14

5. Andrés Camero, Jamal Toutouh, and Enrique Alba. "Random Error Sampling-based Recurrent Neural Network Architecture Optimization". In: *Engineering Applications of Artificial Intelligence* 96 (2020), p. 103946. ISSN: 0952-1976. DOI: 10.1016/j.engappai.2020.103946
   • JCR COMPUTER SCIENCE, ARTIFICIAL INTELLIGENCE – SCIE, **Q1**, ranking 33/136, 2019 JIF = 4.201

UNIVERSIDAD DE MÁLAGA

6. Andrés Camero, Jamal Toutouh, and Enrique Alba. "Low-Cost Recurrent Neural Network Expected Performance Evaluation". In: *arXiv preprint arXiv:1805.07159* (2018)
   • Second round revision in Machine Learning, JCR COMPUTER SCIENCE, ARTIFICIAL INTELLIGENCE – SCIE, **Q2**, ranking 57/136, 2019 JIF = 2.672

C.2 Publications in the Proceedings of International Conferences

7. Andrés Camero et al. "Evolutionary Deep Learning for Car Park Occupancy Prediction in Smart Cities". In: *Learning and Intelligent OptimizatioN Conference LION*. 2018. DOI: 10.1007/978-3-030-05348-2_32
   • H5-index = 12, and it has been **cited more than 40 times**

8. A. Camero et al. "Waste Generation Prediction in Smart Cities through Deep Neuroevolution". In: *Congreso Iberoamericano de Ciudades Inteligentes (ICSC-CITIES 2018)*. 2018. DOI: 10.1007/978-3-030-12804-3_15
   • First Ibero-American Congress on Information Management and Big Data

9. A. Camero et al. "Tile Map Size Optimization for Real World Routing by Using Differential Evolution". In: *2017 IEEE Congress on Evolutionary Computation, CEC 2017 - Proceedings*. 2017. ISBN: 9781509046010. DOI: 10.1109/CEC.2017.7969478
   • H5-index = 68

C.3 Other Publications

10. Andrés Camero et al. "Bayesian Neural Architecture Search Using A Training-Free Performance Metric". In: *arXiv preprint arXiv:2001.10726* (2020)
    • Under review

11. Andrés Camero, Jamal Toutouh, and Enrique Alba. "Comparing Deep Recurrent Networks Based on the MAE Random Sampling, a First Approach". In: *Advances in Artificial Intelligence, Conference of the Spanish Association for Artificial Intelligence (CAEPIA)*. ed. by Francisco Herrera et al. Cham: Springer International Publishing, 2018, pp. 24–33. DOI: 10.1007/978-3-030-00374-6_3
    • Lecture Notes in Artificial Intelligence, 18th Conference of the Spanish Association for Artificial Intelligence

12. Andrés Camero and Enrique Alba. "Neuroevolucion Profunda: Aplicaciones en Ciudades Inteligentes". In: *XVIII Conferencia de la Asociacion Española para la Inteligencia Artificial*. 2018, pp. 1387–1392 Conference of the Spanish Association for Artificial Intelligence
    • **Second Prize** for the best student work in *Doctoral Consortium CAEPIA 2018*

13. A. Camero, J. Toutouh, and E. alba. "DLOPT: Deep Learning Optimization Library". In: *arXiv preprint arXiv:1807.03523* (2018)
    • Software library available on GitHub (https://github.com/acamero/dlopt)

UNIVERSIDAD DE MÁLAGA

C.4 Publications not Used to Support this Thesis

It is worth noticing that during the development of this research, we contributed to the publication of the following study. Nonetheless, we have decided to let it out of the scope of this compendium.

14. José Á Morell, Andrés Camero, and Enrique Alba. "JSDoop and TensorFlow. js: Volunteer Distributed Web Browser-Based Neural Network Training". In: *IEEE Access* 7 (2019), pp. 158671–158684. DOI: 10.1109/ACCESS.2019.2950287
    • JCR COMPUTER SCIENCE, INFORMATION SYSTEMS – SCIE, Q1, ranking 35/156, 2019 JIF = 3.745

UNIVERSIDAD DE MÁLAGA