

IntechOpen

Recent Trends in
Artificial Neural Networks
from Training to Prediction

*Edited by Ali Sadollah
and Carlos M. Travieso-Gonzalez*



Recent Trends in Artificial Neural Networks - from Training to Prediction

*Edited by Ali Sadollah
and Carlos M. Travieso-Gonzalez*

Published in London, United Kingdom



IntechOpen





Supporting open minds since 2005



Recent Trends in Artificial Neural Networks – from Training to Prediction

<http://dx.doi.org/10.5772/intechopen.77409>

Edited by Ali Sadollah and Carlos M. Travieso-Gonzalez

Contributors

Silvia Lopes de Sena Taglialenha, Rubén Augusto Romero Lázaro, Soumya Ghosh, Mrinmoy Majumder, Mohd Arfian Ismail, Vitaliy Mezhuyev, Mohd Saberi Mohamad, Shahreen Kasim, Ashraf Osman Ibrahim, Octavian Dumitru, Gottfried Schwarz, Mihai Datcu, Meisam Gordan, Zubaidah Ismail, Zainah Ibrahim, Hashim Huzaifa, Helton Maia Peixoto, Rafael Marrocos Magalhães, Richardson Menezes, Paola Sánchez-Sánchez, José Rafael García-González, Leidy Perez Coronell, Miguel Melgarejo, Diego Mayorga, Nelson Obregón, Cristian Rodriguez

© The Editor(s) and the Author(s) 2020

The rights of the editor(s) and the author(s) have been asserted in accordance with the Copyright, Designs and Patents Act 1988. All rights to the book as a whole are reserved by INTECHOPEN LIMITED. The book as a whole (compilation) cannot be reproduced, distributed or used for commercial or non-commercial purposes without INTECHOPEN LIMITED's written permission. Enquiries concerning the use of the book should be directed to INTECHOPEN LIMITED rights and permissions department (permissions@intechopen.com).

Violations are liable to prosecution under the governing Copyright Law.



Individual chapters of this publication are distributed under the terms of the Creative Commons Attribution 3.0 Unported License which permits commercial use, distribution and reproduction of the individual chapters, provided the original author(s) and source publication are appropriately acknowledged. If so indicated, certain images may not be included under the Creative Commons license. In such cases users will need to obtain permission from the license holder to reproduce the material. More details and guidelines concerning content reuse and adaptation can be found at <http://www.intechopen.com/copyright-policy.html>.

Notice

Statements and opinions expressed in the chapters are these of the individual contributors and not necessarily those of the editors or publisher. No responsibility is accepted for the accuracy of information contained in the published chapters. The publisher assumes no responsibility for any damage or injury to persons or property arising out of the use of any materials, instructions, methods or ideas contained in the book.

First published in London, United Kingdom, 2020 by IntechOpen

IntechOpen is the global imprint of INTECHOPEN LIMITED, registered in England and Wales, registration number: 11086078, 7th floor, 10 Lower Thames Street, London, EC3R 6AF, United Kingdom

Printed in Croatia

British Library Cataloguing-in-Publication Data

A catalogue record for this book is available from the British Library

Additional hard and PDF copies can be obtained from orders@intechopen.com

Recent Trends in Artificial Neural Networks – from Training to Prediction

Edited by Ali Sadollah and Carlos M. Travieso-Gonzalez

p. cm.

Print ISBN 978-1-78985-419-0

Online ISBN 978-1-78985-420-6

eBook (PDF) ISBN 978-1-78985-859-4

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

4,600+

Open access books available

120,000+

International authors and editors

135M+

Downloads

151

Countries delivered to

Our authors are among the
Top 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Meet the editors



Ali Sadollah received his BS degree in mechanical engineering, solid states, from Azad University, Semnan Branch, Iran in 2007 and MS degree in mechanical engineering, applied mechanics, from the University of Semnan, Semnan, Iran in 2010. He obtained his Ph.D. at the Faculty of Engineering, the University of Malaya, Kuala Lumpur, Malaysia in 2013. Also, he has served as a postdoctoral research fellow for more than 2 years at Korea University, Seoul, South Korea. In 2016 for a year, he has served as a research staff at Nanyang Technological University, Singapore. Afterward, he was a guest assistant professor at Sharif University of Technology, Tehran, Iran for 2 years. Currently, he is an assistant professor at the department of mechanical engineering at University of Science and Culture, Tehran, Iran. His research interests include algorithm development, optimization and metaheuristics, applications of soft computing methods in engineering, artificial neural networks, and computational solid mechanics.



Carlos M. Travieso-González received his MSc degree in Telecommunication Engineering at the Polytechnic University of Catalonia, Spain, in 1997, and his PhD degree at the University of Las Palmas de Gran Canaria (ULPGC-Spain) in 2002. He is Full Professor of Signal Processing and Pattern Recognition and Head of the Signals and Communications Department at ULPGC, teaching signal processing and learning theory since 2001. His research includes biometrics, biomedical signals and images, data mining, classification systems, signal and image processing, machine learning, and environmental intelligence. He has researched in 51 international and Spanish research projects, some of them as head researcher. He is co-author of four books, co-editor of 24 proceedings books, guest editor for eight JCR-ISI international journals, and author of up to 24 book chapters. He has over 440 papers published in international journals and conferences (72 of them indexed on JCR-ISI-Web of Science). He has published seven patents in the Spanish Patent and Trademark Office. He has been supervisor of eight PhD theses (12 more are under supervision) and 130 masters theses. He is founder of the IEEE IWOBI conference series and president of its Steering Committee, the InnoEducaTIC conference series, and the APPIS conference series. He is an evaluator of project proposals for the European Union (H2020), Medical Research Council (MRC-UK), Spanish government (ANECA-Spain), Research National Agency (ANR-France), DAAD (Germany), Argentinian government, and Colombian institutions. He has been a reviewer in different indexed international journals (70) and conferences (220) since 2001. He has been a member of the IASTED Technical Committee on Image Processing since 2007 and a member of the IASTED Technical Committee on Artificial Intelligence and Expert Systems since 2011. He will be ACM-APPIS 2021 General Chair and IEEE-IWOBI 2020 and 2020, and was ACM-APPIS 2020 General Chair, IEEE-IWOBI 2019 General Chair, APPIS 2019 General Chair, IEEE-IWOBI 2018

General Chair, APPIS 2018 General Chair, InnoEducaTIC 2017 General Chair, IEEE-IWOBI 2017 General Chair, IEEE-IWOBI 2015 General Chair, InnoEducaTIC 2014 General Chair, IEEE-IWOBI 2014 General Chair, IEEE-INES 2013 General Chair, NoLISP 2011 General Chair, JRBP 2012 General Chair, and IEEE-ICCST 2005 Co-Chair. He is associate editor of the *Computational Intelligence and Neuroscience* journal (Hindawi—Q2 JCR-ISI). He was Vice-Dean of the Higher Technical School of Telecommunication Engineers in ULPGC from 2004 to 2010, and Vice-Dean of Graduate and Postgraduate Studies from March 2013 to November 2017. He has won “Catedra Telefonica” Awards in Modality of Knowledge Transfer in the 2017, 2018, and 2019 editions.

Contents

Preface	XIII
Section 1	
Time Series and Artificial Neural Networks	1
Chapter 1	3
Time Series from Clustering: An Approach to Forecast Crime Patterns <i>by Miguel Melgarejo, Cristian Rodriguez, Diego Mayorga and Nelson Obregón</i>	
Chapter 2	23
Encountered Problems of Time Series with Neural Networks: Models and Architectures <i>by Paola Andrea Sánchez-Sánchez, José Rafael García-González and Leidy Haidy Perez Coronell</i>	
Section 2	
Metaheuristics and Artificial Neural Networks	41
Chapter 3	43
Electric Transmission Network Expansion Planning with the Metaheuristic Variable Neighbourhood Search <i>by Silvia Lopes de Sena Tagliapietra and Rubén Augusto Romero Lázaro</i>	
Chapter 4	61
An Improved Algorithm for Optimising the Production of Biochemical Systems <i>by Mohd Arfian Ismail, Vitaliy Mezhuyev, Mohd Saberi Mohamad, Shahreen Kasim and Ashraf Osman Ibrahim</i>	
Section 3	
Applications of Artificial Neural Networks	79
Chapter 5	81
Object Recognition Using Convolutional Neural Networks <i>by Richardson Santiago Teles de Menezes, Rafael Marrocos Magalhaes and Helton Maia</i>	

Chapter 6	93
Prediction of Wave Energy Potential in India: A Fuzzy-ANN Approach <i>by Soumya Ghosh and Mrinmoy Majumder</i>	
Chapter 7	105
Deep Learning Training and Benchmarks for Earth Observation Images: Data Sets, Features, and Procedures <i>by Mihai Datcu, Gottfried Schwarz and Corneliu Octavian Dumitru</i>	
Chapter 8	117
Data Mining Technology for Structural Control Systems: Concept, Development, and Comparison <i>by Meisam Gordan, Zubaidah Ismail, Zainah Ibrahim and Huzaifa Hashim</i>	

Preface

In recent years, applications and utilization of artificial neural networks (ANNs) have increased drastically. Applications range from prediction of wave energy potential, object recognition, biochemical systems, earth observation images, structural control systems, forecasting crime patterns, time series problems, and training electric transmission network.

In this book, ANNs and applicable nature of ANNs and its importance are highlighted by explaining different aspects of ANNs. This book starts with time series problems, describing models and an application for clustering, and then proceeds with the usage of metaheuristic optimization methods for reducing error level a given network. The book ends with some real-life and practical applications of ANNs to various fields of studies from energy saving to structural control systems.

I would like to thank all the authors and scholars for their precious contributions to this book. Special thanks go to Intech Open and Author Service Manager Mr. Gordan Tot for their support and patience.

Ali Sadollah, PhD

Department of Mechanical Engineering,
University of Science and Culture,
Tehran, Iran

Carlos M. Travieso-Gonzalez

University of Las Palmas de Gran Canaria,
Spain

Section 1

Time Series and Artificial Neural Networks

Time Series from Clustering: An Approach to Forecast Crime Patterns

*Miguel Melgarejo, Cristian Rodriguez, Diego Mayorga
and Nelson Obregón*

Abstract

This chapter presents an approach to forecast criminal patterns that combines the time series from clustering method with a computational intelligence-based prediction. In this approach, clusters of criminal events are parametrized according to simple geometric prototypes. Cluster dynamics are captured as a set of time series. The size of this set corresponds to the number of clusters multiplied by the number of parameters per cluster. One of the main drawbacks of clustering is the difficulty of defining the optimal number of clusters. The paper also deals with this problem by introducing a validation index of dynamic partitions of crime events that relates the optimal number of clusters with the foreseeability of time series by means of non-linear analysis. The method as well as the validation index was tested over two cases of reported urban crime. Our results showed that crime clusters can be predicted by forecasting their representative time series using an evolutionary adaptive neural fuzzy inference system. Thus, we argue that the foreseeability of these series can be anticipated satisfactorily by means of the proposed index.

Keywords: crime, crime pattern theory, Fuzzy clustering, neuro-fuzzy systems, evolutionary computation

1. Introduction

A crime is an event that emerges from opportunities configured by the interaction of offenders, victims, and the surrounding environment [1]. The process behind a crime event has a decisional nature which evaluates the benefits and risks for the offender [2]. Because of the social value in understanding and preventing crime, it has been studied from different perspectives. It has been noted that crime is a complex phenomenon [3] since criminal activity is connected to the complex dimension of social systems and their actors [4].

Environmental criminology recognizes that crime is not uniformly distributed over space, time, or society [5]. Finding rules that explain the non-randomness of crime dynamics has become an intense research area. Typically, stochastic process analysis has been used in the study of crime dynamics [6–8]. However, other epistemological approaches, like fuzzy set theory [9], non-classical topology [10], and complex network theory [11] have been also applied to describe this phenomenon.

Crime pattern theory points out that crime forms patterns in space, time, and society [5, 10]. A pattern is the interconnection between objects, rules, or processes. This interconnection can be either physical or conceptual. This theory deals with the problem of forecasting when and where a criminal event will occur. The observation of patterns can come from evidence or theoretical considerations. Therefore, the analysis of criminal patterns can be described in terms of agents, rules, or clusters of events taking as a reference the structure of the urban form [12].

Crime is ubiquitous in modern cities [13]. It has been observed that there are urban areas with high crime concentration [8]. Therefore, the term space-time dynamics of crime indicates how crime patterns evolve in time and space. A recent work [11] approaches the analysis of crime dynamics by suggesting that robberies are geographically correlated with urban form. A previous work addressed a similar perspective by means of fuzzy topology [10]. In this case, it was observed that crime patterns correlate to the fuzzy edges of neighborhoods, disperse into vulnerable neighborhoods, and concentrate on some main roads.

Some techniques for non-hierarchical clustering have been employed to detect spatial patterns of criminal activity [14], including basic fuzzy clustering (i.e., Fuzzy C-means algorithm) [9]. However, the problems of criminal directionality and crime dynamics using this perspective have been recently addressed in a work by Mayorga et al. [15]. Fuzzy clustering algorithms for spatio-temporal data have been introduced recently. However, these algorithms are mainly focused on clustering of time series (CTS). An enhanced version of the Fuzzy C-means algorithm was proposed to consider both spatial and temporal components of data [16]. This algorithm deals with the clustering of time series produced by spatial sources (i.e., sensors, monitoring stations, etc.). The method is well adjusted to cluster time series that come from a structured sampling of spatial variables. However, when analyzing criminal events, time series provided by sensors are not available, only discrete points of criminal activity in space and time. Thus, the algorithm is not suitable for analyzing this kind of data. Another spatio-temporal fuzzy clustering algorithm was reported in a study by Ji et al. [17]. Clusters are assigned over time series by introducing a switching function that establishes the correspondence between a section of a time series and a cluster in the partition. This algorithm also assumes that time series are available, which is not how spatio-temporal criminal data are collected and studied.

This work deals with clustering the dynamics of criminal events and to forecast it. It contributes by introducing a method that: (1) uses a clustering reorganization algorithm that tracks the dynamics of crime clusters by producing time series of their geometric parameters, (2) sets the optimal number of clusters by minimizing a fuzzy partition index that quantifies the predictability of time series, and (3) forecasts the time series by means of evolutionary-fuzzy predictors. Conceptually, the main contribution of this work is focused on strengthening the time series from clustering (TSC) method in forecasting crime patterns and connecting the quality of a dynamic partition of crime events with concepts of non-linear analysis.

The method is applied over two independent study cases: (1) house burglary in San Francisco, USA and (2) cellular phone robbery in Bogota DC, Colombia. A comparison between the best and worst situations predicted by the introduced index is also provided in each case, giving a preliminary validation. Results reveal that our approach is promising in terms of prediction capabilities, which motivates its application to forecast spatial crime patterns over fine temporal scales. Hence, this method may be considered as a working tool in the practice of predictive policing [18].

The paper is organized as follows: Section 2 describes our method in detail, reviews our clustering organization algorithm, introduces the dynamic fuzzy

partition quality index, and describes the forecasting approach. Section 3 presents our results for the two study cases. Section 4 discusses our main findings and presents some conclusions.

2. Method

The overall TSC method is depicted in **Figure 1**. Reported spatial events are organized and filtered in frames according to a time scale (i.e., daily, weekly, etc.). A clustering algorithm is applied over these data frames in order to detect and track spatial patterns. These clusters are synthesized in terms of relevant spatial variables computed from each data frame, resulting in several time series. This process is repeated until the maximum number of clusters is reached. Non-linear signal analysis is used to compute a foreseeability index for all time series. The optimal number of clusters is selected as the one that minimizes this index. Selected time series are characterized in order to perform their forecasting. These stages are described in detail as follows.

2.1 Data organization

Reported criminal events are grouped in time frames. These events must have a specified date, and longitude and latitude of occurrence to be organized. Time can be defined daily, weekly, monthly, etc., depending on the number of events in the database. Time frames can be generated independently (i.e., without sharing any events) or with some amount of overlapping (i.e., sharing some events).

2.2 Clustering reorganization algorithm

Following data organization, clustering of criminal events in a given number of clusters is performed. The result is a set of time series that represents the spatio-temporal dynamics of crime. The number of time series is equivalent to the predefined number of clusters C multiplied by the number of parameters per cluster. It is important to ensure that these resulting time series have spatial and temporal structure. The clustering reorganization algorithm (CRA) was proposed in [15]. This algorithm ensures that the order of the identified clusters remains the

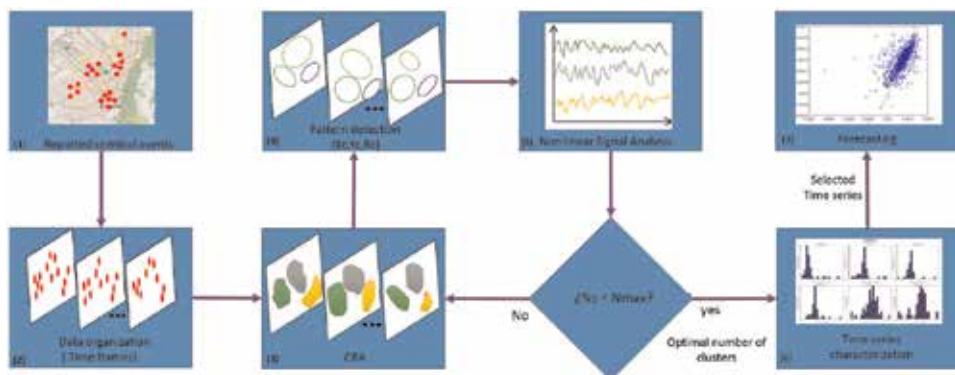


Figure 1. The time series from clustering method uses a clustering reorganization algorithm over reported data to produce clusters of crime events that evolve in time and space. The evolution is represented by means of time series of the clusters' parameters. A designed index relates the number of clusters with the foreseeability of time series by using non-linear analysis.

same throughout the time. In CRA, the clusters are identified by the Fuzzy C-Means algorithm (FCM) [19]. The outline of clusters can also be determined by several other clustering algorithms, such as the Gustafson-Kessel [20], among others. The identification of the clustering algorithm is related to the possible prototypes that can be found in the data frames. These prototypes are related to the urban form where the events take place [11].

The FCM and other algorithms alike initialize their parameters randomly. In this case, the centers of the clusters, their order and the membership values vary from one clustering experiment to another. Because of this random initialization, the clusters identified will not remain in the same zone, nor will the order of the fuzzy partitions created be preserved throughout time.

Therefore, if a study of C clusters is carried out, there must be certainty that the order of the partitions is maintained. It was considered for this case that the spatio-temporal trends in crime tend to be regular due to the normal population behavior. Thus, if a cluster is identified in a time frame, call it C_i , then in all future time frames clusters must be identified near that previously identified cluster. Euclidean distance is taken as a basis for the evaluation between the centers of the clusters of the different time frames (Time frame 0 and Time frame k). The CRA obtains a time series from clustering in five steps. These steps are: Initialization, Iteration, Distance Evaluation, Discarding non-minima, and Assignment.

First, in the initialization stage, the CRA requires the number of clusters to be set, the k -th time frame in which the dataset is in, as well as, the center guide matrix. The center guide determines the order of the clusters in the future time frames. As a second step, the CRA ensures that throughout all time frames, the number of clusters is maintained. In this step, the algorithm verifies that for each reorganization process carried out, each center identified in the time frame k is assigned to a different cluster in the center guide (time frame 0). If the number of clusters is not maintained, this stage is responsible for executing the FCM algorithm once more to re-evaluate the organization of the clusters.

Third, in the distance evaluation stage, the process to measure the Euclidean distance $d_{ij}^{(k)}$ between the center of clusters identified in time frames t_0 and t_k , where i and j represent the cluster order of the k -th frame and the first frame, respectively, takes place as follows:

$$d_{ij}^{(k)} = \sum_{j=1}^C \sqrt{\left(x_j^{(0)} - x_i^{(k)}\right)^2 + \left(y_j^{(0)} - y_i^{(k)}\right)^2}, \quad i = 1, 2, \dots, C. \quad (1)$$

where $d_{ij}^{(k)}$ is the Euclidean distance between the center of the cluster j in time 0 and the cluster i in time k . Based on this distance evaluation, a criterion for the reorganization is established, where the clusters order is assigned according to the minimum distance of each cluster to the centers in the center guide matrix.

Even though clusters will tend to occupy the same zones, in certain cases there may be some clusters that are identified further away from their usual locations. To prevent the CRA confusing the cluster organization, the “discarding non-minima” stage is introduced. In this stage, it is assumed that the i -th center cr_i in time frame k to the closest j -th center cr_j in the time frame 0 is selected for the order in which the i -th cluster is organized.

Finally, the assignment stage takes place once the iteration condition is met. The centers and the membership values have already been assigned in the correct order, according to the identification of the clusters in the first time frame. This stage assigns the correct order to the variables of the centers of the clusters as well as to the membership value of each event. By doing so for each time frame, once the

whole time window is processed there will be a set of time series that represents the spatio-temporal dynamics of the groups that have been identified by the clustering algorithm.

An example of three time series that surrogate the dynamics of a cluster is depicted in **Figure 2**. The spatial dynamics of a cluster is presented as a circle that moves in different frames and whose radius also changes. The dynamics of this cluster is summarized by three time series: $X(t)$ displacement of the cluster centroid in dimension x , $Y(t)$ displacement of the cluster centroid in dimension y , and $R(t)$ radius of the cluster measured as the Euclidean distance from its centroid to the farthest criminal event that belongs to the cluster.

2.3 Non-linear signal processing and optimal number of clusters

2.3.1 The MIG index

The memory, information, and geometry (MIG) index is proposed in this work as a scalar that quantifies the predictability of a time series obtained from the CRA. The index is constructed as follows:

$$q = h^I \frac{G}{M} \quad (2)$$

The predictability of a time series becomes more plausible as the MIG index is minimized. M evaluates the amount of non-linear correlation inside the time series (i.e., Memory). I is an indicator of how much information is produced by the signal. G accounts for the size of the phase space in which the dynamics can be embedded (i.e., Geometry).

In practical terms, the MIG index for a time series $S(t)$ can be evaluated from statistics grounded in non-linear signal processing and chaos theory as:

$$q_s = q(S(t)) = e^{\lambda} \frac{D}{\tau} \quad (3)$$

where λ corresponds to the estimated Largest Lyapunov Exponent (LLE) of $S(t)$, D is the size of the embedding space of the series obtained from the estimation of

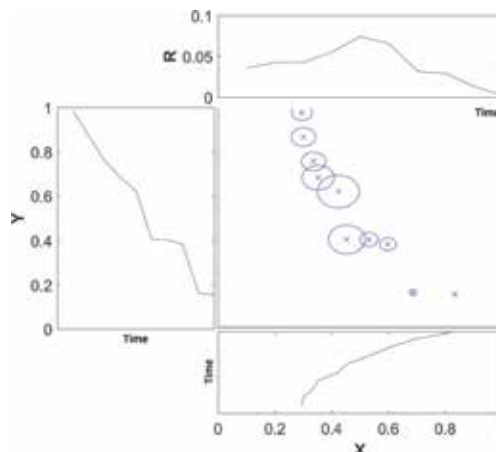


Figure 2. Example of time series that describe a cluster dynamics. Parameters of a circular cluster tracked by the CRA evolve in time producing three time series that correspond to: Horizontal displacement $X(t)$, vertical displacement $Y(t)$, and radius of the cluster $R(t)$.

the False Nearest Neighbors in the attractor dynamics of the signal and τ is the correlation lag located in the first minimum of the mutual average information of $S(t)$. A detailed explanation about the computation of these quantities can be found in [21], while a practical estimation of LLEs is proposed in [22].

2.3.2 Optimal number of clusters

Dynamics of the i -th cluster is represented by three signals (i.e., $X_i(t)$, $Y_i(t)$, and $R_i(t)$), then the MIG index q_C for the dynamics of a partition with C clusters is computed as follows:

$$q_C = \frac{1}{C} \sum_{i=1}^C q_{X_i} + \frac{1}{C} \sum_{i=1}^C q_{Y_i} + \frac{1}{C} \sum_{i=1}^C q_{R_i} \quad (4)$$

The optimal number of clusters C_{opt} is obtained by minimizing q_C :

$$C_{opt} = \arg\left(\min\left(q_{C_j}\right)\right) \quad C_j = 2, 3, \dots, C_{max} \quad (5)$$

where C_{max} is the largest number of clusters considered in the optimization process.

The minimum MIG index q_C^{opt} suggests that in average the time series produced when $C_j = C_{opt}$ is more predictable than in any other case. Thus the dynamics should be studied over $3 * C_{opt}$ signals.

2.4 Time series characterization

MIG index is proposed as an a-priori estimation of the predictability of a time series produced by the CRA. The signals that belong to the most and least predictable scenarios should be forecasted so the MIG validity can be corroborated, however it is computationally expensive. Instead, these signals are characterized through several statistics (independent from the MIG index) in order to select some as the most representative signals for each scenario.

2.4.1 Characterization

Characterization is performed through a selected group of statistics: the first and second estimated statistical moments of the distribution p of the series $S(t)$, the Shannon entropy and the number of lags beyond which the autocorrelation of $S(t)$ is effectively zero. The estimators [23] of the two first statistical moments are computed as follows:

$$\mu = \frac{1}{L} \sum_{k=1}^L S_k \quad (6)$$

$$\sigma = \frac{1}{L-1} \sum_{k=1}^L |S_k - \mu|^2 \quad (7)$$

where S_k is the discrete representation of $S(t)$ and L is the number of samples.

The third measure, the entropy described in (8), is related to the information uncertainty inside the series. A high entropy value is interpreted as a low-predictability variable.

$$E = - \sum_{i=1}^L p(S) * \log(p(S)) \quad (8)$$

The autocorrelation function is shown in Eq. (9). The proposed measure represents a confidence bound of the number of regressors with a linear dependence on the signal. It establishes the grade of possible foreseeability in the basis of its periodical behavior as long as forecasting crime is given in terms of stochastic processes [24].

$$r_h = \frac{c_h}{c_0} \quad (9)$$

$$c_h = \frac{1}{L-1} \sum_{k=1}^{L-h} (S_k - \bar{S})(S_{k+h} - \bar{S})$$

\bar{S} the average of the signal. The number of lags Lg is obtained when c_h drops below $0.2c_0$ for $h > 0$.

2.4.2 Series analysis

The chosen statistics are applied over time series whereby a four-dimensional vector (Eq. (10)) is obtained per signal:

$$v_S = [\mu, \sigma, E, Lg] \quad (10)$$

where S is the class identification of the corresponding series: X —displacement of the cluster in dimension x , Y —displacement of the cluster in dimension y , and R —radius of the cluster. These vectors are arranged in matrices as in Eq. (11). In this way, the analysis is done for each matrix separately:

$$M_S = \begin{pmatrix} v_{S_1} \\ \vdots \\ v_{S_C} \end{pmatrix} \quad (11)$$

where C represents the number of clusters.

- Normalization: the aim is to find the most representative signals (i.e., X, Y, R). The statistics are considered as a way to describe each series, but most of them have a dependence of the scale avoiding a comparison in value. Consequently, a difference of one order of magnitude could be significant in the lags of autocorrelation, but insignificant in the mean. Therefore, normalization is a way to establish a common reference. In this case, the normalization interval is $(0, 1]$ avoiding the value 0. Each M_S matrix is independently normalized.
- Finding the average vectors: an average vector is built from the normalized matrices. As per the last step, this process is independently applied. It is important to emphasize that in a determined clustering, there is an average vector \bar{v}_S for each M_S :

$$\bar{v}_S = \left(\frac{1}{C} \sum_{i=1}^C v_{S_i} \right) \quad (12)$$

- Determining distances: the distance between each row of the matrices M_S and the corresponding average vector is computed. Thereby, the Euclidean distance is calculated for all vectors v_{S_i} as follows:

$$d_{\bar{v}_S v_{S_i}} = \left((\bar{v}_S - v_{S_i})(\bar{v}_S - v_{S_i})^T \right)^{1/2} \quad (13)$$

The matrices D_S are organized by collecting the distances calculated from the matrices M_S .

$$D_S = \begin{pmatrix} d_{\bar{v}_S v_{S_1}} \\ \vdots \\ d_{\bar{v}_S v_{S_C}} \end{pmatrix} \quad (14)$$

- The fittest vectors: finally, the series corresponding to the position of the minimum value in each matrix D_S is chosen as the most representative signal. Thus, there are three representative signals obtained from D_X , D_Y , and D_R which are referenced as $X_f(t)$, $Y_f(t)$, and $R_f(t)$, respectively. These three signals are taken in the forecasting stage as representation of the clustering dynamics for a given number of clusters. Particularly, the representative signals for the best and worst conditions of the MIG index are considered in this work.

2.5 Evolutionary-fuzzy forecasting of representative time series

Forecasting of representative time series is carried out by means of a custom memetic algorithm [25, 26]. The heuristic was proposed in [27] as the combination of the differential evolution algorithm [28] and the adaptive neuro-fuzzy inference system (ANFIS) [29]. The general flow graph of the algorithm is shown in **Figure 3**.

2.5.1 Differential memetic neuro-fuzzy algorithm

A memetic algorithm allows to take advantage of both global and local search. In this manner, the optimization space can be explored widely and deeply. The differential evolution algorithm (highlighted in orange) is chosen as the global optimizer, on account of its use for optimizing multidimensional real-valued functions. In addition, it has been successfully applied for model optimization of complex systems [30]. Several variants of this algorithm have been proposed in literature, taking into account the multiplicity of elitism strategies, chromosome representations, and mutation operators. ANFIS (highlighted in pink) is implemented as the supervised learning strategy of the memetic algorithm. It uses the gradient of the objective function on the basis of its differentiability to search for solutions around a locality of the optimization landscape. ANFIS is supported on Sugeno-type fuzzy systems with gaussian membership functions in the inputs [29, 31]. In addition, the proposed fuzzy system has m rules, which also corresponds to the number of membership functions in each input and the amount of singleton fuzzy sets in the output. Each rule relates the $r - th$ group of membership functions in the inputs with the i th singleton value in the output [30].

As shown in the flowgraph in **Figure 3**, when the population is adapted by ANFIS, it is necessary to re-evaluate the solutions and later, according to the fitness

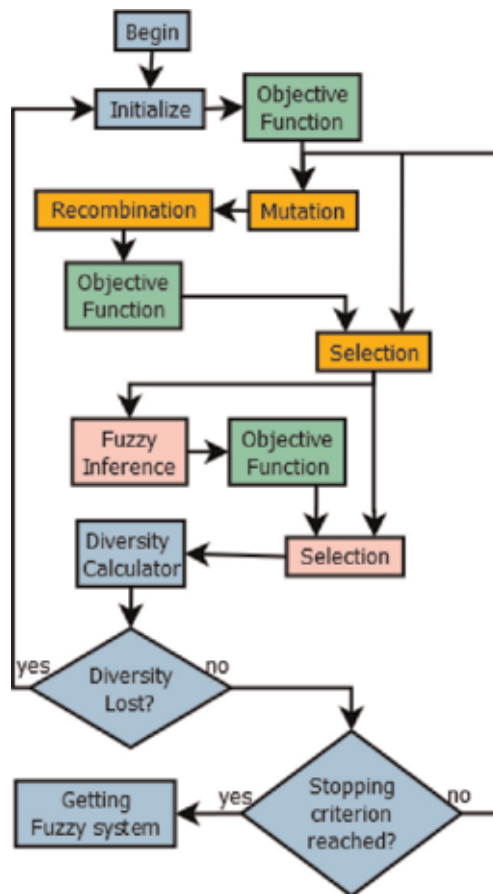


Figure 3. The evolutionary algorithm used to tune fuzzy forecasters. This algorithm uses the script of the differential evolution algorithm with a step of local search provided by an adaptive neural fuzzy inference system.

function, select them. This re-evaluation must be carried out since ANFIS relies on the RMSE to evaluate the solutions but the memetic algorithm relies on the fitness function. Thus, an interesting individual for ANFIS may not necessarily be good according to its fitness score.

In order to guarantee population diversity, and avoiding a fast-convergence caused by ANFIS, the diversity calculator considered is the FANO factor [26], which calculates the average variation of the mean in the membership functions according to Eq. (15). The population is re-initialized if the FANO factor is greater than a given threshold:

$$\hat{F} = \left(\frac{1}{n * m} \sum_{i=1}^{n * m} \frac{1}{j-1} \sum_{f=1}^j (D_{fl} - \bar{D}_i)^2}{\bar{D}_l} \right) * \frac{1}{gen} \quad (15)$$

where

$$\bar{D}_l = \frac{1}{j} \sum_{i=1}^j D_{li}$$

And $n * m$ is the number of membership functions in the inputs, D_{li} is the average of the i -th membership function of the l -th individual, and gen is the current generation.

2.5.2 Fitness function

The root mean squared error (RMSE) index used by ANFIS may not be appropriate for guiding the global search of an evolutionary algorithm [30] while dealing with complex behaviors. Thus, a problem-oriented fitness function to evaluate candidate solutions must be designed. This function is significant because it provides the criteria to judge a solution and its performance. In addition, it is imperative to include several performance indicators to check solutions in a more precise and proper manner. This process is carried out by minimizing the following expression:

$$f = \frac{(1 - NSE) * MAE}{POCID} \quad (16)$$

The fitness function is composed of three indexes. Each one has been chosen to guide the evolution regarding features in the best solution. Thus, many possible solutions that do not have a near-zero mean error but can forecast the series properly, will be explored.

The Nash-Sutcliffe efficiency (NSE) is conceived as an overall performance measure. It is a normalized statistic that determines the relative magnitude of the residual variance compared to the measured data variance [32]. A value of 1 corresponds to a perfect match of the modeled \hat{d} to the observed data d .

$$NSE = 1 - \frac{\sum_{k=1}^N (d_k - \hat{d}_k)^2}{\sum_{k=1}^N (d_k - \bar{d})^2} \quad (17)$$

The mean absolute error (MAE) represents a clear interpretation of the absolute difference between the series and its forecasting, relative to the series [33]. It is chosen instead of RMSE because the series has several peaks and it is necessary to rest importance on them in the forecasting. If it was not done, the fitness function would penalize errors in the lowest values (i.e., near the mean) more than in the highest ones.

$$MAE = \frac{1}{N} \sum_{i=1}^N \left| \frac{d_k - \hat{d}_k}{d_k} \right| \quad (18)$$

Prediction of change in direction (POCID) is a non-linear index included to ensure that a forecast that does not fit the series well, but follows its direction changes reliably, gets a good score [34]. As a multiplicative inverse, it causes an increase in the fitness function as large as how far the fitness score is from the optimal value.

$$POCID = \frac{\sum_{k=1}^N B_k}{N} \quad (19)$$

where

$$B_k = \begin{cases} 1, & \text{if } (d_k - d_{k-1})(\hat{d}_k - \hat{d}_{k-1}) \geq 0 \\ 0, & \text{otherwise} \end{cases} \quad (20)$$

$NSE = 1$ and $MAE = 0$ represent the two roots of f , hence these are enough reasons to consider a solution as the best, but the POCID operates as a non-linear penalizer.

3. Results

The method outlined in the previous section was applied to forecast criminal patterns in two cities: San Francisco, USA and Bogota, Colombia. Results obtained from evolved fuzzy predictors are described in this section. In addition, evidence is presented from these cases about the validity of the MIG index for setting the optimal number of clusters of a dynamic partition of crime events.

3.1 Results for San Francisco, USA

3.1.1 Data organization

The criminal dataset for the city of San Francisco was obtained online through the open data services provided by the local government of the city. The dataset used in this work registers about 70,000 criminal events between years 2003 and 2015. Each criminal register contains attributes such as latitude, longitude, time, date, type of crime, among others. For the purposes of this study, only house burglary registers were considered, and from each register only latitude, longitude, and date were taken into account. Each spatial register was projected by means of the universal mercator system taken the location of the city of San Francisco as reference. Relative distances were expressed in meters. Time frames of criminal activity were created by aggregating the crime events of the last 7 days. Frames iterate from 1 day to the next. A total of 3195 time frames were produced.

3.1.2 Optimal number of clusters

Optimization of the MIG index was carried out over time series from clustering with $C = 2 \dots 16$, as shown in **Figure 4**. There was not be a clear criteria to set the maximum number of clusters to be considered in the optimization of any clustering validation index. In this case, the maximum was determined by taking into account that San Francisco city is divided in 10 police districts. Thus, the optimization was computed considering just three additional clustering settings (i.e. $C = 12, 16, 20$). It was observed for these values that the MIG index diverged too fast, since in $C = 20$, it reached a value greater than the first maximum by about two orders of magnitude. The optimal MIG index is obtained for $C = 4$, whose value is around 30% of the maximum. However, the index found in $C = 5$ is quite similar to the optimum. It is expected to find a similar predictability potential for these two cases.

The difference between the extreme cases may be explained by examining **Figure 5**. Note that for $C_{opt} = 4$, big areas are covered so that clusters concentrate a greater number of crime events. Therefore, the spatial variability of clusters in the optimal case is less subject to fluctuations than that of the worst case. In other

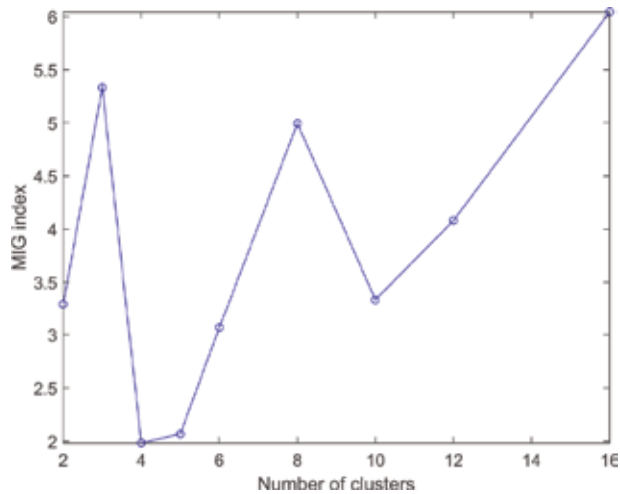


Figure 4. MIG index computed for house burglaries in the city of San Francisco, USA. The minimum of the MIG index appears at four clusters, where it is expected that time series from clustering are the most predictable.

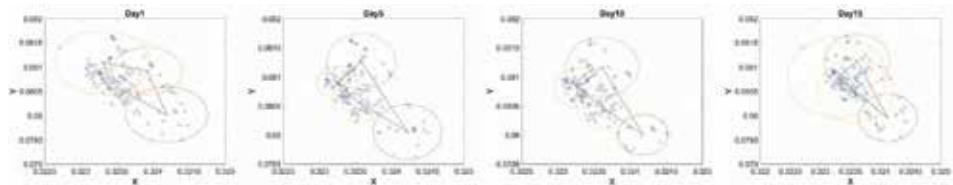


Figure 5. An example of cluster dynamics for house burglaries in the city of San Francisco, USA. In this example, four clusters were tracked with the clustering reorganization algorithm (CRA).

words, the dynamics of clusters in the optimum case would exhibit the lowest level of disorder.

An example of cluster dynamics for house burglaries in the city of San Francisco, USA. In this example, four clusters were tracked with the clustering reorganization algorithm (CRA).

3.1.3 Characterization of time series

According to the MIG index, only clustering results for $C = 4$ and $C = 16$ groups were considered. These cases represent the most and least predictable dynamic partitions. It is necessary to choose a representative time series (Rep) in the two scenarios, denoting them as the optimal scenario (i.e., minimum MIG index) and the control scenario (i.e., maximum MIG index). **Table 1** summarizes some results from this process. In column S , the field “[i]” refers to the cluster number. As shown in this table, selected series are from different clusters since the first aim of the forecasting in this study is to have a preliminary validation of the MIG index instead of developing an exhaustive forecast. Representative series (Rep) are selected by finding the minimum distance between their characterization vector and the mean vector of a class S (i.e., X , Y , and R) according to Eqs. (10)–(14).

3.1.4 Forecasting results

Once the series were selected, the memetic algorithm was configured to compute the forecasting with 70% of the data (i.e., 2236 days) and compute the

validation with the other 30% of the data (i.e., 959 days). In order to initialize the algorithm in such a way that the search space could be explored widely with the least possible computational cost, guaranteeing the simplicity of the solutions, a set of experiments was launched for the combination of parameters presented in **Table 2**. The number of generations and population size were selected as the maximum values in which at least a difference of 0.1% between consecutive generations was found in the fitness function of the fittest solutions. Number of Epochs was chosen to avoid frequent loss of diversity so that the FANO factor was close to the selected threshold. Regarding the fuzzy predictor, the number of regressors in the inputs and the number of rules were selected by finding the maximum values in which the Pearson coefficient for the forecast and real data did not change in more than 2%.

From selected configuration values, due to the random initialization of the memetic algorithm, over 10,000 experiments were run per series. **Table 3** summarizes the performance of the best fuzzy predictors found by the memetic algorithm. Pearson's correlation coefficient between the model and data for the optimal scenario was the greatest in both training and validation. The same observation can be stated for the Nash coefficient. The relative difference between Pearson coefficients of the optimal and control scenarios was about 17% in validation whereas for the Nash index, the relative difference was 26%. Regarding the estimated LLE, smaller values were obtained for the optimal scenario, and even a negative LLE was obtained. Visual results are depicted in **Figure 6**. Selected time series exhibited an interesting texture, which was more accentuated in the control scenario since more peaks appeared randomly and recurrence was not easily observed.

S	μ		$\sigma \times 10^{-8}$		$E \times 10^9$		Lg		D_s
	Rep	Mean	Rep	Mean	Rep	Mean	Rep	Mean	Norm
x [1]	0.323	0.323	1.139	1.534	4.692	4.821	9	12.75	0.052
y [2]	0.0810	0.0806	2.3229	4.230	1.765	2.4516	7	11	0.337
R [3]	5.23e-04	5.14e-04	1.5880	1.574	4.292	4.4145	7	19.75	0.044

These series are a representative sample of the entire, set which contains 12 time series (four clusters each one with three parameters).

Table 1.
 Features of representative time series obtained from the TSC method for house burglaries in the city of San Francisco, USA, considering four clusters.

Parameter	Range	Step	Value
Number of generations	50–500	50	250
Population size	10–50	10	20
Epochs number	1–20	1	5
Number of regressors	3–12	1	8
Number of rules	4–24	4	8

The first three parameters were used to configure the differential evolution algorithm whereas the last two were used to adjust the Adaptive Neural Fuzzy Inference System.

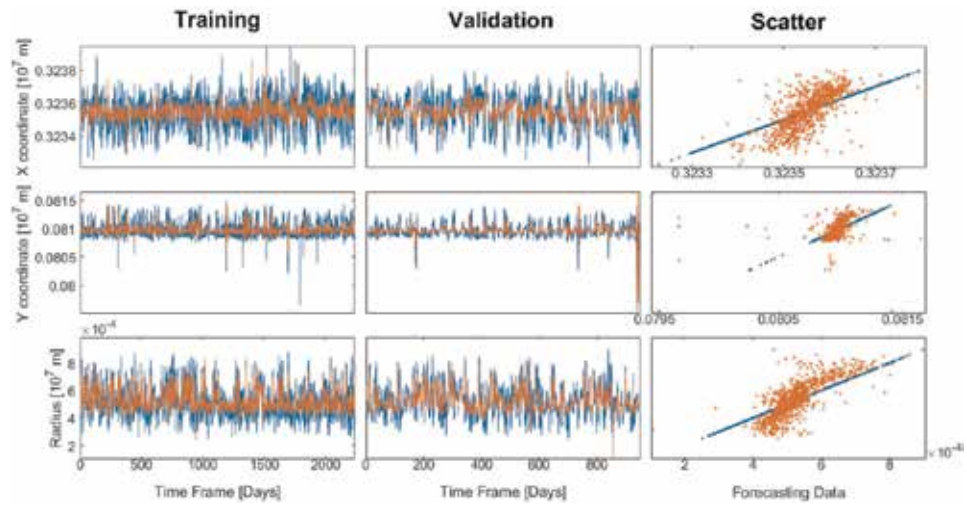
Table 2.
 Parameters of the evolutionary fuzzy predictor used for the city of San Francisco, USA.

S	Pearson coefficient				Nash		LLE	
	Training		Validation		4	16	4	16
	4	16	4	16				
x	0.6099	0.3986	0.6942	0.4581	0.3430	0.1534	0.0339	-0.0359
y	0.5447	0.7114	0.4982	0.5546	0.2908	0.5021	-0.0118	0.0796
R	0.8353	0.6645	0.7803	0.6260	0.6296	0.2779	0.0388	0.0677
Average	0.6633	0.5915	0.6576	0.5463	0.4212	0.3111	x	x

Dynamic partitions with minimum (4 clusters) and maximum (16 clusters) MIG index were considered.

Table 3.

Forecasting indices for representative time series of two dynamic partitions (house burglaries in the city of San Francisco, USA).

**Figure 6.**

Forecasting results for the three representative time series of the cluster dynamics (House burglaries in the city of San Francisco, USA). Series were obtained from a dynamic partition with four clusters.

3.2 Results for Bogota, Colombia

3.2.1 Data organization

The dataset for the city of Bogota, Colombia was provided by the Non-Governmental Organization (NGO) “Fundacion ideas para la paz,” which contains about 25,000 events of cellular phone robbery registered between the years of 2012 and 2015. Criminal registers contain X-coordinates, Y-coordinates, and dates of events. No transformation was required since the coordinates were already expressed in a geographical system adapted to the city. All differential spatial measurements were processed in meters. As in the previous case, time frames were created by the aggregation of the criminal events recorded in the last 7 days. A total amount of 1417 time frames was generated.

3.2.2 Optimal number of clusters

According to **Figure 7**, the MIG index for Bogota series exhibited a global minimum at $C_{opt} = 3$, showing non-convex behavior with respect to the number of

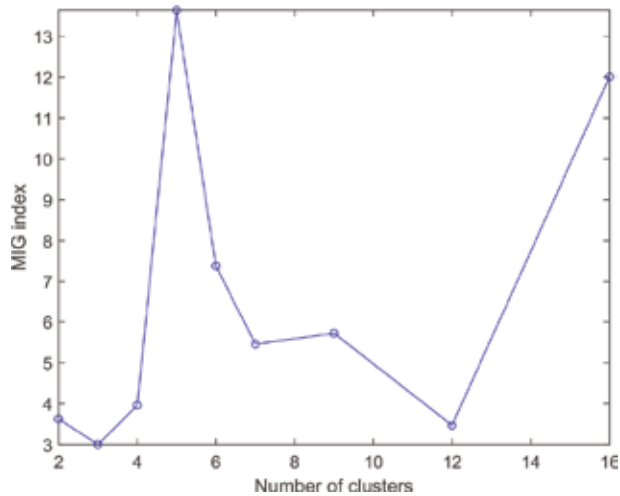


Figure 7. MIG index computed for cellular phone robbery in the city of Bogota, Colombia. The minimum of the MIG index appears at four clusters, where it is expected that time series from clustering are the most predictable.

clusters. Note the similarity of the optimal MIG index and the one obtained for $C = 12$. The optimization process was carried out for $C = 2 \dots 20$; however, the last result with $C = 20$ was omitted since the MIG index grew exaggeratedly.

A sample of the cluster dynamics is presented in **Figure 8** for the optimal case. However, it is not a straightforward task to infer the result of the optimization process from this figure. Centroids of clusters are moving, as can be noticed from the change in shape of the connecting polygon. Note that the radiuses of the clusters changed in size so different areas were covered from one time frame to another.

3.2.3 Characterization of time series

Characterization of time series of dynamic partitions are summarized in **Table 4**. The field “[i]” refers to the cluster number. For these series it is supposed a-priori that the three-group clustering is more predictable than the five-group, considering the respective entropies. This result is in accordance with the MIG index optimization. Regarding the radius series, the number of lags is higher with respect to the series collection. Representative series (Rep) were selected by finding the minimum distance between their characterization vector and the mean vector of a class S (i.e., X , Y , and R) according to Eqs. (10)–(14).

3.2.4 Forecasting results

Table 5 presents the combination of parameters that were used for running the optimization of the fuzzy forecasters. In this case, 70% of generated samples (i.e.,

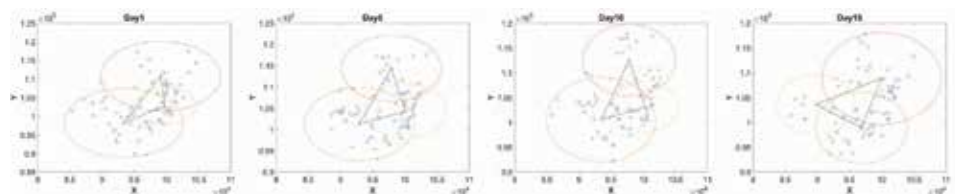


Figure 8. An example of cluster dynamics for cellular phone robbery in the city of Bogota, Colombia. In this example three clusters were tracked with the clustering reorganization algorithm (CRA).

S	$\mu \times 10^4$		$\sigma \times 10^6$		$E \times 10^{-5}$		Lg		D_S
	Rep	Mean	Rep	Mean	Rep	Mean	Rep	Mean	Norm
x [3]	9.8	9.7	8.115	7.251	1.036	1.4336	7	9	0.251
y [3]	10.9	10.4	11.951	11.265	0.894	0.755	6	8	0.099
R [3]	9.8	0.6	3.854	4.4214	1.723	1.484	163	158	0.15

These series are a representative sample of the entire set, which contains nine time series (three clusters each one with three parameters).

Table 4.

Features of representative time series obtained from the TSC method for cellular phone robbery in the city of Bogota, Colombia, considering three clusters.

Parameter	Range	Step	Value
Number of generations	100–400	50	200
Population size	10–50	10	30
Epochs number	1–20	1	4
Number of regressors	3–12	1	8
Number of rules	4–24	4	8

The first three parameters configured the differential evolution algorithm, whereas the last two were adjusted for the Adaptive Neural Fuzzy Inference System.

Table 5.

Parameters of the evolutionary-fuzzy predictor used for the city of Bogota, Colombia.

992 days) were used for training, and the remaining 30% for validation (i.e., 425 days). The best forecasting results for Bogota series are presented in **Table 6**. The Pearson correlation coefficient between the model and real data was greater for the optimal scenario (i.e., three clusters) with respect to the control scenario (i.e. five clusters) in both training and validation. The relative difference was about 18 and 12%, respectively. In terms of the Nash index, the optimal scenario exhibited the highest performance with a relative difference of 31%. Estimated LLEs were smaller in the optimal scenario with a negative LLE in the case of the $X(t)$ signal. **Figure 9** depicts visual results. It can be seen that predicted signals in the optimal scenario are correlated to the real ones. Signals in the control scenario exhibited more random peaks, although the recurrence is similar to the optimal case.

Signal	Pearson Coefficient				2*Nash		2*LLE	
	Training		Validation		3	5	3	5
	3	5	3	5				
x	0.8443	0.7388	0.9154	0.6833	0.6546	0.5025	-1.5102	-0.3230
y	0.7238	0.6833	0.4009	0.4755	0.5069	0.2795	1.5157	1.5100
R	0.8879	0.5025	0.7538	0.6441	0.7740	0.5499	1.2457	2.1515
Average	0.8186	0.6708	0.6900	0.6010	0.6452	0.4440	x	x

Dynamic partitions with minimum (three clusters) and maximum (five clusters) MIG index were considered.

Table 6.

Forecasting indices for representative time series of two dynamic partitions of cellular phone robbery in the city of Bogota, Colombia.

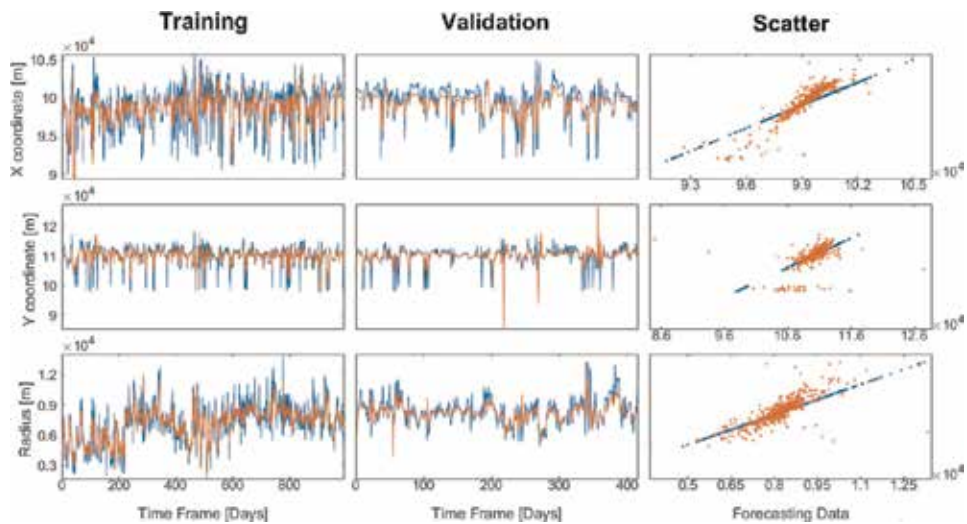


Figure 9. Forecasting results for the three representative time series of the cluster dynamics (cellular phone robbery in the city of Bogota, Colombia). Series were obtained from a dynamic partition with three clusters.

4. Conclusion

Qualitatively speaking, similar results were obtained for the two study cases. In both cases, the forecasting of time series from the optimal scenario outperformed that of the control scenario in terms of two statistical indices (i.e., Pearson correlation and Nash coefficients) and one from chaos theory (i.e., LLE). Therefore, we argue that the MIG index might be useful to evaluate the goodness of a dynamic partition of crime events. Moreover, it may give a confidential a-priori insight about the predictability of series synthesized from TSC. Hence, this method produces coherent series that preserve a temporal structure, which a forecasting method can take advantage of.

TSC series for both cities exhibited an interesting behavior in terms of their texture. No evident periodicity was observed and abundant peaks appeared over the observed time window. Positive LLEs computed in these signals revealed the presence of a possible chaotic nature of the phenomena. Chaotic texture of these series speaks about non-stationary spatial crime patterns that evolve continuously producing information. Thus, this observation reflects a footprint of complexity for urban crime as noted in previous studies [35].

As a future work, the proposed method will be tested over other dynamic phenomena characterized by non-uniform sampling of relevant variables in both space and time. The method will be also refined by considering other techniques such as auto-encoder deep neural networks among others.

Conflict of interest

The authors declare no conflict of interest.

Author details

Miguel Melgarejo^{1*†}, Cristian Rodriguez^{1†}, Diego Mayorga^{1†} and Nelson Obregón^{2†}

1 Laboratory for Automation and Computational Intelligence, Universidad Distrital Francisco José de Caldas, Bogotá DC, Colombia

2 Water Institute, Pontifical Xaverian University, Bogotá DC, Colombia

*Address all correspondence to: mmelgarejo@udistrital.edu.co

†All authors are contributed equally.

IntechOpen

© 2019 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/3.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. 

References

- [1] Felson M. Routine activity approach. In: *Environmental Criminology and Crime Analysis*. Abingdon, UK: Routledge; 2008. p. 7077
- [2] Cornish D, Clarke R. The rational choice perspective. In: *Environmental Criminology and Crime Analysis*. Abingdon, UK: Routledge; 2008. p. 2145
- [3] D’Orsogna M, Perc M. Statistical physics of crime: A review. *Physics of Life Reviews*. 2014;**12**:1-21
- [4] Perc M, Donnay K, Helbing D. Understanding recurrent crime as system-immanent collective behavior. *PLoS One*. 2013;**8**:e76063
- [5] Brantingham PJ, Brantingham P. Crime pattern theory. In: *Environmental Criminology and Crime Analysis*. New York: Willian Publishing; 2008. pp. 78-93
- [6] Short M et al. A statistical model of criminal behavior. *M3AS*. 2008;**18**: 1249-1267
- [7] Rey S, Mack E, Koschinsky J. Exploratory space-time analysis of burglary patterns. *Journal of Quantitative Criminology*. 2012;**28**(3): 509-531
- [8] Mohler G. Marked point process hotspot maps for homicide and gun crime prediction in Chicago. *International Journal of Forecasting*. 2014;**30**(3):491-497
- [9] Grubestic T. On the application of fuzzy clustering for crime hot spot detection. *Journal of Quantitative Criminology*. 2006;**22**(1):77-105
- [10] Brantingham P et al. Crime analysis at multiple scales of aggregation: A topological approach. In: *Putting Crime in Its Place*. New York: Springer; 2009. pp. 87-107
- [11] Davies T, Johnson S. Examining the relationship between road structure and burglary risk via quantitative network analysis. *Journal of Quantitative Criminology*. 2015;**31**(3):481-507
- [12] Malleson N, Andresen M. Spatio-temporal crime hotspots and the ambient population. *Crime Science*. 2015;**4**(10):1-8
- [13] Bettencourt L et al. Urban scaling and its deviations: Revealing the structure of wealth, innovation and crime across cities. *PLoS One*. 2010; **5**(11):20-22
- [14] Murray A, Grubestic T, Leitner M. Exploring spatial patterns of crime using non-hierarchical cluster analysis. In: *Crime Modeling and Mapping Using Geospatial Technologies*. Vol. 8. Netherlands: Springer; 2013. pp. 105-124
- [15] Mayorga D, Melgarejo M, Obregon N. A fuzzy clustering based method for the spatiotemporal analysis of criminal patterns. In: *2016 IEEE International Conference on Fuzzy Systems*; 2016. pp. 738-744
- [16] Izakian H, Pedrycz W, Jamal I. Clustering spatio-temporal data: An augmented fuzzy C-means. *IEEE Transactions on Fuzzy Systems*. 2013; **21**(5):855-868
- [17] Ji M, Xie F, Ping Y. A dynamic fuzzy cluster algorithm for time series. *Abstract and Applied Analysis*. 2013; **2013**:1-7
- [18] Hardyns W, Rummens A. Predictive policing as a new tool for law enforcement? Recent developments and challenges. *European Journal on Criminal Policy and Research*. 2018;**24**:201
- [19] Bezdek JC. *Pattern Recognition with Fuzzy Objective Function Algorithms*.

New York: Springer Science, Business Media; 2013

[20] Babuska R. Fuzzy modeling for control. In: International Series in Intelligent Technologies. Netherlands: Springer; 1998

[21] Abarbanel H. Analysis of Observed Chaotic Data. New York: Springer; 1996

[22] Rosenstein MT et al. A practical method for calculating largest Lyapunov exponents from small data sets. *Physica D*. 1993;**65**:117-134

[23] Ayyub B, McCuen R. Probability, Statistics, and Reliability for Engineers and Scientists. Boca Raton: Chapman and Hill/CRC Press; 2002. pp. 65-72

[24] Ting S, Gang C. Selection of fuzzy time series model based on autocorrelation theory. In: 29th Chinese Control and Decision Conference; Chongqing, China; 2017. pp. 4365-4369

[25] Neri F, Cotta C. Memetic algorithms and memetic computing optimization: A literature review. *Swarm and Evolutionary Computation*. 2012;**2**:1-14

[26] Moscato P, Cotta C. A Gentle Introduction to Memetic Algorithms. Boston MA: Glover, Fred and Kochenberger; 2003. pp. 105-144

[27] Rodriguez C, Mayorga D, Melgarejo M. Forecasting time series from clustering by a memetic differential fuzzy approach: An application to crime prediction. In: 2017 IEEE Symposium Series on Computational Intelligence (SSCI); Honolulu, HI; 2017

[28] Storn R, Price K. Differential evolution: A simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*. 1997;**11**:341-359

[29] Jang J. ANFIS: Adaptive-network-based fuzzy inference system. *IEEE Transactions on Systems, Man, and Cybernetics*. 1993;**23**(3):665-685

[30] Chivata J et al. Complex system modeling using TSK fuzzy cellular automata and differential evolution. In: 2013 IEEE International Conference on Fuzzy Systems; Hyderabad; 2013. pp. 1-5

[31] Mendel J. Fuzzy logic systems for engineering: A tutorial. *Proceedings of the IEEE*. 1995;**83**(3):345-377

[32] Nash J, Sutcliffe J. River flow forecasting through. Part I: A conceptual models discussion of principles. *Journal of Hydrology*. 1970;**10**(3):282-290

[33] Chai T, Draxler R. Root mean square error (RMSE) or mean absolute error (MAE)? Arguments against avoiding RMSE in the literature. *Geoscientific Model Development*. 2014;**7**:1247-1250

[34] Silva D, Alves G, Mattos Neto P, Ferreira T. Measurement of fitness function efficiency using data envelopment analysis. *Expert Systems with Applications*. 2014;**41**(16): 7147-7160

[35] Oliveira M, Bastos-Filho C, Menezes R. The scaling of crime concentration in cities. *PLoS One*. 2017; **12**:113

Encountered Problems of Time Series with Neural Networks: Models and Architectures

Paola Andrea Sánchez-Sánchez, José Rafael García-González and Leidy Haidy Perez Coronell

Abstract

The growing interest in the development of forecasting applications with neural networks is denoted by the publication of more than 10,000 research articles present in the literature. However, the high number of factors included in the configuration of the network, the training process, validation and forecasting, and the sample of data, which must be determined in order to achieve an adequate network model for forecasting, converts neural networks in an unstable technique, given that any change in training or in some parameter produces great changes in the prediction. In this chapter, an analysis of the problematic around the factors that affect the construction of the neural network models is made and that often present inconsistent results, and the fields that require additional research are highlighted.

Keywords: time series, prediction of neural networks, learning algorithms

1. Introduction

The time series forecasting has received a lot of attention in recent decades, due to the growing need to have effective tools that facilitate decision making and overcome the theoretical, conceptual, and practical limitations presented by traditional approaches. The classification of forecasting methods from a statistical point of view, in general, has two aspects, one oriented to causal methods, such as regression and intervention models, and the other to time series, where mobile averages, exponential smoothing, ARIMA models, and neural networks are included. Under this current, the forecast is oriented only to the task of predicting the behavior, prioritizing forward vision and thus obviating many important steps in the model construction process; while the modeling is oriented to find the global structure, model and formulas, which explain the behavior of the data generating process and can be used to predict trends of future behavior (long term), as well as to understand the past. This last vision allows the construction of solid models in its foundation and under which the forecast is seen as an additional step.

The representation of time series with dynamics of nonlinear behavior has acquired great weight in the last decades, because many authors agree in affirming that the real world series present nonlinear behaviors, and the approximation that can be done with linear models, it is inadequate [1–3]. Although approximations have been made with statistical models (an extensive compilation of these is

presented by [4–6]), its representation is difficult to restrict its use to a functional form a priori, for which neural networks have proven to be a valuable tool since they allow to extract the unknown nonlinear dynamics present between the explanatory variables and the series, without the need to perform any assumptions.

The growing interest in the development of forecasting applications with neural networks is denoted by the publication of more than 10,000 research articles in the literature [7]. However, as stated by Zhang et al. [8], inconsistent results about the performance of neural networks in the prediction of time series are often reported in the literature. Many conclusions are obtained from empirical studies, thus presenting limited results that often cannot be extended to general applications and that are not replicable. Cases where the neural network presents a worse performance than linear statistical models or other models may be due to the fact that the series studied do not present high volatilities, that the neural network used to compare was not adequately trained, that the criterion of selection of the best model is not comparable, or that the configuration used is not adequate to the characteristics of the data. Whereas, many of the publications that indicate superior performance of neural networks are related to novel paradigms or extensions of existing methods, architectures, and training algorithms, but lack a reliable and valid evaluation of the empirical evidence of their performance. The high number of factors included in the configuration of the network, the training process, validation and forecast, and the sample of data, which is required to determine to achieve a suitable network model for the forecast, makes neural networks a technique unstable, given that any change in training or in some parameter produces large changes in the prediction [9]. In this chapter, an analysis of the problematic environment is made to the factors that affect the construction of neural network models and that often present inconsistent results.

Empirical studies that allow the prediction of time series with particular characteristics such as seasonal patterns, trends, and dynamic behavior have been reported in the literature [10–12]; however, few contributions have been made in the development of systematic methodologies that allow representing time series with neural networks on specific conditions, limiting the modeling process to ad-hoc techniques, instead of scientific approaches that follow a methodology and process of replicable modeling.

In the last decade, there has been a considerable number of isolated contributions focused on specific aspects, for which a unified vision has not been presented; Zhang et al. [8] made a deep revision until 1996. This chapter is an effort to evaluate the works proposed in the literature and clarify their contributions and limitations in the task of forecasting with neural networks, highlighting the fields that require additional research.

Although some efforts aimed at the formalization of time series forecasting models with neural networks have been carried out, at a theoretical level, there are few advances obtained [13], which evidences a need to have systematic research about of modeling and forecasting of time series with neural networks.

The objective of this chapter is to delve into the problem of forecasting time series with neural networks, through an analysis of the contributions present in the literature and an identification of the difficulties underlying the task of forecasting, thus highlighting the open field research.

2. Motivation of the study

The time series forecasting is considered a generic problem to many disciplines, which has been approached with different models [14]. Formally, the objective of

the time series forecasting is to find a flexible mathematical functional form that approximates with sufficient precision the data generating process, in such a way that it appropriately represents the different regular and irregular patterns that the series may present, allowing the constructed representation to extrapolate future behavior [15]. However, the choice of the appropriate model for each series depends on the characteristics of the time series, and its usefulness is associated with the degree of similarity between the dynamics of the series generating process and the mathematical formulation that is made of it under the premise that the data dictate the tool to be used [16].

As pointed out by Granger and Terasvirta [2], the construction of a model that relates a variable to its own history and/or to the history of other explanatory variables of its behavior can be carried out through a variety of alternatives. These depend both on the functional form by which the relationship is approximated and on the relationship between these variables. Although, each modeler is autonomous in the choice of the modeling tool, in cases where there are relations of a non-linear order, there are limitations in the use of certain types of tools, moreover, this same reason leads to the absence of a method that is the best for all cases. The question that arises is then, how to properly specify the functional form in the presence of non-linear relationships between the time series and the explanatory variables of its behavior.

The representation of time series with dynamics of nonlinear behavior has acquired great weight in the last decades, because many authors agree in affirming that the real world series present nonlinear behaviors, and the approximation that can be done with linear models, it is inadequate [1–3], among others. The approach of series with the stated characteristics has been made, among others, from statistical models, combined or hybrid models and neural networks. The complexity in the representation of non-linear relationships lies in the fact that in most cases, there are not enough physical or economic laws that allow us to specify a suitable functional form for their representation.

The literature has proposed a wide range of statistical models for the representation of series with nonlinear behavior such as bilinear models autoregressive thresholds—TAR, autoregressive soft transition—STAR [17, 18], autoregressive conditional heteroscedasticity—ARCH [19], and its generalized form—GARCH [20]; a comprehensive compilation of these is presented by [4–6]. Although the stated models have proved useful in particular problems, they are not universally applicable, since they limit the form of non-linearity present in the data to empirical specifications of the characteristics of the series based on the available information [2]; its success in practical cases depends on the degree to which the model used manages to represent the characteristics of the series studied. However, the formulation of each family of these models requires the specification of an appropriate type of non-linearity, which is a difficult task compared to the construction of linear models, since there are many possibilities (wide variety of possible non-linear functions), more parameters to be calculated, and more errors can be made [21, 22].

Likewise, in the prediction of time series, it is universally accepted that a simple method is not the best in all situations [23–25]. This is because real-world problems are often complex in nature and a model of this kind may not be adequate to capture different patterns. Empirical studies suggest that by combining different models, the accuracy of the representation may be better than for the individual case [26–28]. Therefore, the union of models with different characteristics increases the possibility of capturing different patterns in the data and provides a more appropriate representation of the time series. The hybrid modeling then arises, naturally as the union of similar or different techniques with complementary characteristics.

In the forecast literature, several combinations of methods have been proposed. However, many of them use similar methods, and this is how different studies about hybrid linear modeling techniques are found in the traditional literature. Although this type of combinations has demonstrated its ability to improve the accuracy of the representations made, it is considered that a more effective route could be based on models with different characteristics. Both theoretical and empirical evidence suggest that the combination of dissimilar models, or those that strongly disagree with others, leads to a decrease in model errors [29, 30] and allows, in addition, to reduce the uncertainty of this one [31]. The hybrid model is thus, more robust to estimate the possible changes in the structure of the data.

Numerous applications have been proposed in the literature based on combinations of linear models with computational intelligence [32–39]. However, the main criticisms of these works is that they do not contemplate the need to integrate subjective information into models, which, like traditional statistical models, require a preprocessing of the series, which is aimed at eliminating the visible components of this one and that require the determination of a large number of parameters, which are not economically explainable.

Neural networks seen as a non-parametric non-linear regression technique have emerged as attractive alternatives to the problem posed, since they allow extracting the unknown nonlinear dynamics present between the explanatory variables and the series, without the need to make any kind of assumptions. From this family of techniques, multi-layer perceptron networks—MLP, understood as a non-linear statistical regression model, have received great attention among researchers from the computational intelligence and statistics community.

The attractiveness of neural networks in the prediction of time series is their ability to identify hidden dependencies based on a finite sample, especially of a non-linear order, which gives them the recognition of universal approximation of functions [3, 40–42]. Perhaps the main advantage of this approach over other models is that they do not start from a priori assumptions about the functional relationship of the series and its explanatory variables, a highly desirable characteristic in cases where the mechanism generating the data is unknown and unstable [43], in addition to its high generalization capacity allows to learn behaviors and extrapolate them, which leads to better forecasts [5].

For artificial intelligence, as well as for operation research, the time series forecasting with neural networks is seen as a problem of error minimization, which consists of adjusting the parameters of the neural network in order to minimize the error between the real value and the output obtained. Although, this criterion allows obtaining models whose output is increasingly closer to the desired one, it is to the detriment of the parsimony of the model, since it leads to more complex representations (a large number of parameters). From the statistical point of view, a criterion based solely on the reduction of the error is not the most optimal, it is necessary a development oriented to the formalization of the model, which requires the fulfillment of certain properties that are not always taken into account, such as the stability of the calculated parameters, the coherence between the series and the model, the consistency with the previous knowledge and the predictive capacity of the model.

The evident interest in the use of neural networks in the prediction of time series has led to the emergence of an enormous research activity in the field. Crone and Kourentzes [7] reveal more than 5000 publications in prediction of time series with neural networks (see also publications [39, 44, 45]), and journals in fields with econometrics, statistics, engineering, and artificial intelligence, even being the central topic in special editions, such as the case of *Neurocomputing* with “*Special issue on evolving solution with neural networks*” published in October 2003 [46] and

the *International Journal of Forecasting* with “Special issue on forecasting with artificial neural networks and computational intelligence” published in 2011.

In order to establish the relevance of the prediction of time series with neural networks, a search was made through *Science Direct* of the Journals that publish articles related to the topic. **Table 1** and **Figure 1** present a compilation of the 10 Journals with more publications and also relate the number of articles published in the years 2015–2019, 2010–2014, 2005–2009, 2000–2004 and 1999 and earlier, which is identified using keywords: (*Forecasting o Prediction, Neural Networks, and Time Series*).

An analysis of **Table 1** and **Figure 1** shows the following facts:

- The number of publications reported on the subject is increasing, being representative the drastic growth reported in the last 5 years (2015–2019), which is evident in all the magazines listed.
- There is a greater participation in journals pertaining to or related to the fields of engineering and artificial intelligence.
- Journals with high number of published articles, *Neurocomputing*, *Applied Soft Computing*, *Procedia Computer Science*, and *Expert Systems with Applications*, are closely related to the topic, both from contributions in the field of neural networks, and time series forecasting.

Many comparisons have been made between neural networks and statistical models in order to measure the prediction performance of both approaches. As stated by Zhang et al. [8]:

“There are many inconsistent reports in the literature on the performance of ANNs for forecasting tasks. The main reason is that a large number of factors including network structure, training method, and sample data may affect the forecasting ability of the networks.”

Journal	Articles identified using keywords (forecasting or prediction, neural networks, and time series)					Total
	2015–2019	2010–2014	2005–2009	2000–2004	1999 and antes	
<i>Energy</i>	308	83	17	3	2	413
<i>Applied energy</i>	297	90	10	4	—	401
<i>Neurocomputing</i>	254	148	88	46	37	573
<i>Renewable and sustainable energy reviews</i>	241	74	14	—	1	330
<i>Applied soft computing</i>	238	132	41	5	—	416
<i>Journal of hydrology</i>	233	166	102	34	5	540
<i>Expert systems with applications</i>	226	364	188	20	11	809
<i>Procedia computer science</i>	212	77	—	—	—	289
<i>Renewable energy</i>	191	49	22	5	3	270
<i>Energy procedia</i>	155	41	—	—	—	196
	2355	1224	482	117	59	4237

Table 1. Journals that publish time series forecast articles with neural networks.

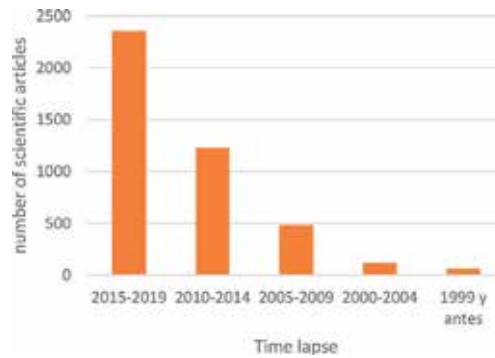


Figure 1.
Published articles for forecasting time series with neural networks.

Such inconsistencies make neural networks an unstable method, given that any change in training or in some parameter produces large changes in prediction [9]. Some key factors where mixed results are presented are:

- Need for data preprocessing (scaling, transformation, simple and seasonal differentiation, etc.) [10–12, 47, 48].
- Criteria for the selection of input variables [15, 22].
- Criteria for the selection of the network configuration. Complexity vs. Parsimony (number of internal layers [40–42], neurons in each layer [22]).
- Estimation of the parameters (learning algorithms, stop criteria, etc.).
- Criteria for selecting the best model [43].
- Diagnostic tests and acceptance.
- Tests on the residuals. Consistency of linear tests.
- Properties of the model: stability of the parameters, mean and variance series versus model.
- Predictive capacity of the model.
- Presence of regular patterns such as: trends, seasonal, and cyclical patterns [10–12].
- Presence of irregular patterns such as: structural changes, atypical data, effect of calendar days, etc. [3, 49, 50].

Cases where the neural network presents a worse performance than linear statistical models or other models, may be due to the fact that the series studied do not present a great disturbance, that the neural network used to compare was not adequately trained, that the criterion of selection of the best model is not comparable, or that the configuration used is not adequate to the characteristics of the data. Many conclusions about the performance of neural networks are obtained from empirical studies, thus presenting limited results that often cannot be extended to general applications.

However, there are few systematic researches about the modeling and prediction of time series with neural networks and the theoretical advances obtained [13], and this is perhaps the primary cause of the inconsistencies reported in the literature.

Many of the optimistic publications that indicate superior performance of neural networks are related to novel paradigms or extensions of existing methods, architectures and training algorithms, but lack a reliable and valid evaluation of the empirical evidence of their performance. Few contributions have been made in the systematic development of methodologies that allow representing time series with neural networks on specific conditions, limiting the modeling process to ad-hoc techniques, instead of scientific approaches that follow a methodology and replicable modeling process. A consequence of this is that, despite the empirical findings, neural network models are not fully accepted in many forecast areas. The previous discussion leads us to think that, although progress has been made in the field, there are still topics open to investigate. The question of whether, because, and on what conditions the models of neural networks are better is still valid.

3. Difficulties in the prediction of time series with neural networks

The design of an artificial neural network is intended to ensure that for certain network inputs, it is capable of generating a desired output. For this, in addition to a suitable network topology (architecture), a learning or training process is required, which allows modifying the weights of the neurons until finding a configuration according to the relationship measured by some criterion and thus estimating the parameters of the network, a process that is considered critical in the field of neural networks [8, 43]. Model selection is not a trivial task in forecasting linear models and is particularly difficult in non-linear models, such as neural networks. Because the set of parameters to be estimated is typically large, neural networks often suffer from over-training problems. That is, they fit the training data very well but produce poor results in the forecast.

To mitigate the effect of over-training, the available data set is often divided into three parts: training, validation, and testing or prediction. The training and validation sets are used to build the neural network model and then be evaluated with the test set. The training set is used to estimate the parameters of an alternative number of neural network specifications (networks with different numbers of inputs and hidden neurons). The generalization capacity of the network is evaluated with the validation set. The network model that performs best in the validation set is selected as the final model. The validity and utility of the model is then tested using the test set. Often this last set is used for forecasting purposes, and the network's generalization capacity for unknown data is evaluated.

The criterion of selecting the model based on the best performance of the validation set, however, does not guarantee that the model has a good fit in the forecast set, and the selection of the appropriate amount of data in each set can also affect performance. This is how a large training set can lead to over-training. Granger [21] suggests that at least 20% of the data be used as a test set; however, there is no general guide on how to partition the set of observations, so that optimal results are guaranteed.

Zhang et al. [22] states that the size of the training set has limited effects on the performance of the network, where, for the sizes investigated by the authors, there is no significant difference in the performance of the forecast. These results are perhaps due to the forecasting method used, with little difference for prediction one step ahead, and marked for multi-step forecast, in which case large differences in the results are expected in the case of different sizes of the training, validation, and test sets.

Although, as a criterion for the selection of the best model, the minimization of some error function is often used, such as mean square error (MSE), absolute average deviation (MAD), cost functions [51], or even expert knowledge [52], because the performance of each measure is not the same, since they can favor or penalize certain characteristics in the data, and that, in the case of expert knowledge is not always easy to acquire; approaches based on the use of machine learning [53, 54] and meta-learning [55–59] have been reported in the literature, which show advantages by allowing an automatic process of model selection based on the parallel evaluation of multiple network architectures, but they are limited to the execution of certain architectures and their implementation is complex. Other studies related to the topic include Qi and Zhang [43] who investigate the well-known criteria of AIC [60], BIC [61], square root of the mean square error (RMSE), absolute average percentage deviation (MAPE), and direction of occurrence (DA). The amplified panorama of the techniques for selecting the best model reflects that, despite the effort made, there is not a strong criterion for adequate selection.

Another widespread criticism that is often made to neural networks is the high number of parameters that must be experimentally selected to generate the desired output, such as: the selection of input variables to the neural network from a usually large set of possible entries; the selection of the internal architecture of the network; and the estimation of the values associated with the weights of the connections. For each of the problems mentioned, different approaches to its solution have been proposed in the literature.

The selection of the input variables depends to a large extent on the knowledge that the modeler possesses about the time series, and it is the task of the latter to choose according to some previously fixed criterion the need of each variable within the model. Although there is no systematic way to determine the set of inputs accepted by the research community, recent studies have suggested the use of rational procedures, based on the use of decisional analysis, or traditional statistical methods, such as autocorrelation functions [62]; however, the use of the latter is disregarded since the functions are based on linear approaches and not neural networks do not express by themselves the components of moving averages (MA) of the model. Mixed results about the benefits of including many or few input variables are also reported in the literature. Tang et al. [63] report the benefits of using a large set of input variables, while Lachtermacher and Fuller [15] report the same results for multistep forecasting, but opposed in forecasting a step forward. Zhang et al. [22] said that the number of input variables in a neural network model for prediction is much more important than the number of hidden neurons. Other techniques based on heuristic analysis of the importance of each lag, statistical tests of non-linear dependence Lagrange multipliers, [64, 65]; radio of likelihood, [66]; Biespectro, [67], criteria for identifying the model, such as AIC [5] or evolutionary algorithms [68, 69] they have also been proposals.

The selection of the internal configuration of the neural network (number of hidden layers and neurons in each layer) is perhaps the most difficult process in the construction of the model where more different approaches have been proposed in the literature, demonstrating in this way the interest of the scientific community to solve this problem.

Regarding the number of hidden layers, theoretically a neural network with a hidden capacity and a sufficient number of neurons can approximate the accuracy of a continuous function in a compact domain. However, in practice, some authors say that the use of a hidden layer when the time series is continuous, and twice if there is some type of discontinuity [41, 42]. However, other research has shown that a network with two hidden layers can result in a more compact architecture and with a high efficiency than networks with a single hidden layer [70–72]. Increasing the number of hidden layers only increases computational time and the danger of overtraining.

With respect to the number of hidden neurons, a small number means that the network cannot adequately learn the relationships in the data, while a large number causes the network to memorize the data with a poor generalization and little utility for prediction. Some authors propose that the number of hidden neurons should be based on the number of input variables; however, this criterion is in turn related to the extension of the time series and the sets of training, validation, and prediction. Given that the value of the weights in each neuron depends on the degree of error between the desired value and that predicted by the network, the selection of the optimal number of hidden neurons is directly associated with the training process used.

The training of a neural network is a problem of non-restricted non-linear minimization in which the weights of the network are iteratively modified in order to minimize the error between the desired output and the obtained one. Several methods have been proposed in the literature for the training of the neural network, going through the classical gradient descendant techniques [73], which have convergence problems and are robust, adaptive dynamic optimization [74, 75], Quickprop [76], Levenberg–Marquardt [77], Cuasi-Newton, BFGS, GRG2 [78], among others. However, the joint selection of hidden neurons and the training process has led to the development of fixed, constructive, and destructive methods, where those based on constructive algorithms have certain advantages over others, since they allow evaluating the convenience of adding or not adding a new one. Neuron to the network, during training, according to it decreases the term of the error, which makes them more efficient methods, although with high computational cost [79]. Other developments such as pruning algorithms (*pruning algorithm*) [77, 80–82], Bayesian algorithms, based on Genetic algorithms as the GANN, neural networks with rugged assemblies, assembled learning [83–86], and meta-learning [9, 87] have also shown good results in the task of finding the optimal architecture of the network; however, these methods are usually more complex and difficult to implement. Furthermore, none of them can guarantee to find the optimal global solution and they are not universally applicable for all real forecasting problems, thus designing a proper neural network.

The efficiency of the prediction with neural networks has been evidenced through the applications published in the literature; however, the power of the prediction produced is limited to the degree of stability of the time series and can fail when it presents complex dynamic behaviors. This is how representations that use dynamic character models, such as neural networks with recurrence Elman, Jordan, etc., emerge as an alternative solution [88–91], which due to the possibility of accumulating dynamic behaviors are able to allow more adequate forecasts. The recurrence feature allows forward and backward connections (recurrent or feedback), forming cycles within the network architecture, which uses previous states as a basis for the current state, and allowing to preserve an internal memory of the behavior of the data, which facilitates the learning of dynamic relationships. However, their main criticism lies in the need they impose an efficient training algorithm that allows them to capture the dynamics of the series, its use being computationally complex. Potentially useful models to address the problem of series with dynamic behavior arise from the combination of different architectures in the input of the multilayer perceptron.

The problem that arises goes beyond the simple estimation of each model in light of the characteristics of each series. Although it is recognized that there is much experience gained in multilayer perceptron neural networks, there are still many theoretical, methodological, and empirical problems open about the use of such models. These general problems are related to the aspects listed below, and for which many of the recommendations given in the literature are contradictory [92–98]:

- There is no systematic way accepted in the literature to determine the appropriate set of inputs to the neural network.
- There is no general guide to partition the set of observations in training, validation and forecast, in such a way that optimal results are guaranteed.
- The effects that factors such as partition in training sets, validation and forecast, preprocessing, transfer function, etc., in different forecasting methods are unknown or unclear.
- There are no clear indications that allow to express a priori which transfer function should be used in the neural network model according to the characteristics of the time series.
- There is no clarity about procedures oriented to the selection of neurons in the hidden layer that in turn allow to minimize the training time of the network.
- There are no empirical, methodological or theoretical reasons to prefer a specific model among several alternatives.
- There is no agreement on how to select the final model when several alternatives are considered.
- It is not clear when and how to transform the data before performing the modeling.
- There is no clarity about the necessity of eliminating or not eliminating trend and seasonal components in neural network models.
- It is difficult to incorporate qualitative, subjective, and contextual information in the forecasts.
- There is little understanding of the statistical properties of different neural network architectures.
- There is no clarity about which are the most adequate procedures for the estimation, validation, and testing of different neural network architectures.
- There is no clarity on how to combine forecasts from several alternative models, and if there are gains derived from this practice.
- There are no or no clarity in the criteria for evaluating the performance of different neural network architectures.
- There is no clarity about whether and under what criteria, different architectures of neural networks allow the handling of dynamic behaviors in the data.

4. Conclusions

In this chapter, the need to have adequate models of neural networks for the prediction of time series has been identified, and this task has been exposed as a difficult, relevant, and timely problem.

A critical step in the forecast process is the selection of the set of input variables. At this point the decision of which lags of the series to include is fundamental for the result and depends on the available information and knowledge. Obviating the need for prior knowledge about the series, the choice of candidate lags to be included in the model should be based on a heuristic analysis of the importance of each lag, a statistical test of non-linear dependence, criteria for identifying the model or evolutionary algorithms, however, before such options the mixed results reported in the literature show that there is no consensus about what is the appropriate procedure for this purpose.

As previously emphasized, in the literature there are no clear indications about the best practices for choosing the size of the training, test, and prediction sets. Often the size is a predefined parameter in the construction of the neural network model or it is chosen randomly; however, there is no study that demonstrates the effect that this decision entails, moreover, this may be related to the forecasting method used.

Likewise, there is a close relationship between the selection of the internal configuration, especially the hidden neurons, and the training process of the neural network. The consensus about the use of a hidden layer when the data of the time series are continuous and two when there is discontinuity, and of the advantages of the functions sigmoidia and hyperbolic tangent in the transfer of the hidden layer, reflects a deep investigation of such topics.

It is often used as a criterion for the selection of the best model based on the error of prediction, expert knowledge or criteria of information; however, the limitations that they manifest and the mixed results reported in their use, in addition to the limited results reported with other techniques, which do not allow conclusive conclusions about their use.

The consideration of characteristic factors of the time series that can affect the evolution of the neural network model such as the length of the time series, the frequency of the observations, the presence of regular and irregular patterns, and the scale of the data, must be included in the process of building the neural network model. The discussion of whether a preprocessing oriented to the stabilization of the series is necessary in non-linear models, and even more, in neural networks, is a topic that is still valid, and depends to a large extent on the type of data that is modeled. The abilities exhibited by neural networks allow, in the first instance, to avoid pre-processing via data transformation. However, it is not yet clear whether, under a correct network construction and training procedure, a prior process of elimination of seasonal trends and patterns is necessary. Scaling is always preferable given its advantages of reducing training patterns and leading to more accurate results.

Likewise, the benefits that different neural network architectures have in relation to nonlinear relationships in the data have been discussed. Neural network models, by themselves, facilitate the representation of non-linear characteristics, without the need for a priori knowledge about such relationships, and such consideration is always desirable in models for real time series; however, it is not. In addition, their performance in the face of dynamic behavior in the data, the exposed architectures have been developed as an extension of neural network models and not explicitly as time series models, so there is no theoretical foundation for the construction of these, nor rigorous studies that allow to assess their performance in time series with the stated characteristics.

Conflict of interest

The authors declare no conflict of interest.

Author details

Paola Andrea Sánchez-Sánchez*, José Rafael García-González
and Leidy Haidy Perez Coronell
Universidad Simón Bolívar, Barranquilla, Colombia

*Address all correspondence to: psanchez9@unisimonbolivar.edu.co

IntechOpen

© 2019 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/3.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. 

References

- [1] Zhang P. An investigation of neural networks for linear time-series forecasting. *Computers & Operations Research*. 2001;**28**(12):1183-1202
- [2] Granger C, Terasvirta T. *Modelling Nonlinear Economic Relationships*. Oxford: Oxford University Press; 1993
- [3] Franses P, Van Dijk D. *Non-Linear Time Series Models in Empirical Finance*. UK: Cambridge University Press; 2000
- [4] Tong H. *Non-Linear Time Series: A Dynamical System Approach*. Oxford: Oxford Statistical Science Series; 1990
- [5] De Gooijer I, Kumar K. Some recent developments in non-linear modelling, testing, and forecasting. *International Journal of Forecasting*. 1992;**8**:135-156
- [6] Peña D. Second-generation time-series models: A comment on 'Some advances in non-linear and adaptive modelling in time-series analysis' by Tiao and Tsay. *Journal of Forecasting*. 1994;**13**:133-140
- [7] Crone S, and Kourentzes N. Input-variable Specification for Neural Networks - An Analysis of Forecasting low and high Time Series Frequency. *Proceedings of the International Joint Conference on Neural Networks, (IJCNN'09)*. in press. 2009
- [8] Zhang P, Patuwo B, Hu M. Forecasting with artificial neural networks: the state of the art. *International Journal of Forecasting*. 1998;**14**(1):35-62
- [9] Yu L, Wang S, Lai K. A neural-network-based nonlinear metamodeling approach to financial time series forecasting. *Applied Soft Computing*. 2009;**9**:563-574
- [10] Franses P, Draisma G. Recognizing changing seasonal patterns using artificial neural networks. *Journal of Econometrics*. 1997;**81**(1):273-280
- [11] Qi M, Zhang P. Trend time-series modeling and forecasting with neural networks. *IEEE Transactions on Neural Networks*. 2008;**19**(5):808-816
- [12] Zhang P, Qi M. Neural network forecasting for seasonal and trend time series. *European Journal of Operational Research*. 2005;**160**:501-514
- [13] Trapletti A. *On Neural Networks as Time Series Models*. Universidad Técnica de Wien; 2000
- [14] Kasabov N. *Foundations of Neural Networks, Fuzzy Systems, and Knowledge Engineering*. 2nd ed. Massachusetts: The MIT Press Cambridge; 1998
- [15] Lachtermacher G, Fuller J. Backpropagation in time-series forecasting. *Journal of Forecasting*. 1995;**14**:381-393
- [16] Meade N. Evidence for selection of forecasting methods. *Journal of Forecasting*. 2000;**19**:515-535
- [17] Granger C, Anderson A. *An Introduction to Bilinear Time Series Models*. Gottingen: Vandenhoeck and Ruprecht; 1978
- [18] Tong H, Lim K. Threshold autoregressive, limit cycles and cyclical data. *Journal of the Royal Statistical Society, Series B*. 1980;**42**(3):245-292
- [19] Engle R. Autoregressive conditional heteroskedasticity with estimates of the variance of UK inflation. *Econometrica*. 1982;**50**:987-1008
- [20] Bollerslev T. Generalised autoregressive conditional heteroscedasticity. *Journal of Econometrics*. 1986;**31**:307-327

- [21] Granger C. Strategies for modelling nonlinear time-series relationships. *Economic Record*. 1993;**69**(206):233-238
- [22] Zhang P, Patuwo E, Hu M. A simulation study of artificial neural networks for nonlinear time-series forecasting. *Computers and Operations Research*. 2001;**28**(4):381-396
- [23] Chatfield C. What is the “best” method of forecasting? *Journal of Applied Statistics*. 1988;**15**:19-39
- [24] Jenkins G. Some practical aspects of forecasting in organisations. *Journal of Forecasting*. 1982;**1**:3-21
- [25] Makridakis S, Anderson A, Carbone R, Fildes R, Hibon M, Lewandowski R, et al. The accuracy of extrapolation (time series) methods: Results of a forecasting competition. *Journal of Forecasting*. 1982;**1**:111-153
- [26] Clemen R. Combining forecasts: A review and annotated bibliography with discussion. *International Journal of Forecasting*. 1989;**5**:559-608
- [27] Makridakis S, Chatfield C, Hibon M, Lawrence M, Mills T, Ord K, et al. The M2 competition: A real-time judgmentally based forecasting competition. *Journal of Forecasting*. 1993;**9**:5-22
- [28] Newbold P, Granger C. Experience with forecasting univariate time series and the combination of forecasts (with discussion). *Journal of the Royal Statistical Society*. 1974;**137**:131-164
- [29] Granger C. Combining forecasts- twenty years later. *Journal of Forecasting*. 1989;**8**:167-173
- [30] Krogh A, Vedelsby J. Neural network ensembles, cross validation, and active learning. *Advances in Neural Information Processing Systems*. 1995;**7**:231-238
- [31] Chatfield C. Model uncertainty and forecast accuracy. *Journal of Forecasting*. 1996;**15**:495-508
- [32] Bates J, Granger C. The combination of forecasts. *Operational Research Quarterly*. 1969;**20**:451-468
- [33] Davison M, Anderson C, Anderson K. Development of a hybrid model for electrical power spot prices. *IEEE Transactions on Power Systems*. 2002;**2**:17
- [34] Luxhoj J, Riis J, Stensballe B. A hybrid econometric-neural network modeling approach for sales forecasting. *International Journal of Production Economics*. 1996;**43**:175-192
- [35] Makridakis S. Why combining works? *International Journal of Forecasting*. 1989;**5**:601-603
- [36] Palm F, Zellner A. To combine or not to combine? issues of combining forecasts. *Journal of Forecasting*. 1992;**11**:687-701
- [37] Reid D. Combining three estimates of gross domestic product. *Economica*. 1968;**35**:431-444
- [38] Winkler R. Combining forecasts: A philosophical basis and some current issues. *International Journal of Forecasting*. 1989;**5**:605-609
- [39] Zhang P. Time series forecasting using a hybrid ARIMA and neural network model. *Neurocomputing*. 2003;**50**:159-175
- [40] Cybenko G. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems*. 1989;**2**:303-314
- [41] Hornik K. Approximation capability of multilayer feedforward networks. *Neural Networks*. 1991;**4**:251-257
- [42] Hornik K, Stinchcombe M, White H. Multilayer feedforward

- networks are universal approximators. *Neural Networks*. 1989;2(5):359-366
- [43] Qi M, Zhang P. An investigation of model selection criteria for neural network time series forecasting. *European Journal of Operational Research*. 2001;132:666-680
- [44] Adya M, Collopy F. How effective are neural networks at forecasting and prediction? A review and evaluation. *Journal of Forecasting*. 1998;17:481-495
- [45] Hill T, O'Connor M, Remus W. Neural network models for time series forecasts. *Management Science*. 1996;42:1082-1092
- [46] Fanni A, Uncini A. Special issue on evolving solution with neural networks. *Neurocomputing*. 2003;55(3-4):417-419
- [47] Faraway J, Chatfield C. Time series forecasting with neural networks: a comparative study using the airline data. *Applied Statistics*. 1998;47:231-250
- [48] Nelson M, Hill T, Remus T, O'Connor M. Time series forecasting using NNs: Should the data be deseasonalized first? *Journal of Forecasting*. 1999;18:359-367
- [49] Hill T, Marquez L, O'Connor M, Remus W. Artificial neural networks for forecasting and decision making. *International Journal of Forecasting*. 1994;10:5-15
- [50] Tkacz G, Hu S. Forecasting GDP Growth Using Artificial Neural Networks. Bank of Canada; 1999
- [51] Tashman L. Out-of-sample tests of forecasting accuracy: An analysis and review. *International Journal of Forecasting*. 2000;16:437-450
- [52] Adya M, Collopy F, Armstrong J, Kennedy M. Automatic identification of time series features for rule-based forecasting. *International Journal of Forecasting*. 2001;17(2):143-157
- [53] Arinze B. Selecting appropriate forecasting models using rule induction. *Omega-International Journal of Management Science*. 1994;22(6):647-658
- [54] Venkatachalan A, Sohl J. An intelligent model selection and forecasting system. *Journal of Forecasting*. 1999;18:167-180
- [55] Giraud-Carrier R, Brazdil P. Introduction to the special issue on meta-learning. *Machine Learning*. 2004;54(3):187-193
- [56] Santos P, Ludermir T, Prudencio R. Selection of time series forecasting models based on performance information. In: *Proceedings of the 4th International Conference on Hybrid Intelligent Systems*. 2004. pp. 366-371
- [57] Santos P, Ludermir T, Prudencio R. Selecting neural network forecasting models using the zoomed-ranking approach. In: *Proceedings of the 10th Brazilian Symposium on Neural Networks SBRN '08*. 2008. pp. 165-170
- [58] Soares C, Brazdil P. Zoomed Ranking – Selection of classification algorithms based on relevant performance information. *Lecture Notes in Computer Science*. 1910;2000:126-135
- [59] Vilalta R, Drissi Y. A perspective view and survey of meta-learning. *Journal of Artificial Intelligence Review*. 2002;18(2):77-95
- [60] Akaike H. A new look at statistical model identification. *IEEE Transactions on Automatic Control*. 1974;9:716-723
- [61] Schwarz G. Estimating the dimension of a model. *The Annals of Statistics*. 1978;6:461-464

- [62] Tang Z, Fishwick P. Feedforward neural nets as models for time series forecasting. *ORSA Journal on Computing*. 1993;5(4):374-385
- [63] Tang Z, Almeida C, Fishwick P. Time series forecasting using neural networks vs Box-Jenkins methodology. *Simulation*. 1991;57(5):303-310
- [64] Luukkonen R, Saikkonen P, Terasvirta T. Testing linearity in univariate time series models. *Scandinavian Journal of Statistics*. 1988;15:161-175
- [65] Saikkonen P, Luukkonen R. Lagrange multiplier tests for testing non-linearities in time series models. *Scandinavian Journal of Statistics*. 1988;15:55-68
- [66] Chan W, Tong H. On tests for non-linearity in time series analysis. *Journal of Forecasting*. 1986;5:217-228
- [67] Hinich M. Testing for Gaussianity and linearity of a stationary time series. *Journal of Time Series Analysis*. 1982;3:169-176
- [68] Happel B, Murre J. The design and evolution of modular neural network architectures. *Neural Networks*. 1994;7:985-1004
- [69] Schiffmann W, Joost M, Werner R. Application of genetic algorithms to the construction of topologies for multilayer perceptron. In: *Proceedings of the International Conference on Artificial Neural Networks and Genetic Algorithms*. 1993. pp. 675-682
- [70] Srinivasan D, Liew A, Chang C. A neural network short-term load forecaster. *Electric Power Systems Research*. 1994;28:227-234
- [71] Zhang X. Time series analysis and prediction by neural networks. *Optimization Methods and Software*. 1994;4:151-170
- [72] Chester D. Why two hidden layers are better than one. In: *Proceedings of the International Joint Conference on Neural Networks*. 1990. pp. 1265-1268
- [73] Bishop C. *Neural Networks for Pattern Recognition*. Oxford University Press; 1995
- [74] Pack D, El-Sharkawi M, Marks R, Atlas L. Electric load forecasting using an artificial neural network. *IEEE Transactions on Power Systems*. 1991;6(2):442-449
- [75] Yu X, Chen G, Cheng S. Dynamic learning rate optimization of the backpropagation algorithm. *IEEE Transactions on Neural Networks*. 1995;6(3):669-677
- [76] Falhman S. Faster-learning variations of back-propagation: An empirical study. In: *de Proceedings of the 1988 Connectionist Models Summer School*. 1989. pp. 38-51
- [77] Cottrell M, Girard B, Girard Y, Mangeas M, Muller C. Neural modeling for time series: a statistical stepwise method for weight elimination. *IEEE Transactions on Neural Networks*. 1995;6(6):1355-1364
- [78] Lasdon L, Waren A. *GRG2 User's Guide*. Austin: School of Business Administration, University of Texas; 1986
- [79] Weigend A, Rumelhart D, Huberman B. Generalization by weight-elimination with application to forecasting. *Advances in Neural Information Processing Systems*. 1991;3:875-882
- [80] Karnin E. A simple procedure for pruning back-propagation trained neural networks. *IEEE Transactions on Neural Networks*. 1990;1(2):239-245

- [81] Reed R. Pruning algorithms a survey. *IEEE Transactions on Neural Networks*. 1993;**4**:740-747
- [82] Siestema J, Dow R. Neural net pruning – why and how. In: *Proceedings of the IEEE International Conference on Neural Networks*. Vol. 1. 1998. pp. 325-333
- [83] Breiman L. *Combining predictors de Combining Artificial Neural Nets—Ensemble and Modular Multi-Net Systems*. Berlin: Springer; 1999. pp. 31-50
- [84] Carney J, Cunningham P. Tuning diversity in bagged ensembles. *International Journal of Neural Systems*. 2000;**10**:267-280
- [85] Hansen L, Salamon P. Neural network ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 1990;**12**:993-1001
- [86] Naftaly U, Intrator N, Horn D. Optimal ensemble averaging of neural networks. *Network: Computation in Neural Systems*. 1997;**8**:283-296
- [87] Chan P, Stolfo S. Metalearning for multistrategy and parallel learning. In: *Proceedings of the Second International Workshop on Multistrategy Learning*. 1993. pp. 150-165
- [88] Connor J, Atlas L, Martin D. *Recurrent Networks and NARMA Modeling de Advances in Neural Information Processing Systems*. Morgan Kaufmann Publishers, Inc. 1991;**119**:301-308
- [89] Kuan C, Liu T. Forecasting exchange rates using feedforward and recurrent neural networks. *Journal of Applied Econometrics*. 1995;**10**:347-364
- [90] Najand M, Bond C. Structural models of exchange rate determination. *Journal of Multinational Financial Management*. 2000;**10**:15-27
- [91] Tenti P. Forecasting foreign exchange rates using recurrent neural networks. *Applied Artificial Intelligence*. 1996;**10**:567-581
- [92] Caire P, Hatabian G, Muller C. Progress in forecasting by neural networks. In: *Proceedings of the International Joint Conference on Neural Networks*. Vol. 2. 1992. pp. 540-545
- [93] Ong P, Zainuddin Z. Optimizing wavelet neural networks using modified cuckoo search for multi-step ahead chaotic time series prediction. *Applied Soft Computing*. 2019;**80**:374-386
- [94] Zhanga Y, Wanga X, Tang H. An improved Elman neural network with piecewise weighted gradient for time series prediction. *Neurocomputing*. 2019;**359**:199-208
- [95] Wang L, Wang Z, Qu H, Liu S. Optimal forecast combination based on neural networks for time series forecasting. *Applied Soft Computing*. 2018;**66**:1-17
- [96] Lopez-Martin M, Carro B, Sanchez-Esguevillas A. Neural network architecture based on gradient boosting for IoT traffic prediction. *Future Generation Computer Systems*. 2019;**100**:656-673
- [97] Zurbarán M, Sanmartin P. Efectos de la Comunicación en una Red Ad-Hoc. *Investigación e Innovación en Ingenierías*. 2016;**4**(1):26-31
- [98] Tealab A. Time series forecasting using artificial neural networks methodologies: A systematic review. *Future Computing and Informatics Journal*. 2018;**3**(2):334-340

Section 2

Metaheuristics and Artificial Neural Networks

Electric Transmission Network Expansion Planning with the Metaheuristic Variable Neighbourhood Search

*Silvia Lopes de Sena Taglialenha
and Rubén Augusto Romero Lázaro*

Abstract

This paper presents a new method to solve the static long-term power transmission network expansion planning (TNEP) problem that uses the metaheuristic variable neighbourhood search (VNS). The TNEP is a large-scale, complex mixed-integer nonlinear programming problem that consists of determining the optimum expansion in the network to meet a forecasted demand. VNS changes structure neighbourhood within a local algorithm and makes the choices of implementation that integrate intensification and/or diversification strategies during the search process. The initial solution is obtained by a heuristic nonlinear mixed integer which takes two Kirchhoff's laws (transportation and the DC models have been used). Several tests are performed on Graver's 6-bus, IEEE 24-bus and Southern Brazilian systems displaying the applicability of the proposed method, and results show that the proposed method has a significant performance in comparison with some studies addressed in common literature.

Keywords: transmission network expansion planning, variable neighbourhood search algorithm, metaheuristic algorithm, power system planning, combinatorial optimization

1. Introduction

Due to consumption growth of electrical power, the need of increasing the existing transmission network power flow capacity is evident. This expansion can be a dynamic or static performance. The static long-term power transmission network expansion planning (TNEP) problem consists of determining the minimum cost planning which specifies the number and the locations of transmission lines to meet a forecasted demand while satisfying the balance between generation and load and other operational constraints [1]. Transmission investments are very capital intensive and have long useful lives, so transmission investment decisions have a long-standing impact on the power system as a whole; therefore TNEP has become an important component of power system planning, and its solution is used to guide future investment in transmission equipment.

The pioneering work on transmission expansion planning is reported in [2], and since then TNEP literature has been vast and reports that there are usually considered various solution methods that depend on the mathematical model formulation [3]. A state of the art, which was obtained from the review of the most interesting models found in the international technical literature, is presented in [4]. In [5] TNEP is reviewed from different aspects such as modelling, solving methods, reliability, distributed generation, electricity market, uncertainties, line congestion and reactive power planning. A critical review focusing on its most recent developments and a taxonomy of modelling decisions and solution to TNEP are presented in [6].

The convenient mathematical modelling to indicate the appropriate operation would be the representation of the problem by mathematical relationships of the AC load flow, typically used for the electric system operation analysis [1]. However, this modelling is more difficult to be used in an efficient way in transmission network planning, due to its non-convex and nonlinear nature. Consequently, the mathematical modelling in its most accurate representation is the direct current (DC) model, which considers Kirchhoff's voltage (KVL) and current (KCL) laws just for balance and active power flow. In this case, the resulting problem is a nonlinear mixed-integer programming with high complexity for large systems, presenting combinatorial explosion of the number of alternative solutions, with extra difficulty of presenting many local optima, which most of the time are of poor quality [3].

A more simplified modelling is the so-called transportation model (TM) which just enforces the KLC at all existent nodes [2]. In this case the resulting problem is an integer linear programming problem which is normally easier to solve than the DC model although it maintains the combinatorial characteristic of the original problem [3].

It is still possible to consider hybrid models which combine characteristics of the DC model and the transportation model. In this model it is assumed that KCL constraints are satisfied for all nodes of the network, whereas the constraint which represents Ohm's law (and indirectly KVL) is satisfied only by the existing circuits (and not necessarily by the added circuits) [3].

Technical literature related to the TNEP proposes many solution methods that can be classified into mathematical optimization, heuristic and metaheuristic approaches [7]. Techniques such as dynamic programming [8], linear programming [2], nonlinear programming [9], mixed-integer programming [10], branch and bound [11], hierarchical decomposition [12] and Benders decomposition [13] have been used and are categorized as mathematical-based approaches. But these techniques demand large computing time due to the dimensionality curse of this kind of problem. Heuristic methods emerged as an alternative to classical optimization methods, and their use has been very attractive since they were able to find good feasible solutions demanding less computational effort.

Some heuristic approaches have been proposed using constructive heuristic algorithms (CHA) [10, 14–16] and the forward-backward approach [17]. Metaheuristic methods emerged as an alternative to the two previous approaches, producing high-quality solutions with moderate computing time. Genetic algorithms [18, 19], greedy randomized adaptive search procedure [13], tabu search [20, 21], simulated annealing [20, 22], GRASP [23], scatter search [24] and grey wolf optimization algorithm [25] have been used to solve the TNEP problem, among other metaheuristic optimization techniques. It is important to point out that they cannot guarantee the global optimal solution to the TNEP problem.

A varied bibliography regarding the theory and application of metaheuristics can be found in [26, 27]. Other applications of metaheuristics appear in [28].

Considering that exact methods of optimization to TNEP are not efficient to big data problems, this paper presents a novel metaheuristic method that considers the so-called variable neighbourhood search (VNS) to solve the TNEP problem considering DC model. The VNS metaheuristic was presented in the middle of the 1990s, by Mladenovic and Hansen [29], and represents a significantly different proposal compared to other metaheuristics. The fundamental idea of the VNS algorithm is based on a basic principle: to explore the space of solutions by systematic changes of neighbourhood structures during the search process. Thus, the transition through the search space of the problem is always accomplished with an improvement of the objective function, and, therefore, the transition is not allowed for a solution of worse quality as occurs with most of the metaheuristics [29].

The VNS algorithm was used with success in the optimization of several problems of operational research [26, 27, 29, 30], but it is still insignificant in the optimization of problems related to the operation and the planning of electric power systems. The VNS was used to TNEP considering transportation model in [31, 32].

This paper is organized as follows: Initially the mathematical model for TNEP problem and the VNS metaheuristic are presented. After, the developed VNS algorithm to solve the TNEP problem is described. Later, obtained results are presented and commented. Finally, conclusions are drawn.

2. Mathematical model of TNEP

The mathematical formulation of the TNEP for the DC model is given by Eqs. (1)–(8) and performs as a nonlinear mixed-integer programming problem [3]:

$$\text{Min } v = \sum_{i,j \in \Omega} c_{ij} \cdot n_{ij} \quad (1)$$

$$AF + G = D \quad (2)$$

$$f_{ij} - \gamma_{ij} (n_{ij}^0 + n_{ij}) (\theta_i - \theta_j) = 0 \quad (3)$$

$$|f_{ij}| \leq (n_{ij}^0 + n_{ij}) \bar{f}_{ij} \quad (4)$$

$$0 \leq g \leq \bar{g} \quad (5)$$

$$0 \leq n_{ij} \leq \bar{n}_{ij} \quad (6)$$

$$n_{ij} \geq 0 \text{ and integer } \forall (i,j) \in \Omega \quad (7)$$

$$f_{ij}, \theta_j \text{ unbounded } \forall (i,j) \in \Omega \quad (8)$$

where v is the total investment value for a predefined horizon; c_{ij} is the cost of a circuit or facility that can be added in the branch (i,j) ; n_{ij} is the number of circuits added during the optimization process; n_{ij}^0 is the number of existing circuits in the initial topology; γ_{ij} is the susceptance of the branch (i,j) ; θ_i is phase angle at the bus i ; F is the vector of power flow with components f_{ij} ; \bar{f}_{ij} is the transmission capacity of a circuit through branch (i,j) ; A is the transposed incidence branch-node matrix of the power system; G is a vector with elements g_k (power generation at bus k) with maximum values \bar{g}_k ; \bar{n}_{ij} is the maximum number of circuits that can be added to the branch (i,j) ; Ω is the set of all branches where it is possible to add new circuits.

Eq. (1) that contains the sum of the investments costs is the objective function. The KCL is framed in Eq. (2), and the Ohm's law is expressed in Eq. (3) which

implicitly takes into consideration Kirchhoff's voltage law (KVL). Inequalities Eq. (4) represent capacity constraints for transmission lines, whereas the absolute value is necessary since power can flow in both directions. Other constraints Eqs. (6)–(8) represent operational limits of the generators, maximum limit for the addition of circuits per branch and integrality demand of the variables n_{ij} , respectively.

The model Eqs. (1)–(8) cannot be solved by using traditional algorithms, and there is no efficient method for solving these kinds of problems directly. Therefore, metaheuristics become suitable optimization tools for finding optimal and suboptimal solutions for the TNEP problem when it is considered complex power systems (big instances).

A more simplified model called the transport model can be considered, which contemplates only Kirchhoff's current law and could be obtained by relaxing the nonlinear constraint Eq. (3) of the DC model described above [3]. In this case, the resulting model is an integer linear programming problem. Even though it is linear, it is still very difficult to find the optimal solution for large and complex systems. The transport model was the first systematic proposal of mathematical modelling used with great success in the problem of planning of transmission systems. The model was proposed by Garver [2] and has represented the beginning of systematic research in the area of transmission system planning.

Another model that has been considered for the PPEST is the linear hybrid model (LHM) which combines characteristics of the DC model and the transport model. This model, in a simpler formulation, preserves the linear properties of the transport model, considering Kirchhoff's current law in all nodes of the network and KVL only in the circuits in the base network (not necessarily in the circuits that will be added) [3, 10]. The LHM is framed by Eqs. (9)–(17):

$$\text{Min } v = \sum_{i, j \in \Omega} c_{ij} \cdot n_{ij} \quad (9)$$

$$AF + A^0 F^0 + G = D \quad (10)$$

$$f_{ij}^0 - \gamma_{ij} n_{ij}^0 (\theta_i - \theta_j) = 0, \forall (i, j) \in \Omega_0 \quad (11)$$

$$|f_{ij}^0| \leq (n_{ij}^0) \bar{f}_{ij}, \forall (i, j) \in \Omega_0 \quad (12)$$

$$|f_{ij}| \leq (n_{ij}^0 + n_{ij}) \bar{f}_{ij}, \forall (i, j) \in \Omega \quad (13)$$

$$0 \leq g \leq \bar{g} \quad (14)$$

$$0 \leq n_{ij} \leq \bar{n}_{ij}, \forall (i, j) \in \Omega \quad (15)$$

$$f_{ij}^0 \text{ unbounded}, \forall (i, j) \in \Omega_0 \quad (16)$$

$$f_{ij}, \theta_j \text{ unbounded}, \forall (i, j) \in \Omega \quad (17)$$

where A^0 is the transposed incidence branch-node matrix of the base topology in previous iterations of the algorithm system; F^0 is the vector of base power flow with components f_{ij}^0 ; n_{ij}^0 is the circuits added during the iterative process to the base case; Ω_0 is the set of all the circuits added during the iterative process and all of the prime circuits of the base case.

The LHM was originally proposed in [10] whose authors present a mathematical modelling Eqs. (9)–(17) which specifies that the portion of the electric system corresponding to the circuits existing in the base configuration must satisfy the two Kirchhoff's laws and the other corresponding part from new circuits must satisfy only Kirchhoff's current law.

The LHM Eqs. (9)–(17) will be considered as a sensitivity indicator to the proposed heuristic algorithm.

3. Metaheuristic VNS

A metaheuristic is a search strategy that orchestrates an interaction between local improvement procedures and higher local strategies to create a process capable of escaping from local optima and performing a robust optimization method for complex problems. This search is performed by means of transitions in the search space from an initial solution or a set of initial solution. In this context, the main difference among the diverse metaheuristic techniques is the strategy used to carry out the transitions within the search space. VNS is a metaheuristic that systematically exploits the idea of neighbourhood change to find local-optimal solutions and to leave those local optima. In that fundamental aspect, VNS is significantly different from other metaheuristics. Most metaheuristics accept the degradation of the current solution as a strategy to leave a local-optimal solution. The VNS algorithm does not accept this possibility [26].

The VNS algorithm changes the neighbourhood as a way of leaving local-optimal solutions. During this process, the current solution is also the incumbent, which does not happen with other metaheuristics. Thus, it is possible to state that the VNS algorithm performs a set of transitions in the search space of a problem and at each step this transition is performed for the new incumbent. If the process finds a local optimum, then the VNS algorithm changes the neighbourhood in order to leave from that local optimum and to achieve the new incumbent. As a consequence of this strategy, if the VNS algorithm finds the global optimum, the search stops at that point, eliminating any chance of leaving it. This behaviour does not occur with other metaheuristics.

The strategy of the VNS algorithm is inspired by three important facts [29]:

Fact 1—A minimum with regard to one neighbourhood structure is not necessary for another.

Fact 2—A global minimum is a local minimum with regard to all possible neighbourhood structures.

Fact 3—For many problems, a local minimum with regard to one or several neighbourhoods is relatively close to each other.

The latter is particularly important in the formulation of the VNS algorithm. This empirical fact implies that a local-optimal solution often provides important information regarding the global one, especially if the local-optimal solution presents excellent quality. It is also an empirical fact that local-optimal solutions are generally concentrated in specific regions of the search space. If local-optimal solutions were to be uniformly distributed in the search space, all metaheuristics would become inefficient. Consequently, if a local optimum is found in the same region where the global optimum is, then the VNS metaheuristic has better chances of finding this global optimum. On the other hand, if the global optimum pertains to another region, then the only possibility to find it is to implement a diversification process. For this reason, equilibrium between intensification and diversification during the search process can be important in a metaheuristic.

There is another important aspect related to the quality of the local optimum that should be part of the implementation logic of a VNS algorithm. A local optimum with a better-quality objective function is not necessarily more suitable for trying to find the global optimum. Let x_a and x_b be two local-optimal solutions with $f(x_a) < f(x_b)$ for the minimization problem. Considering the traditional analysis, it can be concluded that x_a is a local optimum with better quality than x_b .

If these solutions are to be used for initiating (or reinitiating) the search process, then it can be affirmed that the solution presenting internal characteristics closer to those of the global optimum is the most suitable for initiating (or reinitiating) the search and, consequently, solution should not necessarily be chosen.

Thus, for instance, considering the TNEP problem, the local-optimal solution with the largest number of n_{ij} elements equal to the optimal solution is the most appropriate for initiating (or reinitiating) the search. It is evident that in normal conditions, the optimal solution is unknown. However, there are some problems where the optimal solution is known, and there are also various heuristic algorithms to find local-optimal solutions for this problem.

In this way, the previous observation can be used to identify the heuristic algorithm that produces best-quality local-optimal solutions for initiating the search using the VNS algorithm. This type of behaviour occurs in the TNEP problem where for some instances (power systems) optimal solutions are known and various constructive heuristic algorithms used to find excellent local-optimal solutions are available. Thus, the best constructive heuristic algorithm to be incorporated into the solution structure of a VNS algorithm can be identified.

Let $N_k, k = 1, \dots, k_{max}$ be a finite set of preselected neighbourhood structures, and let $N_k(x)$ be a set of solutions or neighbours in the k th neighbourhood of x .

An optimal solution x_{opt} (or global minimum) is a solution where the minimum of Eqs. (9)–(17) is achieved.

A solution x' is a local minimum of Eqs. (1)–(8) with regard to $N_k(x)$, if there is no solution $x'' \in N_k(x) \subseteq X$, such that $f(x'') < f(x')$.

Thus, the idea is to define a set of neighbourhood structures that can be used in a deterministic, random or both deterministic and random manners. These different forms of using the neighbourhood structure lead to VNS algorithms with different performances.

There are various proposals of VNS algorithms that can be used independently or in an integrated manner forming more complex VNS structures. The simplest form of a VNS algorithm is the variable neighbourhood descent (VND). The VND algorithm is based on previously mentioned Fact 1, i.e. the local minimum for a given move is not necessarily the local minimum for another type of move [29]. In this way, the local optimum x' in the neighbourhood $N_1(x)$ is not necessarily equal to the local optimum x'' of x' to the neighbourhood $N_2(x)$.

The VND algorithm takes on the form shown in **Figure 1**.

This algorithm can be integrated into a more complex structure of the VNS algorithm.

For example, the sept (a) in **Figure 1** could be replaced by randomly generating a solution neighbour x' of $x(x' \in N_k(x))$; and the resulting algorithm is called the reduced variable neighbourhood search (RVNS). In the RVNS, usually, the

Initialization: Select the set of neighborhood structures $N_k, k = 1, \dots, k_{max}$ that will be used in the descent; Find an initial solution x ;
 Repeat the following sequence until no improvement is obtained:
 (1) Set $k=1$;
 (2) Repeat the following steps until $k = k_{max}$:
 (a) Exploration: Find the best neighbor x' of $x(x' \in N_k(x))$;
 (b) Move or not:
 If the solution x' thus obtained is better than x , set $x = x'$ and $k = 1$;
 Otherwise, set $k = k + 1$.

Figure 1.
VND algorithm [33].

Initialization: Select the set of neighborhood structures $N_k, = 1, \dots, k_{max}$;
Find an initial solution x ;
Choose a stopping condition;
Repeat the following sequence until no improvement is obtained:
(1) Set $k = 1$;
(2) Repeat the following steps until $k = k_{max}$:
(a) Randomly generate a solution neighbor x' of $(x' \in N_k(x))$;
(b) Local search: Apply a local search method with x' as an initial solution;
Denote with x'' the obtained local optimum;
(c) Move or not:
 If the solution x'' thus obtained is better than x , set $x = x'$ and $k = 1$;
 Otherwise, set $k = k + 1$.

Figure 2.
BVNS framework [33].

neighbourhoods will be nested, i.e. each one contains the previous. Then a point is chosen at random in the first neighbourhood. If its value is better than that of the incumbent (i.e. $f(x') < f(x)$), the search is recentred there ($x' \leftarrow x$). Otherwise, one proceeds to the next neighbourhood. After all neighbourhoods have been considered, one begins again with the first, until a stopping condition is met.

The RVNS algorithm chooses neighbours more dynamically by selecting those from all neighbourhood structures (diversification) and prioritizing the first neighbourhood structure (intensification) during the initial stages of the search. Nevertheless, an important component of the RVNS structure is its capacity for finding new promising regions from a local optimum. The RVNS algorithm can also be used independently or be integrated into a more complex structure of the VNS algorithm.

More efficient VNS algorithms can be formulated by integrating those characteristics of the VND algorithm that allow local quality optima to be found and those of the RVNS algorithm that allow new promising regions from a local optimum to be found. Thus, by merging those characteristics, two types of VNS algorithms that generally exhibit excellent performance can be formulated. These algorithms are called the basic variable neighbourhood search (BVNS) and the general variable neighbourhood search (GVNS).

The BVNS algorithm combines a local search with systematic changes of neighbourhood around the local optimum found in [33]. The structure of the BVNS algorithm is presented in **Figure 2**.

The logical procedure adopted by the BVNS is very interesting. Firstly, k neighbourhood structures should be chosen. The optimization process is initiated from a solution x and the corresponding neighbourhood $N_1(x)$. Then, a neighbour x' of x in $N_1(x)$ is randomly selected. From x' , a local search process to find the local optimum x'' is started.

In this context, three cases may occur:

1. If x'' it is equal to x' , one already was the local optimum of the valley and, consequently, a change of neighbourhood level should be performed ($N_2(x)$ in this case).
2. If x'' is worse than x' , then the local optimum with less quality than the incumbent x was found, and a change of neighbourhood should also be carried out.

3. If x'' is better than x' , it means that a better solution than the incumbent was found, and, consequently, the incumbent should be updated; the search should be reinitiated from the new incumbent while remaining in the neighbourhood N_1 .

Whenever the local search finds a new incumbent, at any iteration of the process, the neighbourhood $N_1(x)$ should be considered again. Also, whenever the local search finds an equal or worse quality solution than the incumbent, a change towards a more complex neighbourhood should be performed. This strategy and the random choice of the incumbent x' neighbour avoid cycling and allow local optima which are distant from the current incumbent to be found.

The local search of the BVNS algorithm can be any heuristic strategy. Nonetheless, the local search can also use a strategy of the VNS algorithm. Therefore, the BVNS algorithm can be transformed into a more general algorithm called general variable neighbourhood search (GVNS). The GVNS algorithm is obtained through the generalization of the BVNS algorithm by simply using a VND algorithm as a local search and using a RVNS algorithm to improve the initial solution required to begin the search.

All observations made for the BVNS algorithm remain valid for the GVNS algorithm. As mentioned previously, the fundamental change corresponds to the improvement stage of the initial solution using an RVNS algorithm and a VND algorithm for the local search stage.

Since the VNS algorithm can be implemented in various ways, a family of VNS algorithms can also be implemented. In [26, 30, 33] diverse types of VNS algorithms are analysed. In this work, only one of these algorithms is presented. There are other more complex algorithms or structures based on the logic of the VNS algorithm that are out of the scope of this work. Those algorithms can be found in [30, 33].

4. Modified VNS for TNEP

In this section the application of our proposed VNS to the TNEP will be described. The GVNS described in **Figure 3** will be used considering the following steps that will be explained in detail in sequence:

- Step 1—Initial solution: Considering a heuristic algorithm to determine an initial solution.
 - Step 2—Definition of neighbourhoods: Characterization of each neighbourhood and determination of their elements.
 - Step 3—Improvement: Improve the initial solution by using a RVNS algorithm.
 - Step 4—Local search: Apply some local search to determine the best configuration for each current solution neighbourhood.
- Step 1: Initial solution

To determine a DC initial solution to TNEP, the constructive heuristic algorithm (CHA) presented by Villasana-Garver-Salon (VGS) [10] is considered. This algorithm iteratively chooses a new circuit to be added to the system considering a step-by-set procedure that uses a sensitivity index (given in Eq. (18)) that plays a key role in the CHA. The iteratively process continues until a feasible solution is achieved; that means that there is no need for new circuit additions:

Initialization:
 Select the set of neighborhood structures N_k , $k = 1, \dots, k_{max}$ that will be used in the shaking phase;
 Select the set of neighborhood structures N_l , $l = 1, \dots, l_{max}$ that will be used in the local search;
 Find an initial solution x and improve it by RVNS.
 Choose a stopping condition.
 Repeat the following sequence until no improvement is obtained:
 (1) Set $k = 1$;
 (2) Repeat the following steps until $k = k_{max}$:
 (a) Randomly generate a solution neighbor x' of $(x' \in N_k(x))$;
 (b) Local search by VND:
 (b1) Set $l = 1$;
 (b2) Repeat the following steps until $l = l_{max}$:
 • Exploration: Find the best neighbor x'' of x' in $N_l(x')$;
 • Move or not: If $f(x'') < f(x')$, set $x' = x''$ and $l = 1$;
 Otherwise, set $l = l + 1$.
 (c) Move or not: If this optimum is better than the incumbent, set $x = x''$ and continue the search with $N_1(k = 1)$; Otherwise, set $k = k + 1$.

Figure 3.
 GVNS framework [33].

$$IS = \text{Max} \left\{ s_{ij} = n_{ij} \overline{f_{ij}} : n_{ij} \neq 0 \right\} \quad (18)$$

Generally, for large and complex systems, the derived solutions are local-optimal [10]. The VGS can be summarized by the following steps:

- VGS1: Take a base topology as a current solution, and resolve the HML Eqs. (10)–(17) considering that all of the circuits of the current solution must follow both Kirchhoff's laws.
- VGS2: Solve LP for the HML using the current solution. If the LP solution indicates that the system is adequately operating with the new additions and $v = 0$, then stop. A new solution for the DC model was found. Go to step 4.
- VGS3: Identify the most attractive circuit considering the sensitivity in Eq. (18). Update the current solution with the chosen circuit, update n_{ij}^0 and Ω_0 , and go to step 2.

All of the added circuits represent the solution of the CHA. It can be noted that although the VGS uses a hybrid linear model to identify the best circuit for addition in an iterative process, it complies with both of Kirchhoff's laws after adding a new circuit; thus, the final solution is also feasible in DC.

Example 1: considering Graver's system [34] that includes six transmission lines and six buses with a 760-MW demand for base topology, which is shown in **Figure 4a**, after has applied the VGS it gave the topology in **Figure 4b**, with $v = 130.000$ m.u.

Step 2: Definition of neighbourhoods

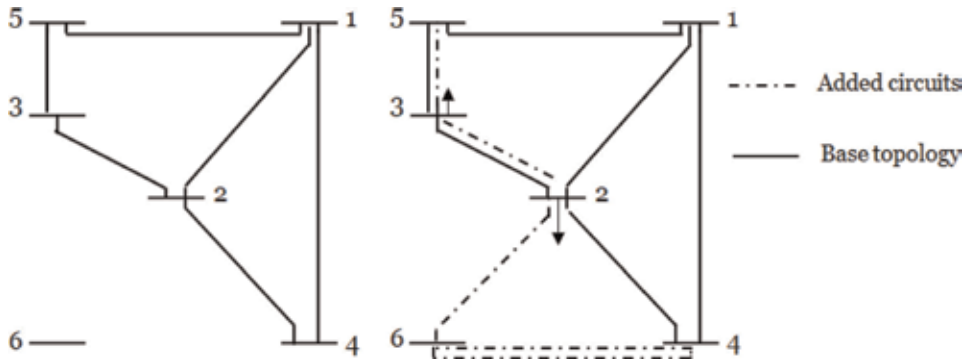


Figure 4. Base topology and VGS solution for Graver's system. (a) Base topology and (b) Initial solution by VGS.

Given solution x , the structures of neighbourhood within the solution space can be defined by Eq. (19):

$$N_k(x) = \{x' \in S : d(x, x') = k, k = 1, \dots, k_{max}\} \quad (19)$$

where $d(x, x') = k$ is the quantity of branches with a different number of added circuits in the solutions x and x' .

For example, given solutions x , x' and x'' from **Figure 5a–c**, respectively, which are coded in **Figure 6**, $d(x, x') = 1$, and $d(x, x'') = 2$. So, solution x' is a neighbour of x in $N_1(x)$, and solution x'' is a neighbour of x in $N_2(x)$.

Neighbour x' is obtained from x by adding a circuit in branch 8 (buses 3–6), whereas the neighbour x'' was obtained from x by adding one circuit in branch 7 (buses 3–5) and removing one circuit in branch 9 (buses 4–6). In the same way, the neighbours in the other k neighbourhoods can be obtained.

Step 3: Improvement of the initial solution

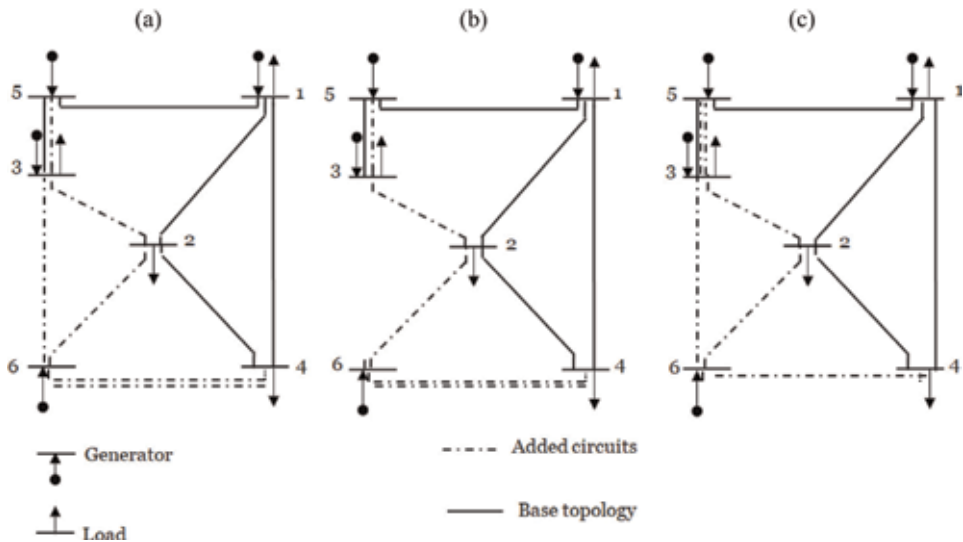


Figure 5. Neighbourhood characterization.

	1 1 2	2 1 4	3 1 5	4 2 3	5 2 4	6 2 6	7 3 5	8 3 6	9 4 6	branches bus i to bus j
x :	0	0	0	1	0	1	1	1	2	added circuits
$x^{i'}$:	0	0	0	1	0	1	1	0	2	
$x^{j'}$:	0	0	0	1	0	1	2	1	1	

Figure 6.
 x , x^i and x^j neighbours codification.

Considering $k_{max} = 5$ and the initial solution obtained in step 1, a local improvement search using a GVNS described in **Figure 3** is applied considering the HLM Eqs. (9)–(17).

In $N_1(x)$, sort all added circuits in cost-decreasing order, remove the circuit having the maximum cost, and verify the operation using the HLM model. If such removal keeps a feasible solution which indicates that the system is in adequate operation condition (i.e. $v = 0$ after HML solution), remove that circuit; otherwise, keep the circuit. Repeat the process of simulating circuit removal until all of the added circuits have been tested.

At the end of the process, all the added circuits that were not removed represent the improved solution.

As for the remaining neighbourhoods, the cost variations due to changes (cost difference between entering and leaving circuits) are calculated, and only the changes that exhibit negative variation are simulated (the HLM is solved). If the simulation points out a feasible configuration, then it is a candidate to be used by updating the current configuration. If the new configuration is unfeasible, then the simulation is cancelled.

It is important to elucidate that the movement only be carried out if the new configuration is better than the incumbent and that in this step the procedure only accepts movements that lead to feasible solutions.

The stop criterion corresponds to the maximum number of solved HLM.

Step 4: Local search

The local search is based in VND described in **Figure 1**.

5. Results

To illustrate the effectiveness of the proposed method, three problems are considered: the Garver 6-bus, the IEEE 24-bus and the Brazilian Southern 46-bus systems.

Full data can be found in [34–36], respectively. Planning could be done with (r) or without (w) generation rescheduling, resulting in these following cases that have been widely used to validate results of new methods [2, 10, 15, 16, 20, 34]; Da [21–24, 31, 32, 35, 36]:

- Case 1w: Garver 6-bus system without rescheduling
- Case 1r: Garver 6-bus system with rescheduling
- Case 2w: IEEE 24-bus system without rescheduling

Cases	Initial solution			GVNS solution			
	Added circuits	Total cost ($\times 1.000$)	PLs required	Added circuits	Total cost ($\times 1.000$)	PLs required	k_{max}
1W	$x_{1-3} = 3, x_{1-5} = 1, x_{2-3} = 1, x_{4-6} = 3$	244	9	$x_{2-6} = 4, x_{3-5} = 1, x_{4-6} = 2$	200	27	3
1r	$x_{23} = 1, x_{26} = 1, x_{35} = 1, x_{46} = 2$	130	6	$x_{35} = 1, x_{46} = 3$	110	19	2
2W	$x_{3-24} = 1, x_{10-12} = 2, x_{6-10} = 1, x_{7-8} = 2,$ $x_{10-12} = 1, x_{12-13} = 1, x_{14-16} = 1,$ $x_{15-24} = 1, x_{16-17} = 1$	476	12	$x_{3-24} = 1, x_{6-10} = 1, x_{7-8} = 2,$ $x_{9-11} = 1, x_{10-12} = 1, x_{14-16} = 2, x_{16-17} = 1$	392	1776	3
2r	$x_{1-5} = 1, x_{3-24} = 1, x_{6-7} = 2, x_{6-10} = 1,$ $x_{7-8} = 1, x_{10-11} = 1, x_{14-16} = 2, x_{15-16} = 1,$ $x_{15-21} = 1, x_{15-24} = 1, x_{16-17} = 2, x_{17-18} = 1$	618	16	$x_{3-24} = 1, x_{6-10} = 1, x_{7-8} = 2, x_{9-11} = 1,$ $x_{10-12} = 1, x_{14-16} = 2, x_{16-17} = 1$	342	361	2
3W	$x_{5-6} = 2, x_{20-21} = 2, x_{24-25} = 2, x_{25-32} = 1,$ $x_{28-31} = 1, x_{31-41} = 1, x_{40-41} = 1, x_{42-43} = 1,$ $x_{46-6} = 1$	166.041	17	$x_{5-6} = 2, x_{19-25} = 1, x_{20-21} = 1, x_{24-25} = 2,$ $x_{26-29} = 3, x_{28-30} = 1, x_{29-30} = 2, x_{31-32} = 1,$ $x_{42-43} = 2, x_{46-6} = 1$	154.420		5
3r	$x_{5-6} = 2, x_{19-21} = 1, x_{20-21} = 2, x_{20-23} = 1,$ $x_{46-6} = 1$	95.795	8	$x_{2-5} = 1, x_{5-6} = 2, x_{13-20} = 1, x_{20-21} = 2,$ $x_{20-23} = 1, x_{42-43} = 1, x_{46-6} = 2$	72.870	497	3

Table 1.
Obtained results.

- Case 2r: IEEE 24-bus system with rescheduling
- Case 3w: Brazilian Southern 46-bus system without rescheduling
- Case 3r: Brazilian Southern 46-bus system with rescheduling

The Brazilian Southern is a real referred system originally formed by 46 buses and 66 circuits in the base topology, 79 candidate paths and 6.880 MW as expected demand [35].

For reducing the size of the considered neighbourhoods, only those added circuits operating below 70% of their capacity were considered to be candidate circuits for removal.

Table 1 shows the results. The proposed method was more efficient than the methods shown in [15, 20], since it requires less number of linear programming resolutions.

6. Conclusions

In this paper an efficient new method based on variable neighbourhood search has been proposed for transmission networking problem planning considering the DC model whose mathematical formulation is nonlinear and mixed integer. The TNEP is a multimodal problem of high complexity for medium and large systems and cannot be solved by exact algorithms in reasonable computational times.

The proposed method systematically exploits the idea of neighbourhood change to find local-optimal solutions and to leave those local optima. It was observed that the definition of neighbourhood structures plays an important role to the convergence of the VNS algorithm applied to TNEP.

The proposed method was tested in the Garver 6-bus, in the IEEE 24-bus and in the Brazilian Southern 46-bus systems, and the results got more chance of finding better solutions than mathematical optimization techniques and find local-optimal solution requiring fewer solved linear problems.

As further research directions, new strategies for reducing the size of the neighbourhood such as using adjacency lists to avoid adding new lines in isolated circuits and different kinds of structure neighbourhoods could be developed.

Author details

Silvia Lopes de Sena Taglialenha^{1*} and Rubén Augusto Romero Lázaro²

1 Federal University of Santa Catarina, Technological Center of Joinville, Joinville, Brazil

2 Electrical Engineering at FEIS-UNESP-Ilha Solteira, Solteira, Brazil

*Address all correspondence to: s.taglialenha@ufsc.br

IntechOpen

© 2019 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/3.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. 

References

- [1] Sullivan RL. Power System Planning. New York: McGraw-Hil; 1977
- [2] Garver LL. Transmission network estimation using linear programming. IEEE Transaction Apparatus Systems. 1970;**89**:1688-1697
- [3] Romero R, Monticelli A, Garcia A, Haffner S. Test systems and mathematical models for transmission network expansion planning. IEE Proceedings Generation, Transmission and Distribution. 2002;**149**(1):27-36
- [4] Latorre G, Cruz RD, Areiza JM, Villegas A. Classification of publications and models on transmission expansion planning. IEEE Transactions on Power Systems. 2003;**18**(2):938-946
- [5] Hemmati R, Hooshmand RA, Khodabakhshian A. State-of-the-art of transmission expansion planning: Comprehensive review. Renewable and Sustainable Energy Reviews. 2013;**23**: 312-319. DOI: 10.1016/j.rser.2013.03.015
- [6] Lumbreras S, Ramos A. The new challenges to transmission expansion planning. Survey of recent practice and literature review. Electric Power Systems Research. 2016;**134**:19-29
- [7] Lee CW, Ng SKK, Zhong J, Wu FF. Transmission Expansion Planning From Past to Future. In: IEEE PES Power Systems Conference and Exposition; Atlanta, GA; 2006. pp. 257-265
- [8] Dusonchet YP, El-Abiad A. Transmission planning using discrete dynamic optimizing. IEEE Transactions on Power Apparatus and Systems. 1973; **PAS-92**(4):1358-1137. DOI: 10.1109/TPAS.1973.293543
- [9] Al-Hamouz ZM, Al-Faraj AS. Transmission expansion planning using nonlinear programming. In: IEEE/PES Transmission and Distribution Conference and Exhibition; Vol. 1. Yokohama, Japan: IEEE; 2002. pp. 50-55. DOI: 10.1109/TDC.2002. 1178259
- [10] Villasana R, Garver LL, Salon SJ. Transmission network planning using linear programming. IEEE Transactions on Power Systems. 1985;**104**:349-356
- [11] Haffner S, Monticelli A, Garcia A, Mantovani J, Romero R. Branch and bound algorithm for transmission system expansion planning using a transportation model. IEE Proceedings - Generation, Transmission and Distribution, V. 2000;**147**(3):149-156
- [12] Romero R, Monticelli A. A hierarchical decomposition approach for transmission network expansion planning. IEEE Transactions on Power Systems. 1994;**9**:373-380
- [13] Binato S, Pereira MVF, Granville S. A new benders decomposition approach to solve power transmission network design problems. IEEE Transactions on Power Apparatus and Systems. 2001;**16**: 235-240
- [14] Romero R, Rocha C, Mantovani JRS, Sanchez IG. Constructive heuristic algorithm for the DC model in network transmission expansion planning. IEE Proceedings-Generation, Transmission and Distribution. 2005;**152**(2):277-282
- [15] Romero R, Rider M, Silva I. A metaheuristic to solve the transmission expansion planning. IEEE Transactions on Power Systems. 2007;**22**:2289-2291
- [16] Da Silva EL, Gil HA, Areiza JM. Transmission network expansion planning under an improved genetic algorithm. IEEE Transactions on Power Systems. 2000;**15**(4):1168-1117
- [17] Seifi H, Sepasian MS, Haghghat H, Foroud AA, Yousefi GR, Rae S.

Multi-voltage approach to long-term network expansion planning. IET Generation Transmission and Distribution. 2007;**1**:9. DOI: 10.1049/iet-gtd:20070092

[18] Gallego RA, Monticelli A, Romero R. Transmission expansion planning by extended genetic algorithm. IEE Proceedings - Generation, Transmission and Distribution. 1998;**145**(3):329-335

[19] da Silva EL, Gil HA, Areiza JM. Transmission network expansion planning under an improved genetic algorithm. IEEE Transactions on Power Systems. 2000;**15**:1168-1175

[20] Gallego RA, Monticelli A, Romero R. Comparative studies of non-convex optimization methods for transmission network expansion planning. IEEE Transactions on Power Systems. 1998; **13**(3):822-828

[21] Da Silva EL, Areiza JM, Oliveira GC, Binato S. Transmission network expansion planning under a tabu search approach. IEEE Transactions on Power Systems. 2001;**16**(1):62-68

[22] Gallego RA, Alves AB, Monticelli A, Romero R. Parallel simulated annealing applied to long term transmission expansion planning. IEEE Transactions on Power Systems. 1997;**12**(12):181-187

[23] Faria HJ, Binato S, Resende MGC, Falcão DM. Power transmission network design by greedy randomized adaptive path relinking. IEEE Transactions on Power Systems. 2005; **20**(1):43-49

[24] Mori H, Shimomugi K. Network expansion planning with scatter search. In: IEEE International Conference on Systems, Man and Cybernetics. ISIC; 2007. pp. 3749-3754

[25] Khandelwal A, Bhargava A, Sharma A, et al. Modified grey wolf optimization algorithm for transmission network expansion planning problem.

Arabian Journal for Science and Engineering. 2018;**43**:2899. DOI: 10.1007/s13369-017-2967-3

[26] Glover F, Kochenberger GA. Handbook of Metaheuristics. Kluwer Academic Publishers; 2003

[27] Yang XS. Review of meta-heuristics and generalized evolutionary walk algorithm. International Journal of Bio-Inspired Computation. 2011;**3**(2):77-84

[28] Li Y, Gong G, Li N. Recent advances in modelling and optimizing complex systems based on intelligent algorithms. International Journal of Industrial Engineering: Theory, Applications and Practice. 2018;**25**(6):779-799

[29] Mladenovic N, Hansen P. Variable neighborhood search. Computers and Operations Research. 1997;**24**(11): 1097-1100

[30] Hansen P, Mladenovic N. Variable neighborhood search: Principles and applications. European Journal of Operational Research. 2001;**130**: 449-467

[31] Tagliailenha SLS. Novas Aplicações de Meta heurísticas na Solução do Problema de Planejamento da Expansão do Sistema de Transmissão de Energia Elétrica [thesis]. 2008. DEE-FEIS-UNESP, Ilha Solteira

[32] Tagliailenha SLS, Fernandes CWN, Silva VMD. Variable neighborhood search for transmission network expansion planning problem. In: Borsato M et al, editors. Transdisciplinary Engineering: Crossing Boundaries. ISPE TE. 2016;**2016**:543-552. DOI: 10.3233/978-1-61499-703-0-543

[33] Hansen P, Mladenovic N. A tutorial on variable neighbourhood search. Les Cahiers du GERAD, G-2003-46; 2003

[34] Haffner S, Monticelli A, Garcia A, Mantovani J, Romero R. Branch and

bound algorithm for transmission system expansion planning using a transportation model. IEE Proceedings-Generation, Transmission and Distribution. 2000;**147**(3):149-156

[35] Oliveira GC, Costa APC, Binato S. Large scale transmission network planning using optimization and heuristic techniques. IEEE Transactions on Power Systems. 1995;**10**:1828-1834

[36] Risheng F, Hill DJ. A new strategy for transmission expansion in competitive electricity markets. IEEE Transactions on Power Systems. 2003;**18**(1):374-380

An Improved Algorithm for Optimising the Production of Biochemical Systems

*Mohd Arfian Ismail, Vitaliy Mezhuyev,
Mohd Saberi Mohamad, Shahreen Kasim
and Ashraf Osman Ibrahim*

Abstract

This chapter presents an improved method for constrained optimisation of biochemical systems production. The aim of the proposed method is to maximise its production and, at the same time, to minimise the total amount of chemical concentrations involved in producing the best production. The proposed method models biochemical systems with ordinary differential equations. The optimisation process became complex for the large size of biochemical systems that contain many chemicals. In addition, several constraints as the steady-state constraint and the constraint of chemical concentrations also contributed to the computational complexity and difficulty in the optimisation process. This chapter considers the biochemical systems as a nonlinear equations system. To solve the nonlinear equations system, the Newton method was applied. Then, both genetic algorithm and cooperative co-evolutionary algorithm were applied to fine-tune the components in the biochemical systems to maximise the production and minimise the total amount of chemical concentrations involved. Two biochemical systems were used, namely the ethanol production in the *Saccharomyces cerevisiae* pathway and the *tryptophan* production in the *Escherichia coli* pathway. In evaluating the performance of the proposed method, several comparisons with other works were performed, and the proposed method demonstrated its effectiveness in maximising the production and minimising the total amount of chemical concentrations involved.

Keywords: biochemical systems production, constrained optimisation, computational intelligence, cooperative co-evolutionary algorithm, genetic algorithm, Newton method

1. Introduction

Computational systems biology is a field of biological study that combines the knowledge of science and engineering. The objective of this field is to model the behaviour of biochemical reactions through a computational approach. Within this field, the structures and complexity of biological processes can be investigated as a system [1]. Therefore, computational systems biology enables the scientist to represent the biological process as a system. This allows the biochemical process in a

living cell to be manipulated as a real factory and gives a way for scientists to improve the cell production (microbial production).

Integrating the knowledge of microbial production with genomic techniques and biotechnology processes creates the ability to manipulate a living cell to act like a real cell factory, thus opening new doors for researchers seeking to improve microbial productions [2]. One example of improving the microbial production is the optimisation of a biochemical systems production. Generally, biochemical systems can be defined as a series of chemical reactions found in a microorganism cell. With the knowledge of microbial production and genomic techniques, the biochemical systems can be represented as a dynamic mathematical model such as the Michaelis-Menten type [3], the stoichiometric approach [4], flux-balance analysis [5], metabolic control analysis [6] and biochemical systems theory (BST) [7]. Among these various choices, this work uses the BST representation to model the biochemical system. An advantage of using the BST is that prior knowledge of the mechanisms for each reaction is not required in order to build equations and the mathematical models can be designed by identifying the reactants and their interconnections [7]. For that reason, a canonical form that uses an ordinary differential equation (ODE) representation is suitable for modelling biochemical systems [1].

The optimisation of the biochemical systems production is a biotechnological process that aims to improve production by fine-tuning the chemical reaction. Besides that, the total amount of chemical concentrations involved also needs to be taken into account [8, 9]. To date, many studies have been carried out to develop methods for the optimisation of the biochemical systems production. Researchers tend to use the computational methods due to the flexibility of the mathematical models allowing to reduce the required costs and time. Popular methods used are the linear programming method (Vera et al. 2010; Xu 2012) and the geometric programming method [10, 11]. These methods depend on the definitions of the decision variables and the equality and inequality constraints, which could cause a convergence problem if the definition process is not performed well [12]. In order to overcome this problem, the present study was carried out using the stochastic method. The stochastic method operates on an evolving set of candidate solutions. In the evolving process, the candidate solutions are modified by the stochastic operator to produce the next generation. Using the stochastic operator, the search direction is determined by a random method, which makes it more efficient and robust [13]. In addition, the stochastic method does not rely on the manipulation of the objective function and constraints or the initialisation of a feasible point [14]. There are many stochastic methods that can be adopted for the optimisation process, among which is the genetic algorithm (GA) that has been widely found to be the most suitable method [15–17]. The GA works by representing the chemical reaction in the biochemical systems as a chromosome. The chromosome is then evolved and modified by a crossover and mutation process, with the intention to improve the solution.

As mentioned above, this chapter uses the BST method to model biochemical systems. Within the BST, two representations are typically used, namely, the S-system and generalised mass action (GMA). This study employs the GMA representation due to its ability to represent the nonlinearity of a biochemical systems and superior performance in optimisation [10]. The GMA uses the power law function, which is an ODE to model the biochemical systems. Applying only the GA for the optimisation of biochemical systems is not sufficient as the GA only fine-tunes the chemical concentrations. Therefore, a method is needed to deal with the biochemical systems. Implementing the Newton method for the biochemical systems is a good choice because the GMA model that represents the biochemical systems can be viewed as a nonlinear equations system [8, 18–22]. It also has been

found that the Newton method is suitable for the nonlinear equations system due to the convergence speed, simplicity and ease of use [23, 24].

Using the Newton method with the GA in optimising the biochemical systems production is a good choice because the Newton method deals with the biochemical systems, while the GA is used to fine-tune the chemical concentrations by representing the chemical concentrations into a chromosome. However, several problems do occur when dealing with large biochemical systems that contain many chemicals and has complex structures where it makes the representation of the solution become complex and difficult to evaluate. Hence, a method is needed in order to overcome these problems by simplifying the representation of the solution. Using the cooperative co-evolutionary algorithm (CCA) is a good choice because it has the ability to simplify the representation of the candidate solution by decomposing a single chromosome into multiple sub-chromosomes [17, 25, 26].

In this chapter, a hybrid method known as the advanced Newton cooperative genetic algorithm (ANCGA) that combined the Newton optimisation method; the GA and the CCA were presented. This method models biochemical systems as a system of nonlinear equations and applies the Newton method to solve the system. In the optimisation process, the GA and the CCA were used to represent the variables in a nonlinear system in order to search the best solution. The GA was used to maximise the production, while the CCA was used to minimise the total amount of chemical concentrations involved. The ANCGA that proposed in this study is the improvement of the existing method [17]. The reason of proposing the ANCGA is due to the previous algorithm that takes longer time for the optimisation process. Moreover, the performance of the previous work can be improved in terms of maximising the production and minimising the total amount of chemical concentrations involved. In order to do that, this work introduces a concept of external population. The external population was used to store the best solution found in every generation. The reason of using this concept was to avoid the best solution found in every generation from being lost during the reproduction process. The methods used in this study are presented in the following order. Firstly, the proposed method is explained in detail. Case studies of the fermentation pathway in *Saccharomyces cerevisiae* (*S. cerevisiae*) and the *tryptophan* (*trp*) of biosynthesis in *Escherichia coli* (*E. coli*) are then presented. Following that, the results are discussed, and a brief conclusion is made.

2. The proposed method

This section describes the proposed ANCGA in detail. The ANCGA is proposed in order to improve the performance of the previous method [17] in terms of computational time. In addition, the ANCGA is hope to improve the performance of the previous method [17] in maximising the production and minimising the total amount of chemical concentrations involved. **Figure 1** shows the flowchart of ANCGA. The ANCGA operates by treating the biochemical systems as a system of nonlinear equations and then uses the Newton method in solving the nonlinear equations system. Then, the GA and CCA were used in the optimisation process. The detailed operation of the ANCGA is described in the following steps:

Step 1—randomly generate the initial n sub-chromosomes in m sub-populations and create an empty external population. The number of sub-populations (m) must be the same to the number of variables in the nonlinear equations system. The sub-chromosomes represent the variables in the nonlinear equations system. The sub-chromosome is in the binary format.

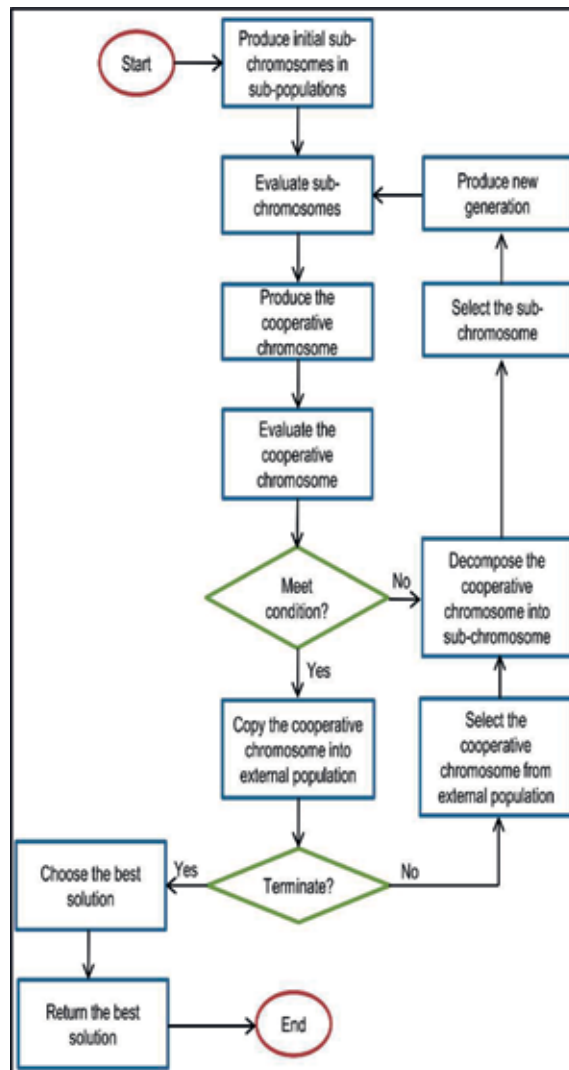


Figure 1.
The flow chart of ANCGA.

Step 2—evaluate the sub-chromosome. The evaluation process starts when a representative from every sub-population is selected to produce a complete solution that is known as a cooperative chromosome. The selection of representatives is based on their fitness value, where the lowest values are selected first. This process is known as the sub-chromosome evaluation. The objective of this process is to minimise the total amount of chemical concentrations involved by letting representatives that have the lowest fitness values from every sub-population to be combined together.

Step 3—produce the cooperative chromosome. The cooperative chromosome is produced after all the selected representatives are combined together. The cooperative chromosome is the complete solution. The formation of the cooperative chromosome is depicted in **Figure 2**.

Step 4—evaluate the cooperative chromosome. In this step, the cooperative chromosome is tested. The evaluation process starts with an encoding of the cooperative chromosome into variables in the nonlinear equations system. Then, the

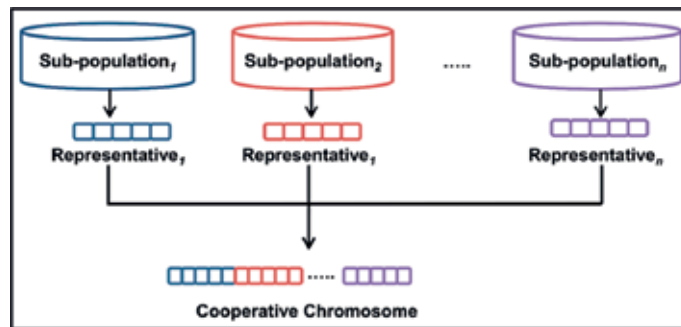


Figure 2.
The formation of the cooperative chromosome.

Newton method is used to solve the nonlinear equations system. In the evaluation process, a condition might occur depending on whether or not the cooperative chromosome follows the set of constraints. If the cooperative chromosome follows the constraints, then the procedure goes ahead to Step 8; if not, it goes to Step 5.

Step 5—decompose the cooperative chromosome into sub-chromosomes. After solving the nonlinear equations system using the Newton method, the variables in the nonlinear equations system are decoded back into the cooperative chromosome form. Then, the cooperative chromosome is decomposed into multiple sub-chromosomes. After that, all the sub-chromosomes are sent back to their own sub-populations in order to perform selection and reproduction.

Step 6—select a pair of sub-chromosome for the reproduction process. The selection process is based on their fitness value, where the lowest fitness value is selected first.

Step 7—produce new generations. In this step, the genetic operators of crossover and mutation are applied on the selected sub-chromosomes in order to produce new generations. This process is performed up to the last sub-chromosome. Then, the new generation is processed again, starting from Step 2.

Step 8—copy the cooperative chromosome into the external population. The process is performed by copying selected cooperative chromosome that fulfil the constraints and put the selected cooperative chromosome into external population. This process is intended to keep the best solution in every generation and prevent it from being lost in the reproduction process (Step 7). At this stage, two conditions may occur: either the maximum number of generations is reached or the maximum number of cooperative chromosomes in the external population is achieved. If these two conditions are fulfilled, the procedure jumps to Step 10; otherwise, the procedure continues to the next step. During this process, if the maximum number of cooperative chromosomes in the external population is reached before the maximum number of generations is achieved, the cooperative chromosome that has the lowest fitness value is deleted and replaced by a newly copied cooperative chromosome. However, if the maximum number of generations is reached before the maximum number of cooperative chromosomes in the external population is achieved, the procedure moves to Step 10.

Step 9—select some of the cooperative chromosomes from the external population. This process refers to the elitism of external population concept. The elitism of external population concept works where some (with y probability) of the cooperative chromosomes from the external population are selected and combined with the current sub-chromosomes. The selection process is based on their fitness value, where the cooperative chromosomes from the external population that have the highest fitness value are selected first. Then, this process goes back to Step 5.

Step 10—choose the best solution. The best solution is chosen among all the cooperative chromosomes in the external population. The selection is based on the fitness values of the cooperative chromosomes, where the cooperative chromosome with the highest fitness value is chosen.

Step 11—return to the best solution. This step decodes the selected cooperative chromosome into its real value (the variable in the nonlinear equations system) and gives the best solution set.

3. Case studies

In this section, the effectiveness and efficiency of the ANCGA is demonstrated. The effectiveness of the proposed method refers to the ability of the ANCGA to obtain the best result, while the efficiency refers to the ability of the ANCGA to maintain its performance in producing the best result in several case studies. Two case studies were used, namely, the *S. cerevisiae* pathway and the *E. coli* pathway. In order to test the performance of the ANCGA, a Java program based on two Java libraries, namely, jMetal [27] and JAMA of the version 1.0.3, was developed. The jMetal library can be downloaded from <http://jmetal.sourceforge.net/index.html>, while the JAMA version 1.0.3 can be accessed at <http://math.nist.gov/javanumerics/jama/>.

3.1 Case study 1: the ethanol production in *S. cerevisiae* pathway

In this case study, the ANCGA was used to optimise ethanol production in the *S. cerevisiae* pathway. The GA was used to represent the chemical reactions in the *S. cerevisiae* pathway, which were metabolites and enzymes. Details of the metabolites and enzymes, including the initial steady-state values, are presented in **Table 1**. The pathway was suspended in a cell culture at p. 4.5 and had the following ODE models [28].

Metabolite/enzyme	Symbol	Initial steady-state value
Glc _{in}	X ₁	0.0345
G6P	X ₂	1.0110
FDP	X ₃	9.1440
PEP	X ₄	0.0095
ATP	X ₅	1.1278
V _{in}	Y ₁	19.70
V _{HK}	Y ₂	68.50
V _{PFK}	Y ₃	31.70
V _{GAPD}	Y ₄	49.90
V _{PK}	Y ₅	3440.00
V _{Carb}	Y ₆	14.31
V _{Gro}	Y ₇	203.00
V _{ATPase}	Y ₈	25.10

Table 1.
Details of metabolite and enzymes in case study 1.

$$\begin{aligned}
 \frac{dX_1}{dt} &= V_{in} - V_{HK} \\
 \frac{dX_2}{dt} &= V_{HK} - V_{PFK} - V_{Carb} \\
 \frac{dX_3}{dt} &= V_{PFK} - V_{GAPD} - 0.5V_{Gro} \\
 \frac{dX_4}{dt} &= 2V_{GAPD} - V_{PK} \\
 \frac{dX_5}{dt} &= 2V_{GAPD} + V_{PK} - V_{HK} - V_{Carb} - V_{PFK} - V_{ATPase}
 \end{aligned} \tag{1}$$

Eq. (2) shows the fluxes at the steady-state condition.

$$\begin{aligned}
 V_{in} &= 0.8122X_2^{-0.2344}Y_1 \\
 V_{HK} &= 2.8632X_1^{0.7464}X_5^{0.0243}Y_2 \\
 V_{PFK} &= 0.5232X_2^{0.7318}X_5^{-0.3941}Y_3 \\
 V_{Carb} &= 8.904 \times 10^{-4}X_2^{8.6107}Y_6 \\
 V_{GAPD} &= 7.6092 \times 10^{-2}X_3^{0.6159}X_5^{0.1308}Y_4 \\
 V_{gro} &= 9.272 \times 10^{-2}X_3^{0.05}X_4^{0.533}X_5^{-0.0822}Y_7 \\
 V_{PK} &= 9.471 \times 10^{-2}X_3^{0.05}X_4^{0.533}X_5^{-0.0822}Y_5 \\
 V_{ATPase} &= X_5X_8
 \end{aligned} \tag{2}$$

For the total amount of chemical concentration involved, it can be formulated as follows:

$$\min F_2 = \sum_{j=1}^5 X_j + \sum_{k=1}^6 X_k \tag{3}$$

In the optimisation process, the GMA model was treated as a nonlinear equations system, where all the GMA models were set to be equal to 0. This gave all the ODE models in Eq. (1) the following forms:

$$\begin{aligned}
 V_{in} - V_{HK} &= 0 \\
 V_{HK} - V_{PFK} - V_{Pol} &= 0 \\
 V_{PFK} - V_{GAPD} - 0.5V_{Gol} &= 0 \\
 2V_{GAPD} - V_{PK} &= 0 \\
 2V_{GAPD} + V_{PK} - V_{HK} - V_{Pol} - V_{PFK} - V_{ATPase} &= 0
 \end{aligned} \tag{4}$$

For the metabolite concentration constraint, the constraint was set to 20% from the steady-state value, which was in the range between 0.8 and 1.2 [8, 29]. Thus, the constraint for this case study became as follows:

$$X_k^{0.8} \leq X_k \leq X_k^{1.2} \quad k = 1, 2, 3, 4, 5 \tag{5}$$

Meanwhile, the enzyme concentration constraint was set in the range between 0 and 50 from the steady-state value [8, 29]. The enzyme concentration constraint can be formulated as follows:

$$Y_k^0 \leq Y_k \leq Y_k^{50} \quad k = 1, 2, 3, 4, 5, 8 \tag{6}$$

3.2 Case study 2: the tryptophan biosynthesis in *E. coli* pathway

For case study 2, the ANCGA was used to optimise the end product of this pathway, which was *trp* production. The complete description of this pathway was provided by Xiu and colleagues [30]. The GMA models of this pathway are given as follows:

$$\begin{aligned}\frac{dX_1}{dt} &= V_{11} - V_{12} \\ \frac{dX_2}{dt} &= V_{21} - V_{22} \\ \frac{dX_3}{dt} &= V_{31} - V_{32} - V_{33} - V_{34}\end{aligned}\quad (7)$$

where X_1 is the mRNA concentration, X_2 is the enzyme concentration and X_3 is the *trp* concentration. The rates of all reactions in this pathway at steady state are given as follows:

$$\begin{aligned}V_{11} &= 0.6403X_3^{-5.87 \times 10^{-4}}X_5^{-0.8332} \\ V_{12} &= 1.0233X_1X_4^{0.0035}X_{11}^{0.9965} \\ V_{21} &= X_1 \\ V_{22} &= 1.4854X_2X_4^{-0.1349}X_{12}^{0.8651} \\ V_{31} &= 0.5534X_2X_3^{-0.5573}X_6^{0.5573} \\ V_{32} &= X_3X_4 \\ V_{33} &= 0.9942X_3^{7.0426 \times 10^{-4}}X_7 \\ V_{34} &= 0.8925X_3^{3.5 \times 10^{-6}}X_4^{0.9760}X_8X_9^{-0.0240}X_{10}^{-3.5 \times 10^{-6}}\end{aligned}\quad (8)$$

The *trp* production in this case study is given by the reaction V_{34} [31]. This leads to optimisation that can be formulated as follows:

$$\max F = V_{34}\quad (9)$$

For the total amount of chemical concentrations involved, it can be formulated as follows:

$$\min F_2 = \sum_{k=1}^5 X_k + X_8\quad (10)$$

Similar to case study 1, the GMA model was set to be equal to 0, thus Eq. (8) became as follows:

$$\begin{aligned}V_{11} - V_{12} &= 0 \\ V_{21} - V_{22} &= 0 \\ V_{31} - V_{32} - V_{33} - V_{34} &= 0\end{aligned}\quad (11)$$

In this case study, the GA and CCA only represent several chemical concentrations. This was because not all chemical concentrations were being tuned [1, 10, 11]. The chemical concentrations that tuned were X_1 up to X_6 and X_8 . These chemical concentrations including their initial steady states are summarised in **Table 2**. For the other chemical concentrations which were X_7 and X_9 up to X_{13} , fixed values were used [1, 10, 11]. Eq. (12) lists the range of these chemicals.

Reaction	Initial steady-state value
X_1	0.0345
X_2	1.0110
X_3	9.1440
X_4	0.0095
X_5	1.1278
X_6	19.70
X_8	25.10

Table 2.
 Summary of reaction concentrations in case study 2.

$$\begin{aligned}
 X_k^{0.8} \leq X_k \leq X_k^{1.2} \quad k = 1, 2, 3 \\
 0 \leq X_4 \leq 0.00624 \\
 4 \leq X_5 \leq 10 \\
 500 \leq X_6 \leq 5000 \\
 X_7 = 0.0022X_5 \\
 0 \leq X_8 \leq 1000 \\
 X_9 = 7.5 \\
 X_{10} = 0.005 \\
 X_{11} = 0.9 \\
 X_{12} = 0.02 \\
 X_{13} = 0
 \end{aligned} \tag{12}$$

4. Results and discussion

In performing the experiments, many parameter settings were used. The list of all parameter settings used in this study is listed in **Table 3**, whereas **Table 4** presents the parameter settings in producing the best result. The binary coding was used to represent the chemical concentrations. For the Newton method, fixed

Parameter	Rate
Number of sub-populations	Depend on the number of variables in nonlinear equations system
Number of sub-chromosomes in sub-population	[100,110,120,130,140,150]
Number of chromosomes in external population P	[100,110,120,130,140,150]
Maximum number of generations	[100,110,120,130,140,150]
Crossover rate	[0.1,0.2,0.3,0.4,0.5]
Mutation rate	[0.1,0.2,0.3,0.4,0.5]
Elitism rate	[0.1,0.2,0.3,0.4,0.5]

Table 3.
 List of all parameter settings used.

Parameter	Case study 1	Case study 2
Number of sub-populations	11	7
Number of sub-chromosomes in sub-population	150	140
Number of chromosomes in external population P	100	100
Maximum number of generations	150	130
Crossover rate	0.3	0.4
Mutation rate	0.1	0.1
Elitism rate	0.2	0.2

Table 4.
Parameter settings in producing optimum solution.

Variables	Best solution 1	Average
X_1	1.1240	0.9951
X_2	1.0322	1.0018
X_3	0.9900	1.0053
X_4	1.1407	1.1297
X_5	1.0001	0.9831
Y_1	49.8103	49.9793
Y_2	45.3702	45.0767
Y_3	45.3452	49.8103
Y_4	48.5112	47.4064
Y_5	49.4448	49.3426
Y_8	49.7563	49.7876
F_1	53.0200	52.7499
F_2	293.5249	294.5178

Table 5.
The full result of case study 1.

parameters were used, namely, 50 for the maximum number of iterations and 10^{-6} for tolerance.

The full results obtained by the ANCGA when applied on *S. cerevisiae* pathway are given in **Table 5**. At the best solution, the ANCGA was able to increase the F_1 (ethanol production) up to 53.02 bigger than its initial steady-state value. For the F_2 (total amount of chemical concentrations involved), the proposed method was able to reduce it to 293.5249. All metabolites and enzymes fulfilled their constraints, with all the metabolites staying in the range of 0.8–1.2, while all the enzymes were in the range of 0–50. The performance of the ANCGA was assessed by comparing the result obtained by ANCGA with other works, and the comparison results are listed in **Table 6**. As shown in the table, the ANCGA produced higher results as compared to other methods. In addition, to verify the results achieved by the ANCGA, an average of 100 independent runs was recorded. The results are summarised in **Table 5**. It shows that the average result for the metabolites and enzymes fulfilled their constraints, whereby they were in their optimum range, thus leading to the conclusion that the ANCGA is able to produce reliable results. It can be said that the ANCGA can produce higher production of ethanol as compared to the methods used in other studies.

Work by	F_1	F_2
Xu [11]	52.38	297.664
Rodriguez-Acosta et al. [29]	52.31	295.270
Previous method [17]	52.91	294.800
ANCGA	53.02	293.5249

Table 6.
 Comparison with other works for case study 1.

Variables	Best solution	Average
X_1	0.8064	1.0742
X_2	0.8046	1.1085
X_3	0.8000	0.8000
X_4	0.0054	0.0054
X_5	4.0116	4.4694
X_6	5000	5000
X_8	1000	1000
F_1	3.9774	3.9616
F_2	6006.4280	6007.4575

Table 7.
 The full result of case study 2.

Work by	F_1	F_2
Marin-Sanguino et al. [10]	3.062	6006.1412
Vera et al. [1]	3.05	6007.1314
Xu [11]	3.946	6007.7814
Previous method [17]	3.9759	6006.5581
ANCGA	3.9774	6006.4280

Table 8.
 Comparison with other works for case study 2.

The full results of the *E. coli* pathway are presented in **Table 7**. The ANCGA was able to improve the F_1 (production of *trp*) to 3.9774 from its initial steady state. Meanwhile, the proposed method was able to reduce the F_2 (total amount of chemical concentrations involved) to 6006.4280. All variables representing the chemical reaction followed their constraints and were in the optimum range. To assess the performance of the ANCGA, the results achieved were compared to the results of other methods, with the details of the comparison shown in **Table 8**. As presented in the table, the F_1 of the ANCGA was higher when compared to the methods employed in other works. Similar to the previous case study, 100 experiments were conducted, and the average result was calculated in order to validate the ANCGA results. **Table 7** presents the average result. From the data in **Table 7**, it can be concluded that the ANCGA is reliable in performing the optimisation of this pathway because the average of all the variables follows their constraints. From the observations presented in **Tables 7** and **8**, it can be concluded that the ANCGA is effective in optimising the *trp* production as well as producing reliable results.

The external population concept used by ANCGA can be validated by comparing it with the previous method proposed in [17]. The aim of the external population concept was to reduce the computational time and the number of generations. To learn the effect of the external population concept, several experiments were conducted. To investigate the decrease in the number of generations, F_1 was set to 52.5 for case study 1 and 3.90 for case study 2. After F_1 was achieved, the process was stopped. This helped to investigate which method required more generations in achieving the target production. **Figures 3 and 4** illustrate the comparisons of all case studies. In both figures, the maximum number of the external population was smaller as compared to the maximum number of the previous method in achieving F_1 . This was caused by the concept of external population that was introduced in this study. By using this concept, the best solutions found in the iteration process could be maintained and thus enabled the number of generations to be reduced. In addition, it was found that this concept tended to converge faster than the previous method. This meant that the use of the external population concept allowed faster search of the best solution. In conclusion, the external population concept had an impact in reducing the number of generations and helped in faster convergence as compared to previous methods. To determine the statistical significance between the proposed method and previous methods, the paired t-test and the Wilcoxon signed-rank test were used. The result of the statistical tests showed that all p-values were <0.05 , thus confirming that the proposed method significantly improved the previous method.

Meanwhile, to investigate the decrease in computational time, the maximum number of generations was not set, but F_1 was set to 52.5 for case study 1 and 3.9 for case study 2. After F_1 was achieved, the process was terminated. **Table 9** lists the

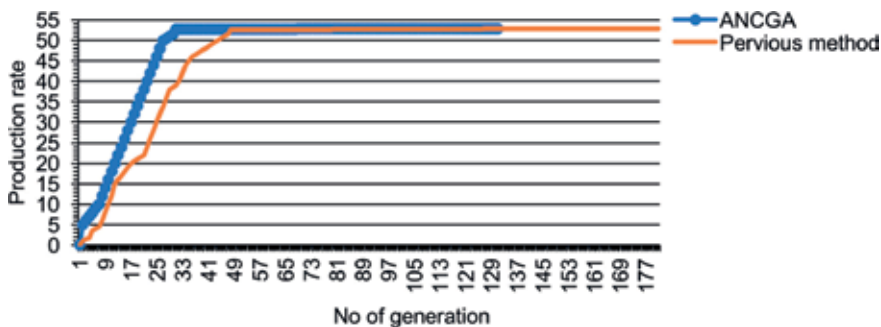


Figure 3.
The comparison of results of elitism concept and non-elitism concept for case study 1.

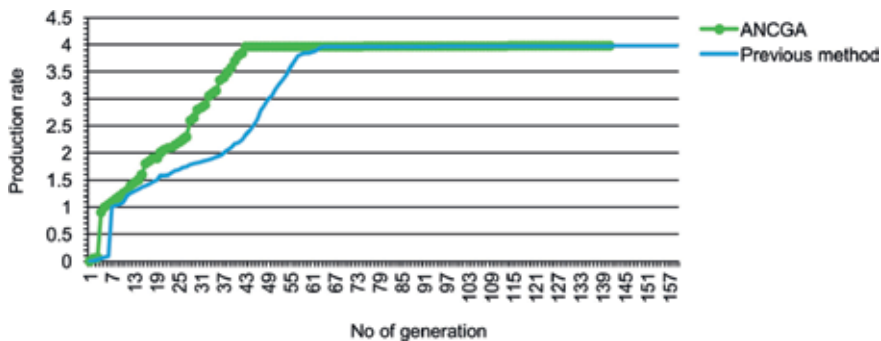


Figure 4.
The comparison of results of elitism concept and non-elitism concept for case study 2.

Method	Case study 1	Case study 2
ANCGA	75.56	38.07
Previous method [17]	80.40	40.45

Table 9.
The computation times obtained (in second).

computational time results, and it was found that the ANCGA required less time as compared to the method in [17]. This situation occurred because the high-quality solutions were stored in the external population and then combined with the current solution in the optimisation process. Copying the high-quality solutions into the external population prevented them from being lost (because the optimisation process involving crossover and mutation operation could lose the high-quality solutions). By storing the high-quality solutions into the external population, it would be able to keep the best solution until the optimisation process stopped. To determine the significant improvement of the proposed method against the previous method, the paired t-test and the Wilcoxon signed-rank test were used. The p-value from both tests was <0.05 . From this finding, the proposed method and the previous method were statistically different from each other, and the improvement of the proposed method could be accepted.

5. Conclusion

Improving production has become an important issue in the optimisation of biochemical systems. Many factors need to be considered to ensure optimal production. In this work, a hybrid method for constraint optimisation of the biochemical systems production known as the ANCGA was presented. The ANCGA was developed based on a previous method [17], where the ANCGA combined the Newton method, GA and CCA. This study introduced a concept of external population. The aim of this concept was to reduce computational time. In this work, the biochemical system was modelled by a nonlinear equations system. In the optimisation process, the Newton method was employed to deal with a system of nonlinear equations. The GA and CCA were then applied to fine-tune the chemical concentration value in the nonlinear system in order to search for the best solution. During the optimisation process, the high-quality solutions were copied and stored into the external population. The purpose of this process was to avoid the loss of high-quality solutions during the optimisation process. Then, some solutions from the external population were mixed with the next generation of solutions. By doing this, the computational time and number of generations were reduced. In the present study, the proposed method was applied on two case studies, and better results were obtained as compared to the methods presented in other works. In addition, the results were validated, and they demonstrated that the constraints of all the components in the biochemical system were fulfilled. Thus, it can be concluded that the performance of the ANCGA is effective and reliable in producing the best result.

Acknowledgements

Special appreciation to Universiti Malaysia Pahang for the sponsorship of this study by approving the RDU Grant Vot No. RDU180307. Special thanks to the reviewers and editor who reviewed this manuscript.

Author details

Mohd Arfian Ismail^{1*}, Vitaliy Mezhuyev¹, Mohd Saberi Mohamad^{2,3},
Shahreen Kasim⁴ and Ashraf Osman Ibrahim^{5,6}

1 Faculty of Computer Systems and Software Engineering, Universiti Malaysia Pahang, Gambang, Pahang, Malaysia

2 Institute For Artificial Intelligence and Big Data, Universiti Malaysia Kelantan, City Campus, Kota Bharu, Kelantan, Malaysia

3 Faculty of Bioengineering and Technology, Universiti Malaysia Kelantan, Jeli Campus, Jeli, Kelantan, Malaysia

4 Soft Computing and Data Mining Centre, Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn, Johor, Malaysia

5 Faculty of Computer Science and Information Technology, Alzaiem Alazhari University, Khartoum North, Sudan

6 Arab Open University, Khartoum, Sudan

*Address all correspondence to: arfian@ump.edu.my

IntechOpen

© 2019 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/3.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. 

References

- [1] Vera J, Gonzalez-Alcon C, Marin-Sanguino A, Torres N. Optimization of biochemical systems through mathematical programming: Methods and applications. *Computers & Operations Research*. 2010;**37**(8): 1427-1438
- [2] Sowa SW, Baldea M, Contreras LM. Optimizing metabolite production using periodic oscillations. *PLoS Computational Biology*. 2014;**10**(6): e1003658
- [3] Sakamoto N. Characterization of the transit and transition times for a pathway unit of Michaelis-Menten mechanism. *Biochimica et Biophysica Acta (BBA) - General Subjects*. 2003; **1623**(1):6-12
- [4] Planes FJ, Beasley JE. A critical examination of stoichiometric and path-finding approaches to metabolic pathways. *Briefings in Bioinformatics*. 2008;**9**(5):422-436
- [5] Salleh A, Mohamad M, Deris S, Illias R. Identifying minimal genomes and essential genes in metabolic model using flux balance analysis. In: Selamat A, Nguyen N, Haron H, editors. *Intelligent Information and Database Systems SE - 43*. Vol. 7802. Berlin, Heidelberg: Springer; 2013. pp. 414-423
- [6] Fell D. Metabolic control analysis. In: Alberghina L, Westerhoff HV, editors. *Systems Biology SE - 80*. Vol. 13. Berlin, Heidelberg: Springer; 2005. pp. 69-80
- [7] Voit EO. Biochemical systems theory: A review. *ISRN Biomathematics*. 2013; **2013**:1-15
- [8] Link H, Vera J, Weuster-Botz D, Darias NT, Franco-Lara E. Multi-objective steady state optimization of biochemical reaction networks using a constrained genetic algorithm. *Computers and Chemical Engineering*. 2008;**32**(8):1707-1713
- [9] Xu G. Bi-objective optimization of biochemical systems by linear programming. *Applied Mathematics and Computation*. 2012;**218**(14): 7562-7572
- [10] Marin-Sanguino A, Voit EO, Gonzalez-Alcon C, Torres NV. Optimization of biotechnological systems through geometric programming. *Theoretical Biology and Medical Modelling*. 2007;**4**:38-54
- [11] Xu G. Steady-state optimization of biochemical systems through geometric programming. *European Journal of Operational Research*. 2013;**225**(1): 12-20
- [12] Mariano AP et al. Optimization strategies based on sequential quadratic programming applied for a fermentation process for butanol production. *Applied Biochemistry and Biotechnology*. 2009;**159**(2):366-381
- [13] Balsa-Canto E, Banga JR, Egea JA, Fernandez-Villaverde A, Hijas-Liste GM. Global optimization in systems biology: Stochastic methods and their applications. In: Goryanin II, Goryachev AB, editors. *Advances in Systems Biology*. Vol. 736. New York: Springer; 2012. pp. 409-424
- [14] Mariano AP et al. Genetic algorithms (binary and real codes) for the optimisation of a fermentation process for butanol production. *International Journal of Chemical Reactor Engineering*. 2010;**8**. DOI: 10.2202/1542-6580.2333
- [15] Elsayed SM, Sarker RA, Essam DL. A new genetic algorithm for solving optimization problems. *Engineering*

Applications of Artificial Intelligence. 2014;**27**:57-69

[16] Deng H et al. The application of multiobjective genetic algorithm to the parameter optimization of single-well potential stochastic resonance algorithm aimed at simultaneous determination of multiple weak chromatographic peaks. *The Scientific World Journal*. 2014;**2014**

[17] Ismail MA, Deris S, Mohamad MS, Abdullah A. A newton cooperative genetic algorithm method for in silico optimization of metabolic pathway production. *PLoS One*. 2015;**10**(5): e0126199

[18] Grosan C, Abraham A. A new approach for solving nonlinear equations systems. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*. 2008;**38**(3): 698-714

[19] Luo Y-Z, Tang G-J, Zhou L-N. Hybrid approach for solving systems of nonlinear equations using chaos optimization and quasi-Newton method. *Applied Soft Computing*. 2008;**8**(2):1068-1073

[20] Babaei M. A general approach to approximate solutions of nonlinear differential equations using particle swarm optimization. *Applied Soft Computing*. 2013;**13**(7):3354-3365

[21] Ramos H, Monteiro MTT. A new approach based on the Newton's method to solve systems of nonlinear equations. *Journal of Computational and Applied Mathematics*. 2017;**318**:3-13

[22] Ahmad F, Tohidi E, Carrasco JA. A parameterized multi-step Newton method for solving systems of nonlinear equations. *Numerical Algorithms*. 2016;**71**(3):631-653

[23] Liu C-S, Atluri SN. A novel time integration method for solving a large

system of non-linear algebraic equations. *Computer Modeling in Engineering and Sciences*. 2008;**31**(2): 71-83

[24] Taheri S, Mammadov M. Solving systems of nonlinear equations using a globally convergent optimization algorithm. *Global Journal of Technology & Optimization*. 2013;**3**:132-138

[25] Gu J, Gu M, Cao C, Gu X. A novel competitive co-evolutionary quantum genetic algorithm for stochastic job shop scheduling problem. *Computers and Operations Research*. 2010;**37**(5): 927-937

[26] Ismail MA, Asmuni H, Othman MR. The fuzzy cooperative genetic algorithm (FCoGA): The optimisation of a fuzzy model through incorporation of a cooperative coevolutionary method. *Journal of Computing*. 2011;**3**(11):81-90

[27] Durillo JJ, Nebro AJ. jMetal: A Java framework for multi-objective optimization. *Advances in Engineering Software*. 2011;**42**(10):760-771

[28] Galazzo JL, Bailey JE. Fermentation pathway kinetics and metabolic flux control in suspended and immobilized *Saccharomyces cerevisiae*. *Enzyme and Microbial Technology*. 1990;**12**(3): 162-172

[29] Rodriguez-Acosta F, Regalado CM, Torres NV. Non-linear optimization of biotechnological processes by stochastic algorithms: Application to the maximization of the production rate of ethanol, glycerol and carbohydrates by *Saccharomyces cerevisiae*. *Journal of Biotechnology*. 1999;**65**(1):15-28

[30] Xiu Z-L, Zeng A-P, Deckwer W-D. Model analysis concerning the effects of growth rate and intracellular tryptophan level on the stability and dynamics of tryptophan biosynthesis in bacteria. *Journal of Biotechnology*. 1997;**58**(2): 125-140

[31] Marin-Sanguino A, Torres NV.
Optimization of tryptophan production
in bacteria. Design of a strategy for
genetic manipulation of the tryptophan
operon for tryptophan flux
maximization. *Biotechnology Progress*.
2000;**16**(2):133-145

Section 3

Applications of Artificial
Neural Networks

Object Recognition Using Convolutional Neural Networks

Richardson Santiago Teles de Menezes,

Rafael Marrocos Magalhaes and Helton Maia

Abstract

This chapter intends to present the main techniques for detecting objects within images. In recent years there have been remarkable advances in areas such as machine learning and pattern recognition, both using convolutional neural networks (CNNs). It is mainly due to the increased parallel processing power provided by graphics processing units (GPUs). In this chapter, the reader will understand the details of the state-of-the-art algorithms for object detection in images, namely, faster region convolutional neural network (Faster RCNN), you only look once (YOLO), and single shot multibox detector (SSD). We will present the advantages and disadvantages of each technique from a series of comparative tests. For this, we will use metrics such as accuracy, training difficulty, and characteristics to implement the algorithms. In this chapter, we intend to contribute to a better understanding of the state of the art in machine learning and convolutional networks for solving problems involving computational vision and object detection.

Keywords: machine learning, convolutional neural network, object detection

1. Introduction

There are fascinating problems with computer vision, such as image classification and object detection, both of which are part of an area called object recognition. For these types of issues, there has been a robust scientific development in the last years, mainly due to the advances of convolutional neural networks, deep learning techniques, and the increase of the parallelism processing power offered by the graphics processing units (GPUs). The image classification problem is the task of assigning to an input image one label from a fixed set of categories. This classification problem is central within computer vision because, despite its simplicity, there are a wide variety of practical applications and has multiple uses, such as labeling skin cancer images [1], use of high-resolution images to detect natural disasters such as floods, volcanoes, and severe droughts, noting the impacts and damage caused [2–4].

The performance of image classification algorithms crucially relies on the features used to feed them [5]. It means that the progress of image classification techniques using machine learning relied heavily on the engineering of selecting the essential features of the images that make up the database. Thus, obtaining these resources has become a daunting task, resulting in increased complexity and computational cost. Commonly, two independent steps are required for image

classification, feature extraction, and learning algorithm choice, and this has been widely developed and enhanced using support vector machines (SVMs).

The SVM algorithm, when considered as part of the supervised learning approach, is often used for tasks as classification, regression, and outlier detection [6]. The most attractive feature of this algorithm is that its learning mechanism for multiple objects is simpler to be analyzed mathematically than traditional neural network architecture, thus allowing to complex alterations with known effects on the core features of the algorithm [7]. In essence, an SVM maps the training data to higher-dimensional feature space and constructs a separation hyperplane with maximum margin, producing a nonlinear separation boundary in the input space [8].

Today, the most robust object classification and detection algorithms use deep learning architectures, with many specialized layers for automating the filtering and feature extraction process. Machine learning algorithms such as linear regression, support vector machines, and decision trees all have its peculiarities in the learning process, but fundamentally they all apply similar steps: make a prediction, receive a correction, and adjust the prediction mechanism based on the correction, at a high level, making it quite similar to how a human learns. Deep learning has appeared bringing a new approach to the problem, which attempted to overcome previous drawbacks by learning abstraction in data following a stratified description paradigm based on a nonlinear transformation [9]. A key advantage of deep learning is its ability to perform semi-supervised or unsupervised feature extraction over massive datasets.

The ability to learn the feature extraction step present in deep learning-based algorithms comes from the extensive use of convolutional neural networks (ConvNet or CNN). In this context, convolution is a specialized type of linear operation and can be seen as the simple application of a filter to a determined input [10]. Repeated application of the same filter to an input results in a map of activations called a feature map, indicating the locations and strength of a detected feature in the input by tweaking the parameters of the convolution. The network can adjust itself to reduce the error and therefore learn the best parameters to extract relevant information on the database.

Many deep neural network (DNN)-based object detectors have been proposed in the last few years [11, 12]. This research investigates the performance of state-of-the-art DNN models of SSD and Faster RCNN applied to a classical detection problem where the algorithms were trained to identify several animals in images; furthermore to exemplify the application in scientific research, the YOLO network was trained to solve the mice tracking problem. The following sections describe the DNN models mentioned earlier in more details [13–15].

2. Object detection techniques

2.1 Single shot multibox detector

The single shot multibox detector [13] is one of the best detectors in terms of speed and accuracy comprising two main steps, feature map extraction and convolutional filter applications, to detect objects.

The SSD architecture builds on the VGG-16 network [16], and this choice was made based on the strong performance in high-quality image classification tasks and the popularity of the network in problems where transfer learning is involved.

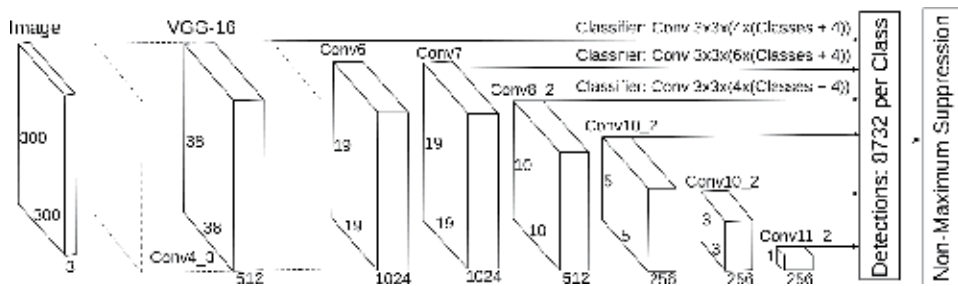


Figure 1. The SSD network has several feature layers to the end of the base network, which predicts the offsets to default boxes of different scales, aspect ratios, and their associated confidences. Figure based on [13].

Instead of the original VGG fully connected layers, a set of auxiliary convolutional layers change the model, thus enabling to extract features at multiple scales and progressively decrease the size of the input to each subsequent layer.

The bounding box generation considers the application of matching pre-computed, fixed-size bounding boxes called *priors* with the original distribution of ground truth boxes. These *priors* are selected to keep the intersection over union (IoU) ratio equal to or greater than 0.5.

The overall loss function defined in Eq. (1) is a linear combination of the confidence loss, which measures how confident the network is of the computed bounding box using categorical cross-entropy and location loss, which measures how far away the networks predicted bounding boxes are from the ground truth ones using L2 norm.

$$L(x, c, l, g) = \frac{1}{N} (L_{conf}(x, c) + \alpha L_{loc}(x, l, g)) \quad (1)$$

where N is the number of matched default boxes and L_{conf} and L_{loc} are the confidence and location loss, respectively, as defined in [13]. **Figure 1** depicts how to apply the convolutional kernels to an input image in the SSD architecture.

2.2 You only look once

You only look once [14] is a state-of-the-art object detection algorithm which targets real-time applications, and unlike some of the competitors, it is not a traditional classifier purposed as an object detector.

YOLO works by dividing the input image into a grid of $S \times S$ cells, where each of these cells is responsible for five bounding boxes predictions that describe the rectangle around the object. It also outputs a confidence score, which is a measure of the certainty that an object was enclosed. Therefore the score does not have any relation with the kind of object present in the box, only with the box's shape.

For each predicted bounding box, a class it's also predicted working just like a regular classifier giving resulting in a probability distribution over all the possible classes. The confidence score for the bounding box and the class prediction combines into one final score that specifies the probability for each box includes a specific type of object. Given these design choices, most of the boxes will have low confidence scores, so only the boxes whose final score is beyond a threshold are kept.

Eq. (2) states the loss function minimized by the training step in the YOLO algorithm.

$$\begin{aligned}
& \lambda_{coord} \sum_{i=0}^{s^2} \sum_{j=0}^B 1_{ij}^{obj} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right] \\
& + \lambda_{coord} \sum_{i=0}^{s^2} \sum_{j=0}^B 1_{ij}^{obj} \left[(\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2 \right] \\
& + \sum_{i=0}^{s^2} \sum_{j=0}^B 1_{ij}^{obj} (C_i - \hat{C}_i)^2 + \lambda_{coord} \sum_{i=0}^{s^2} \sum_{j=0}^B 1_{ij}^{obj} (C_i - \hat{C}_i)^2 + \sum_{i=0}^{s^2} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2
\end{aligned} \tag{2}$$

where 1_i^{obj} indicates if an object appears in cell i and 1_{ij}^{obj} denotes the j^{th} bounding box predictor in cell i responsible for that prediction; x , y , w , h , and C denote the coordinates that represent the center of the box relative to the bounds of the grid cell. The width and height predictions are relative to the whole image. Finally, C denotes the confidence prediction, that is, the IoU between the predicted box and any ground truth box.

Figure 2 describes how the YOLO network process as image. Initially, the input gets passed through a CNN producing the bounding boxes with its perspectives confidences scores and generating the class probability map. Finally, the results of the previous steps are combined to form the final predictions.

2.3 Faster region convolutional neural network

The faster region convolutional neural network [15] is another state-of-the-art CNN-based deep learning object detection approach. In this architecture, the network takes the provided input image into a convolutional network which provides a convolutional feature map. Instead of using the selective search algorithm to identify the region proposals made in previous iterations [18, 19], a separate network is used to learn and predict these regions. The predicted region proposals are then reshaped using a region of interest (ROI) pooling layer, which is then used to

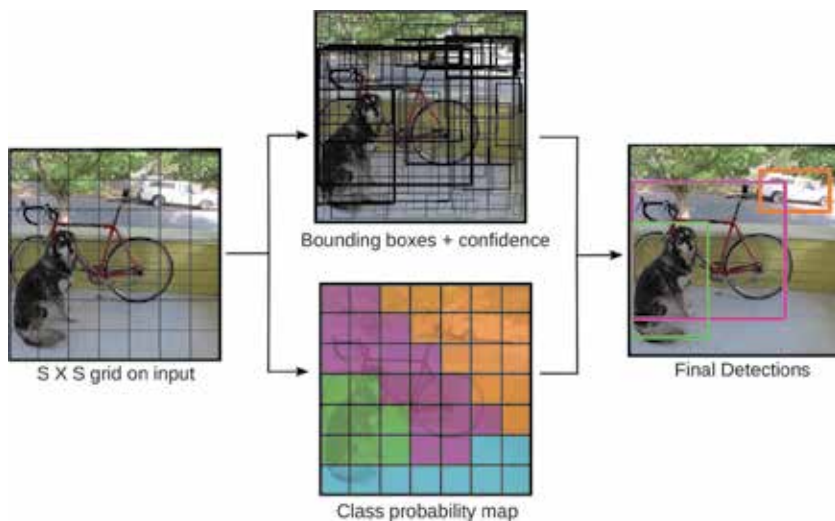


Figure 2. YOLO model detection as a regression problem [17]. Thus the input image is divided into a $S \times S$ grid and for each grid cell, B bounding boxes, confidence for those boxes, and C class probabilities are predicted. These encoded predictions are as an $S \times S \times (B * 5 + C)$ tensor. Figure based on [17].

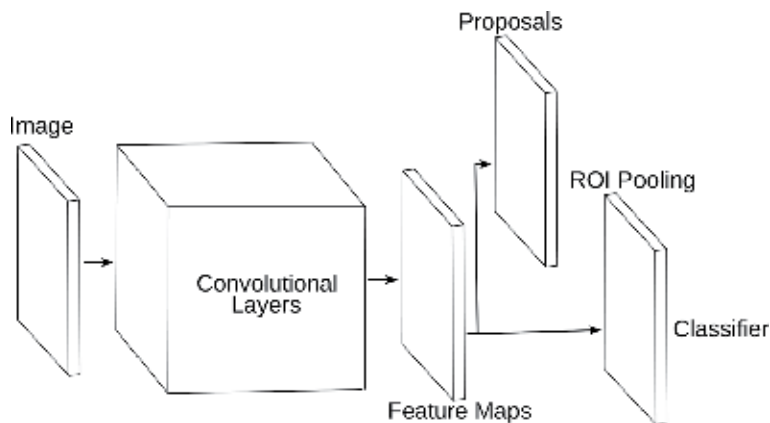


Figure 3. Faster RCNN acts as a single, unified network for object detection [15]. The region proposal network module serves as the “attention” of this unified network. Figure based on [15].

classify the image within the proposed region and predict the offset values for the bounding boxes.

The strategy behind the region proposal network (RPN) training is to use a binary label for each anchor, so the number one will represent the presence of an object and number zero the absence; with this strategy any IoU over 0.7 determines the object’s presence and below 0.3 indicates the object’s absence.

Thus a multitask loss function shown in Eq. (3) is minimized during the training phase.

$$L(\{p_i\}, \{t_i\}) = \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p_i^*) + \lambda \frac{1}{N_{reg}} \sum_i p_i^* L_{reg}(t_i, t_i^*) \quad (3)$$

where i is the index of the anchor in the batch, p_i is the predicted probability of being an object, p_i^* is the ground truth probability of the anchor, t_i is the predicted bounding box coordinate, t_i^* is the ground truth bounding box coordinate, and L_{cls} and L_{reg} are the classification and regression loss, respectively.

Figure 3 depicts the unified network for object detection implemented in the Faster RCNN architecture. Using the recently popular terminology of neural networks with “attention” mechanisms [20], the region proposal network module tells the Fast RCNN module where to look [15].

3. Datasets

A sample of the PASCAL VOC [21] dataset is used to exemplify the use of SSD and RCNN object detection algorithms. A sample of 6 classes of the 20 available were selected. **Table 1** describes the sample size selected for each class.

The images presented in the dataset were randomly divided as follows: 1911 for training corresponding to 50%, 1126 for validation corresponding to 25% and test also corresponding to 25%.

To further illustrate the applications of such algorithms in scientific research, the dataset used for the YOLO network presented in [22] was also analyzed. As described in [22], the dataset is composed of images from three researches that involve behavioral experiments with mice:

Class	Number of images
Bird	811
Cat	1128
Dog	1341
Horse	526
Sheep	357
Total	4163

Table 1.
SSD and RCNN network dataset description.

Dataset	Number of images	Resolution
Ethological evaluation [23]	3707	640 × 480
Automated home-cage [24]	3073	320 × 240
CRIM13 [25]	6842	656 × 490
Total	13,622	

Table 2.
Description of the dataset for use with the YOLO network as earlier used in [22].

- **Ethological evaluation** [23]: This research presents new metrics for chronic stress models of social defeat in mice.
- **Automated home-cage** [24]: This study introduces a trainable computer vision system that allows the automated analysis of complex mouse behaviors; they are eat, drink, groom, hang, micromovement, rear, rest, and walk.
- **Caltech Resident-Intruder Mouse dataset (CRIM13)** [25]: It has videos recorded with superior and synchronized lateral visualization of pairs of mice involved in social behavior in 13 different actions.

Table 2 describes the sample size selected from each of the datasets used in this paper. For the ethological evaluation [23], 3707 frames were used, captured in a top view of the arena of social interaction experiments among mice. For the automated home-cage [24], a sample of 3073 frames was selected from a side view of behavioral experiments. For the CRIM13 [25], a sample of 6842 frames was selected, 3492 from a side view and 3350 from a top view.

The same dataset division used in [22] was also reproduced resulting in 6811 images for training, 3405 for validation, and 3406 for the test.

4. Material and methods for object detection

In this work, the previously described SDD and Faster RCNN networks are compared in the task of localization and tracking of six species of animals in diversified environments. Having accurate, detailed, and up-to-date information about the location and behavior of animals in the wild would improve our ability to study and conserve ecosystems [26]. Additionally, results from the YOLO network, reproduced from [22], to detect and track mice in videos are recorded during

behavioral neuroscience experiments. The task of mice detection consists of determining the location in the image where the animals are present, for each frame acquired.

The computational development here presented was performed on a computer with CPU AMD Athlon II X2 B22 at 2.8GHz, 8GB of RAM, NVIDIA GeForce GTX 1070 8GB GPU, Ubuntu 18.04 LTS as OS, CUDA 9, and CuDNN 7. Our approach used the convolutional networks described in Section 2.

5. Results and conclusion

The results obtained for the SSD and Faster RCNN networks in the experiments were based on the analysis of 4163 images, organized according to the dataset described in Section 3.

Figure 4(a) depicts the increasing development of the mean average precision values in the epochs of training. Both architectures reached high mean average precision (mAP) while successfully minimizing the values of their respective loss functions. The Faster RCNN network presented higher and better stability in precision, which can be seen by the smoothness in its curve. **Figure 4(b)** is a box plot of the time spent by each network on the classification of a single image, whereas the SSD came ahead with 17 ± 2 ms as the mean and standard deviation values, and the Faster RCNN translated its higher computational complexity in the execution time with 30 ± 2 ms as the mean and standard deviation values, respectively.

Table 3 presents more results related to object detection performance. First, it shows the mean average precision, which is the mean value of the average precisions for each class, where average precision is the average value of 11 points on the precision-recall curve for each possible threshold, that is, all the probability of detection for the same class (Precision-Recall evaluation according to the terms described in the PASCAL VOC [21]).

Figure 5 shows some selected examples of object detection results on the dataset used. Each output box is associated with a category label and a softmax score in $[0, 1]$. A score threshold of 0.5 is used to display these images.

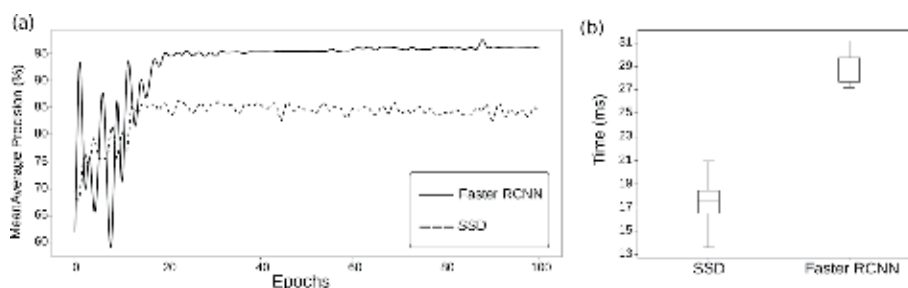


Figure 4. (a) Comparison of the mAP models during the training phase. (b) Time spent to execute each architecture on a single image.

Network	Framework	Mean average precision (%)
Fast RCNN	GluonCV [27]	96.07
SSD	GluonCV [27]	84.35

Table 3. Mean average precision results after 100 epochs of training.

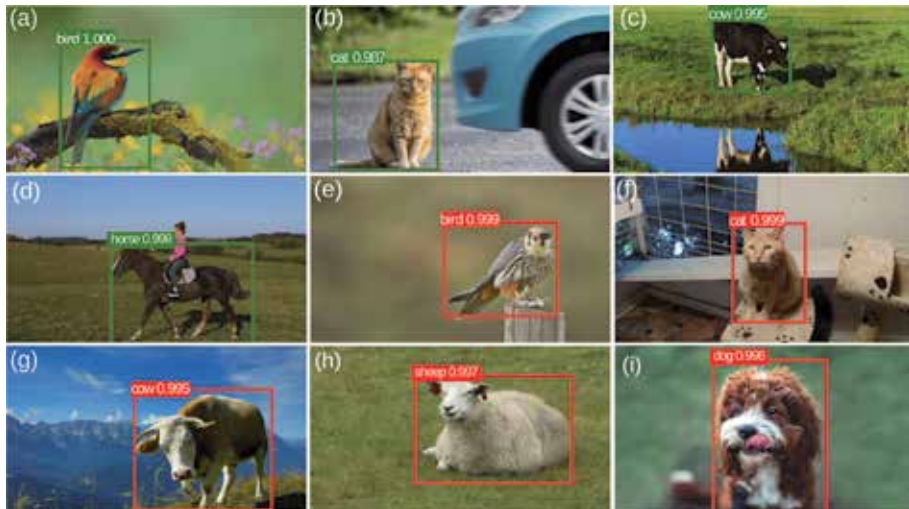


Figure 5. Output examples of the networks. (a)–(d) refer to SSD and (e)–(i) to Faster RCNN.

Our approach, as in [22], also used two versions of the YOLO network to detect mice within three different experimental setups. The results obtained were based on the analysis of 13,622 images, organized according to the dataset described in Section 3.

The first version of YOLO being trained was the YOLO Full network which uses the *Darknet-53* [14] convolutional architecture that comprises 53 convolutional layers. Such a model was trained as described in [17], starting from an ImageNet [28] pre-trained model. Each model requires 235 MB of storage size. We used a batch of eight images, a momentum of 0.9, and weight decay of 5×10^{-4} . The model took 140 hours to be trained.

A smaller and faster YOLO alternative was also trained and named as YOLO Tiny. To speed up the process, this “tiny” version comprises only a portion of the *Darknet-53* [14] resources: 23 convolutional layers. Each model requires only 34 MB of storage size. The network training follows as described in [17], fine-tuning an ImageNet [28] pre-trained model. We used a batch of 64 images, a momentum of 0.9, and weight decay of 5×10^{-4} . The model took 18 hours to be trained.

Figure 6 shows the comparison of the two YOLO models used, YOLO Full and Tiny. **Figure 6(a)** shows high accuracy of the Full architecture with small oscillations of the accuracy curve during the training. In **Figure 6(b)**, the high accuracy is maintained from the earliest times and remains practically unchanged up to the limit number of epochs. Both architectures reached high mean average precision values while successfully minimizing the values of their loss function. The Tiny version of the YOLO network presented better stability in precision, which can be seen by the smoothness in its curve. The results show that the mean average precision reached by this re-implementation was 90.79 and 90.75% for the Full and Tiny versions of YOLO, respectively. The use of the Tiny version is a good alternative for experimental designs that require real-time response.

Figure 6(c) is a bar graph showing the mean time spent on the classification of a single image in both architectures. The smaller size of the Tiny version gets a direct translation in execution time, having 0.08 ± 0.06 s as the mean and standard deviation values, whereas the Full version has 0.36 ± 0.16 s as the mean and standard deviation values, respectively.

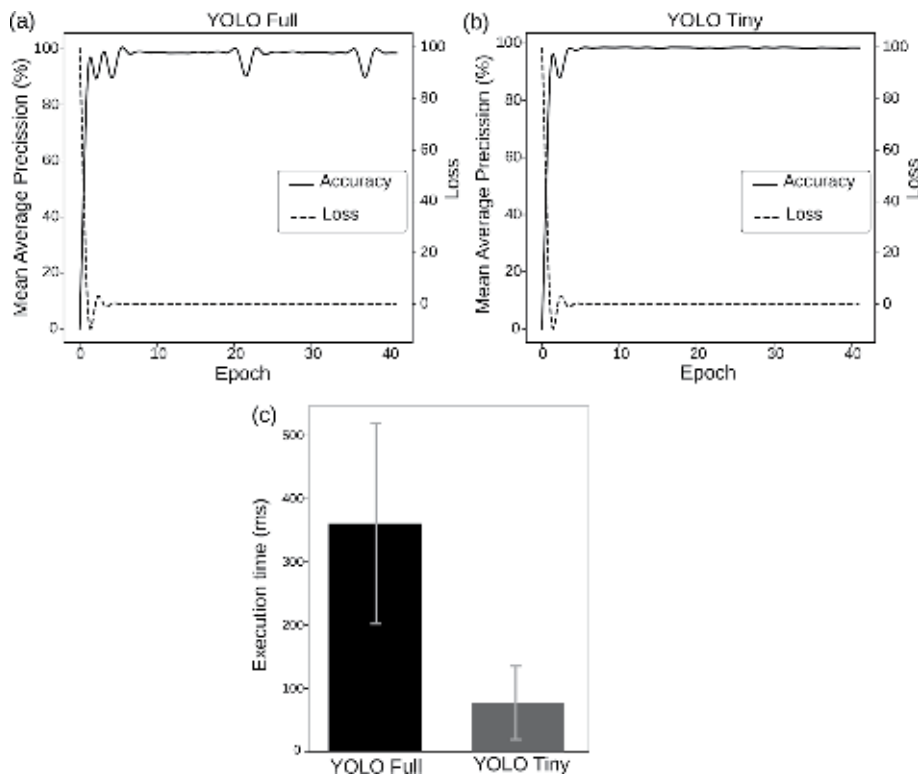


Figure 6. (a) and (b) YOLO architecture evolution in mean average precision and minimization of the loss function during the training phase. (c) GPU time required to obtain the classification of an image in each of the networks.

Given the aforementioned small difference between the two versions of the YOLO object detector, the possibility of designing real-time systems for experiments involving animal tracking is closer to reality with the Tiny architecture. Derived from the smaller demand for computing power, systems where actions are taken while the experiment takes place can be designed without the need for human intervention.

Figure 7 shows some examples, resulting from mice tracking performed on the three different datasets used. Thus, it is possible to verify the operation of mouse tracking in different scenarios. In (a)–(c), the black mouse appears over a white background, the video is recorded from a top view camera in a typical configuration in behavioral experiments. For Figures (d)–(f), the camera was positioned on the side of the experimental box; the algorithm performed the tracking correctly for different positions of the animal. Finally, in Figures (g)–(i), the images were recorded by a top-view camera, and it is possible to verify a large amount of information besides the tracked object. However, the algorithm worked very well, even for two animals in the same arena.

This chapter presented an overview of the machine learning techniques using convolutional neural networks for image object detection. The main algorithms for solving this type of problem were presented: Faster RCNN, YOLO, and SSD. To exemplify the functioning of the algorithms, datasets recognized by the scientific literature and in the field of computer vision were selected, tests were performed, and the results were presented, showing the advantages and differences of each of the techniques. This content is expected to serve as a reference for researchers and those interested in this broadly developing area of knowledge.

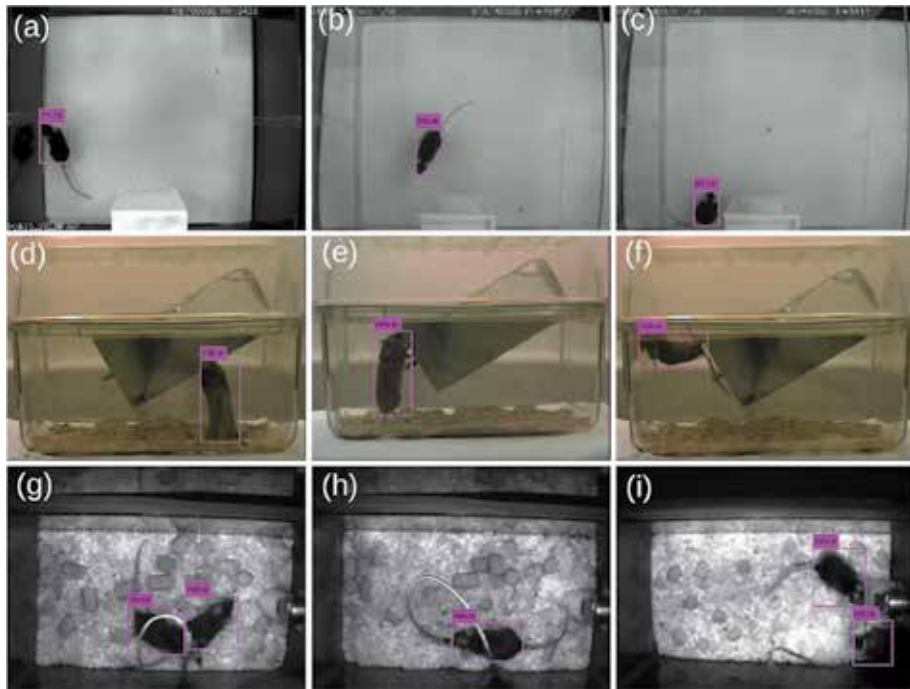


Figure 7. Output examples of the YOLO network. (a)–(c) refer to ethological evaluation [23], (d)–(f) refer to automated home-cage [24], and (g)–(i) refer to CRIM13 [25].

At the moment, we are experiencing the era of machine learning applications, and much should be developed in the coming years from the use and improvement of these techniques. Further improvements in the development of even more specific hardware and fundamental changes in related mathematical theory are expected shortly, making artificial intelligence increasingly present and important to the contemporary world.

Author details


Richardson Santiago Teles de Menezes¹, Rafael Marrocos Magalhaes² and Helton Maia^{1*}

¹ Federal University of Rio Grande do Norte, Brazil

² Federal University of Paraiba, Brazil

*Address all correspondence to: helton.maia@gmail.com

IntechOpen

© 2019 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/3.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. 

References

- [1] Esteva A et al. Dermatologist-level classification of skin cancer with deep neural networks. *Nature*. 2017; **542**(7639):115
- [2] Jayaraman V, Chandrasekhar MG, Rao UR. Managing the natural disasters from space technology inputs. *Acta Astronautica*. 1997;**40**(2–8):291-325
- [3] Leonard M et al. A compound event framework for understanding extreme impacts. *Wiley Interdisciplinary Reviews: Climate Change*. 2014;**5**(1): 113-128
- [4] Kogan FN. Global drought watch from space. *Bulletin of the American Meteorological Society*. 1997;**78**(4): 621-636
- [5] Srinivas S, Sarvadevabhatla RK, Mopuri RK, Prabhu N, Kruthiventi SSS, Venkatesh Babu R. An introduction to deep convolutional neural nets for computer vision. In: *Deep Learning for Medical Image Analysis*. Academic Press; 2017. pp. 25-52
- [6] de Menezes RST, de Azevedo Lima L, Santana O, Henriques-Alves AM, Santa Cruz RM, Maia H. Classification of mice head orientation using support vector machine and histogram of oriented gradients features. In: 2018 International Joint Conference on Neural Networks (IJCNN). IEEE; 2018. pp. 1-6
- [7] Oskoei MA, Gan JQ, Hu H. Adaptive schemes applied to online SVM for BCI data classification. In: 2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE; 2009. pp. 2600-2603
- [8] Hearst MA, Dumais ST, Osuna E, Platt J, Scholkopf B. Support vector machines. *IEEE Intelligent Systems and their Applications*. 1998;**13**(4):1828
- [9] Pan WD, Dong Y, Wu D. Classification of malaria-infected cells using deep convolutional neural networks. In: *Machine Learning: Advanced Techniques and Emerging Applications*. 2018. p. 159
- [10] Goodfellow I, Bengio Y, Courville A. *Deep Learning*. MIT Press; 2016
- [11] Deng L, Hinton G, Kingsbury B. New types of deep neural network learning for speech recognition and related applications: An overview. In: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing. IEEE; 2013. pp. 8599-8603
- [12] Kriegeskorte N. Deep neural networks: A new framework for modeling biological vision and brain information processing. *Annual Review of Vision Science*. 2015;**1**:417-446
- [13] Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu C-Y, et al. SSD: Single shot multibox detector. In: *European Conference on Computer Vision*. Cham: Springer; 2016. pp. 21-37
- [14] Redmon J, Farhadi A. Yolov3: An Incremental Improvement. *arXiv*; 2018
- [15] Ren S, He K, Girshick R, Sun J. Faster r-cnn: Towards real-time object detection with region proposal networks. In: *Advances in Neural Information Processing Systems*. 2015. pp. 91-99
- [16] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*; 2014
- [17] Redmon J, Divvala S, Girshick R, Farhadi A. You only look once: Unified, real-time object detection. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016. pp. 779-788

- [18] Girshick R, Donahue J, Darrell T, Malik J. Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2014. p. 580587
- [19] Girshick R. Fast r-cnn. In: Proceedings of the IEEE International Conference on Computer Vision. 2015. pp. 1440-1448
- [20] Chorowski JK, Bahdanau D, Serdyuk D, Cho K, Bengio Y. Attention-based models for speech recognition. In: Advances in Neural Information Processing Systems. 2015. pp. 577-585
- [21] Everingham M et al. The Pascal visual object classes (VOC) challenge. International Journal of Computer Vision. 2010;**88**(2):303-338
- [22] Peixoto HM, Teles RS, Luiz JVA, Henriques-Alves AM, Santa Cruz RM. Mice Tracking Using the YOLO Algorithm. Vol. 7. PeerJ Preprints; 2019. p. e27880v1
- [23] Henriques-Alves AM, Queiroz CM. Ethological evaluation of the effects of social defeat stress in mice: Beyond the social interaction ratio. Frontiers in Behavioral Neuroscience. 2016;**9**:364
- [24] Jhuang H et al. Automated home-cage behavioural phenotyping of mice. Nature Communications. 2010;**1**:68
- [25] Burgos-Artizzu XP, Dollár P, Lin D, Anderson DJ, Perona P. Social behavior recognition in continuous video. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition. IEEE; 2012. pp. 1322-1329
- [26] Norouzzadeh MS et al. Automatically identifying, counting, and describing wild animals in camera-trap images with deep learning. Proceedings of the National Academy of Sciences of the United States of America. 2018;**115**(25):E5716-E5725
- [27] Guo J, He H, He T, Lausen L, Li M, Lin H, et al. GluonCV and GluonNLP: Deep Learning in Computer Vision and Natural Language Processing. arXiv preprint arXiv:1907.04433
- [28] Deng J, Dong W, Socher R, Li L-J, Li K, Fei-Fei L. ImageNet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. IEEE; 2009. pp. 248-255
- [29] Chen X-L et al. Remote sensing image-based analysis of the relationship between urban heat island and land use/cover changes. Remote Sensing of Environment. 2006;**104**(2):133-146

Prediction of Wave Energy Potential in India: A Fuzzy-ANN Approach

Soumya Ghosh and Mrinmoy Majumder

Abstract

The conversion efficiency of wave energy converters is not only unsatisfactory but also expensive, which is why the popularity of wave energy as an alternative to conventional energy sources is subjacent. This means that besides wave height and period, there are many other factors which influence the amount of “utilizable” wave energy potential. The present study attempts to identify these important factors and predict power potential as a function of these factors. Accordingly, a polynomial neural network was utilized, and fuzzy logic was applied to identify the most important factors. According to the results, wave height was found to have the maximum importance followed by wave period, water depth, and salinity. In total, 12 different neural network models were developed to predict the same output, among which the model with all of the 4 inputs was found to have optimal performance.

Keywords: wave energy, power potential, fuzzy logic, artificial neural network

1. Introduction

Wave energy is considered as one of the most promising marine renewable resources, with global worldwide wave power estimated at around 2 TW [1]. Several renewable energy-generating sources such as wave power, tides, and current which are associated with marine have always been misunderstood though it has strong predictability and other physical properties [2]. Wave energy presents a number of advantages with respect to other CO₂-free energy sources—high-power density, a relatively high utilization factor, and last, but not the least, low environmental and visual impact [3]. Wave energy resource assessments fall into two categories. Renewable energy is continually available, but due to the complexity of conversion and storage procedures and uncertainty in their availability, such sources of energy have till now been used with caution [4]. Most of the drawbacks were found to vary with location. Some of the advantages are high-energy density [5] and good predictability as well as reduced negative environmental impacts on beaches [6], the marine ecosystems [7], and the wave climate [8]. If we consider the energy consumption, then India ranks four just after the United States, China, and Russia. Electricity consumption in India is expected to rise to around 2280 BkWh by 2021–2022 and to around 4500 BkWh by 2031–2032 [9]. Various methods have been used to estimate wave power potential, but most of them are subjective and

linear and cannot be adapted to various situations. In the present study, a new method for estimating wave power potential is proposed; it is an objective, cognitive, and unbiased method which estimates the wave energy potential of a location considering the most important nonlinearity.

1.1 Objective

The objective of my study was multi-criteria decision-making (MCDM) methods like fuzzy logic decision-making (FLDM), and cognitive methods like group method of data handling (GMDH) were utilized which incorporate both objectivity and adaptability in the predictive method. As far as the authors know, fuzzy-based MCDM cascaded with GMDH has not previously been used to estimate wave power potential.

1.2 Future aspect

Cognitive study of site variety for wave energy power plant was infrequently attempted, and that is why the authors of the present study tried to propose a novel methodology in selection of most favorable sites for wave energy generation by MCDM and ANN technologies. Finally, the consideration of another multi-criteria decision-making method instead of fuzzy for evaluating the decision alternatives and the comparison of the results with the ones of the present study could represent a subject for future research.

2. Methodology

The new method comprises two steps:

- I. Application of MCDM, i.e., FLDM, to find the weight of importance
- II. Application of GMDH to provide a predictive infrastructure for making the method resource independent

Sections 2.1 and 2.2 discuss the strengths, weaknesses, and applicability of the method in this study.

2.1 Fuzzy logic

Fuzzy set theory was first introduced as the mathematical programming of the primary works [10]. Fuzzy logic resembles human analysis in its use of inaccurate information to create decisions. Many such problems can be formulated as the minimization of functionals defined over a class of admissible domains. Nondeterministic condition deceits both design variables and allowable limits. A stochastic problem can be transformed into its deterministic form by using expected value and the chance-constrained programming technique. Thus, fuzzy mathematical formulation could be a substitution of this [11]. The advantage of fuzzy logic lies in the depiction of importance for similarly important factors by fuzzy scale, and disadvantages are only found in the qualitative variables which can be used. Fuzzy logic could be applied to such problems as determining a suitable location for a biogas plant, geothermal potential, and the control design of power management [12].

2.2 Group method of data handling (GMDH)

The self-adaptive heuristic ANN based method is one of the learning machine approaches based on the polynomial theory of complex systems, designed by Ivakhnenko (1971). Generally, the first-order (linear) Kolmogorov-Gabor polynomial including n nodes can be used as transfer function [13]:

$$Y = f(x_1, x_2, \dots, x_n) = a_0 + a_1x_1 + a_2x_2 + \dots + a_nx_n \quad (1)$$

where Y is the middle candidate solution, x is a given initial solutions, and a is the vector of coefficients or weights. New middle candidate solutions can be obtained according to the inputs of the current layer and the transfer function.

Self-organizing models of optimal convolution is constructed by inductive algorithm which was supervised by original GMDH. It is totally based on the input-output relationships of a given dataset, without the need for user interference. The GMDH network is known as a self-organized approach that solves various complex problems in nonlinear systems [14].

The main advantage of the GMDH model is in building analytical functions within feed-forward networks based on quadratic polynomials whose weighting coefficients are obtained using the regression method [15].

3. Methodology

The methods were used to estimate the rank of importance of the parameters based on the study objective shown in **Figure 1**. The procedures to estimate the wave power potential involve the application of the MCDM method to estimate the priority value of the parameters and GMDH to reveal the relationship between the input and output parameters.

The MCDM methodology deduces the importance of the parameters based on their citation frequency, expert inputs, and availability of data. All three methods were used to estimate the rank of importance of the parameters based on the study objective and on criteria like efficiency and cost. The detailed hierarchy of MCDM methodologies is shown in **Figure 2**. The model uses fuzzy logic to determine

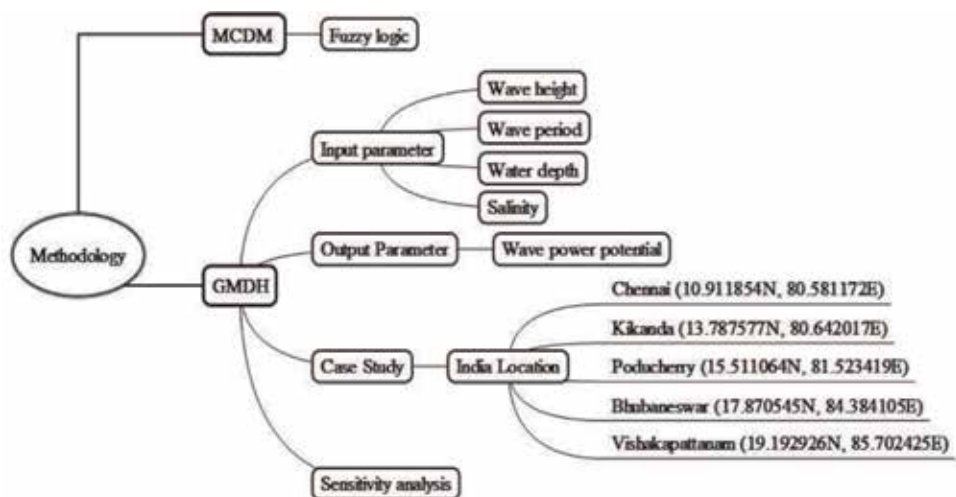


Figure 1.
 Schematic diagram of present investigation.

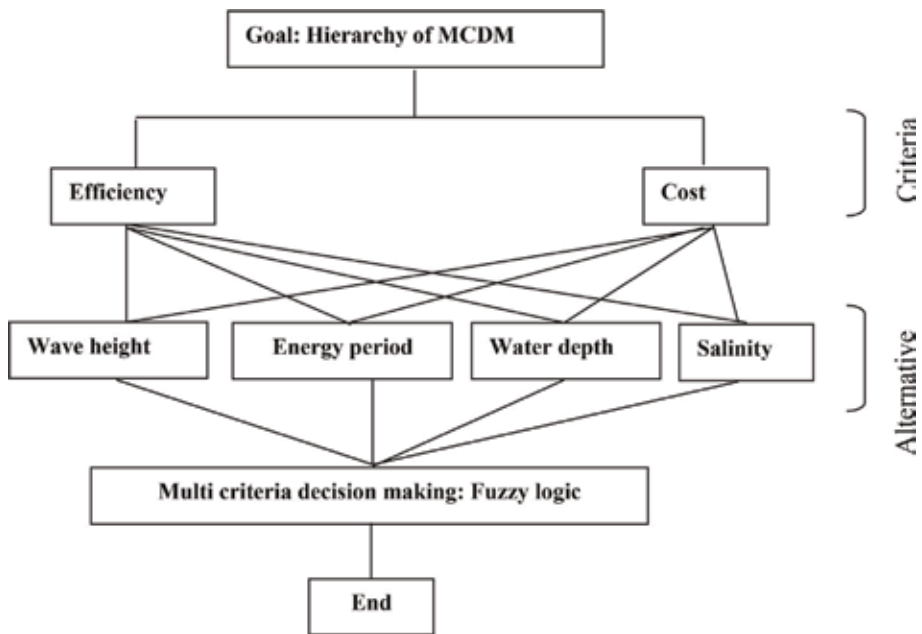


Figure 2.
Figure showing the hierarchy of the MCDM methodologies.

weights of importance as derived from the rank of importance and the aggregation method.

The model uses fuzzy logic to determine weights of importance as derived from the rank of importance and the aggregation method.

3.1 Case study

Figure 3 presents the geographical locations of five points (locations 1–5), which are used to define the wave energy potential of different locations. The data of wave

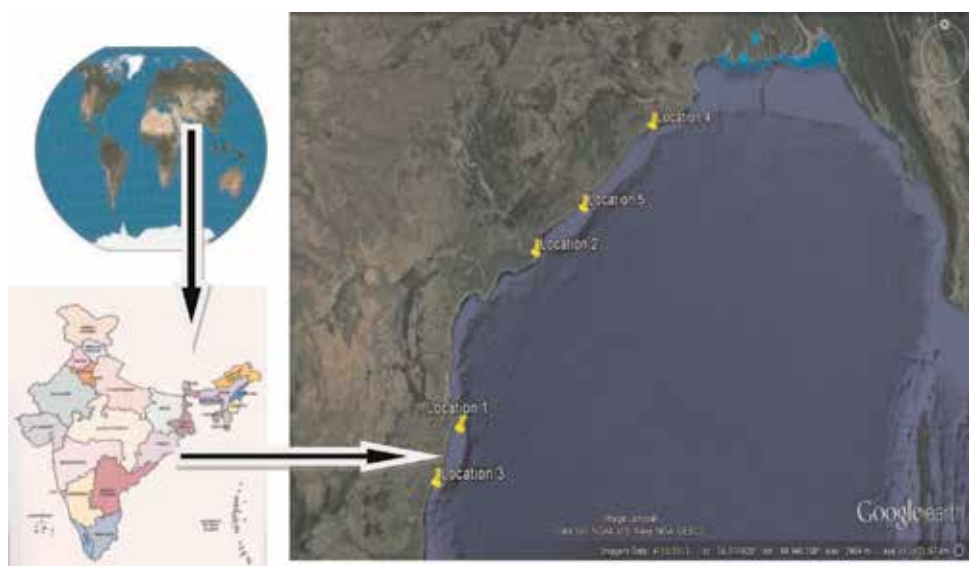


Figure 3.
Locations of the study area.

Parameters	Indian scenario				
	Location 1: Chennai (10.911854, 80.581172E)	Location 2: Kikanda (13.787577, 80.642017E)	Location 3: Puducherry (15.511064 N, 81.523419E)	Location 4: Bhubaneswar (17.870545 N, 84.384105E)	Location 5: Visakhapatnam (19.192926 N, 85.702425E)
Wave height (m)	3.5	2.6	3.1	2.4	1.6
Wave period (s)	6.2	7.4	6.5	8.2	8.4
Water depth (m)	3000	2500	700	1400	2200
Salinity (psu)	34.7	33.5	35.6	32.8	33.2

Table 1.
 Magnitude of the parameter with respect to the selected location.

height, wind speed, and water depth for the five locations were collected from the National Data Buoy Center. The most recent reanalysis dataset was produced by the European Centre for Medium-Range Weather Forecasts (ECMWF) [16]. The five locations in the Bay of Bengal (BoB) are used to estimate the wave power potential in an Indian scenario. The wave height (H_s) and wave period (T_e) are obtained from the spectral moment as shown in Eqs. (2) and (3):

Significant of wave height (H_s)

$$H_s = 4\sqrt{m_0} \quad (2)$$

$$\text{Energy period } (T_e) = \frac{m_{-1}}{m_0} \quad (3)$$

More details about locations 1–5 are provided in **Table 1**, where the corresponding water depth and the geographical coordinates are indicated for each of the five selected locations.

3.2 Development of the cognitive method

In total, 12 GMDH-based models were developed with the same 4 inputs and 1 output as wave power potential. The numbers of inputs were varied from three to five where transformation of input and output data was conducted by the use of tangent and cube root functions. The top three parameters were identified with the help of the fuzzy logic MCDM method. According to the EPI, 3 models which were found to be better than the 12 models developed for the present study were selected for further validation.

The performance of all 12 models was analyzed by aean absolute error (MAE) [17] and correlation (R) [18]. The former metrics are known to be inversely proportional to model accuracy, whereas the other metrics are directly proportional to model performance. The performance of the model during the checking (c) or testing phase is a more important indicator of model reliability than the performance of the model in the training (t) phase [19]. Performance of the three selected models was tested for reliability with the help of root-mean-square error (RMSE), mean relative error (MRE), correlation (R), and percent bias (PBIAS) between the predicted and observed data. The equivalent performance index (EPI) was prepared to represent the performance of the models (see Eq. (4)).

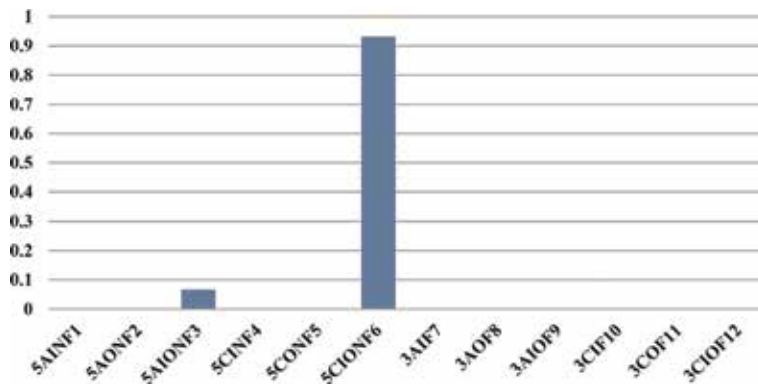


Figure 4.
The 12 models developed for prediction of suitable site selection.

$$EPI = \frac{R}{MAE + MRE + RMSE + PBIAS} \quad (4)$$

The names of the models considered in the study are given in **Figure 4**. The nomenclature was prepared by placing the number of inputs as the first letter followed by the initial letter of the training algorithm, the data transformation function, and lastly the model number.

3.3 Sensitivity analysis

The sensitivity of the better model among the models considered in the study was also tested to verify whether the importance of the input parameters are imbedded into the model result.

4. Results and discussion

Figure 5 shows the score and the rank of the criteria based on the fuzzy logic method. Literature surveys and wave heights were found to be the most important criterion and alternative, respectively, whereas data availability and salinity were identified as the least important criterion and alternative, respectively. According to the results from the MCDM, it can be observed that the wave height (0.4084) and salinity (0.3897) have the highest and lowest importance, respectively, with respect to location selection for wave power plants in **Figure 5**. The performance analysis of the 12 models prepared for prediction is depicted in **Figure 4**.

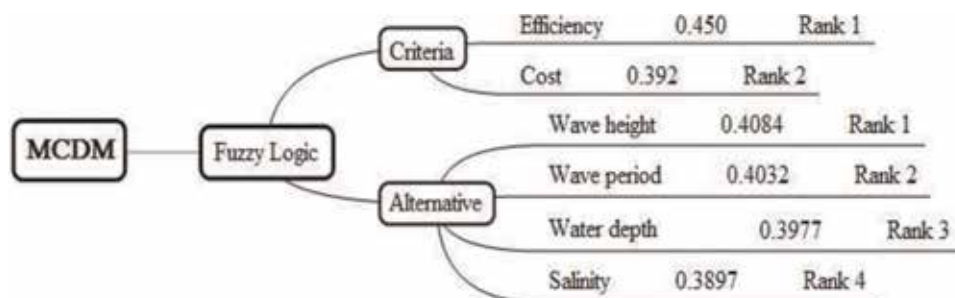


Figure 5.
Fuzzy logic results.

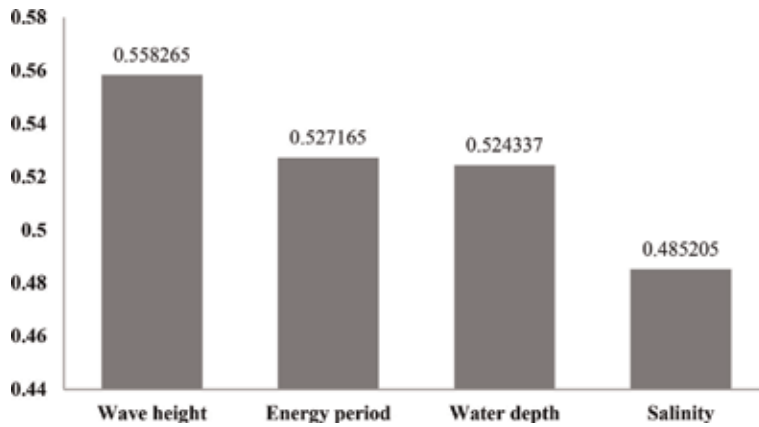


Figure 6.
 Figure showing the sensitivity analysis of input variable.

The result of the sensitivity analysis is shown in **Figure 6**, and case study results are shown in **Table 2**.

Figure 7 shows the comparison of predicted and observed output during the training and testing. The performance analysis of the 12 models revealed that the developed model no. “5CIONF6” was the most consistent model among all the models in the study. The most important models were trained with GMDH, the input and output was transformed by the cube root function, and all five variables were used as input.

Location	Wave height	Energy period	Water depth	Salinity	Indicator	Rank
Location 1: Chennai	0.26515	0.16893	0.30612	0.20435	0.01177	1
Location 2: Kikanda	0.19696	0.20163	0.25510	0.19729	0.00724	3
Location 3: Puducherry	0.23484	0.17711	0.07142	0.20965	0.00975	2
Location 4: Bhubaneswar	0.18181	0.22343	0.14285	0.19316	0.00780	4
Location 5: Visakhapatnam	0.12121	0.22888	0.22448	0.19552	0.00261	5

Table 2.
 The performance analysis of the locations for wave energy power potential in the case study area.

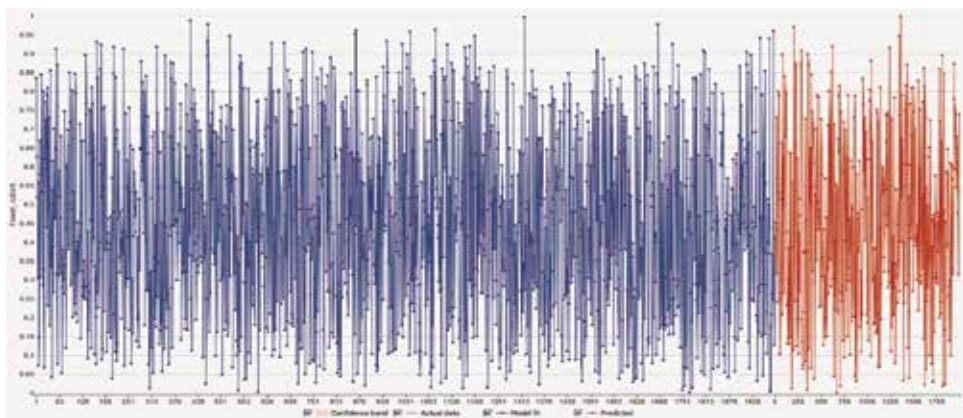


Figure 7.
 The comparison of actual and predicted value of the index both with training and testing data.

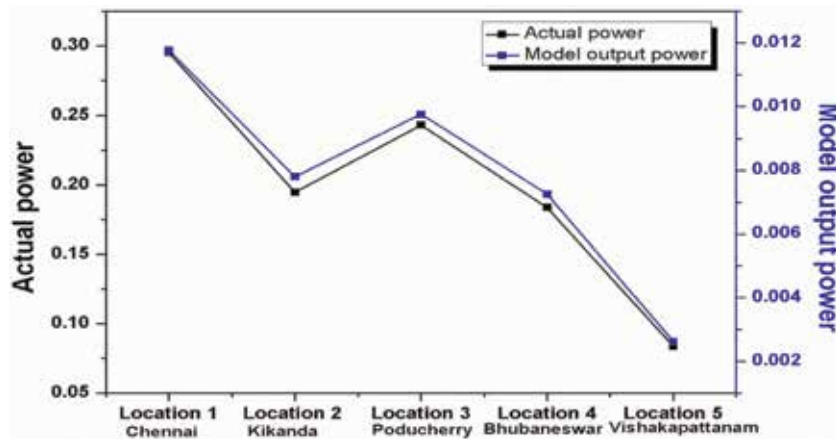


Figure 8.
Location-based model output indicator vs. actual power potential.

Figure 6 depicts that the sensitivity analysis of the model in wave height is maximum and least important of salinity. **Figure 8** depicts the prediction of power potential in form locations of the eastern coastline of India, as predicted in the selected model output with combined the actual power of wave power equation. This model was satisfactory in our objective.

4.1 Study area

In the investigation, the quality of five locations for installation of wave stations was determined by the new methodology. Location 1 (Chennai) has greater practicality than four alternative locations for utilization of wave energy potential. The wave power potential per meter of wave crest of the five locations was also calculated as recommended by [20] in Eq. (5).

$$P = \frac{\rho g^2}{64\pi} H_{mo}^2 T_e \approx \left(0.5 \frac{kw}{m^3s}\right) H_{mo}^2 T_e \quad (5)$$

where P is the wave power per unit crest length (kw/m), ρ is the sea water density (kg/m^3), g is the gravitational acceleration (m/s^2), H_s is the significant wave height (m), and T_e is the energy period (s).

For the Indian scenario, the power potentials of five locations were found to be 12385.224 (kw), 8157.452 (kw), 10186.215 (kw), 7702.158 (kw), and 3506.674 (kw). The model output values, locations 1–5, were found to be equal to 0.011777, 0.007245, 0.009758, 0.007801, and 0.002619, respectively. The power potential and the model value were found to be consistent with each other. According to the graph, the model output power and locations are based on the normalized value of power potential shown in **Figure 8**. The values were 0.295324188, 0.194513469, 0.242889081, 0.183657038, and 0.083616223, by locations 1–5, respectively.

5. Conclusion

The present study attempts to predict the wave energy potential of different coastal regions with the help of the four most relevant factors. The study utilized fuzzy MCDM and GMDH models to develop a framework to predict the wave

energy potential. In total, four factors were identified as the most important in regard to the calculation of wave energy potential, as found from the literature survey. In total, 12 different models were developed by varying the inputs within these 4 factors and power potential as output. The data representing various scenarios was generated and used to train the models. The arc tangent function was used in six cases to transfer the data of either input or output or both. Performance metrics like RMSE, MAE, PBIAS, and R were used to find the equivalent performance of the models. The model with all the factors as input was found to be most efficient among all the other 11 models. The accuracy of the model was found to be above 99.99%. The power potential of five different locations on the Indian coastal belt was used as a case study. The model output and the result from the power potential equation were compared and found to be coherent with each other, although magnitude of the results is well apart.

Nomenclature

ANN	artificial neural network
MCDM	multi-criteria decision-making
FLDM	fuzzy logic decision-making method
GMDH	group method of data handling
NSE	Nash-Sutcliffe model efficiency coefficient
PBIAS	percent bias
RSR	RMSE-observation standard deviation ratio
R	correlation
PI	performance index

Author details

Soumya Ghosh* and Mrinmoy Majumder
School of Hydro-Informatics Engineering, National Institute of Technology
Agartala, Tripura, India

*Address all correspondence to: soumyaee@gmail.com

IntechOpen

© 2019 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/3.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. 

References

- [1] Gunn K, Stock-Williams C. Quantifying the global wave power resource. *Renewable Energy*. 2012;**44**: 296-304
- [2] Henfridsson U, Neimane V, Strand K, Kapper R, Bernhoff H, Danielsson O, et al. Wave energy potential in the Baltic Sea and the Danish part of the North Sea, with reflections on the Skagerrak. *Journal of Renewable Energy*. 2007; **32**(12):2069-2084
- [3] Mackay EB, Bahaj AS, Challenor PG. Uncertainty in wave energy resource assessment. Part 2: Variability and predictability. *Journal of Renewable Energy*. 2010;**35**(8):1809-1819
- [4] Xie WT, Dai YJ, Wang RZ, Sumathy K. Concentrated solar energy applications using Fresnel lenses: A review. *Journal of Renewable and Sustainable Energy Reviews*. 2011;**15**(6): 2588-2606
- [5] Iglesias LM, Carballo R, Castro A, Fraguera JA, Frigaard P. Wave energy potential in Galicia (NW Spain). *Journal of Renewable Energy*. 2009;**34**:2323-2333
- [6] Abanades J, Greaves D, Iglesias G. Wave farm impact on the beach profile: A case study. *Journal of Coastal Engineering*. 2014;**86**:36-44
- [7] Azzellino A, Conley D, Vicinanza D, Kofoed JP. Marine renewable energies: Perspectives and implications for marine ecosystems. *The Scientific World Journal*. 2013;**2013**:1-3
- [8] Veigas M, Ramos V, Iglesias GA. Wave farm for an island: Detailed effects on the nearshore wave climate. *Journal of Energy*. 2014;**69**:801-812
- [9] Garg P. Energy scenario and vision 2020 in India. *Journal of Sustainable Energy and Environment*. Aug 2012;**3** (1):7-17
- [10] Zimmermann HJ. *Fuzzy Set Theory and its Applications*. 2nd ed. Boston, Dordrecht, London: Kluwer Academic Publishers; 1991
- [11] Sevkli M. An application of the fuzzy ELECTRE method for supplier selection. *Journal of International Journal of Production Research*. 2010; **48**(12):3393-3405
- [12] Franco C, Bojesen M, Leth Hougaard J, Nielsen K. A fuzzy approach to a multiple criteria and geographical information system for decision support on suitable locations for biogas plants. *Journal of Applied Energy*. 2015;**140**:304-315
- [13] Anastasakis L, Mort N. The development of self-organization techniques in modelling: A review of the group method of data handling (GMDH). Technical Report. University of Sheffield, Department of Automatic Control and Systems Engineering; 2001
- [14] Hwang HS. Fuzzy GMDH-type neural network model and its application to forecasting of mobile communication. *Journal of Computers and Industrial Engineering*. 2006;**50**(4): 450-457
- [15] Kalantary F, Ardalani H, Nariman-Zadeh N. An investigation on the Su-N SPT correlation using GMDH type neural networks and genetic algorithms. *Journal of Engineering Geology*. 2009; **109**(1):144-155
- [16] de Antonio FO. Wave energy utilization: A review of the technologies. *Journal of Renewable and Sustainable Energy Reviews*. 2010;**14**(3):899-918
- [17] Willmott CJ, Matsuura K. Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model

performance. *Journal of Climate Research*. 2005;**30**(1):79-82

[18] Pascual-González J, Guillén-Gosálbez G, Mateo-Sanz JM, Jiménez-Esteller L. Statistical analysis of the EcoInvent database to uncover relationships between life cycle impact assessment metrics. *Journal of Cleaner Production*. 2016;**112**:359-368

[19] Noori N, Kali L. Coupling SWAT and ANN models for enhanced daily stream flow prediction. *Journal of Hydrology*. 2016;**533**:141-151

[20] Ghosh S, Chakraborty T, Saha S, Majumder M, Pal M. Development of the location suitability index for wave energy production by ANN and MCDM techniques. *Renewable and Sustainable Energy Reviews*. 2016;**59**:1017-1028

Deep Learning Training and Benchmarks for Earth Observation Images: Data Sets, Features, and Procedures

*Mihai Datcu, Gottfried Schwarz
and Corneliu Octavian Dumitru*

Abstract

Deep learning methods are often used for image classification or local object segmentation. The corresponding test and validation data sets are an integral part of the learning process and also of the algorithm performance evaluation. High and particularly very high-resolution Earth observation (EO) applications based on satellite images primarily aim at the semantic labeling of land cover structures or objects as well as of temporal evolution classes. However, one of the main EO objectives is physical parameter retrievals such as temperatures, precipitation, and crop yield predictions. Therefore, we need reliably labeled data sets and tools to train the developed algorithms and to assess the performance of our deep learning paradigms. Generally, imaging sensors generate a visually understandable representation of the observed scene. However, this does not hold for many EO images, where the recorded images only depict a spectral subset of the scattered light field, thus generating an indirect signature of the imaged object. This spots the load of EO image understanding, as a new and particular challenge of Machine Learning (ML) and Artificial Intelligence (AI). This chapter reviews and analyses the new approaches of EO imaging leveraging the recent advances in physical process-based ML and AI methods and signal processing.

Keywords: Earth observation, synthetic aperture radar, multispectral, machine learning, deep learning

1. Introduction

This chapter introduces the basic properties, features, and models for very specific Earth observation (EO) cases recorded by very high-resolution (VHR) multispectral, Synthetic Aperture Radar (SAR), and multi-temporal observations. Further, we describe and discuss procedures and machine learning-based tools to generate large semantic training and benchmarking data sets. The particularities of relative data set biases and cross-data set generalization are reviewed, and an algorithmic analysis frame is introduced. Finally, we review and analyze several examples of EO benchmarking data sets.

In the following, we describe what has to be taken into account when we want to benchmark the classification results of satellite images, in particular the classification capabilities, throughputs, and accuracies offered by modern machine learning and artificial intelligence approaches.

Our underlying goal is the identification and understanding of the semantic content of satellite images and their application-oriented interpretation from a user perspective. In order to determine the actual performance of automated image classification routines, we need to find and select test data and to analyze the performance of our classification and interpretation routines in an automated environment.

A particular point to be understood is what type of data exists for remote sensing images that we want to classify. We are faced with long processing chains for the scientific analysis of image data, starting with uncalibrated “raw” sensor data, followed by dedicated calibration steps, subsequent feature extraction, object identification and annotation, and ending with quantitative scientific research and findings about the processes and effects being monitored in the geophysical environment of our planet with respect to climate change, disaster risks, crop yield predictions, etc.

In addition, we have to mention that free and open-access satellite products have revolutionized the role of remote sensing in Earth system studies. In our case, the data being used are based on multispectral (i.e., multi-color) sensors such as Landsat with 7 bands, Sentinel-2 [4] with 13 bands, Sentinel-3 with 21 bands, and MODIS with 36 bands but also SAR sensors such as Sentinel-1 [6], TerraSAR-X [26] or RADARSAT. For a better understanding of their imaging potential, we will describe the most important parameters of these images. For multispectral sensors, there exists several well-known and publicly available land cover benchmarking data sets comprising typical remote sensing image patches, while comparable SAR benchmarking data sets are very scarce and dedicated.

The main aspects being treated are:

- ML paradigms to support the semantic annotation of very large data sets, that is, using hybrid methods integrating Support Vector Machines (SVMs), Bayesian, and Deep Neural Networks (DNNs) algorithms in active learning paradigms by using initially small and controllable training data sets, and progressively growing the volume of labeled data by transfer learning.
- Proposing solutions to the semantic aspects of the spatial annotations for different sensor resolutions and spatial scales.
- Discussing the implications of the sensory and semantic gaps.

In this chapter, we assume that we can rely on already processed data with sufficient calibration accuracy and accurate annotation allowing us to understand all imaging parameters and their accuracy. We also assume that we can profit from reliably documented image data and that we can continue with data analytics for image understanding and high-level interpretation without any further precautions.

The latter steps have to be organized systematically in order to guarantee reliable results. A common strategy is to split these tasks into three phases, namely initial basic software functionality testing; second, training and optimizing of the software parameters by means of selected reference data, and finally, benchmarking of the overall software functionality such as processing speed and attainable results. This systematic approach leads to quantifiable and comparable results as described in the following sections.

During the last years, the field of deep learning had an explosive expansion in many domains with predominance in computer vision, speech recognition, and text analysis. For example, during 2019, more than 500 articles per month have been published in the field of deep learning. Thus, any reports on the state of the art hardly can follow this development. In Ref. [1], published in January 2019, more than 330 references were analyzed reviewing the theoretical and architectural aspects for Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), including Long Short-Term Memories (LSTMs) and Gated Recurrent Units (GRUs), Auto-Encoders (AEs), Deep Belief Networks (DBNs), Generative Adversarial Networks (GANs), and Deep Reinforcement Learning (DRL). The review paper [1] also summarizes 20 deep learning frameworks, two standard development kits, 49 benchmark data sets in all domains, from which three are dedicated to hyperspectral remote sensing. In addition, Ball et al. [2] describe the landscape of deep learning from all perspectives, theory, tools, applications, and challenges as of 2017. This article analyzes 419 references. A more recent overview from April 2019 [3] summarizes more than 170 references reporting on applications of deep learning in remote sensing.

2. Remote sensing images

Typical remote sensing images acquired by aircraft or satellite platforms can be characterized based on the operational capabilities of these platforms (such as their flight path, their capabilities for instrument pointing, and the on-board data storage and data downlink capacities), the type of instruments and their sensors (such as optical images with distinctive spectral bands [4, 5] or radar images such as synthetic aperture radars [6]), and opportunities for the repetitive acquisition of geographically overlapping image time series (for instance, for vegetation monitoring to predict optimal crop harvesting dates).

Current images can provide raw data with more than eight bits per sample, can perform initial data processing and annotation already on board, and can downlink compressed data with error correcting codes. After downlinking the image data to ground stations, the received data will be stored and processed by dedicated computing facilities. A common remote sensing strategy is to perform a systematic level-by-level processing (generating so-called products that comprise image data together with metadata documenting relevant image acquisition and processing parameters).

A common conventional approach is to follow a unified concept, where Level-0 products contain unprocessed but re-ordered detector data; Level-1 data represent radiometrically calibrated intensity images, while Level-2 data are geometrically corrected and map-projected data. Level-3 data are higher level products such as semantic maps or overlapping time-series data. In general, users have access to different product levels and can access and download selected products from databases via image catalogs and so-called quick-look (also called thumb-nail) images.

Some additional products have to be generated interactively by the users. Typical examples are image content classifications and trend analyses following mathematical approaches. Today, these interactive steps migrate from purely interactive and simple tools to commonly accepted machine learning tools. At the moment, the majority of machine learning tools use “deep” learning approaches; here, the problem is decomposed into several layers to find a good representation of image content categories [7]. These aspects will be dealt with in more detail in Section 4.

What we have to outline first are some important parameters of remote sensing images. One critical point of typical remote sensing images is their enormous size

calling for big data environments with powerful processors and large data stores. A second important point is the geometrical and radiometrical resolution of the image pixels, resulting in different target types that can be identified and discriminated during classification. While the typical pixel-to-pixel spacing of air-borne cameras corresponds to centimeters on the ground, space-borne instruments with high resolution flown on low polar orbits mostly lie in the range of half a meter to a few meters. In contrast, imaging from more distant geostationary or geosynchronous orbits results in low-resolution images. As for the number of brightness levels of each pixel, modern cameras often provide more than eight bits of resolution. **Table 1** shows some typical parameters of current satellites with imaging instruments.

Further, the pixels of an image can be complemented by additional information obtained by feature extraction and automated object identification (used as image content descriptors) as well as publicly available information from auxiliary external databases (e.g., geographical references or geophysical parameters). These data allow the provision of accurate quantitative results in physical units; however, one has to be aware of the fact that while many phenomena become visible, some internal relationships may remain invisible without dedicated additional investigations. **Table 1** shows some typical parameters of current satellite images.

In addition to the standard image products as described above, any additional automated or interactive analysis and interpretation of remote sensing images calls for intelligent strategies how to quickly select distinct and representative images, how to generate image time series, to extract features, to identify objects, to recognize hitherto hidden relationships and correlations, to exploit statistical descriptive models describing additional relationships, and to apply techniques for the annotation and visualization of global/local image properties (that have to be stored and administered in databases).

While typical traditional image content analysis tools either use full images, sequences of small image patches, collections of mid-size image segments or countless individual pixels together with routines from already established toolboxes (e.g., Orfeo [9]), or advanced machine learning approaches exploiting innovative machine learning strategies, as for instance, transfer learning [8] or the use of adversarial networks [10]. However, any use of advanced approaches requires the

High-resolution imaging instruments	Optical cameras and spectrometers	SAR instruments
Image size (typ. lines × columns)	$10^4 \times 10^4$ pixels	$10^4 \times 10^4$ pixels
Bands (typ.)	300 to 1000 nm and infrared bands	C-band, X-band, L-band, etc.
Spatial resolution (typ.)	Sub-meter to tens of m	Meters to several meters
Target areas (typ.)	Land, ocean, ice, atmosphere	Land, ocean, ice
Special modes (typ.)	Dynamical targeting stereo views fusion of bands	V and H polarization scan modes interferometry
Pixel types (typ.)	Detector counts reflectances	Complex-valued “detected data” (amplitudes or intensities)
Important parameters (typ.)	Number of overlapping bands	Viewing/incidence angle polar or geo. orbit

Table 1.
Typical imaging parameters of current satellites.

Clipping of outliers and de-noising
Color coding of brightness levels
Histogram manipulation (e.g., stretching)
Normalization and contrast enhancement
Box-car filtering (e.g., high-pass filtering, smoothing)
Transformations and filtering of coefficients
Analysis of pixel statistics and use of computer vision algorithms (e.g., histograms of gradients, local binary patterns, speeded-up robust features)
Feature extraction and classification (edges, corners, ridges, texture, color, interest points, shapes)
Extraction of content-oriented regions and objects

Table 2.
Typical capabilities of traditional image content analysis tools.

preparation and conduction of tests that allow a benchmarking of the new software routines, notably methods and tools to generate and analyze data for testing, training, verification, and final benchmarking. These testing activities have to be supported by efficient visualization tools.

As can be seen from **Table 2**, there exist already quite a number of traditional image content analysis tools. Some of them generate pre-processed images for subsequent analysis by human image interpreters, while others allow the identification and extraction of objects. However, these tools do not yet exploit the most recent automated machine learning techniques.

3. Machine learning, artificial intelligence, and data science for remote sensing

Currently, we see a lot of public interest in machine learning (ML), artificial intelligence (AI), and data science (DS). We have to make sure what we mean by these buzzwords:

- ML is often used if we describe technical developments where a computer system is trained and used to find and classify objects in data sets. A prominent example is the identification and interpretation of traffic signs for automated driving, typically use cases where a computer system is coupled with a camera and other sensors, and the traffic signs have to be recognized independent of different illumination and weather conditions, a vast range of potential driving speeds, varying distances and perspectives, other cars moving within the field of view of the camera, supplementary information provided by text panels or adjacent traffic signs, and constraints to be observed such as the maximum reasonable processing time. In essence, we can consider these applications as a reduction of many image pixels into single features (from a given list of cases and options) or a combination of features (e.g., max speed of 30 mph except on weekends). In most cases, the ML software is tested and trained by many typical examples as well as counterexamples.
- AI combines the full functionality of ML with additional decision-making and reaction capabilities. This additional decision-making can be implemented by continuous understanding of the current overall situation, the extraction of reactions from given rule sets (supported by continuously updated

parameters), and the handling of unexpected emergency cases. In the case of autonomous driving, one can think of a lane change on a motorway after a reason for a lane change has been found, and from a number of alternative reactions, when a lane change appears as the best reaction. Then the current situation has to be checked when a lane change becomes possible, and a sequence of subsequent actions is executed.

- DS as a scientific and technical discipline of its own shall provide all guiding principles that are needed from end-to-end system design up to data analytics and image understanding—including the system layout and verification, the selection of components and tools, the implementation and installation of the components and their verification, and the benchmarking of the full functionality. In the case of remote sensing applications, we also have to include all aspects of sensor calibration, comparisons with the findings of other researchers via Internet, and traceable scientific data interpretation.

As our applications mostly use cases dealing with remote sensing images, we can limit ourselves to the main ML paradigms that support the semantic annotation of very large data sets. Based on the current state-of-the-art developments, we consider that there are three currently important fundamental and internationally accepted image classification approaches for remote sensing applications and two additional learning principles useful for satellite images:

- *Bayesian networks*: a Bayesian network consists of a probabilistic graphical model representing a set of variables together with their conditional dependencies. It can be used for parameter learning and is based on traditional formulas derived by Bayes [11].
- *Support Vector Machines (SVMs)*: SVMs support classification and regression tasks by identifying basic support points that are used to define a robust separation plane between all sample points. In general, the resulting separation plane is a hyperplane with nonlinear characteristics. In order to obtain a separation plane with linear characteristics, the sample points are mapped into a higher-dimensional system with linear characteristics. This mapping exploits so-called kernel functions [12]. A well-known SVM software package is [13], which also explains how to train and verify a new SVM.
- *Neural Networks*: neural networks follow the concept of biological neurons that trigger a positive response if the input signal corresponds to a known object. Thus, technical implementations mostly consist of three levels, namely a visible input layer followed by an internal processing layer that is not visible to the user (in principle, an artificial neural network), and a visible output layer. An extension of general neural networks are deep neural networks; here, the processing layer is split into several linked internal sublayers that allow a more detailed analysis of the input data (e.g., on selected scales). The internal network parameters (i.e., the filter coefficients) are derived (“learned”) by means of typical (and atypical) image samples and manual labeling by users [11].
- *Active learning*: this learning strategy combines automated learning with interactive steps involving the user during important decisions. This can be

accomplished by a visualization interface where a user can select or deselect image patches that do belong to or do not belong to a specific target class. For further details, see [14].

- *Transfer learning*: the idea of transfer learning is to train a network for a given task and then to exploit or “translate” the resulting network parameters to another use case. A typical example cited in [8] is the use of knowledge gained, while learning to recognize cars in images is applied when trying to recognize trucks.

One of the most critical points for satellite image classification is the dependence of the classification results on the resolution (pixel spacing) of the images. Experiences gained by many authors demonstrate that the identified classes and their local assignment within image patches are strongly resolution-dependent as higher resolution will often lead to a higher number of visible and identified semantic categories. Thus, the performance of any semantic interpretation of images must be considered as a data-dependent metric: this potential difficulty should prevent us from blind-folded direct performance comparisons.

Another similar point to be mentioned is the risk of sensory and semantic gaps encountered during image classification. Sensory gaps result from cases where a sensing instrument cannot measure the full range of potential cases with all their physical effects and details that could exist in a real-world scene and that we cannot record and identify with uniform confidence. A similar potential pitfall for image understanding can result from semantic gaps. For instance, during interactive labeling by test persons, different people could assign different categories to image patches due to their educational background, professional experiences, etc. For further details, see [15].

The number of available approaches, algorithms, and tools is growing continuously. Some examples have become very widespread in academia such as Caffe [16], TensorFlow [17], and PyTorch [18]. In contrast to these established solutions, a large number of fresh publications are submitted every day. As an example, the ArXiv preprint repository [19] collects in its “computer science” and “statistics” directories hundreds of new machine learning papers per day.

4. Networks for deep learning

Many experiments with image classification systems have shown that traditional single-level (“shallow”) algorithms are less performant than multi-level (“deep”) concepts where distinct filtering operations are applied on each level, and the results of the previous levels can be used on each deeper level; the final result will be obtained by combining the specific results of each separate level. The reason for the better performance of multi-level algorithms is that one can apply distinct filters specifically tailored to each level. Typical examples are multi-resolution filters that detect image characteristics on several scales: when we look at satellite images of urban settlements, then a business district normally has larger high-rise buildings and broader streets than a residential suburb with interspersed low-rise buildings and individual gardens.

From a high-level perspective, we can say that learning works best with deep learning approaches exploiting dedicated “network” structures. Here, we understand networks as design structures of the data flows and the arrangement of pixel handling steps governing the processing of our images. This concept also supports

more intricate label assignment concepts such as primary labels defining the main category of an image patch supplemented by secondary labels that provide additional information about “mixed classes” or supplementary spatial details of a given image patch.

In the meantime, some types of networks have emerged that have proven their robustness in the case of satellite images to be annotated semantically. In the following, we list four types of networks that have proven their usefulness for satellite image interpretation:

- *Deep Neural Networks* (DNNs): as described in [20], these networks consist of several layers and comprise an input layer, an output layer, and at least one hidden layer in between. Each layer performs dedicated pixel processing. The corresponding training phase can be understood as deep learning.
- *Recursive Neural Networks* (*not to be confused with recurrent neural networks; both network types appear as RNNs*): when we have structured input data, these data can be efficiently handled by recursive neural networks that are often being used for speech processing and understanding. Recursive neural networks can also be used for natural scenes such as images containing recursive structures [23]. RNN algorithms identify the units that an image contains and how the units interact. Thus, one can use RNNs for semantic scene segmentation and annotation.
- *Convolutional Neural Networks* (CNNs): these networks have been conceived for low-error classification of big images with a very large number of classes. As described by [21], one can classify more than a million images and assign more than 1000 different classes. This is accomplished internally by five convolutional layers, three fully connected layers, and a million internal parameters. To reduce overfitting, the method applies regularization by disregarding offending elements (“dropout method”).
- *Generative Adversarial Networks* (GANs): an adversarial network allows the mutual training of two competing multilayer perceptron models G and D following an adversarial process: G determines the data distribution, while D estimates the probability that a sample comes from training data rather than from D . In addition, D maps the high-dimensional input data to semantic category labels. For further details, see [24].

Besides the network types listed above, we also need an overall algorithmic architecture embedding the networks. For our applications, a “U” approach has proven to be a useful concept for satellite image content analysis. A “U” approach contains a descending branch followed by an ascending branch and is conceived for handling a progressively shrinking number of elements until a final core element (a main category) is found, followed by stepwise complementary semantic information. Further details can be found in [21].

In our experience, most general remote sensing applications can be solved efficiently by CNNs or similar approaches. However, quite a number of innovative alternatives have been proposed during the last years, for example, common auto-encoders, recursive approaches for time series, and adversarial networks for fast learning with only a few examples. In our case, we suggest to use CNNs for non-critical satellite image applications, while highly complicated or time-critical applications could call for innovative approaches as already described above.

5. Training and benchmarking

When we train a classification network and verify its performance, the main goal is to train the system for correct category assignments resp. semantic annotations (labels), that is, to add supplementary information to each satellite image patch that we analyze.

The semantic annotations can either be learned in a preparatory phase or be taken from catalogs of already existing categories. If we aim at long-term analyses of satellite images, a good approach is to use the same catalogs during the entire lifetime of the analysis or to re-run the entire system with updated catalogs.

The easiest approach is to select typical examples for each category and to assign the given labels to all new image data. However, if we follow this straightforward approach, we will probably encounter some difficulties when image patches with unexpected content arrive. A first remedy is to add an additional “unknown” category and to assign this label to all image patches that do not fit well to one of the given categories. Further, experience with machine learning systems has shown that good classification results can also be reached when we systematically select positive as well as negative examples (i.e., counterexamples) for each category leading to a comprehensive coverage and understanding of each category. This process can be accomplished manually by knowledgeable operators (i.e., image interpretation experts) [22]. Another approach is data augmentation: If we do not have sufficient examples of a necessary category, one can create additional realistic data by simply flipping or rotating already available images.

This simple example leads us to systematic methods for a database creation. One has to find a comprehensive and fairly balanced set of examples that covers the expected total variety of cases. Thus, we avoid so-called database biases [23]. In addition, one has to make sure that the inclusion of additional examples does not lead to overfitting or excessive runtimes. This can be accomplished by setting up a validation testbed where these potential pitfalls can be tested, trained, and where the final performance of the created database structure can be verified. One has to be aware of the fact that database access times may strongly depend on the available computer systems, their interconnections, and the selected type of database.

These approaches led to a number of publicly available databases with label annotations for civilian remote sensing data. There are several semantically annotated databases based on optical (most often multispectral) data, while there are only a few databases based on SAR data. Some advanced remote sensing database examples are [25–27]. Of course, their general applicability and transferability depend on the actual image resolution, the imaging geometry, and the noise content of the images. Current state-of-the-art systems are being assessed based on end-to-end tests covering also inter alia practical aspects such as the runtime depending on the database design and the selected test images, the amount and organization of available labels, the correctness of the obtained annotations, and the overall implementation and validation effort.

6. Perspectives

As for remote sensing images, there exist already several semantically annotated collections of typical high-resolution satellite images—a number of collections of optical images and a few collections of SAR images. However, these collections often seem to be potpourris of interesting snapshots rather than systematically selected samples based on regionally typical target classes and their visibility as a

function of different instrument types. The situation is aggravated by the current lack of systematically selected benchmarking data that could be used as well-known reference data for quality and performance assessments such as classification tasks or throughput testing.

These deficiencies have to be solved in the near future as more and more high-resolution images become publicly available, while the end-users already expect reliable automated image classification and content understanding results for more and more high-level applications. We can expect that the progress in deep learning will also lead to much progress in many other fields of image processing, even beyond the field of remote sensing; thus, remote sensing should be aware of what is published by the image processing and environmental protection communities at large.

7. Conclusions

While high-resolution imaging has made much progress for many remote sensing applications, standardized image classification benchmarking still deserves more progress. On the one hand, several benchmarking concepts and tools could still be gleaned from other disciplines; on the other hand, an optimal solution of test cases for SAR image interpretation still needs more progress in basic approaches of how to verify actual image classification results and the identification of dubious cases.

Acknowledgements

We appreciate the cooperation with Politehnica University of Bucharest (UPB) in Romania and our project partners from the European H2020 projects CANDELA (under grant agreement No. 776193) and ExtremeEarth (under grant agreement No. 825258).

Author details


Mihai Datcu^{1,2*}, Gottfried Schwarz¹ and Corneliu Octavian Dumitru¹

¹ German Aerospace Center (DLR), Remote Sensing Technology Institute, Wessling, Germany

² Politehnica University of Bucharest, Bucharest, Romania

*Address all correspondence to: mihai.datcu@dlr.de

IntechOpen

© 2020 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/3.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. 

References

- [1] Alom Z, Taha T, Yakopcic C, Westberg S, Sidike P, Nasrin S, et al. State-of-the-art survey on deep learning theory and architectures. *Electronics*. 2019;**8**:292. Available at: <https://www.mdpi.com/2079-9292/8/3/292>
- [2] Ball J, Anderson D, Chan CS. A comprehensive survey of deep learning in remote sensing: Theories, tools and challenges for the community. *Journal of Applied Remote Sensing*. 2017;**11**(4):042609. Available at: <https://arxiv.org/abs/1709.00308>
- [3] Ma L, Liu Y, Zhang X, Ye Y, Yin G, Johnson B. Deep learning in remote sensing applications: A meta-analysis and review. *ISPRS Journal of Photogrammetry and Remote Sensing*. 2019;**152**:166-177. Available at: <https://www.sciencedirect.com/science/article/pii/S0924271619301108>
- [4] ESA Sentinel-2. Available at: <https://sentinels.copernicus.eu/web/sentinel/missions/sentinel-2> [Accessed: April 2019]
- [5] ESA Sentinel-3. Available at: <https://sentinels.copernicus.eu/web/sentinel/missions/sentinel-3> [Accessed: April 2019]
- [6] ESA Sentinel-1. Available at: <https://sentinels.copernicus.eu/web/sentinel/missions/sentinel-1> [Accessed: April 2019]
- [7] Deep learning. Available at: <https://www.deeplearningbook.org> [Accessed: April 2019]
- [8] Transfer learning. Available at: https://en.wikipedia.org/wiki/Transfer_learning [Accessed May 2019]
- [9] Orfeo Toolbox. Available at: <https://www.orfeo-toolbox.org/> [Accessed May 2019]
- [10] Adversarial machine learning. Available at: https://en.wikipedia.org/wiki/Adversarial_machine_learning [Accessed May 2019]
- [11] Bayesian network. Available at: https://en.wikipedia.org/wiki/Bayesian_network [Accessed May 2019]
- [12] Support vector machine. Available at: https://en.wikipedia.org/wiki/Support-vector_machine [Accessed May 2019]
- [13] LIBSVM -- A Library for Support Vector Machines. Available at: <http://www.csie.ntu.edu.tw/~cjlin/libsvm/> [Accessed May 2019]
- [14] Active learning. Available at: https://en.wikipedia.org/wiki/Active_learning [Accessed May 2019]
- [15] Bahmanyar R, Murillo A. Evaluating the sensory gap for earth observation images using human perception and an LDA-based computational model. In *Image Processing (ICIP), 2015 IEEE International Conference on* pp. 566-570
- [16] Caffe software. Available at: <https://caffe.berkeleyvision.org/> [Accessed May 2019]
- [17] TensorFlow. Available at: <https://www.tensorflow.org> [Accessed: March 2019]
- [18] PyTorch. Available at: <https://pytorch.org/> [Accessed: March 2019]
- [19] arXiv e-Print archive. Available at: <https://arxiv.org/> [Accessed: March 2019]
- [20] Krizhevsky A, Sutskever I, Hinton GC. ImageNet Classification with Deep Convolutional Neural Networks. Available at: <https://papers.nips.cc/4824-imagenet-with-deep-convolutional-neural-networks.pdf>

- [21] Ronneberger O, Fischer P, Brox T. U-Net: Convolutional Networks for Biomedical Image Segmentation, Medical Image Computing and Computer-Assisted Intervention (MICCAI). Vol. 9351. Basel, Switzerland: Springer International Publishing, LNCS; 2015. pp. 234-241, Available at: <https://arXiv:150504597>
- [22] Murillo Montes de Oca A, Bahmanyar R, Nistor N, Datcu M. Earth observation image semantic bias: A collaborative user annotation approach. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing (JSTARS). 2017;**10**(6):2462-2477. DOI: 10.1109/JSTARS.2017.2697003
- [23] Socher R, Lin C, Ng AY, Manning, CD. Parsing Natural Scenes and Natural Language with Recursive Neural Networks, 28th International Conference on Machine Learning (ICML 2011). Available at: https://nlp.stanford.edu/pubs/SocherLinNgManning_ICML2011.pdf
- [24] Goodfellow IJ, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, et al. Generative Adversarial Nets. Available at: <http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf>
- [25] Xia G-S, Bai X, Ding J, Zhu Z, Belongie S, Luo J, et al. DOTA: A large-scale dataset for object detection in aerial images. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, Utah, USA. 2018
- [26] Dumitru CO, Schwarz G, Datcu M. SAR image land cover datasets for classification benchmarking of temporal changes. IEEE Journal of Selected Topics in Applied Earth Observation and Remote Sensing (JSTARS). 2018;**11**(5):1571-1592
- [27] Sumbul G, Charfuelan M, Demir B, Markl V. BIGEARTHNET: A large-scale benchmark archive for remote sensing image understanding. In: IEEE International Conference on Geoscience and Remote Sensing Symposium (IGARRS), Yokohama, Japan. 2019

Data Mining Technology for Structural Control Systems: Concept, Development, and Comparison

*Meisam Gordan, Zubaidah Ismail, Zainah Ibrahim
and Huzaifa Hashim*

Abstract

Structural control systems are classified into four categories, that is, passive, active, semi-active, and hybrid systems. These systems must be designed in the best way to control harmonic motions imposed to structures. Therefore, a precise powerful computer-based technology is required to increase the damping characteristics of structures. In this direction, data mining has provided numerous solutions to structural damped system problems as an all-inclusive technology due to its computational ability. This chapter provides a broad, yet in-depth, overview in data mining including knowledge view (i.e., concept, functions, and techniques) as well as application view in damped systems, shock absorbers, and harmonic oscillators. To aid the aim, various data mining techniques are classified in three groups, that is, classification-, prediction-, and optimization-based data mining methods, in order to present the development of this technology. According to this categorization, the applications of statistical, machine learning, and artificial intelligence techniques with respect to vibration control system research area are compared. Then, some related examples are detailed in order to indicate the efficiency of data mining algorithms. Last but not least, capabilities and limitations of the most applicable data mining-based methods in structural control systems are presented. To the best of our knowledge, the current research is the first attempt to illustrate the data mining applications in this domain.

Keywords: data mining, structural damped systems, vibration control, machine learning, artificial intelligence, statistical analysis

1. Introduction

In recent years, there has been a vast theoretical and experimental investigations in various problems encountered in different structures, from basic structural components (e.g., beams and plates) to complex structural systems (e.g., bridges and buildings). This is due to the fact that structures are built to support a load, namely, static or dynamic loads, incoming from different forces (e.g., tension, compression, torsion, bending, and shear). In this direction, many structures need to be designed to withstand dynamic loads even though they spend most of the time supporting static loads [1, 2]. Static loads are those that are gradually applied and remain in

place for longer duration of time. These loads are not time dependent. As an illustration, a live load on a structure is considered as a static load. Besides, most of the loadings applied to civil engineering structures, including seismic loadings, are usually considered as equivalent static loads [3, 4]. On the other hand, time-dependent dynamic loads such as machinery vibrations, earthquakes, wind storms, sea waves, and traffic can cause intensive and continuous vibrational motions which can cause changing of the structural properties (i.e., mass, stiffness, or damping) and loading to change in the dynamic responses, such as natural frequencies, mode shapes, and damping ratios [5–8]. Therefore, in-service structural systems in civil engineering such as tall buildings, long hydraulic structures, and long-span bridges are damage-prone under these loads during their service life [9–14]. Moreover, these loads can cause intensive and stable vibrational motions, which can be damaging to human inhabitants. Based on these explanations, vibration is a serious concern in civil structures. It is due to the fact that existence of damage can disturb functionality and safety of the structure. However, the risk of occurrence of structural damage can be decreased by using a controlled vibration system to increase the damping characteristics of the structure. Accordingly, the advantage of using damping device is that damping system can improve the ability of the structure to dissipate a portion of the energy released during a dynamic loading event [15–18].

Over the last few decades, taller and wider structures have been built because of enormous developments in civil engineering area. As mentioned earlier, these structures will be subject to external loads which can cause vibrational problems. Consequently, it is essential to control the vibrational motions to reduce the response and to improve structure performance, safety, flexibility, serviceability, and structural reliability of these structures. Generally, structural control systems include four main groups which are passive, active, semi-active, and hybrid devices. Classification of these energy dissipation supplements is based on their operational mechanisms [19–21].

Data mining is the analysis of datasets to discover the relationships, new correlations, and trends and to extract the useful data in the form of patterns. Therefore, this process has been used to identify valid, valuable, and understandable forms of data [22, 23]. Accordingly, in recent years, this technology has provided various solutions to structural damped systems because of its powerful computational capacity. In this matter, many researchers have studied and examined various data mining techniques for passive, semi-active, active, and hybrid damped systems. In the same line, this chapter attempts to present the recent developments of well-known data mining techniques in vibration control devices. Before going into the details, it is important to point out the fundamental principles of data mining. Hence, data mining concepts including definition, background, functions, and techniques are discussed in the following section. Then, the concepts of applicable algorithms and their applications in damped systems are detailed in Section 3. Furthermore, applicable examples of data mining algorithms are presented for better understanding.

2. Data mining concept

Data can be defined as any fact, number, or text which can be proceeded by a computer. As the obtained pattern through data mining may be very difficult to find, it is sometimes compared to gold mining in rivers (**Figure 1**). The term “gold mining” refers to the search for gold in rocks or sand. Data mining is a search for information and knowledge. The origination of data mining traces back to the development of artificial intelligence in the 1950s. The development of data mining is shown in **Figure 2**.

In general, data mining has two classes which are descriptive mining and predictive mining using various techniques and functions (see **Figure 3** and **Table 1**).

The techniques play important roles to obtain effective models from observations. Besides, data mining techniques have also three main groups which are statistical techniques, machine learning techniques, and artificial intelligence techniques. It is noted that each of these techniques has particular algorithms for running the models to get the best solution. For instance, artificial neural network (ANN), Bayesian analysis, ant colony optimization, ICA, support vector machine, principal component analysis, particle swarm optimization (PSO), genetic algorithm, fuzzy logic, regression analysis, clustering, classification, and decision tree are classified under data mining techniques. Furthermore, the functions of data mining are categorized into clustering, prediction, classification, exploration, and association. The purpose of clustering is to divide the samples into groups with related behavior. The numerical prediction activity determines patterns, rules, or models to predict continuous or discrete target values which can also be used for other functions. Classification is used to recognize several rules which can be applied in future work to determine whether a previously unknown item belongs to a known class. Exploration is used to find out dimensionality of an input data, and, eventually, the association activity is used to frequently detect occurring related objects. Based on their particular utilizations in consequence of their assumptions and drawbacks, one or a combination of some of these tasks can be used to find the hidden information [24–27].



Figure 1.
 Gold mining and data mining.

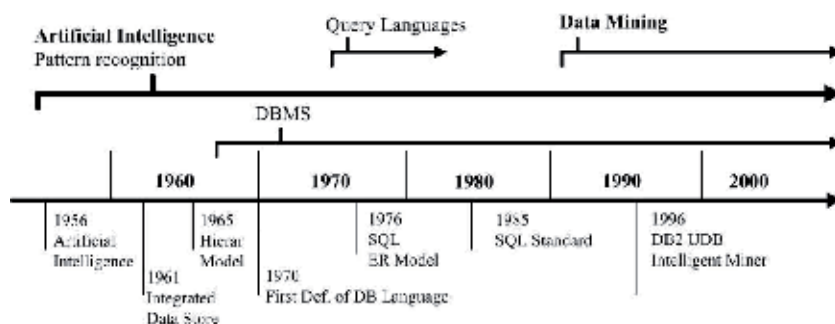


Figure 2.
 History of data mining development.

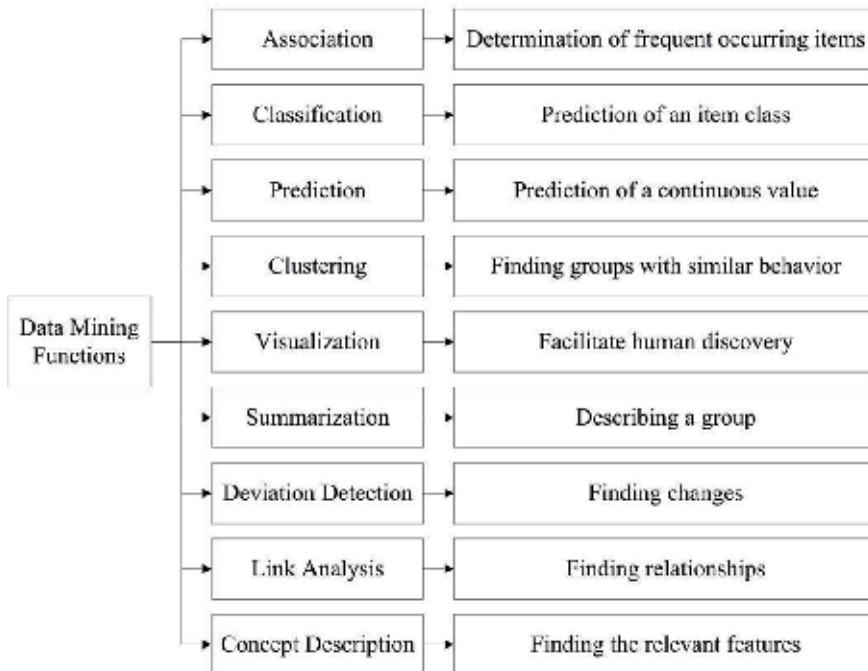


Figure 3.
Data mining functions.

Data mining technique	Category	Learning type
Artificial neural network	Artificial intelligence	Supervised/unsupervised
Support vector machine	Machine learning	Supervised
Decision tree	Statistical	Supervised
Clustering	Statistical	Unsupervised
Principal component analysis	Machine learning	Unsupervised
Regression	Statistical	Supervised
Fuzzy	Artificial intelligence	Supervised/unsupervised
Meta-heuristics	Artificial intelligence	-
Classification	Statistical	Supervised
Bayesian	Machine learning	Supervised

Table 1.
Data mining techniques.

3. Data mining algorithms

3.1 Support vector machine (SVM)

SVM is one of the classification- and prediction-based techniques which was first introduced by Vapnik in 1963 [28]. It works based on learning theory and because of its high accuracy and good generalization capability; it has the potential to produce high-quality predictions in numerous tasks. Therefore, SVM has various

applications which can be found in several areas such as machine learning, data classification, and pattern recognition [29, 30]. Basic models of SVM are linear SVM with linear functions and nonlinear SVM with kernel functions. Moreover, the aim of SVM classifier is to determine a separating hyperplane to divide the given data into two classes (i.e., positive class and negative class) in the optimal form. Therefore, the optimal separating hyperplane is determined by solving an optimization problem [31].

SVM has been used in structural control systems. For instance, a SVM-based semi-active control strategy was reported by [32] for the numerical model of a multi-storey structure. In this study, four seismic waves including the El Centro, Hachinohe, and Kobe waves, as well as the Shanghai artificial wave, whose peak ground accelerations were all scaled to 0.1 g, were taken into consideration. As shown in **Figure 4**, a three-storey shear-type frame structure with dampers was considered as a case study in this work.

The seismic responses of structural top storey with the structure-damper system, structure-SVM system, and no-control device are shown, respectively, in **Figure 5**. It is seen from this figure that the structure-SVM system model has perfectly learned the control effectiveness of the structure-damper system. This observation indicated that the structure-SVM system model was significantly better than the structure-damper system.

In order to further examine the seismic response reduction of the controlled structure using the present algorithm, the displacement response of every floor under these four seismic waves is shown in **Figure 6**. It is seen that under the Hachinohe wave, the peak displacement response of every floor, especially the top floor, with the structure-SVM system model, was remarkably smaller than that with the structure-damper system. The authors verified once again that the proposed structure-SVM system model will render better effectiveness than the structure-damper system.

Comparative results of this study demonstrate that general semi-active dampers designed using the SVM-based semi-active control algorithm was capable of providing the higher level of response reduction.

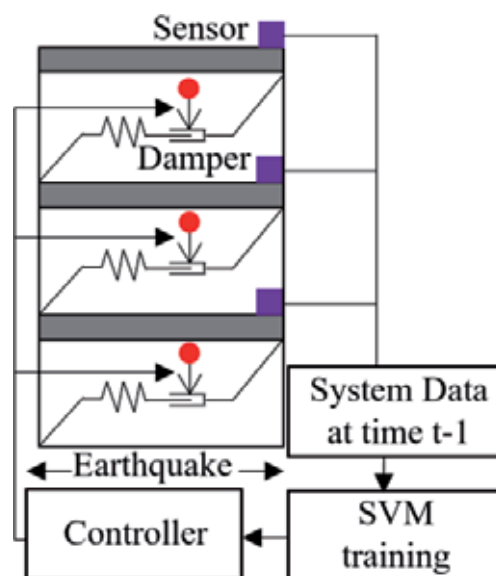


Figure 4. Structure-SVM semi-active control system model and implementation flow chart.

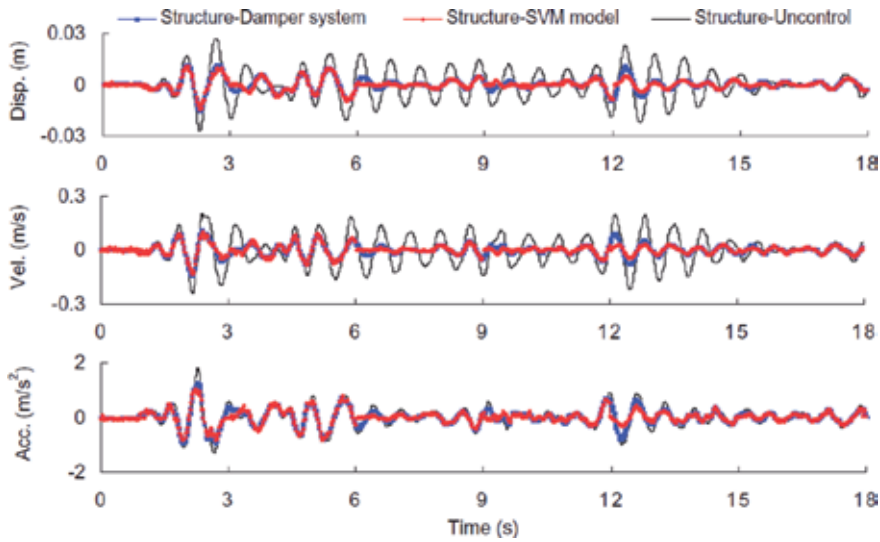


Figure 5. Seismic responses of the structural top storey under the El Centro wave with PGA = 0.1 g using general semi-active dampers and SVM-based semi-active control algorithm.

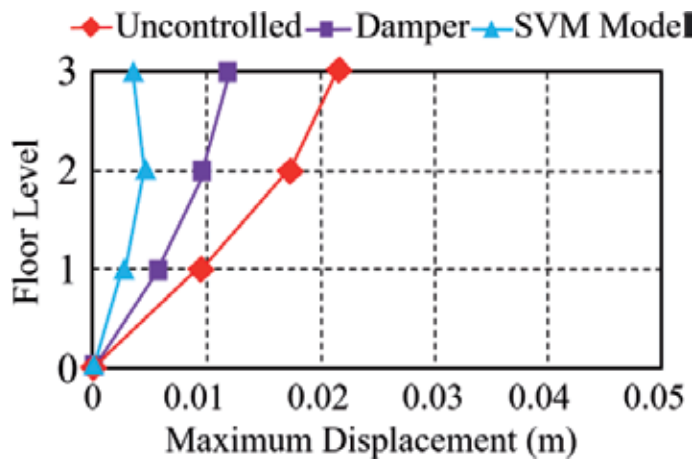


Figure 6. Displacement responses of every storey under the Hachinohe seismic wave using general semi-active dampers and SVM-based semi-active control algorithm.

3.2 Artificial neural network (ANN)

Artificial neural network, which is a self-organizing prediction-based computational technique, was first proposed in the 1980s. This algorithm can solve many functions through pattern recognition [33]. It also can effectively be used to reconstruct nonlinear relationship learning from training [34]. A typical ANN model has two parts, that is, processing units (neurons) and connections between elements [34], in which neurons are located in layers of the network. A layered ANN structure, called multilayer perceptron (MLP), is one of the most widespread ANN methods. Generally, a conventional ANN has three layers which are input layer, hidden layer, and output layer. ANNs also can be categorized by their network topology such as feed forward and feedback or by their learning algorithms such as supervised learning and unsupervised learning [35].

There are a variety of researches focusing on the application of ANN in structural control systems. For instance, according to reports by [36, 37], ANN has a great capacity to improve the functionality of active control systems due to its high pattern recognition capability. It also could be used for semi-active [38, 39] and passive damping systems [40]. The following are the review of some related examples which indicate the applicability of ANN in damping systems.

Suresh et al. [36] applied a nonlinearly parameterized neural network as a novel controller scheme for the active control of earthquake-excited nonlinear base-isolated buildings. Numerical simulations were performed on a full-scale numerical test-bed base-isolated building with an isolation system comprising hysteretic lead-rubber bearings. They showed that the proposed approach could achieve good response reductions for a wide range of near-fault earthquakes, without a corresponding increase in the superstructure response.

Figure 7 demonstrates a neural network model that was developed by [40] which shows the application of ANN in passive damping devices. In this study, the ANN was employed in order to predict the inelastic demand of structural systems with viscoelastic dampers in terms of peak displacement, effective damping, and effective time period. The authors established that the ANN could be effectively used for new designs as well as for checking the response of any retrofitted structure for the chosen design spectrum. In addition, they concluded that artificial neural networks also were useful in quickly deciding the amount of damping and the number of dampers required to reduce the peak displacement and help in restricting further damage.

A smart active control system, called NEURO-FBG combining fiber Bragg grating (FBG) sensors and neural networks, has been proposed by [41] in a steel building. In this study, an attempt has been made to illustrate the development procedure of the converter and controller by means of “NEURO-FBG converter” and “NEURO-FBG controller.” In this regard, the NEURO-FBG smart control system was designed to be a robust and reliable active control system with “smart” performance. To achieve this goal, a specific methodology was proposed comprising three parts, that is, a structural surveillance system, three converters, and a controller (see **Figure 8**). The analytical results show that the NEURO-FBG system could effectively control the response of the structure and provide a more reliable system than ordinary active control. Later on, the authors verified their method using an experimentation [42]. According to their experimental results, the proposed active control system can be successfully applied to buildings.

Figure 9 shows the architecture of an ANN-based real-time force tracking scheme for magnetorheological (MR) dampers, which was applied numerically and

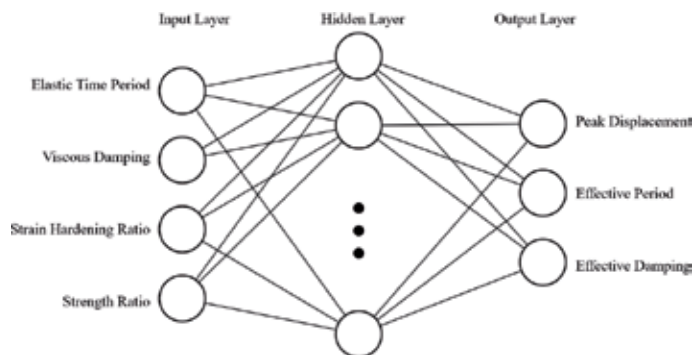


Figure 7.
Neural network model for a passive control system.

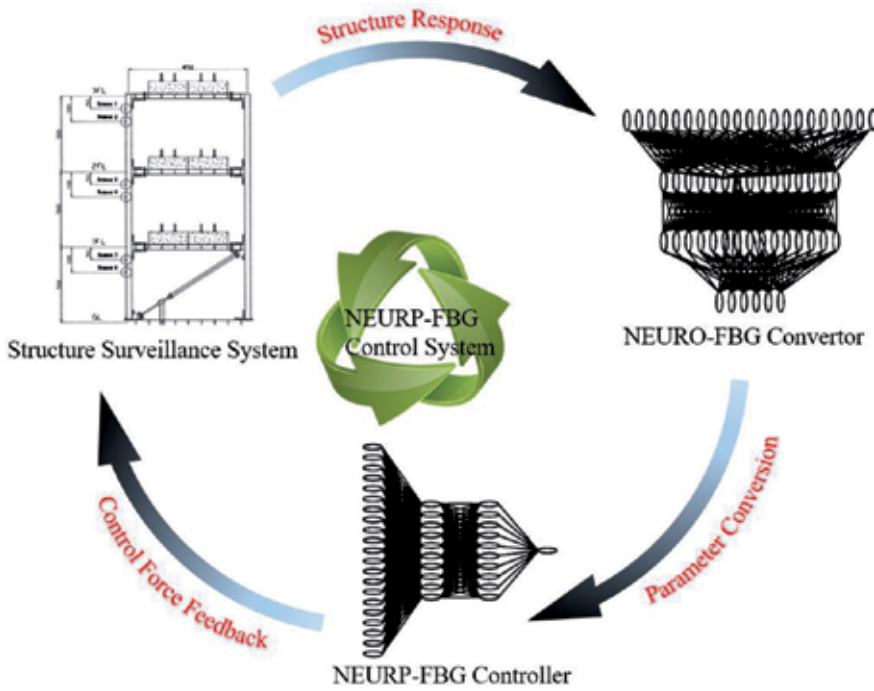


Figure 8.
Block diagram of NEURO-FBG smart control system.

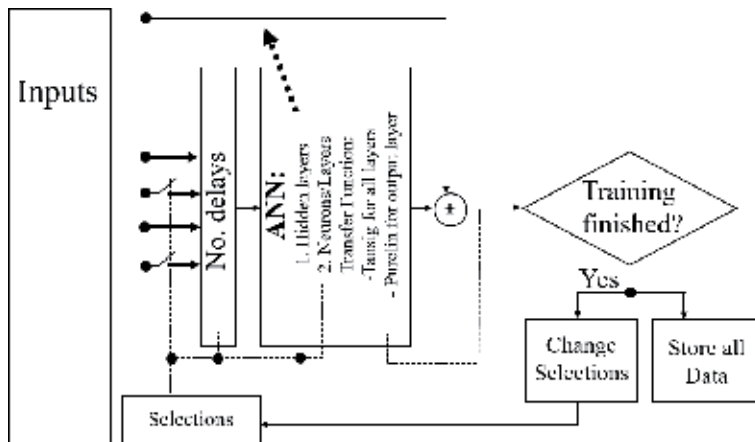


Figure 9.
The architecture of the neural network modeling.

experimentally on a five-storey shear frame by [43]. In this study, the forward and inverse MR damper dynamics were modeled by the neural network method using constant and half-sinusoidal current tests. As it can be seen in **Figure 9**, the ANN modeling was implemented for the forward and inverse MR damper model. It was concluded that the experimental validation of the neural network modeling both the forward and inverse MR damper dynamics showed the accuracy of these models.

A semi-active control strategy that combined a neuro-control system including a multilayer ANN with a back-propagation training algorithm with a smart damper was proposed to reduce seismic responses of structures [38]. A set of numerical simulations was performed to verify the effectiveness of the proposed method.

To aid the aim, two controllers were used, that is, (1) a primary control algorithm based on a cost function, and the sensitivity evaluation algorithm was employed in order to replace an emulator neural network as well as produce the desired active control force and (2) a secondary bang-bang-type controller caused the smart damper to generate the desired active control force, so long as this force was dissipative. It should be noted that cost function is defined as the squared sum of offset between the actual and the desired responses. Therefore, the main purpose was to minimize the cost function during training the network. **Figure 10** demonstrates the diagram of control for semi-active neuro-control using smart damper as well as a three-storey building with a single smart damper which was used as a numerical modeling. The authors showed that the proposed semi-active control system using ANN and smart dampers was a promising tool for control of real structures.

3.3 Fuzzy logic

Fuzzy logic was proposed by Lotfi Zadeh for the first time in 1965. It has been employed in different applications such as pattern recognition, classification, decision-making, etc. [44]. The basic configuration of a fuzzy technique consists of four important components, which are fuzzification, fuzzy rule base, fuzzy inference, and defuzzification. Fuzzification is a mapping from a crisp input to fuzzy membership sets. The fuzzy rule base has set rules of fuzzy variables described by membership functions. Fuzzy inference is a decision-making mechanism of the fuzzy system. The defuzzifier changes the fuzzy consequences from different rules into crisp values [45]. Fuzzy is a model-free technique for structural system identification, where the most important advantages of fuzzy systems are their high parallel implementation, nonlinearity, and being capable of adapting [46]. Applications of fuzzy logic in SHM are detailed in **Table 2**.

Adaptive fuzzy control strategy [47], fuzzy gain scheduling [48], semi-active fuzzy logic control system [49], model-based fuzzy logic controller (MBFLC) [50],

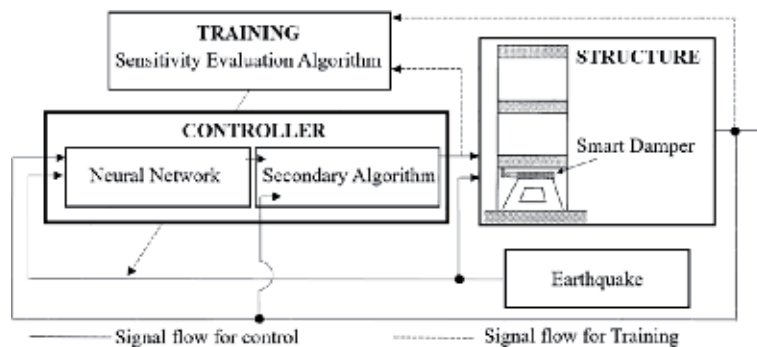


Figure 10.
 Diagram of control for semi-active neuro-control using smart damper.

Case	Wind speed (m/s)	Maximum torsion (rad)
Uncontrolled	55.52	0.02
Controlled with passive TMD	98	0.02
Controlled with STMD-FLC	110	0.0063

Table 2.
 Comparison of the effectiveness of passive TMD and STMD-FLC.

optimal fuzzy logic controller [51], fuzzy controller [52–54], neuro-fuzzy [55–57], genetic fuzzy logic controller (GFLC) [58], fuzzy control strategy based on a neural network forecasting model [59], and wavelet-neuro-fuzzy control [37] are some of the important applications of fuzzy logic in structural control systems. The following examples illustrate the applicability of fuzzy in damping systems.

A semi-active fuzzy control system was introduced by [49] to reduce the seismic responses in variable orifice dampers. In this direction, a numerical study was conducted to investigate the effectiveness of the proposed approach. Results revealed that the fuzzy logic controller (FLC) was capable of improving the structural responses. Another semi-active fuzzy control system comprising a semi-active tuned mass damper (STMD) system with variable damping was proposed by [60] to control the flutter instability of long-span suspension

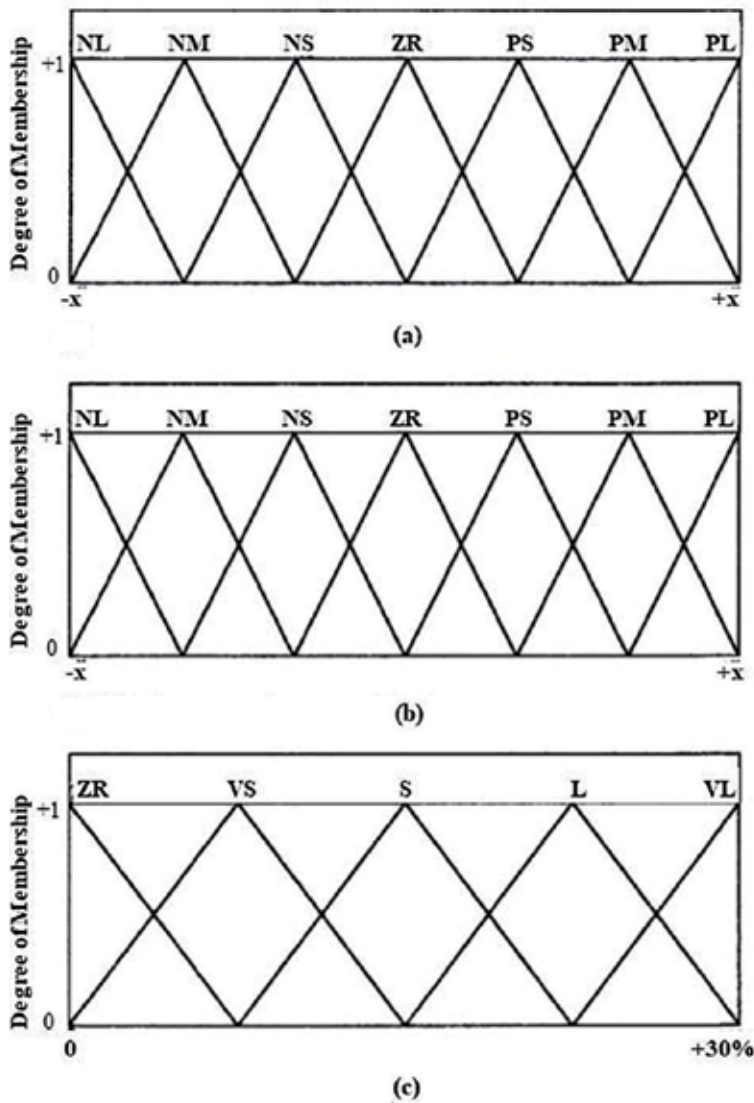


Figure 11. Membership functions of the fuzzy logic controller variables: (a) membership functions for displacement, (b) membership functions for velocity, and (c) membership functions for semi-active tuned mass damper (TMD) damping ratio [60].

bridges. In this study, the variable damping of the system was chosen through a fuzzy logic controller. The STMD-FLC methodology was applied to increase the flutter wind speed of the test structure which was a suspension bridge. To do so, in order to select the level of semi-active damping ratio, a fuzzy logic feedback controller was incorporated into a closed-loop control system. The displacement and velocity quantities were used as the input to the fuzzy logic controller, and the level of STMD damping ratio was its output for each degree of freedom, namely, vertical and torsional (see **Figure 11**). The FLC system was designed based on the Mamdani's fuzzy inference method.

In addition, a comparison of the effectiveness of passive TMD and STMD-FLC was carried out in this research, which is shown in **Table 2**. The table clearly shows the superior performance of semi-active control over the passive control.

The description of the fuzzy input membership function abbreviations is as follows: NL = negative large, NM = negative medium, NS = negative small, ZR = zero, PS = positive small, PM = positive medium, and PL = positive large; and those of the output are as follows: ZR = zero, VS = very small, S = small, L = large, and VL = very large.

Adaptive network-based fuzzy inference system (ANFIS) is a hybrid learning algorithm which combines the back-propagation gradient descent and least squares techniques to generate a fuzzy inference system. The membership functions in ANFIS are adjusted according to a given set of input and output data. The main objective of ANFIS is to integrate the finest features of neural networks and fuzzy systems. Accordingly, the outputs of ANFIS can be seen in two steps, that is, (1) representation of prior knowledge into a set of constraints to reduce the optimization search space from fuzzy system and (2) adaptation of back-propagation to structured network to automate fuzzy control parametric tuning from neural network. Therefore, ANFIS is one of the best trade-offs between neural and fuzzy systems providing smoothness due to the fuzzy control interpolation and adaptability, due to the neural network back-propagation [61].

ANFIS has proven to be an excellent function approximation tool. For example, an ANFIS controller was developed by [61] for reduction of environmentally induced vibration in multiple-degree-of-freedom building structure with MR damper. The systems were excited using two different earthquake random vibration loadings. **Figure 12** illustrates the comparison of the displacement response at the top of the structure with and without control under El Centro and Hachinohe earthquakes. The figure shows that ANFIS clearly could reduce the displacement amplitude in both vibration loadings.

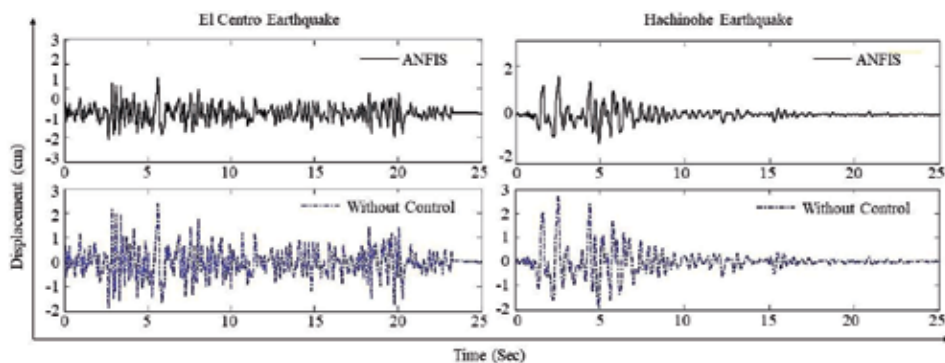


Figure 12.
Displacement response of the structure under El Centro and Hachinohe earthquakes.

3.4 Clustering

Clustering is an unsupervised statistical data analysis technique, which is used in pattern recognition, image analysis, and bioinformatics. This method is employed to divide datasets into separated similar subsets (clusters) according to typical patterns identified in the clustering analysis [62]. In order to have a successful clustering, maximum intra-cluster similarity as well as minimum inter-cluster similarity is required. The K-means is one of the most descriptive partitioning clustering algorithms with a quite reliable effectiveness at local optimum. However, it can be employed only to numerical datasets. Furthermore, K-means has poor handling for data prone to noise and outliers [63]. Clustering can also help to decrease the distance between datasets and improve the similarity of datasets in each cluster [64, 65].

A combination of fuzzy C-means clustering and subtractive clustering has been developed numerically for nonlinear system identification of a seismically excited building-MR damper system. It was demonstrated from the simulation that the proposed fuzzy model is effective in identifying nonlinear behavior of the building-MR damper system subjected to the 1940 El Centro earthquake. **Figure 13** compares the displacement and acceleration responses of the original simulation model with those of the identified model. Note that the original simulation model means an analysis model of the building equipped with an MR damper. As can be seen from the figure, overall good agreements between the original values and the identified model were found in the time histories of both displacement and acceleration responses [66].

3.5 Genetic algorithm (GA)

GA, which is one of the most powerful optimization-based algorithms, was first proposed by John Holland in the 1970s. In GA, a chromosome is used to determine the solution. The chromosome includes a group of genes that optimize parameters. This algorithm employs a random solution from a current population. Then, the next generation will be created using crossover and mutation operators [67]. In general, GA is an attractive tool to optimize difficult problems due to its benefits such as parallelism, convergence to global optima, adaptation, and no need for the gradient of the objective function. Considering these benefits, GA has been successfully applied in optimal design of TMDs [68–70] and multiple tuned liquid column damper (MTLCD) [71], optimization of earthquake energy dissipation system [72], optimization of active control systems in high-rise buildings [73], optimal damper distribution [74], smart control systems [75], etc.

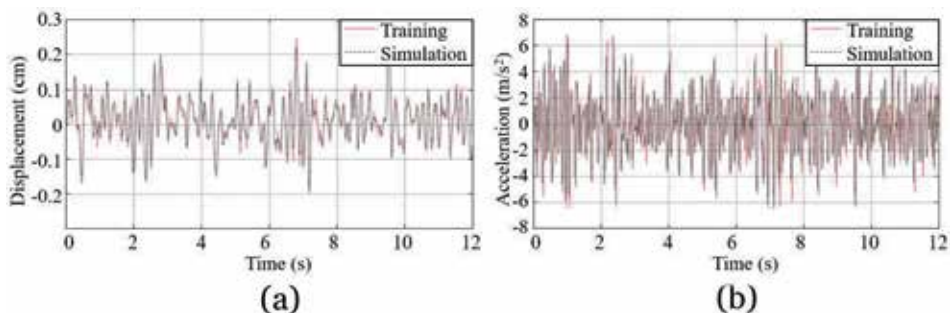


Figure 13. Comparison of original simulation and obtained results from the proposed model. (a) displacement response, (b) acceleration response.

An optimization strategy of hydraulic actuators, that is, an implicit redundant representation (IRR) genetic algorithm with a non-dominated sorting II (NS2) GA, namely, NS2-IRR GA, was implemented numerically by [73] in order to minimize the distribution of control devices in large-scale structures as well as optimize the dynamic responses of structures. It was shown that the proposed NS2-IRR GA-based control system was effective in finding not only optimal locations and numbers of actuators in structures but also minimum responses of the buildings. In the same line, **Figure 14**, which compares the dynamic behavior of the proposed approach with those of the benchmark control system in reducing displacements of the 20-storey building, clearly indicates the effectiveness of GA in minimizing the displacement/drift responses of the building structure.

3.6 Particle swarm optimization

PSO which was first proposed by Kennedy and Eberhart [76] is one of the population-based artificial intelligence optimization-based techniques. The approach was simulated by the social behavior of organisms such as bird flocking to be used as a suitable tool for global optimization [77]. In PSO, a particle represents

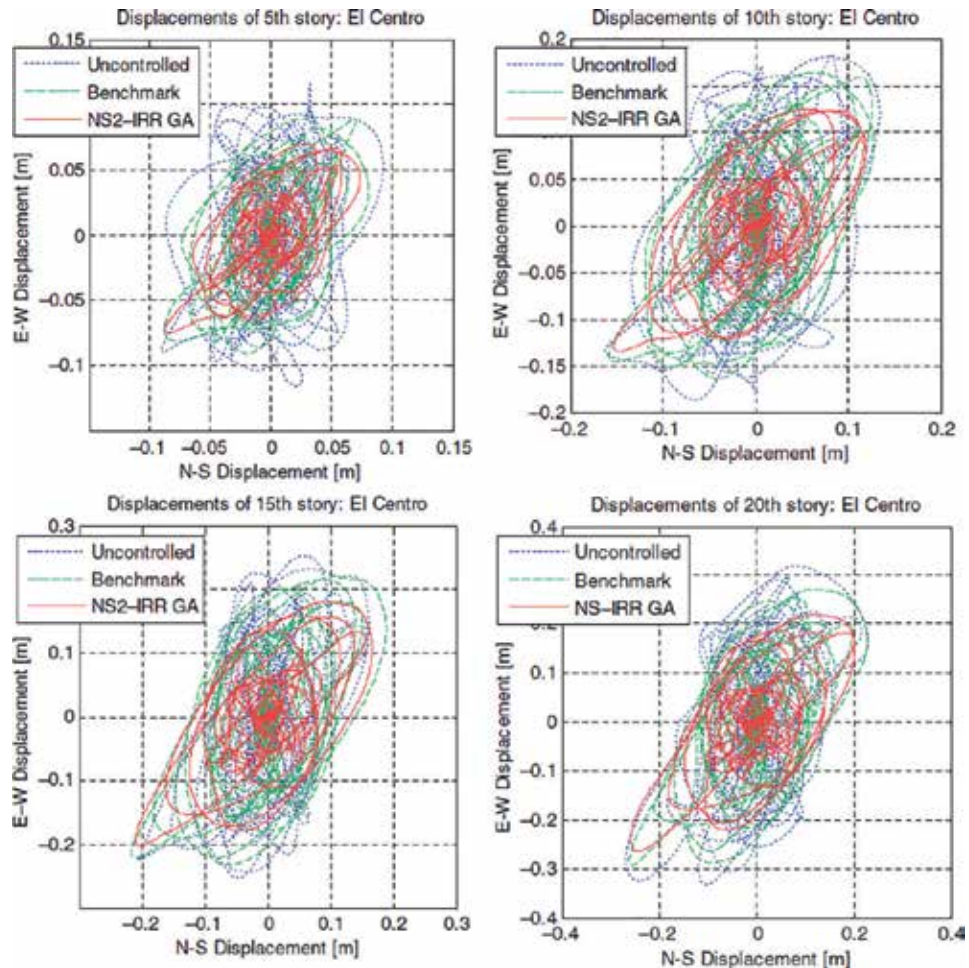


Figure 14. Comparison of displacement responses of benchmark and NS2-IRR genetic algorithm approaches under the El Centro earthquake.

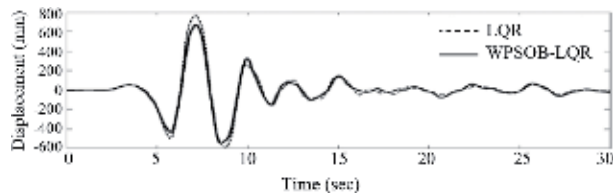


Figure 15.
Results for the 1981 Imperial Valley (El Centro-06): controlled displacement.

a potential solution where each particle has two updatable features: position and velocity. PSO is easy to apply and has a great computational capacity. In comparison with other optimization approaches, PSO is more efficient and requires a fewer numbers of function evaluations while giving better or the same quality of results. However, it has some weaknesses such as trapping into local optimum in a complex search space. Besides, the disability to implement a precise local search around a local optimum is another drawback in PSO [78, 79].

PSO was applied for optimization of the parameters of a TMD-viscously damped system in [80] including the optimum mass ratio, damper damping, and tuning frequency. The results were calculated by means of three numerical examples for different nonstationary ground acceleration systems to demonstrate the efficiency of the proposed method. To this end, the system was subjected to ground accelerations with different PSO-based power spectra in order to minimize either the maximum displacement or acceleration mean square responses. The authors of this research reported that it was quite easy to program the applications of PSO in practical engineering.

Another method called wavelet PSO-based linear quadratic regulator (WPSOB-LQR) was presented numerically by [81] to find the optimal control force of active TMD via PSO-based linear quadratic regulator (LQR) and wavelet analysis. To aid the aim, PSO was used to determine the gain matrices through the online update of the weighting matrices used in the controller while eliminating the trial and error. **Figure 15** shows the time history of displacement response by using pre-developed LQR control method and the proposed WPSOB-LQR approach. As it can be observed from this figure, the displacement was significantly reduced using their proposed method. Moreover, the authors stated that the proposed method was practicable and worthwhile for vibration control of structures.

4. Conclusion

In this chapter, a brief description of data mining has been made, the most applicable techniques were reviewed, and the applications of machine learning, artificial intelligence, and statistical algorithms in structural control systems have been stated. Furthermore, for each technique, an attempt has been made to present several examples to familiarize readers more in the corresponding field as well as presenting an overall background of the researches done by several investigators worldwide. The following are some of the important conclusions.

Fuzzy, GA, and ANN are the most applicable methods in structural control systems. Among all, fuzzy controllers were the most powerful techniques to solve numerous problems. As a matter of fact, fuzzy algorithm could present uncomplicated and strong solutions for control systems in order to modify uncertainty, such as selecting damping ratios and reducing the responses of structures. ANN was another self-organizing technique which has been used to control and predict

the seismic responses of structures with energy dissipation systems. As far as the ANN was concerned, the local minimum point, overlearning, and the excessive dependence on experience in the choice of structures and types were its inevitable limitations, while SVM could get rid of these limitations and has been successfully applied to time series forecasting. Likewise, SVM could provide some special advantages in the fields of small sample issues and nonlinear and high-dimensional pattern recognition. MATLAB was the main program language which has been used to develop data mining techniques in this area.

Acknowledgements

This research was funded by the University of Malaya (UM) and the Ministry of Higher Education (MOHE), Malaysia (Grant numbers: IIRG007A, GPF015A-2018 and RG561-18HTM).

Notes/thanks/other declarations


The authors would like to express their sincere thanks to the University of Malaya and the Ministry of Education, Malaysia, for their support given through research grants.

Author details

Meisam Gordan*, Zubaidah Ismail, Zainah Ibrahim and Huzaifa Hashim
Department of Civil Engineering, University of Malaya, Kuala Lumpur, Malaysia

*Address all correspondence to: meisam.gordan@gmail.com

IntechOpen

© 2019 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/3.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. 

References

- [1] Hanif MU, Ibrahim Z, Jameel M, Ghaedi K, Aslam M. A new approach to estimate damage in concrete beams using non-linearity. *Construction and Building Materials*. 2016;**124**:1081-1089
- [2] Hanif MU, Ibrahim Z, Jameel M, Ghaedi K, Hashim H. Simulation-based non-linear vibration model for damage detection in RC beams. *European Journal of Environmental and Civil Engineering*. March 2019:1-26. DOI: 10.1080/19648189.2019.1578270
- [3] Filiatrault A. *Principles of Passive Supplemental Damping and Seismic Isolation*. Pavia: Iuss Press; 2006
- [4] Ghayeb HH, Razak HA, Sulong NHR. Development and testing of hybrid precast concrete beam-to-column connections under cyclic loading. *Construction and Building Materials*. 2017;**151**:258-278. DOI: 10.1016/j.conbuildmat.2017.06.073
- [5] Gordan M, Ismail Z, Razak HA, Ibrahim Z. Vibration-based structural damage identification using data mining. In: 24th International Congress on Sound and Vibration. London; 2017
- [6] Gordan M, Ghaedi K. Experimental study on the effectiveness of tuned mass damper on a steel frame under harmonic load. In: 4th International Congress on Civil Engineering, Architecture & Urban Development, Shahid Beheshti University, Tehran. Tehran, Iran: Shahid Beheshti University; 2016
- [7] Hanif MU, Ibrahim Z, Ghaedi K, Javanmardi A, Rehman SK. Finite element simulation of damage In RC beams. *Journal of Civil Engineering, Science and Technology*. 2018;**9**:50-57
- [8] Ghaedi K, Ibrahim Z, Javanmardi A. A new metallic bar damper device for seismic energy dissipation of civil structures. IOP Conference Series: Materials Science and Engineering. 2018;**431**:1-7. DOI: 10.1088/1757-899X/431/12/122009
- [9] Javanmardi A, Ibrahim Z, Ghaedi K, Jameel M, Khatibi H, Suhatri M. Seismic response characteristics of a base isolated cable-stayed bridge under moderate and strong ground motions. *Archives of Civil and Mechanical Engineering*. 2017;**17**:419-432. DOI: 10.1016/j.acme.2016.12.002
- [10] Javanmardi A, Ibrahim Z, Ghaedi K, Khan NB, Ghadim HB. Seismic isolation retrofitting solution for an existing steel cable-stayed bridge. *PLoS One*. 2018;**13**:1-22. DOI: 10.1371/journal.pone.0200482
- [11] Ghaedi K, Jameel M, Ibrahim Z, Khanzaei P. Seismic analysis of roller compacted concrete (RCC) dams considering effect of sizes and shapes of galleries. *KSCE Journal of Civil Engineering*. 2016;**20**:261:261-272. DOI: 10.1007/s12205-015-0538-2
- [12] Ghaedi K, Hejazi F, Ibrahim Z, Khanzaei P. Flexible foundation effect on seismic analysis of roller compacted concrete (RCC) dams using finite element method. *KSCE Journal of Civil Engineering*. 2017;**22**:1-13
- [13] Ghaedi K, Ibrahim Z, Adeli H. Invited review: Recent developments in vibration control of building and bridge structures. *Journal of Vibroengineering*. 2017;**19**:3564-3580
- [14] Ghaedi K, Khanzaei P, Vaghei R, Fateh A, Javanmardi A, Gordan M, et al. Reservoir hydrostatic pressure effect on roller compacted concrete (RCC) dams. *Malaysian Construction Research Journal*. 2016;**19**:1-9
- [15] Gordan M. Experimental Investigation of Passive Tuned Mass

Damper and Fluid Viscous Damper on a Slender Two Dimension Steel Frame. Johor, Malaysia: University Technology of Malaysia; 2014

[16] Gordan M, Haddadiasl A, Marsono AK, Md Tap M. Investigation the behavior of a four-storey steel frame using viscous damper. *Applied Mechanics and Materials*. 2015;**735**:149-153

[17] Gordan M, Izadifar M, Haddadiasl A, Ahad J, Abadi R, Mohammadhosseini H. Interaction of across-wind and along-wind with tall buildings. *Australian Journal of Basic and Applied Sciences*. 2014;**8**:96-101

[18] Ghaedi K, Javanmardi A, Gordan M, Hamed K, Abdollah M. Application of 2D and 3D finite element modelling of gravity dams under seismic loading. In: 3rd National Graduate. Conference, Universiti Tenaga Nasional, Kuala Lumpur, Putrajaya Campus; 2015. pp. 264-269

[19] Ghaedi K, Ibrahim Z, Jameel M, Javanmardi A, Khatibi H. Seismic response analysis of fully base-isolated adjacent buildings with segregated foundations. *Advances in Civil Engineering*. 2018;**2018**:1-21. DOI: 10.1155/2018/4517940

[20] Ghaedi K, Ibrahim Z, Javanmardi A, Rupakhety R. Experimental study of a new Bar damper device for vibration control of structures subjected to earthquake loads. *Journal of Earthquake Engineering*. 2018:1-19. DOI: 10.1080/13632469.2018.1515796

[21] Javanmardi A, Ibrahim Z, Ghaedi K, Benisi Ghadim H, Hanif MU. State-of-the-art review of metallic dampers: Testing, development and implementation. *Archives of Computational Methods in Engineering*. 2019:1-24. DOI: 10.1007/s11831-019-09329-9

[22] Miranda T, Correia AG, Santos M, Ribeiro L, Cortez P. New models for strength and deformability parameter calculation in rock masses using data-mining techniques. *International Journal of Geomechanics*. 2011;**11**:44-58

[23] Buchheit RB, Garrett JH Jr, Lee SR, Brahme R. A knowledge discovery case study for the intelligent workplace. *Computing in Civil and Building Engineering*. 2000:914-921. DOI: 10.1061/40513(279)119

[24] Gordan M, Razak HA, Ismail Z, Ghaedi K. Recent developments in damage identification of structures using data mining. *Latin American Journal of Solids and Structures*. 2017;**14**:2373-2401. DOI: 10.1590/1679-78254378

[25] Liao S-H, Chu P-H, Hsiao P-Y. Data mining techniques and applications – A decade review from 2000 to 2011. *Expert Systems with Applications*. 2012;**39**:11303-11311

[26] Alves V, Cremona C, Cury A. On the use of symbolic vibration data for robust structural health monitoring. *Proceedings of the Institution of Civil Engineers*. 2015;**169**:715-723

[27] Gordan M, Razak HA, Ismail Z, Ghaedi K. Data mining based damage identification using imperialist competitive algorithm and artificial neural network. *Latin American Journal of Solids and Structures*. 2018;**15**:1-14

[28] Vapnik V. *The Nature of Statistical Learning Theory*. New York: Springer-Verlag; 1995

[29] He H-X, Yan W. Structural damage detection with wavelet support vector machine: Introduction and applications. *Structural Control and Health Monitoring*. 2007;**14**:162-176

[30] Tinoco J, Gomes Correia A, Cortez P. Support vector machines

applied to uniaxial compressive strength prediction of jet grouting columns. *Computers and Geotechnics*. 2014;**55**:132-140

[31] Kishore B, Satyanarayana MRS, Sujatha K. Efficient fault detection using support vector machine based hybrid expert system. *International Journal of Systems Assurance Engineering and Management*. 2014;34-40. DOI: 10.1007/s13198-014-0281-y

[32] Li C, Liu Q. Support vector machine based semi-active control of structures: A new control strategy. *Structural Design of Tall and Special Buildings*. 2011;**20**:711-720

[33] Ahmed R, El Sayed M, Gadsden SA, Tjong J, Habibi S. Artificial neural network training utilizing the smooth variable structure filter estimation strategy. *Neural Computing and Applications*. 2015;**27**:537-548

[34] Ali A, Amin SE, Ramadan HH, Tolba MF. Enhancement of OMI aerosol optical depth data assimilation using artificial neural network. *Neural Computing and Applications*. 2013;**23**:2267-2279

[35] Azimzadegan T, Khoeini M, Etaat M, Khoshakhlagh A. An artificial neural-network model for impact properties in X70 pipeline steels. *Neural Computing and Applications*. 2012;**23**:1473-1480

[36] Suresh S, Narasimhan S, Nagarajaiah S. Direct adaptive neural controller for the active control of earthquake-excited nonlinear base-isolated buildings. *Structural Control and Health Monitoring*. 2012;**19**:370-384

[37] Mitchell R, Kim Y, El-Korchi T, Cha Y-J. Wavelet-neuro-fuzzy control of hybrid building-active tuned mass damper system under seismic excitations. *Journal of Vibration and Control*. 2012;**19**:1881-1894

[38] Jung H-J, Lee H-J, Yoon W-H, Oh J-W, Lee I-W. Semiactive neurocontrol for seismic response reduction using smart damping strategy. *Journal of Computing in Civil Engineering*. 2004;**18**:277-281

[39] Chen ZH, Ni YQ. On-board identification and control performance verification of an MR damper incorporated with structure. *Journal of Intelligent Material Systems and Structures*. 2011;**22**:1551-1565

[40] Vaidyanathan CV, Kamatchi P, Ravichandran R. Artificial neural networks for predicting the response of structural systems with viscoelastic dampers. *Computer-Aided Civil and Infrastructure Engineering*. 2005;**20**:294-302

[41] Lin T, Chang K, Chung L, Lin Y. Active control with optical fiber sensors and neural networks. I: Theoretical analysis. *Journal of Structural Engineering*. 2006;**132**:1293-1304

[42] Lin T, Chang K, Lin Y. Active control with optical fiber sensors and neural networks. II: experimental verification. *Journal of Structural Engineering*. 2006;**132**:1304-1314

[43] Weber F, Bhowmik S, Høgsberg J. Extended neural network-based scheme for real-time force tracking with magnetorheological dampers. *Structural Control and Health Monitoring*. 2014;**21**:225-247

[44] Rutkowski L. *Flexible Neuro-Fuzzy Systems: Structures, Learning and Performance Evaluation*. Poland: Technical University of Czestochowa, Springer Science & Business Media; 2004

[45] Nyongesa HO. Enhancing neural control systems by fuzzy logic and evolutionary reinforcement. *Neural Computing and Applications*. 1998;**7**:121-130

- [46] Nerves AC, Krishnan R. Active control strategies for tall civil structures. Proceedings of IECON'95-21st Annual Conference on IEEE Industrial Electronics. 1995;2:962-967
- [47] Zhou L, Chang C, Wang L. Adaptive fuzzy control for nonlinear building–Magnetorheological damper system. Journal of Structural Engineering. 2003;129:905-913
- [48] Wongprasert N, Symans MD. Experimental evaluation of adaptive elastomeric Base-isolated structures using variable-orifice fluid dampers. 2005;131:867-877
- [49] Ghaffarzadeh H, Dehrod EA, Talebian N. Semi-active fuzzy control for seismic response reduction of building frames using variable orifice dampers subjected to near-fault earthquakes. Journal of Vibration and Control. 2012;19:1980-1998
- [50] Kim Y, Langari R, Hurlebaus S. Control of a seismically excited benchmark building nonlinear fuzzy control. Journal of Structural Engineering. 2010;136:1023-1026
- [51] Ahlawat AS, Ramaswamy A. Multiobjective optimal fuzzy logic controller driven active and hybrid control systems for seismically excited nonlinear buildings. Journal of Engineering Mechanics. 2002;130:416-423
- [52] Samali B, Al-dawod M, Kwok KCS, Naghdy F. Active control of cross wind response of 76-story tall building using a fuzzy controller. Journal of Engineering Mechanics. 2004;130:492-498
- [53] Soleymani M, Khodadadi M. Adaptive fuzzy controller for active tuned mass damper of a benchmark tall building subjected to seismic and wind loads. Structural Design of Tall and Special Buildings. 2014;23:781-800
- [54] Lin C-J, Yau H-T, Lee C-Y, Tung K-H. System identification and semiactive control of a squeeze-mode magnetorheological damper. IEEE/ASME Transactions on Mechatronics. 2013;18:1691-1701
- [55] Kim H, Roschke PN. Fuzzy control of base-isolation system using multi-objective genetic algorithm. Computer-Aided Civil and Infrastructure Engineering. 2006;21:436-449
- [56] Reigles DG, Symans MD. Supervisory fuzzy control of a base-isolated benchmark building utilizing a neuro-fuzzy model of controllable fluid viscous dampers. Structural Control and Health Monitoring. 2006;13:724-747
- [57] Shook DA, Roschke PN, Ozbulut OE. Superelastic semi-active damping of a base-isolated structure. Structural Control and Health Monitoring. 2008;15:746-768
- [58] Mohtat A, Yousefi-Koma A, Dehghan-Niri E. Active vibration control of seismically excited structures by Atmds: Stability and performance robustness perspective. International Journal of Structural Stability and Dynamics. 2010;10:501-527
- [59] Guo Y-Q, Fei S-M, Xu Z-D. Simulation analysis on intelligent structures with magnetorheological dampers. Journal of Intelligent Material Systems and Structures. 2007;19:715-726
- [60] Pourzeynali S, Datta TK. Semiactive fuzzy logic control of suspension. Journal of Structural Engineering. 2005;131:900-912
- [61] Gu ZQ, Oyadiji SO. Application of MR damper in structural control using ANFIS method. Computers and Structures. 2008;86:427-436. DOI: 10.1016/j.compstruc.2007.02.024

- [62] Ghaedi K, Ibrahim Z. Earthquake prediction. In: Zouaghi T, editor. *Earthquakes - Tectonics, Hazard Risk Mitig.* Rijeka, Croatia: InTech; 2017. pp. 205-227. DOI: 10.5772/65511
- [63] Symeonidis A, Mitkas P. Data mining and knowledge discovery: A brief overview. In: *Agent Intelligence Through Data Mining.* United States: Springer; 2005. pp. 11-40
- [64] Chen TY, Huang JH. Application of data mining in a global optimization algorithm. *Advances in Engineering Software.* 2013;**66**:24-33
- [65] Xiao F, Fan C. Data mining in building automation system for improving building operational performance. *Energy and Buildings.* 2014;**75**:109-118
- [66] Kim Y, Langari R, Hurlebaus S. MIMO fuzzy identification of building-MR damper systems. *Journal of Intelligent Fuzzy Systems.* 2011;**22**:185-205
- [67] Aghajanloo M, Sabziparvar A. Artificial neural network – Genetic algorithm for estimation of crop evapotranspiration in a semi-arid region of Iran. *Neural Computing and Applications.* 2012;**23**:1387-1393
- [68] Pourzeynali S, Salimi S. Robust multi-objective optimization design of active tuned mass damper system to mitigate the vibrations of a high-rise building. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science.* 2014
- [69] Singh MP, Singh S, Moreschi LM. Tuned mass dampers for response control of torsional buildings. *Earthquake Engineering and Structural Dynamics.* 2002;**31**:749-769
- [70] Mohebbi M, Joghataie A. Designing optimal tuned mass dampers for nonlinear frames by distributed genetic algorithms. *Structural Design of Tall and Special Buildings.* 2012;**21**:57-76
- [71] Ahadi P, Mohebbi M, Shakeri K. Using optimal multiple tuned liquid column dampers for mitigating the seismic response of structures. *ISRN Civil Engineering.* 2012;**2012**:1-6
- [72] Hejazi F, Toloue I, Jaafar MS, Noorzaei J. Optimization of earthquake energy dissipation system by genetic algorithm. *Computer-Aided Civil and Infrastructure Engineering.* 2013;**28**:796-810
- [73] Cha Y-J, Kim Y, Raich AM, Agrawal AK. Multi-objective optimization for actuator and sensor layouts of actively controlled 3D buildings. *Journal of Vibration and Control.* 2012;**19**:942-960
- [74] Wongprasert N, Symans MD. Application of a genetic algorithm for optimal damper. *Journal of Engineering Mechanics.* 2004;**130**:401-406
- [75] Lin T, Chu Y, Chang K, Chang C. Renovated controller designed by genetic algorithms. *Earthquake Engineering and Structural Dynamics.* 2009;**38**:457-475
- [76] Kennedy J, Eberhart R. Particle swarm optimization. *Proceedings of ICNN'95- International Conference on Neural Networks.* 1995;**4**:1942-1948
- [77] Ghayeb HH, Razak HA, Sulong NHR, Hanoon AN, Abutaha F, Ibrahim HA, et al. Predicting the mechanical properties of concrete using intelligent techniques to reduce CO2 emissions. *Materiales de Construcción.* 2019;**69**:1-20
- [78] Gholizadeh S, Fattahi F. Design optimization of tall steel buildings by a modified particle. *Structural Design of Tall and Special Buildings.* 2014;**23**:285-301

[79] Gundogdu O, Egrioglu E, Aladag CH, Yolcu U. Multiplicative neuron model artificial neural network based on Gaussian activation function. *Neural Computing and Applications*. 2015;27:927-935

[80] Leung AYT, Zhang H, Cheng CC, Lee YY. Particle swarm optimization of TMD by non-stationary base excitation during earthquake. *Earthquake Engineering and Structural Dynamics*. 2008;37:1223-1246

[81] Amini F, Hazaveh NK, Rad AA. Wavelet PSO-based LQR algorithm for optimal structural control using active tuned mass dampers. *Computer-Aided Civil and Infrastructure Engineering*. 2013;28:542-557

*Edited by Ali Sadollah
and Carlos M. Travieso-Gonzalez*

Artificial intelligence (AI) is everywhere and it's here to stay. Most aspects of our lives are now touched by artificial intelligence in one way or another, from deciding what books or flights to buy online to whether our job applications are successful, whether we receive a bank loan, and even what treatment we receive for cancer. Artificial Neural Networks (ANNs) as a part of AI maintains the capacity to solve problems such as regression and classification with high levels of accuracy.

This book aims to discuss the usage of ANNs for optimal solving of time series applications and clustering. Bounding of optimization methods particularly metaheuristics considered as global optimizers with ANNs make a strong and reliable prediction tool for handling real-life application. This book also demonstrates how different fields of studies utilize ANNs proving its wide reach and relevance.

Published in London, UK

© 2020 IntechOpen
© ktsimage / iStock

IntechOpen

