Mathematical and Computer Modelling 51 (2010) 229–238

Contents lists available at ScienceDirect

Mathematical and Computer Modelling

journal homepage: www.elsevier.com/locate/mcm

Evolving cellular automata rules for multiple-step-ahead prediction of complex binary sequences

A.V. Adamopoulos [a], N.G. Pavlidis [b], M.N. Vrahatis [b,*]

[a] *Medical Physics Laboratory, Department of Medicine, Democritus University of Thrace, GR-681 00 Alexandroupolis, Greece*

[b] *Computational Intelligence Laboratory (CI Lab), Department of Mathematics, University of Patras Artificial Intelligence Research Center (UPAIRC), University of Patras, GR-26110 Patras, Greece*

ARTICLE INFO

Keywords:
Cellular automata
Genetic algorithms
Complex binary sequence prediction
Multiple-step-ahead forecasting

ABSTRACT

Complex binary sequences are generated through the application of simple threshold, linear transformations to the logistic iterative map. Depending primarily on the value of its non-linearity parameter, the logistic map exhibits a great variety of behavior, including stable states, cycling and periodical activity and the period doubling phenomenon that leads to high-order chaos. From the real data sequences, binary sequences are derived. Consecutive L bit sequences are given as input to a cellular automaton with the task to regenerate the subsequent L bits of the binary sequence in precisely L evolution steps. To perform this task a genetic algorithm is employed to evolve cellular automaton rules. Various complex binary sequences are examined, for a variety of initial values and a wide range of values of the non-linearity parameter. The proposed hybrid multiple-step-ahead prediction algorithm, based on a combination of genetic algorithms and cellular automata proved efficient and effective.

1. Introduction

Cellular automata (CA) are decentralized structures of simple and locally interacting elements, named *cells*. A CA starts with an initial configuration that refers to the initial state of its cells and it evolves according to a set of rules [1,2]. The set of rules determines the derived cell states (values) at the next evolution step, for all possible combinations of cell states. CA have been proposed as a novel approach for a large number of problems. Among others, CA have been proposed as models for physical, biological and social systems, games and pattern recognition [3,4]. As systems, CA have proved capable of parallel and emergent computation [5,6].

In this work, simple, bistable, one-dimensional CA are used to predict highly complex binary sequences. The considered binary sequences are derived by applying two linear threshold transformations, proposed in [7], on real-valued sequences obtained from the logistic map. In previous work [8], we considered the capability of CA given as input (initial configuration) the first L bits of a binary sequence of length $2L$, to reproduce the last L bits of the sequence in at most L evolution steps. To this end, a population of CA was evolved using a Genetic Algorithm (GA) [9,10]. The GA evolved the set of rules of the CA [11–13]. Different values of the half-length, L, reaching up to 50 were considered, and the experimental results suggested that the GA was capable of identifying a suitable set of CA rules within a small number of generations. At present, our primary focus is to investigate the extent to which this approach can be employed to perform perfect multiple-step-ahead prediction for all subsequences of a given length L in very long binary sequences.

* Corresponding author.
E-mail addresses: adam@med.duth.gr (A.V. Adamopoulos), npav@math.upatras.gr (N.G. Pavlidis), vrahatis@math.upatras.gr (M.N. Vrahatis).

0895-7177/$ – see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.mcm.2009.08.010

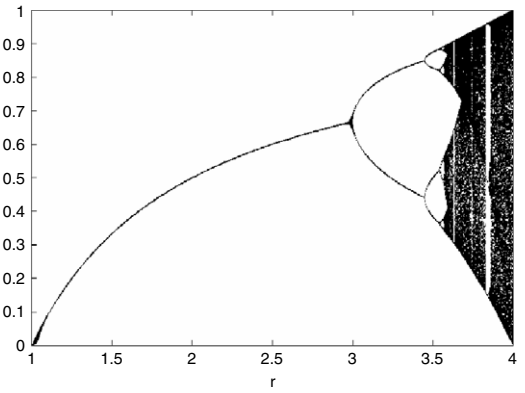

Fig. 1. Bifurcation diagram of the steady states of the logistic map with respect to r.

To the best of our knowledge this is the first attempt to utilize CA to perform prediction of binary sequences. An important advantage of CA is the fact that they are capable of generating an L-step-ahead forecast based only on the L previous values of the sequence. In other words, CA exploit substantially less information to perform this task than alternative prediction methods like artificial neural networks. Furthermore, bistable, one-dimensional CA and GAs are particularly suited to the task of binary sequence prediction. Finally, CA, as well as, the GA employed to evolve the set of rules, are easily implemented, and inherently parallel algorithms, a property that is highly desirable in computationally demanding tasks like the one presently considered. The experimental results reported suggest that the proposed approach is capable of distinguishing regions of (perfect) predictability of a highly complex system. On the other hand, no set of rules was discovered that would allow the perfect predictability of entire binary sequences, with the exception of trivial sequences.

The remaining paper is organized as follows: Section 2 provides a brief discussion on the properties of the logistic map and presents the two binary transformations employed in Section 2.1; while Section 2.2 outlines CA. Next, in Section 3 the obtained experimental results are presented and discussed. The paper ends with concluding remarks and future work in Section 4.

2. Background material and methods

2.1. Binary transformations

Complex binary sequences are generated through a two step procedure. At the first step, the logistic map:

$$x_{n+1} = r\, x_n\, (1 - x_n), \quad n = 0, 1, \ldots \tag{1}$$

is used to generate real-valued sequences. For values of r in the interval $[0, 4]$ and initial value, x_0, in the interval $[0, 1]$ the logistic map is bounded in the interval $[0, 1]$. Despite its simplicity, Eq. (1) yields a variety of dynamical characteristics, which strongly depend on the value of the parameter r [14,15]. The parameter r is an expression of the non-linearity of the system [16,17]. Specifically, for values of the non-linearity parameter r in the range $(0, 3)$ the system reaches a single-state stable value, $x_s = 1 - 1/r$. For r in the range $[3, 3.57)$ the period doubling phenomenon occurs, and the system exhibits cycling (periodical) behavior with increasing cycling period as the value of r increases. This results to a fully chaotic behavior, i.e. infinite period, for values of r in the range $[3.57, 4]$. A plot of the steady state values with respect to r is called a bifurcation diagram. The bifurcation diagram for the logistic map is provided in Fig. 1. It has been shown that the bifurcation diagram of the logistic map is a fractal object.

Having generated the chaotic data, $x_n(x_0, r)$ through Eq. (1), the binary sequence $b_n(x_0, r)$ is derived by applying the transformation [7]:

$$b_n(x_0, r) = \begin{cases} 0, & \text{if } x_n \leqslant 0.5, \\ 1, & \text{if } x_n > 0.5. \end{cases} \tag{2}$$

A second binary transformation, also proposed in [7], was applied on the raw data, Eq. (3). This is also a simple, linear, threshold and binary transformation, but it incorporates a variable threshold, which is set equal to the previous value of the raw data of the logistic map. This transformation is formulated as:

$$b_n(x_0, r) = \begin{cases} 0, & \text{if } x_n \leqslant x_{n-1}, \\ 1, & \text{if } x_n > x_{n-1}. \end{cases} \tag{3}$$

Assuming a binary sequence $b_n(x_0, r)$, consecutive L bit long subsequences are provided as input to the CA. In other words, L bits are used for the construction of the initial configuration of the CA, which corresponds to evolution step zero of the CA.

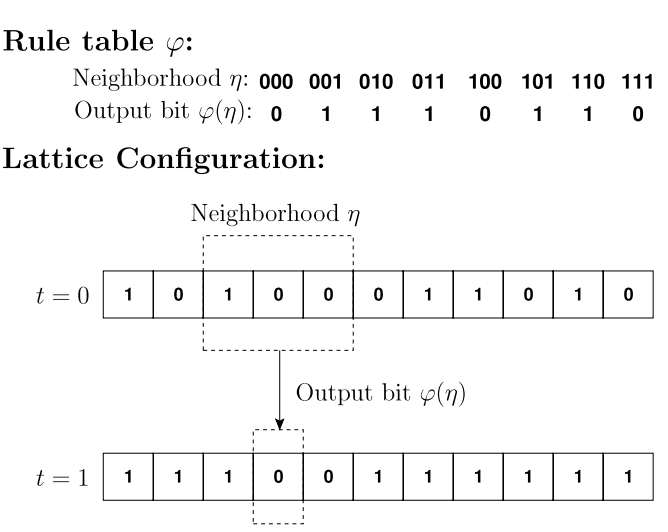

Fig. 2. The components of a one-dimensional, binary-state, $R = 1$ ("elementary") CA, iterated one time step on a configuration with $N = 11$ lattice sites and periodic boundary conditions (i.e. $s_N = s_0$).

The task is to evolve the CA, using a suitable set of rules, for L evolution steps and to investigate if it is able to regenerate the subsequent L bits of $b_n(x_0, r)$, for all the L bit subsequences in $b_n(x_0, r)$. An outline of the workings of one-dimensional CA is provided in the next subsection.

2.2. Cellular automata

A one-dimensional cellular automaton consists of a lattice of N identical finite-state machines (cells), each with an identical topology of local connections to other cells for input and output, along with boundary conditions. Let Σ denote the set of states in a cell's finite-state machine, and $k = |\Sigma|$ denote the number of states per cell. Each cell is indexed by its site number, $i = 0, 1, \ldots, N - 1$.

A cell's state at time t is denoted by s_i^t, where $s_i^t \in \Sigma$. The state, s_i^t, of cell i together with the states of the cells with which i is connected is called the *neighborhood*, η_i^t, of cell i. Each cell obeys the same transition rule $\varphi(\eta_i^t)$, that gives the updated state, $s_i^{t+1} = \varphi(\eta_i^t)$, for cell i as a function of η_i^t. We will drop the indices on s_i^t and η_i^t when we refer to them as general (local) variables. We use s^t to denote the configuration of cell states, $s^t = \{s_0^t, s_1^t, \ldots, s_{N-1}^t\}$, at time t. Thus, a CA $\{\Sigma^N, \varphi\}$ specifies a global map Φ of the configurations:

$$\Phi : \Sigma^N \to \Sigma^N, \quad \text{with } s^{t+1} = \Phi(s^t).$$

In a synchronous CA, a global clock provides an update signal for all cells: at each t all cells synchronously read the states of the cells in their neighborhood and then update their own states according to $s_i^{t+1} = \varphi(\eta_i^t)$.

The neighborhood η is often taken to be spatially symmetric. For one-dimensional CA, $\eta_i = s_{i-R}, \ldots, s_0, \ldots, s_{i+R}$, where R is the CA's radius. Therefore, $\varphi : \Sigma^{2R+1} \to \Sigma$. For small-radius, binary-state CA, in which the number of possible neighborhoods is not too large φ is often displayed as a look-up table, or rule table, that lists every possible η together with the *output bit*, s_i^{t+1}. One-dimensional CA, with $(k, R) = (2, 1)$ are employed in this work. Here, the neighborhood of each cell consists of itself and its two adjacent cells and the boundary conditions are periodic: $s_N = s_0$. An example is shown in Fig. 2. The number of rules of a specific CA depends on the order R (radius) of the cell neighborhood. This parameter determines the number $(2R + 1)$ of cells that a specific cell interacts with in a local manner. For $R = 1$, therefore, the neighborhood of each cell consists of three cells. A three-cell neighborhood, with each cell considered as a bistable element, presents $C = 2^{2R+1} = 2^3 = 8$ distinct combinations. Thus, the adaptation of a set of 8 rules is necessary. In the general case, these C combinations are ordered and numbered from 0 to $(C - 1)$ following the representation of integer numbers in the binary arithmetical system. This is shown in Table 1.

To identify rules that manage to predict satisfactorily L bit binary sequences, the sets of rules of a population of N CA are evolved using a binary encoded GA. Each set of rules is represented as the chromosome of an individual of the GA, with each gene being a binary digit. The length of the chromosome is C, as this is the number of rules that comprise the corresponding set of rules. The binary representation of the GA individuals permits the application of well-known and widely used genetic operators for selection, crossover and mutation [9]. In particular, we used *roulette-wheel selection, one-point crossover*, and *bit-flip mutation*. Recalling our goal, the fitness function for the evaluation of the individuals of the GA counts the number of bits that are successfully predicted after L evolution steps of the CA. The necessity of utilizing a GA must be emphasized. As shown in Table 2, the size, V, of the search space expands exponentially, and becomes enormous even for small values of R.

Table 1
An example of a set of rules for $R = 1$, which results to $C = 8$ distinct rules.

Rule Nr. C:	0	1	2	3	4	5	6	7
	000	001	010	011	100	101	110	111
	1	0	0	1	1	0	1	0

Table 2
Number of possible combinations of CA rules for various values of R.

Radius R	Number of neighbors $H = 2R + 1$	Number of rules $C = 2^H$	Size of search space $V = 2^C$
1	3	8	$2^8 = 256$
2	5	32	$2^{32} = 4294\,967\,296$
3	7	128	$2^{128} = 3.40282 \times 10^{38}$
4	9	512	$2^{512} = 1.34078 \times 10^{154}$

3. Experimental results

The proposed CA trained through GAs was implemented in C++ using the GNU Compiler Collection (gcc) ver. 4.0.3. The method was tested for various binary sequences generated for a large number of combinations of the parameters x_0 and r of the logistic map, Eq. (1), and for both binary transformations, Eqs. (2) and (3). The values of the non-linearity parameter, r, were selected in the range $[3.57, 4]$ for which the logistic map exhibits chaotic behavior. To avoid transient phenomena, if any, the first 10^4 iterations of the map were disregarded. To generate the real-valued sequences from the logistic map, floating point numbers with precision of at least 5000 bits were utilized using the GNU Multiple Precision Arithmetic Library (GMP) ver. 4.1.4 [18]. The total length of the binary sequences, n, was set to 10^6 in all experiments, while the parameter L ranged from 2 to 30.

To measure the fitness of a rule, all consecutive sequences of length L are provided as initial configuration to the CA. Starting from each initial configuration, the CA performs L evolution steps. The resulting configuration is compared with the L bits of the binary sequence, $b_n(x_0, r)$, immediately following the L input bits. The fitness of a rule for a particular input configuration is equal to the number of common bits between the L bits of $b_n(x_0, r)$ immediately following the input configuration, and the L bit long final configuration of the CA. The overall fitness of a rule, is equal to the mean fitness over all the input sequences that are encountered in $b_n(x_0, r)$. The number of input sequences is equal to $(n - 2L + 1)$, where $n = 10^6$ stands for the length of $b_n(x_0, r)$.

3.1. Fixed threshold binary transformation

In this subsection, we present indicative results for binary sequences, obtained by applying the transformation of Eq. (2). In Fig. 3 the distribution of bits with value '1' and '0' for values of the non-linearity parameter, $r \in [3.5, 4]$, is plotted. Evidently, for this binary transformation, an equal distribution of ones and zeros appears to be the exception rather than the rule.

Fig. 4 illustrates the proportion of the binary sequence $b_{10^6}(0.4, 3.6)$ that is perfectly predictable as the number of CA rules considered increases from one to 256. Note that rules are included in descending order of mean predictability (overall fitness), starting from the best performing rule. In case the mean predictability associated with two, or more, CA rules is equal, the rules are included according to their lexicographic ordering. Fig. 4(a) depicts the proportion of the sequence that is perfectly predictable for $L = \{2, \ldots, 10\}$, while Fig. 4(b) corresponds to longer forecasting horizons, $L = \{11, \ldots, 30\}$. As expected, the chaotic nature of the logistic map and the corresponding complexity of the derived binary sequences, reduces the proportion of the sequence that is perfectly predictable as the forecasting horizon increases. This finding is common to all the binary sequences considered. As illustrated, however, the decrease with respect to L is not monotonic. The maximum proportion of the sequence that is predictable when all 256 possible rules are included is 0.7427 and corresponds to $L = 6$. This proportion drops below 0.1 for $L = 13$, and it reaches 0.0034 for $L = 30$. Table 3 reports the mean forecasting performance (μ) of the best performing CA rule for different horizons.

Finally, Fig. 5 provides a visualization of the regions of the phase space of the logistic map with $r = 3.6$ that are perfectly predictable by all the CA rules, when the real-valued sequence undergoes the transformation of Eq. (2). Fig. 5(a) corresponds to forecasting the next six bits ($L = 6$) while in Fig. 5(b) $L = 10$. It is evident from the figure that increasing the forecasting horizon inhibits predictability. A closer inspection of Fig. 5 also reveals that although increasing L from six to ten reduces overall performance (as clearly shown in Fig. 4), the regions of the phase space that are perfectly predictable, become less clearly separable from those that are not. This finding becomes more pronounced in the binary sequences considered next.

Increasing the value of the non-linearity parameter, r, to 3.9, reduces overall predictability, as illustrated in Fig. 6. Fig. 6(a) reports the obtained results for $L = \{2, \ldots, 10\}$, while Fig. 6(b) illustrates the results for $L = \{11, \ldots, 30\}$. The maximum proportion of the series that is predictable when all 256 possible rules are included is 0.7016 and corresponds to $L = 3$. This proportion falls below 0.1 for $L = 10$, and it reaches 10^{-5} for $L = 30$. Table 4 presents the mean predictive ability, μ, of the best performing CA rule for different sequence lengths. A visualization of the regions of the phase space of the logistic map

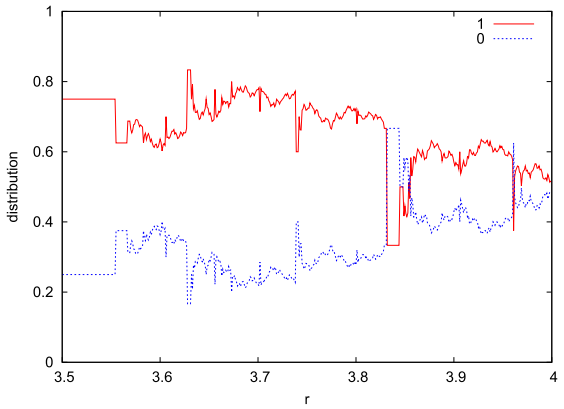

Fig. 3. Distribution of ones (solid red) and zeros (dashed blue) for binary sequences obtained through Eq. (2) for $r \in [3.5, 4]$ with stepsize 10^{-3}. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

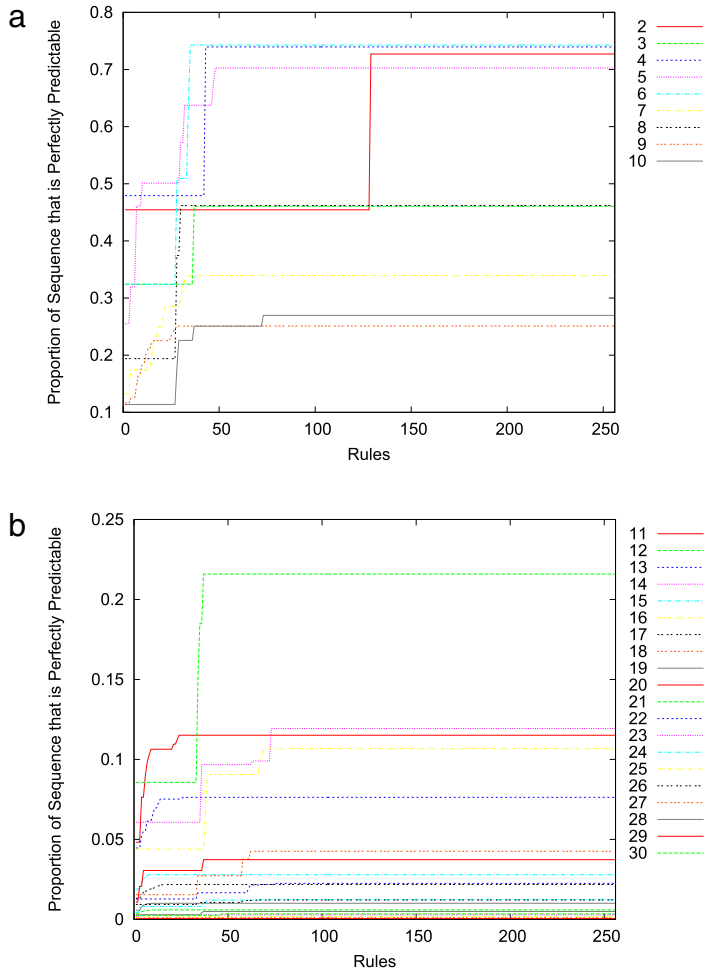

Fig. 4. Proportion of the sequence $b_{10^6}(0.4, 3.6)$, obtained through the transformation of Eq. (2) that is perfectly predictable as additional CA rules are included. (CA rules are added in descending order of fitness): (a) $L = \{1, \ldots, 10\}$, (b) $L = \{11, \ldots, 30\}$. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

for $r = 3.9$ that are perfectly predictable when the original sequence is subjected to the transformation of Eq. (2) is provided in Fig. 7. In this figure the previously discussed phenomenon that wants increasing the forecasting horizon to render the

Table 3
Mean performance of the best CA rule for $b_n(0.4, 3.6)$ generated through Eq. (2) with respect to the prediction horizon L.

L	2	3	4	5	6	7	8	9
μ	1.45434	2.18772	3.47918	4.05752	5.18151	5.69623	6.90867	7.49329
L	10	11	12	13	14	15	16	17
μ	8.63584	9.1991	10.363	10.9815	12.0902	12.70221	13.8174	14.4361
L	18	19	20	21	22	23	24	25
μ	15.5445	16.1412	17.2717	17.8601	18.9989	19.5872	20.726	21.3124
L	26	27	28	29	30			
μ	22.4532	23.058	24.1804	24.773	25.9075			

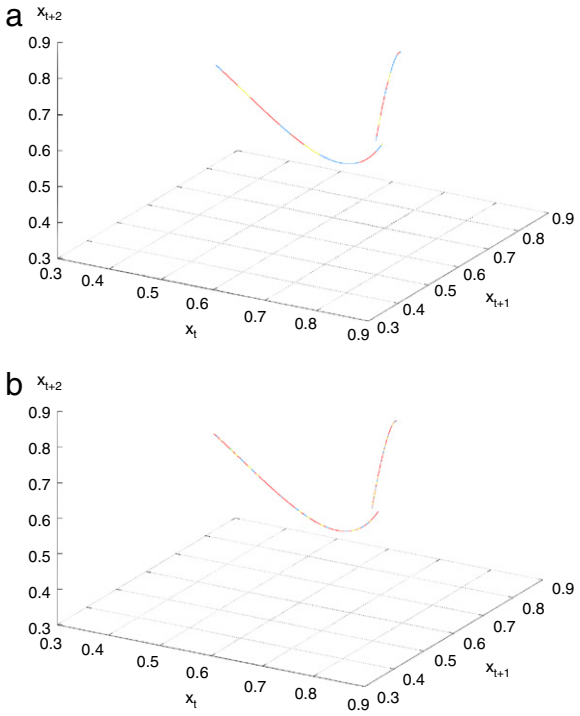

Fig. 5. Regions of perfect predictability on the phase diagram of the logistic map with $r = 3.6$. (a) $L = 6$ blue: best performing rule; blue and yellow: top 30 best performing rules; non-red: all 256 rules. (b) $L = 10$ blue: best performing rule; blue and yellow: top 50 best performing rules; non-red: all 256 rules. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 4
Mean performance of the best CA rule for $b_n(0.4, 3.9)$ generated through Eq. (2) with respect to the prediction horizon L.

L	2	3	4	5	6	7	8	9
μ	1.13995	1.91696	2.32276	2.8808	3.46368	3.98985	4.61595	5.12981
L	10	11	12	13	14	15	16	17
μ	5.69979	6.26977	6.83976	7.40974	7.97972	8.54971	9.11969	9.68968
L	18	19	20	21	22	23	24	25
μ	10.2597	10.8296	11.3996	11.9696	12.5396	13.1096	13.6796	14.2495
L	26	27	28	29	30			
μ	14.8195	15.3895	15.9595	16.5295	17.0994			

regions of the phase space that are perfectly predictable less clearly separable from those that are not (despite the fact that overall predictability deteriorates) is more visible.

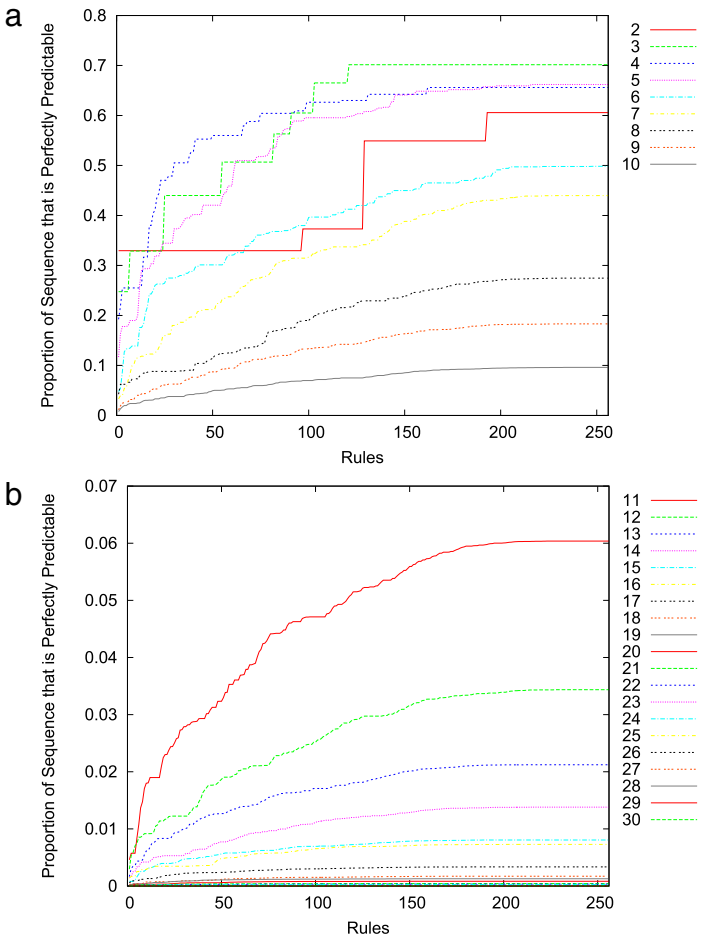

Fig. 6. Proportion of the sequence $b_n(0.4, 3.9)$, obtained through the transformation of Eq. (2) that is perfectly predictable as additional CA rules are included. (CA rules are added in descending order of fitness): (a) $L = \{2, \ldots, 10\}$, (b) $L = \{11, \ldots, 30\}$. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

3.2. Variable threshold binary transformation

In Fig. 8 the distribution of bits with value '1' and '0' for binary sequences obtained through Eq. (3), and values of the non-linearity parameter, $r \in [3.5, 4]$, is plotted. In contrast to the distribution induced by the transformation of Eq. (2), this transformation produces an equal distribution of ones and zeros until r reaches approximately 3.7. As r increases beyond 3.7, the two proportions tend to diverge. This phenomenon has direct implications for the predictability of the binary sequences obtained through this transformation. For example, for $r = 3.6$ a bit with value '1' always follows a bit with value '0', and vice versa. This renders the entire binary sequence perfectly predictable by a large number of rules and to any forecasting horizon, L. Therefore, only results for the case of $r = 4.0$ are presented.

The proportion of the binary sequence $b_{10^6}(0.4, 4.0)$ that is perfectly predictable with respect to the number of CA rules considered is depicted in Fig. 9. As in the previous cases, rules are incorporated in descending order of mean predictability. Furthermore, Fig. 9(a) shows the obtained results for forecasting horizons $L = \{2, \ldots, 10\}$, while Fig. 9(b) illustrates the results for longer forecasting horizons, $L = \{11, \ldots, 30\}$. The mean predictive capability, μ, of the best performing CA rule, for different values of L is reported in Table 5. The regions of the phase space of the logistic map that are perfectly predictable after the raw data undergo the transformation of Eq. (3) are depicted in Fig. 10. Fig. 10 shows that increasing L causes the regions of perfect predictability to become less clearly separable.

4. Concluding remarks

In this work a hybrid evolutionary algorithm for the prediction of highly complex binary sequences is presented. The algorithm incorporates Cellular automata with sets of rules that are suitably codified in order to be evolved by a simple, binary encoded, Genetic Algorithm. A cellular automaton receives as input an L bit long binary pattern and produces a prediction for the values of the subsequent L bits in L evolution steps. To this extent, cellular automata are capable

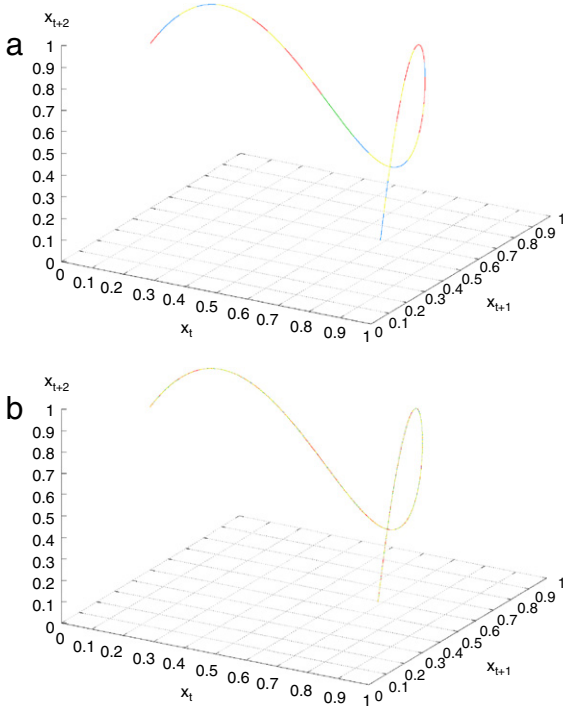

Fig. 7. Regions of perfect predictability on the phase diagram of the logistic map with $r = 3.9$. (a) $L = 3$ blue: best performing rule; blue and yellow: top 100 best performing rules; non-red: all 256 rules. (b) $L = 10$ blue: best performing rule; blue and yellow: top 100 best performing rules; non-red: all 256 rules. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

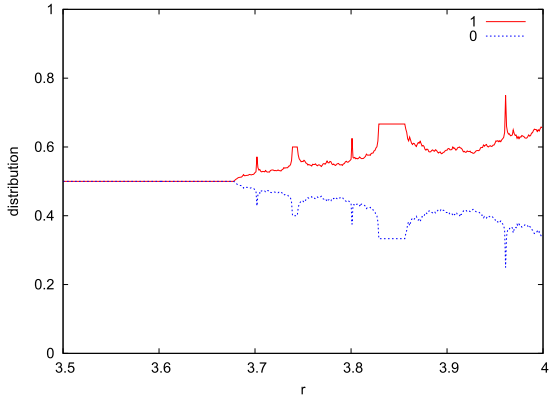

Fig. 8. Distribution of ones (solid red) and zeros (dashed blue) for binary sequences obtained through Eq. (3) for $r \in [3.5, 4]$ with stepsize 10^{-3}. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 5
Mean performance of the best CA rule for $b_n(0.4, 4.0)$ generated through Eq. (3) with respect to the prediction horizon L.

L	2	3	4	5	6	7	8	9
μ	1.33407	2.0011	2.66814	3.33517	4.0022	4.66923	5.33627	6.0033
L	10	11	12	13	14	15	16	17
μ	6.67033	7.33736	8.0044	8.67143	9.33846	10.0055	10.6725	11.3396
L	18	19	20	21	22	23	24	25
μ	12.0066	12.6736	13.3407	14.0077	14.6747	15.3418	16.0088	16.6758
L	26	27	28	29	30			
μ	17.3429	18.0099	18.6769	19.344	20.011			

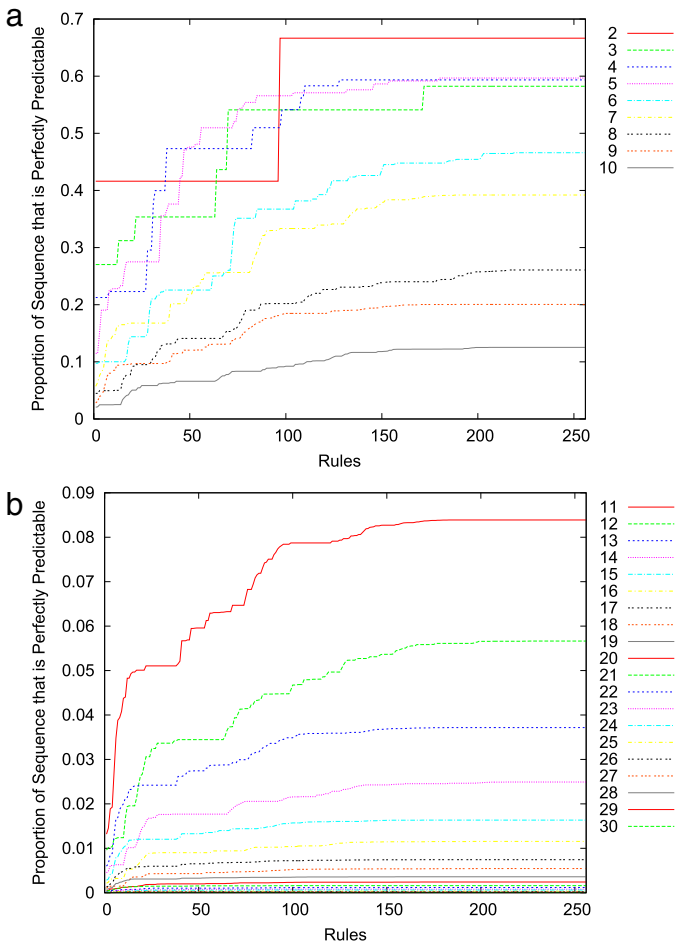

Fig. 9. Proportion of the sequence $b_n(0.4, 4.0)$, obtained through the transformation of Eq. (3) that is perfectly predictable as additional CA rules are included. (CA rules are added in descending order of fitness): (a) $L = \{2, \ldots, 10\}$, (b) $L = \{11, \ldots, 30\}$. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

of generating multiple-step-ahead predictions based on a very limited amount of information compared to alternative methodologies.

To evaluate the algorithm we employed sequences derived from two simple, linear, threshold, transformations of real-valued sequences generated by the logistic map. The logistic map is a well-known system that through period doubling reaches chaotic behavior. The dynamical behavior of this map is heavily influenced by the value of the non-linearity parameter r in Eq. (1). Our findings suggest that the same is true for the predictability of the derived binary sequences. The reported experimental results indicate that the performance of the algorithm is particularly robust with respect to the length of the forecasting horizon. Although no one-dimensional CA with radius one is capable of predicting accurately an entire binary sequence (irrespective of forecasting horizon), the proposed algorithm can detect regions of the phase space that are predictable. An interesting finding verified in all the considered sequences is that although expanding the forecasting horizon reduces predictability, it also blurs the distinction between regions of perfect and imperfect predictability.

In future work we intend to investigate the robustness of the proposed algorithm to the presence of noise in the original data generating process. Robust performance in the presence of noise is particularly important as most real-world time series are contaminated with noise. This method will also be applied to real-world data. The second binary transformation is particularly meaningful in financial time series applications, as it represents the direction of change of a series, a piece of information that is vital in decision making. Finally, we intend to perform a thorough comparative analysis of the algorithm against alternative approaches.

Acknowledgments

This work was partially supported by the Hellenic Ministry of Education and the European Union under research Program PYTHAGORAS-89203.

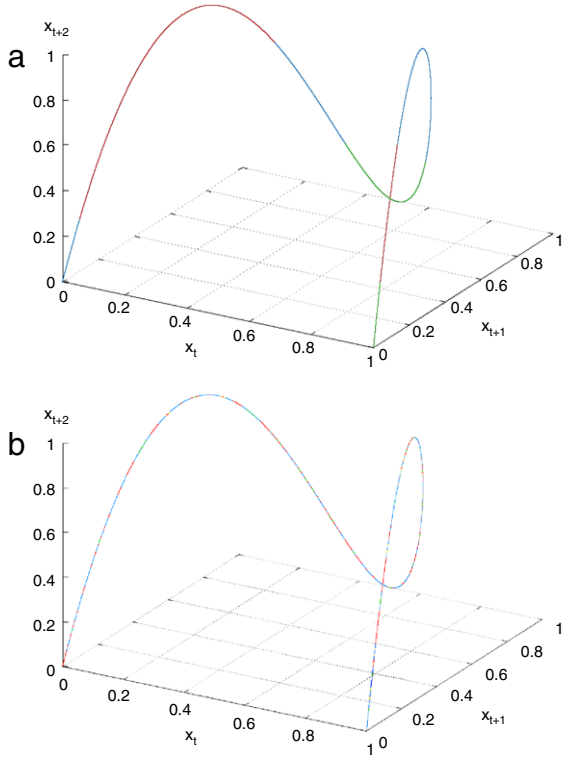

Fig. 10. Regions of perfect predictability on the phase diagram of the logistic map with $r = 4.0$. (a) $L = 2$ blue: best performing rule; non-red: all 256 rules. (b) $L = 5$ blue: top 50 best performing rules; blue and green: top 100 best performing rules; non-red: all 256 rules. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

References

[1] S. Wolfram, Theory and Applications of Cellular Automata, World Scientific, 1986.
[2] S. Wolfram, Cellular automata as models of complexity, Nature 311 (1984) 419–424.
[3] S. Wolfram, Cellular Automata and Complexity, World Scientific, Singapore, 1994.
[4] N. Ganguly, B.K. Sikdar, A. Deutsch, G. Canright, P.P. Chaudhuri, A survey on cellular automata, Tech. Rep., Centre for High Performance Computing, Dresden University of Technology, 2003.
[5] M. Mitchell, Computation in cellular automata: A selected review, in: T. Gramss, S. Bornholdt, M. Gross, M. Mitchell, T. Pellizzari (Eds.), Nonstandard Computation, Wiley-VCH, Weinheim, 1998, pp. 95–140.
[6] J. Crutchfield, M. Mitchell, The evolution of emergent computation, in: Proceedings of the National Academy of Sciences, Vol. 92, USA, 1995, pp. 10742–10746.
[7] N.G. Packard, A genetic learning algorithm for the analysis of complex data, Complex Systems 4 (5) (1990) 543–572.
[8] A.V. Adamopoulos, N.G. Pavlidis, M.N. Vrahatis, Genetic algorithm evolution of cellular automata rules for complex binary sequence prediction, in: T. Simos (Ed.), Proceedings of the International Conference of Computational Methods in Sciences and Engineering, ICCMSE 2005, in: Lecture Series on Computer and Computational Sciences, vol. 4, 2005, pp. 1424–1427.
[9] M. Mitchell, Introduction to Genetic Algorithms, MIT Press, 1996.
[10] J.R. Koza, Genetic Programming: On the Programming of Computers by Means of Natural Selection, MIT Press, 1992.
[11] M. Mitchell, J. Crutchfield, R. Das, Evolving cellular automata with genetic algorithms: A review of recent work, in: Proceedings of the First International Conference on Evolutionary Computation and its Applications, 1996.
[12] M. Mitchell, J.P. Crutchfield, P.T. Hraber, Evolving cellular automata to perform computations: Mechanisms and impediments, Physica D 75 (1994) 361–391.
[13] R. Das, J.P. Crutchfield, M. Mitchell, J.E. Hanson, Evolving globally synchronized cellular automata, in: L. Eshelman (Ed.), Proceedings of the Sixth International Conference on Genetic Algorithms, Morgan Kaufmann, 1995, pp. 336–343.
[14] M. Barnsley, Fractals Everywhere, Academic Press, New York, 1988.
[15] A.V. Adamopoulos, S. Likothanassis, N.G. Pavlidis, M.N. Vrahatis, Short-term prediction of complex binary data, in: T.E. Simos, G. Psihoyios, C. Tsitouras (Eds.), International Conference of Numerical Analysis and Applied Mathematics, ICNAAM 2005, Wiley-VCH Verlag GmbH & Co, 2005, pp. 872–875.
[16] J.L. Casti, Searching for Certainty, Scribners, 1992.
[17] T.P. Meyer, F.C. Richards, N.H. Packard, Learning algorithm for modeling complex spatial dynamics, Physical Review Letters 63 (16) (1989) 1735–1738.
[18] Free Software Foundation Inc., GNU Multiple Precision Arithmetic Library ver. 4.1.4. URL: http://swox.com/gmp/.