Frontiers in Artificial Intelligence and Applications

NEW DIRECTIONS IN NEURAL NETWORKS

18th Italian Workshop on Neural Networks: WIRN 2008

Edited by
Bruno Apolloni
Simone Bassis
Maria Marinaro

IOS Press

New Directions in Neural Networks

18th Italian Workshop on Neural Networks: WIRN 2008

Edited by

Bruno Apolloni

Università degli Studi di Milano,
Dipartimento di Scienze dell'Informazione, Milano, Italy

Simone Bassis

Università degli Studi di Milano,
Dipartimento di Scienze dell'Informazione, Milano, Italy

and

Maria Marinaro

International Institute for Advanced Scientific Studies,
Vietri sul Mare (Salerno), Italy

IOS
Press

Amsterdam • Berlin • Tokyo • Washington, DC

ISBN 978-1-58603-984-4
Library of Congress Control Number: 2009921089

Publisher
IOS Press BV
Nieuwe Hemweg 6B
1013 BG Amsterdam
Netherlands
fax: +31 20 687 0019
e-mail: order@iospress.nl

Distributor in the UK and Ireland
Gazelle Books Services Ltd.
White Cross Mills
Hightown
Lancaster LA1 4XS
United Kingdom
fax: +44 1524 63232
e-mail: sales@gazellebooks.co.uk

Distributor in the USA and Canada
IOS Press, Inc.
4502 Rachael Manor Drive
Fairfax, VA 22032
USA
fax: +1 703 323 3668
e-mail: iosbooks@iospress.com

PRINTED IN THE NETHERLANDS

New Directions in Neural Networks
B. Apolloni et al. (Eds.)
IOS Press, 2009

Preface

The book is a collection of selected papers from the 18th WIRN workshop, the annual meeting of the Italian Neural Networks Society (SIREN). As 18 marks the year young people come of age in Italy, the Society invited two generations of researchers to participate in a common discussion: those new to the field and those with extensive familiarity with the neural paradigm. The challenge lay in understanding what remains of the revolutionary ideas from which neural networks stemmed in the eighties, how these networks have evolved and influenced other research fields, and ultimately what are the new conceptual/methodological frontiers to trespass for a better exploitation of the information carried by data.

From this discussion we selected 27 papers which have been gathered under two general headings, "Models" and "Applications," plus two specific ones, "Economy and Complexity" and "Remote Sensing Image Processing." The editors would like to thank the invited speakers as well as all those who contributed to the success of the workshops with papers of outstanding quality. Finally, special thanks go to the referees for their valuable input.

We are also pleased that non-SIREN member researchers joined us both at the meeting and in this editorial venture, bearing witness to a wide sphere of interest in the debate. We hope, moreover, that the book will serve in making a scientific contribution to the discovery of new forms of cooperative work – so necessary today for the invention of efficient computational systems and new social paradigms too.

November 2008

Bruno Apolloni
Simone Bassis
Maria Marinaro

Organizing-Scientific Commitee

B. Apolloni (Univ. Milano), A. Bertoni (Univ. Milano), N.A. Borghese (Univ. Milano), D.D. Caviglia (Univ. Genova), P. Campadelli (Univ. Milano), F. Celada (Univ. Genova), J. Chanussot (Univ. Grenoble, France), A. Chella (Univ. Palermo), A. M. Colla (ELSAG Genova), M. Corazza (Univ. Venezia), A. Esposito (Univ. Napoli II), G. Facchinetti (Univ. Modena e Reggio Emilia), F.M. Frattale Mascioli (Univ. Roma "La Sapienza"), C. Furlanello (ITC-IRST Trento), S. Giove (Univ. Venezia), M. Gori (Univ. Siena), T. Kavzoğlu (Univ. Gebze, Turkey), M. Marinaro (Univ. Salerno), F. Masulli (Univ. Pisa), C. Morabito (Univ. Reggio Calabria), P. Morasso (Univ. Genova), G. Orlandi (Univ. Roma "La Sapienza"), F. Palmieri (Univ. Napoli), T. Parisini (Univ. Trieste), E. Pasero (Politecnico Torino), A. Petrosino (Univ. Napoli "Parthenope"), V. Piuri (Univ. Milano), P. Richmond (Trinity College, Ireland), R. Serra (Univ. di Modena e Reggio Emilia), F. Sorbello (Univ. Palermo), A. Sperduti (Univ. Padova), R. Tagliaferri (Univ. Salerno), J. Taylor (King's College London), A. Uncini (Univ. Roma)

Referees

Apolloni Bruno, Bassis Simone, Bertoni Alberto, Borghese Alberto, Corazza Marco, Di Gregorio Salvatore, Esposito Anna, Faggini Marisa, Fuller Robert, Gaito Sabrina, Giove Silvio, Gori Marco, Goffredo Haus, LaForesta Fabio, Lombardi Gabriele, Malchiodi Dario, Marinaro Maria, Martone Raffaele, Frattale Mascioli Fabio Massimo, Masulli Francesco, Morabito Carlo, Nicosia Giuseppe, Palmieri Francesco, Pasero Eros, Pasi Gabriella, Rovetta Stefano, Scarpetta Silvia, Serra Roberto, Sorbello Filippo, Sperduti Alessandro, Tagliaferri Roberto, Uncini Aurelio, Valentini Giorgio, Valerio Lorenzo, Vitabile Salvatore, Zippo Antonio, Zoppis Italo

Contents

Preface v
Bruno Apolloni, Simone Bassis and Maria Marinaro

Chapter 1. Application

A Neural Based WSN Mote Trajectory Reconstruction for Mining Periodic Patterns 3
Alfredo Petrosino and Antonino Staiano

Sugeno Model Based Control for Semi Active Vehicle Suspension 12
Francesco Voci and Fabio Massimo Frattale Mascioli

Automatic Classification of Defects in an Industrial Environment 22
N. Alberto Borghese

STLmax Joint Mutual Information for Quantifying Independence in the Epileptic Brain 30
Nadia Mammone, Fabio La Foresta, Francesco C. Morabito and Mario Versaci

Neural Networks for the Parameters Characterization of ECG Dynamical Model 40
Matteo Cacciola, Fabio La Foresta, Francesco C. Morabito and Mario Versaci

Multiresolution Minimization of Renyi's Mutual Information for Fetal-ECG Extraction 50
Fabio La Foresta, Nadia Mammone, Giuseppina Inuso, Francesco C. Morabito and Andrea Azzerboni

Classification of DNA Microarray Data with Random Projection Ensembles of Polynomial SVMs 60
Alberto Bertoni, Raffaella Folgieri and Giorgio Valentini

An Embedded Fingerprints Classification System Based on Weightless Neural Networks 67
Vincenzo Conti, Carmelo Militello, Filippo Sorbello and Salvatore Vitabile

Discovering Regression Data Quality Through Clustering Methods 76
Dario Malchiodi, Simone Bassis and Lorenzo Valerio

Chapter 2. Models

A Flexible Natural Gradient Approach to Blind Separation of Complex Signals 89
Michele Scarpiniti, Daniele Vigliano, Raffaele Parisi and Aurelio Uncini

Single Channel Polyphonic Music Transcription 99
Angelo Ciaramella

Genetic Regulatory Networks and Neural Networks 109
Roberto Serra, Alex Graudenzi and Marco Villani

An Experimental Comparison of Kernel Clustering Methods 118
Maurizio Filippone, Francesco Masulli and Stefano Rovetta

Diffusion Learning and Regularization 127
Marco Gori

Sources of Asymmetric Randomness 138
Bruno Apolloni and Simone Bassis

Random Projections Preserving the Hamming Distance Between Words 148
Stefano Arca, Alberto Bertoni and Giuseppe Lipori

Notes on Factor Graphs 154
Francesco Palmieri

Chapter 3. Economy and Complexity Workshop

Safety Loading in the Annuity Pension Fund Dynamics 165
Marilena Sibillo, Mariarosaria Coppola, Valeria D'Amato and Emilia Di Lorenzo

Early Warning Systems: An Approach via Self Organizing Maps with Applications to Emergent Markets 176
Marina Resta

Optimal Partial Questioning in Large Dimensional AHP 185
Michele Fedrizzi and Silvio Giove

Evaluation of Fuzzy Quantities by Means of a Weighting Function 194
Gisella Facchinetti and Nicoletta Pacchiarotti

Neural Network Sieve Bootstrap Prediction Intervals: Some Real Data Evidence 205
Francesco Giordano, Michele La Rocca and Cira Perna

Fuzzy Interval Net Present Value 214
Marco Corazza and Silvio Giove

Interactive Visualization Tools for Meta-Clustering 223
Ida Bifulco, Francesco Iorio, Francesco Napolitano, Giancarlo Raiconi and Roberto Tagliaferri

Chapter 4. Remote Sensing Images Processing

Remote Sensing Imagery for Soil Characterization: A Wavelet Neural Data Fusion Approach 235
Matteo Cacciola, Francesco Carlo Morabito and Vincenzo Barrile

Application of Self Organizing Maps to Multi-Resolution and Multi-Spectral Remote Sensed Images 245
Ferdinando Giacco, Silvia Scarpetta, Luca Pugliese, Maria Marinaro and Christian Thiel

Comparison of Neural Classification Algorithms Applied to Land Cover Mapping 254
Christian Thiel, Ferdinando Giacco, Friedhelm Schwenker and Günther Palm

Author Index 265

Chapter 1

Application

New Directions in Neural Networks
B. Apolloni et al. (Eds.)
IOS Press, 2009

doi:10.3233/978-1-58603-984-4-3

A Neural Based WSN Mote Trajectory Reconstruction for Mining Periodic Patterns

Alfredo PETROSINO [a] and Antonino STAIANO [a,1]
[a] *Dipartimento di Scienze Applicate, Università di Napoli "Parthenope" Centro Direzionale - Isola C4, I-80143 Napoli, Italy*

Abstract. The problem of reconstruction and mining object trajectories is of interest in the applications of mining transport enterprise data concerning with the route followed by its delivery vans in order to optimize time and space deliveries. The paper investigates the case of Wireless Sensor Network (WSN) technology, not primarily designed for localization, and reports a technique based on recurrent neural networks to reconstruct the trajectory shape of a moving object (a sensor on a Lego train) from the sensor accelerometer data and to recover its localization. The obtained patterns are thus mined to detecting periodic or frequent patterns, exploiting a recently proposed technique based on clustering algorithms and associative rules to assert the ability of the proposed approach to track WSN mote localizations.

Keywords. Wireless Sensor Networks, Spatio-Temporal Data mining, Recurrent Neural Networks

Introduction

The efficient management of spatio-temporal data has gained much interest during the past few years [7,2], mainly due to the rapid advancement in telecommunications which facilitate the collection of large data set of such information. Management and analysis of moving object trajectories are challenging due to the vast amount of collected data and novel types of spatio-temporal queries [5]. In many applications, the movements obey periodic patterns, i.e., the objects follow the same routes (approximately) over regular time intervals. Objects that follow approximate periodic patterns include transportation vehicles (buses, boats, airplanes, trains, etc.), animals, mobile phone users, etc. The problem of discovering periodic patterns from historical object movements is very challenging. Usually, the patterns are not explicitly specified, but have to be discovered from the data. The approximate nature of patterns in the spatio-temporal domain increases the complexity of the mining tasks.

One of the most challenging problems is how discovering periodic patterns from historical object movements, in an independent manner from the adopted technology

[1] Corresponding Author: Antonino Staiano, Dipartimento di Scienze Applicate, Università di Napoli "Parthenope", Centro Direzionale - Isola C4, I-80143 Napoli, Italy; E-mail: antonino.staiano@uniparthenope.it.

that not always may assure the complete covering of the object localizations. As known, a possible solution is to let mobile nodes deploy expensive global positioning systems (GPS) to get their locations. However, many applications require sensor network mobility in the environments where GPS signals may not be available, therefore a GPS solution is not feasible. In recent years Wireless Sensor Networks (WSNs) have emerged as one of the key enablers for a variety of applications such as environment monitoring, vehicle tracking and mapping, and emergency response [8]. The network of such devices, called sensor network, forms a distributed and adaptive platform for data gathering, fusion and aggregation of the observed process.

In this paper we focus our attention on two important and related issues concerning with the tracking of a moving object by adopting WSNs. The aim is to develop and study a small scale system from which to derive, although with some assumptions and constraints, the trajectory of a WSN single moving sensor placed on a Lego train and to mine the periodic patterns from the reconstructed trajectory.

The first issue is how to locate a node's (a moving sensor) position. Although many localization algorithms have been proposed for wireless sensor networks, they assume that the nodes inside the networks are static. Little research has been presented on considering localization in cases where the network cannot be assumed static (see [9] and references therein). Here we describe a procedure to gain some insights to determine the trajectory out of acceleration reading from a moving mote located on the object. Extrapolating a trajectory out of acceleration is very difficult, since the accumulative error of the accelerometer equipping the motes gets too large. If one gets a model of the train tracks, then it might be possible to use the accelerometer information to detect the various turns and use that to keep track of the train's position. To highlight the proposed model property, we made experiments by using a little Lego [10] train that follows different controlled tracks.

The second addressed issue is to proper analyze the spatiotemporal data acquired. Although several algorithms for mining frequent patterns in large databases have been proved very effective for a variety of static data, no one is well suited for spatio-temporal sequences. An interesting and effective approach is described in [2], where an algorithm based on the well known DBScan [4] clustering method and a variant of the algorithm Apriori-TID [1] for the analysis of associative rules is proposed.

The paper is organized as follows: in Section 1 we describe the problem of determining the trajectory out of accelerometer data and the proposed approach based on a recurrent neural network. In Section 2, the problem of mining periodic patterns is introduced and the adopted algorithms are described. Experimental results, showing how to compute the trajectory shape and the coordinates of a Lego train equipped with a Tmote Invent and how discover periodic patterns from the data are described in Section 3. Finally, in Section 4, some concluding remarks and the description of possible applicative scenarios close the paper.

1. Computing trajectory coordinates

Motes are equipped with an accelerometer which provides the base station with the acceleration measures on the $x--y$ axes. One possible solution to derive location information from acceleration is to double integrate acceleration over time to derive the sensor resulting movement. However, the motes have an operating frequency at 50Hz making them very sensitive to any vibration. Furthermore, the battery life cycle is very short so that the accelerometer readings might differ largely even for the same piece of trajectory in different timestamps. This introduces an overall very high noise over the signal so that the integration leads to a very poor representation of the actual trajectory even for very simple piece of route such as a straight line. This led us to consider the trajectory shape reconstruction and then to derive the mote positions by adding some constraints on the train movements on the tracks.

1.1. Trajectory reconstruction

The idea is to recognize pieces of the circuit followed by the train considering that when the train is on a back straight the corresponding acceleration values are near zero, if the train is turning left the acceleration values are negatives, while if the train is turning right the corresponding acceleration values are positives. Therefore, after a preprocessing step to reduce noise over the acceleration signal acquired by the sensor through a FIR filter, an Elman neural network [3,6] is used to recognize pieces of the circuit followed by the train.

1.1.1. Elman network

The Elman network is a recurrent feed-forward neural network. Recurrent neural networks allows to model dynamic mappings according to which the input and output variables change over time and time is not represented explicitly but by the effect it has on processing and not as an additional dimension of the input. This class of neural networks are well suited, among all, for times series prediction [6]. The Elman network differs from a classical feed-forward neural network structure for a further layer placed at the same level of the input layer, called *context unit layer* (for which the weights are fixed to 1), whose task is to hold memory of the hidden layer output at the previous time instant. Therefore, when the input is fed to te network at time t, its output is computed also on the base of the hidden layer neuron activations at time $t-1$, i.e.

$$u_k = g(W_1' I_k + W_1'' y_{k-1}^h + b_1) \qquad y_k = f(W_2 u_k + b_2) \tag{1}$$

where g and f are respectively the hyperbolic tangent and logistic sigmoid functions, W_1' the weight matrix from input to hidden layers, W_1'' the weight matrix from hidden to context layers and W_2 the weight matrix from hidden to output layers. The Elman network is trained with the backpropagation to adjust the hidden-to-output weights and the input-to-hidden weights, while the recurrent connections are fixed at 1 and are not subject to adjustment. Our Elman network takes, each time, an acceleration value as input, from a training set, and tries to recognize if that value correspond to a back straight, a left turn or a right turn on the base of the previous time instants. Each segment is coded by a 1-of-c coding, so three output values are used to code a segment shape class (100 for back straights, 010 for left turns and 001 for right turns). The network is trained over a data set of segments of curve of each direction taken over different real mote trajectories and labelled by hand. Since the network gives evidence of the segment turn but not localizations in the the corresponding $x-y$ coordinates, we need to properly assemble each recognized segment to obtain the shape and the coordinates of the entire circuit.

1.1.2. Circuit segments assembling

First of all, the train speed is assumed to be constant over the entire route making it possible to calculate the coordinates of each recognized circuit segment. In fact, with a constant speed, and knowing that the accelerometer sampling rate is 0.1 seconds, it is possible to compute the coordinates for the back straights through a simple procedure. For left and right turns, it is necessary to establish the train time to go through a left or right curve. Here a further assumption comes, i.e., we consider that each turn is at least a $90°$ curve (corresponding to 4 pieces of Lego tracks) and all its multiples $(180°, 270°)$. Therefore, knowing that the circumference length assembled by the Lego curve tracks (64 cm), it is possible to compute the coordinates of the $90°$ curve.

The procedure determines the positions of the first segment, then it recognizes the next segment on the base of the Elman network output, and finally calculates the positions of the recognized segment updating the route orientation. These steps are iterated for the entire circuit. The starting point is assumed to be the $(0, 0)$ coordinate, updated by the last location obtained for each computed segment. The orientation is updated by considering three conditions, namely, the orientation of the previous segment, the turn (left or right) and the dimension $(90°, 180°, 270°)$. As an example, let us consider a simple elliptic circuit assuming that the train started its path on a back straight. At first step, the algorithm computes the locations of the first circuit segment, next it recognizes that the train turns on the left by $180°$ and calculates its coordinates by combining two $90°$ curves. These curves are properly translated to overlap the origin of the curve with the last location of the previous segment; the orientation is next updated. These steps are iterated for the entire circuit. The step-wise results are shown in Figure 1.

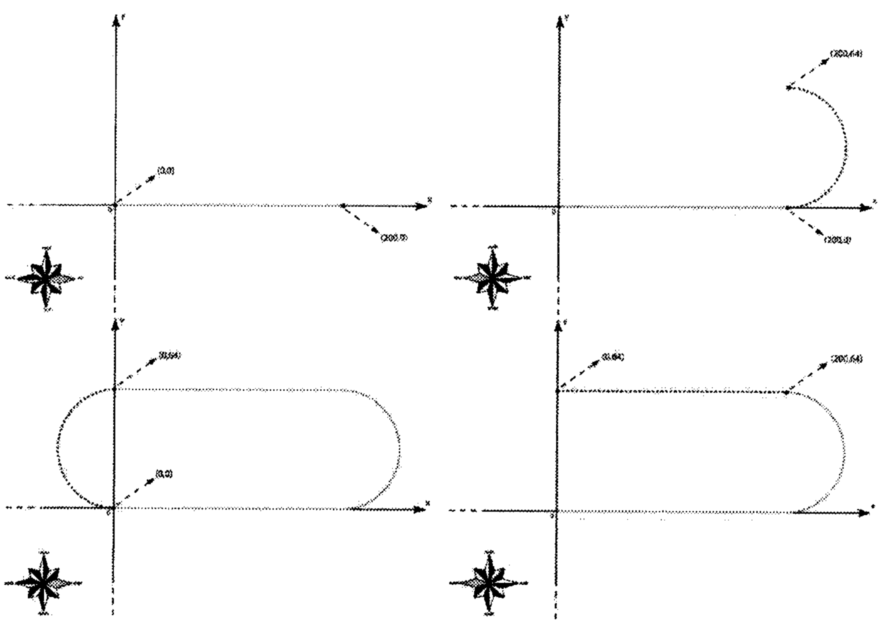

Figure 1. Steps of the reconstruction procedure.

2. Periodic routes detection

The problem may be defined as follows. An object movement is defined as a n-length sequence S of spatial locations, one for each time point (l_i, t_i), $i = 0, \ldots n-1$, where l is the object location at time t. Each location l_i, is a spatial location (x_i, y_i). Given a minimum support min_sup $(0 < min\_sup \leq 1)$ and an integer T (the period), the goal is to discover movement patterns that repeat themselves every T time points. The pattern is defined as a T-length sequence of the form $P = r_0, r_1, \ldots, r_{T-1}$, where r_i is a spatial region or a wild-card character $*$ indicating any region of the whole spatial universe. As an example, the pattern $ABC * *$ denotes that the object is in region A at the starting point, and in the next two time intervals it is in region B and C, respectively, while in the last two time intervals it moved irregularly (at every time point it is in a different region), until a new cycle begins. The patterns are required to be followed by the object in at least α ($\alpha = min\_sup \cdot \lfloor \frac{n}{T} \rfloor$) periodic intervals in S. Since it is unlikely that an object repeats an identical sequence of (x, y) locations precisely, the exact locations are replaced by the regions which contain them.

STPMine1 (SpatioTemporal periodic Pattern Min(E)ing1) is the first algorithm for mining periodic patterns proposed in [2] and is the one we used. In the following we discuss it informally, the interested reader may refer to [2] for a complete description. The algorithm is structured in two parts:

1. First, the algorithm uses DBScan clustering approach to discover 1-length patterns. So doing each spatial location is assigned, in automatic way, a spatial region which is a valid cluster found to which the spatial location belongs. A valid cluster is a cluster containing at least MinPoint elements (MinPoint is a user defined parameter of the algorithm). For each cluster found a 1-length pattern is generated.
2. Starting from the 1-length patterns of the previous step, STPMine1 implements a variant of the Apriori-TID algorithm to find, iteratively, patterns of length greater than 1. For this aim, it generates candidate patterns joining pairs of patterns. Here, eventually invalid candidate pattern are pruned (patterns whose subpatterns are not frequent patterns). Once a valid candidate pattern is generated it is checked whether its regions are still clusters. This is a validation procedure: points at non-* candidate pattern positions could not be a valid cluster. The procedure is iterated until no candidate patterns can formed anymore. The last valid pattern formed will be the discovered frequent pattern.

3. Experimental results

The trajectory reconstruction approach and STPMine1 algorithm for mining periodic patterns have been tested on several circuit shapes. In the following only the obtained results on a Z-shaped circuit (two joining ellipses, see Figure 2, corresponding to 92 spatial locations) are described.

3.1. Trajectory reconstruction

To properly comment the trajectory reconstruction results, some details on parameter setting should be introduced.

Table 1. Circuit segment reconstruction errors. Each column is a circuit segment, i.e., Back straight, Right and Left curves. The length are in cm.

	90°R	B1	90°L	B2	180°R	B3	90°R	B4	90°L	B5	180°R	B6
Track	-	6	-	8	-	8	-	6	-	8	-	9
Length	-	75	-	100	-	100	-	75	-	100	-	112.5
Ref length	-	72	-	97	-	102	-	70	-	98	-	120
Error	-	-3	-	-3	-	+2	-	-5	-	-2	-	+7.5

- Train speed. As stressed in Section 1, one assumption is the constant speed of our train. The actual speed has been measured by an external chronometer by considering the time necessary to go through a one meter length back straight. Several measurements, under the same conditions, have been made. The constant train speed was chosen as the mean speed of $1.14\frac{m}{s}$, over all the measurements.
- Elman network. The training sets have been obtained by acquiring the accelerometer data over two laps of each considered circuit in both directions (east-west and west-east) in order to provide the network with similar circuits with all possible segment directions. The employed network is a one input - three output (1 of c coding) network, with 15 hidden neurons (therefore 15 context unit neurons), 500 learning epochs and a threshold error set to 0.01.

After the training phase, the Elman network has been evaluated on a test set and procedures as descibed above were applied to derive the shape and locations of the entire circuit. Figure 2 illustrates the original and the reconstructed circuit. As shown, the shape of the reconstructed circuit is very close to the original. Table 1 shows the the reconstruction error on each segment, corresponding to an overall error on the entire circuits of 22.5 cm (4% of the total length).

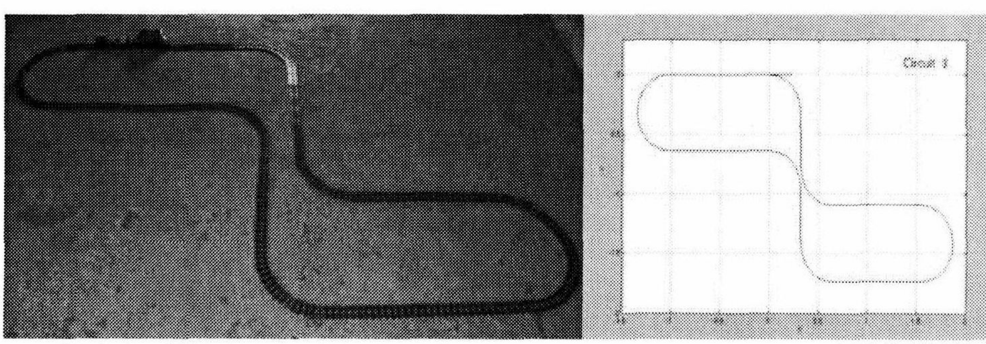

Figure 2. The original Z-shaped circuit (left) and the reconstructed circuit (right).

3.2. *Mining frequent patterns*

Starting from the circuit spatial location computed as described in Section 1, several variants of the same circuit are ncessary to find periodic patterns. Therefore, we generated, from the circuit spatial locations, n similar pseudo-random circuits with different coordinates. The STPMine1 algorithm was applied to the Z-shaped circuit and its generated variants (see Figure 3). In the following we see two different experiments where some

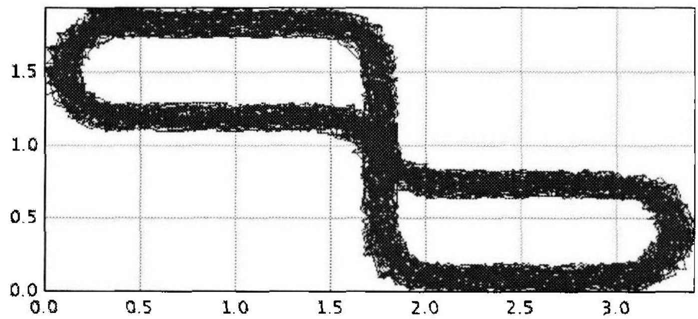

Figure 3. The Z-shaped circuits generated.

STPMine1 critical parameters are changed. In particular, the parameter Eps (points minimum distance to find clusters) of DBScan is changed ($Eps = 0.05$ and $Eps = 0.09$) in order to observe its impact on the discovered patterns. Indeed, an Eps low value causes DBScan to find just few clusters, while, on the other side, a very high value causes DBScan to detect all $1-$length pattern as one cluster. Obviously, this latter case implies high time costs. In Table 2, the results for $Esp = 0.05$ are reported. It is interesting to note the computing time and a frequent pattern of reduced size with respect to the length of the timestamps (92) of the circuit. In Table 3, instead, DBScan carries out a more accurate cluster detection, so the algorithm is able to detect the entire 92 timestamps of the circuit as periodic pattern.