Life-Like Network Automata Descriptor based on Binary Patterns for Network Classification

Lucas C. Ribas*, Jeaneth Machicao+, Odemir M. Bruno+**

(*) Institute of Mathematical and Computer Sciences (ICMC), University of São Paulo (USP), P.O. Box 668, São Carlos, SP P.C. 14560-970, Brazil

(+) Scientific Computing Group. São Carlos Institute of Physics, University of São Paulo, São Carlos - SP, PO Box 369, 13560-970, Brazil.

Abstract

We propose a descriptor based on binary patterns extracted from network-automata time-evolution patterns (TEP) aiming to characterize networks. More, in particular, we explore TEPs descriptors from the Life-Like Network Automata (LLNA), a cellular automaton inspired by the rules of the "Life-Like" family that uses a network as tessellation, and based on its dynamics to extract features for network characterization. In recent work, the LLNA has been introduced as a pattern recognition tool that uses a descriptor based on the histograms of complexity measures such as the entropy, word length, and Lempel-Ziv complexity. However, these descriptors correspond to continuous values, and consequently, their histograms lack of an optimal number of bins, which therefore turns out to be a parametric issue. To overcome this disadvantage, we propose a new descriptor that computes feature vectors formed by discrete binary patterns histograms with different lengths D. Furthermore, we show a statistical improvement of the proposed method compared to earlier approaches such as the original LLNA and classical network structural measurements. Our experimental results show the performance improvement of the proposed method in six synthetic network databases and eight real network databases.

1. Introduction

Currently, with the introduction of data science and the *big data* era, there is a high demand for pattern recognition methods that handle non-linear data. Network science is being increasingly used because of its flexibility and ability to represent and analyze any discrete system such as metabolic networks [17, 14], protein-protein interaction networks [32, 40], social networks [11, 30], ecological networks [15], scientific collaboration networks [29], brain networks [5], etc.

*Corresponding author

Email address: bruno@ifsc.usp.br (Odemir M. Bruno+*)

In this context, networks science and pattern recognition emerge as an important alternative in this scenario. Thus, pattern recognition in networks aims to classify large-scale networks into several classes and distinguishing them according to their intrinsic characteristics [3, 28], instead of focusing on the topology properties of an isolated network, it is interested in the classification, clustering, and comparison between different networks, in which the network is explored as a whole [46]. In recent years the concept of networks has been applied to different pattern recognition problems with promising results, such as computer vision [35, 37, 33, 27, 34], authorship attribution [21, 26, 1], phylogenetic reconstruction [3, 22], among others.

Recently, networks and cellular automata arise as an important combination for data science and pattern recognition. Cellular automata are well-known for its capability to produce complex patterns from even simple rules. These emergent patterns may range from simpler such as homogeneous, stables or periodic structures, to chaotic patterns, and even complex structures [44]. The spatiotemporal patterns formed by cellular automata (CA) dynamics are still marveling a huge number of researchers from diverse fields. For instances, ecologists and biologists have studied the formation of the pigmentation pattern of animals, such as the shell of the species *conus textile* [43], as well as the formation of tumors growth [31], epidemic propagation [38], plant population dynamics [9], colonization processes [2]; Physicists have studied CAs to model crystals growth, such as snowflakes [48] and grain growth [50]. Similarly, chemists have relied on CAs to study chemical reaction diffusions [6]. CAs patterns have been used even in more striking fields, such as cryptography [23, 20], and of course, by computer scientists, e.g. either using image processing tools to classify CAs rules [10, 24] or to propose CAs as a tool for image processing [36].

Although CAs are typically studied in regular tessellations, they can also be explored in irregular structures such as graphs or networks. Recently, the incorporation of a CA dynamics over the network topology, also named as network-automata, has started to gain more attention [49, 45, 25, 39, 28]. Network-automata is a generalization of CA, in which a network (tessellation) is the habitat where a CA simulates artificial life, the set of vertices represent the cells, the edges represent the neighborhood, and a local rule governs their cell states. The dynamic evolution of a network-automaton provides a time-evolution pattern (TEP), which can be visualized by compiling their states (from top to bottom) while it evolves in time, from which intrinsic network properties can be extracted in order to be used in a pattern recognition context.

In a paper published in 2016, Miranda et al. [28] proposed a family of network-automata called as the Life-Like Network Automata (LLNA) (Fig. 1a-c), which has been introduced as a tool for network analysis aimed for pattern recognition applications. In this method, the TEPs patterns formed from the dynamical behavior of this non-linear system are used as features (signatures). Since the LLNA is a CA whose states are binaries, therefore the extracted TEPs are represented as chains of zeros and ones, representing their cell's states (live or dead), e.g. 010100100010100111001. Miranda et al. [28] used a set of three measurements: the Shannon entropy, the word length, and the Lempel-Ziv com-

plexity to calculate histogram distributions in order to obtain the corresponding feature vectors, which are able to characterize the network topology.

Although, the dynamic evolution signature provided by the histograms of the former measurements are well suited for pattern recognition tasks [28], there is a lack of the method to choose the number of bins as a parameter for the construction of the histograms. Since these measurements are composed of continuous values, then there is a need to define an optimal number of bins which will influence in the classification performance and it is necessary to be obtained for any new classification task. For that reason, instead of putting more efforts to find an optimal number of bins, which represents a difficult and expensive task, a more robust manner to extract measurements independently of this parameter issue is needed, which in consequence may lead to over exploit the advantages from these rich patterns aiming to improve the classification performance.

In this paper, we propose an improved manner to extract feature vectors from LLNA time-evolution patterns (see Fig. 1) by means of binary pattern dictionaries, hereinafter called as LLNA-BP. A dictionary β_D is a discrete set $\beta \in [0, 2^D - 1]$ containing all possible combinations of zeros and ones of length D. In order to fill the previous work lack, here we propose to account the frequency of these patterns within a TEP, as can be observed in Fig. 1c, from which a frequency histogram of the discrete set of binary patterns is calculated (see Fig. 1e). In this work, we demonstrate how these measurements improve the results obtained in the earlier work [28].

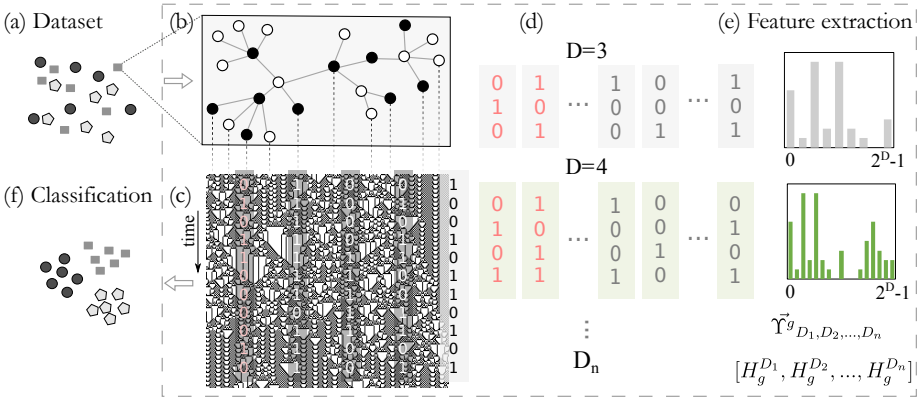

Figure 1: Proposed pattern recognition scheme based on binary patterns extracted from LLNA time-evolution patterns (LLNA-BP). (a) Given a network database problem. (b) For each network, c) the Life-Like network automaton is evolved with a previously selected rule for a given network initialized with random states represented by white (alive or one) and black (dead or zero). From top to bottom, the corresponding time-evolution pattern produced by the LLNA dynamics of the given network. (d) Sequences of zeros and ones are concatenated to produce a binary pattern of different sizes, e.g. $D = 3, D = 4, \ldots D_n$, from which a unique histogram is obtained, respectively, and (e) thus forming the feature vector by using various strategies, e.g. concatenating the global frequency distributions $\vec{\Upsilon}^{g}_{D_1,D_2,...,D_n}$, which are then used for (f) the networks classification task.

We evaluated the performance of the proposed approach using eight different applications concerning pattern recognition in networks. Thus, we used two types of databases, synthetics, and real-world networks. The first database of synthetic networks contains four different models: random (Erdös & Rényi) [13], small-world (Watts & Strogatz [41]), scale-free (Barabási & Albert [4]), and geographical networks (Waxman [42]). The second and third database were obtained based on the latter one: (i) focusing on the dependency of different k degrees and (ii) a noise-perturbed database, where edges were randomly added and removed from the network, respectively. The fourth database contains exclusively networks belonging to the scale-free model with both linear and non-linear preferential attachment and different parameters. Regarding the real-world applications, a fifth database was explored containing two online social networks (Twitter and Google+) [19]. Besides that, we also constructed seven metabolic networks databases based on the biochemical reactions of several species obtained from the Kyoto Encyclopedia of Genes and Genomes database (KEGG) [18]. Therefore, we also applied our method in various classification tasks aiming to distinguish metabolic networks from different biological origins such as the *eukariota* and *bacteria* kingdom, the *animal* domain, *fungi* domain, *plant* domain, *protist* domain, *Firmicutes-Bacillis* phylum and *Actinobacteria* phylum. According to the results, our approach outperformed the results yielded by the LLNA [28] and classical structural measurements of different types such as degree-based, clustering coefficient, paths and degree-correlation [8, 7].

In the remainder of this manuscript, Section 2 presents a detailed description of the LLNA as well as its mathematical definitions. In Section 3, we present a study regarding the histogram formation for different measurements. In Section 4, the configurations of the experiments are detailed. In Section 5, we evaluated the proposed approach regarding the classification task for eight distinct databases with discussions about the results, and, finally, conclusions are presented in Section 6.

2. Background

2.1. Life-Like network automaton

CAs were inspired by the concept of "artificial life", in which its components (cells) interact with each other and their environment, simulating life or death of their cells. Their states are modified over time according to a local rule, depending on the previous states of their neighborhood and its cell state itself, from which rich patterns can emerge.

Formally, a network-automaton $\mathcal{R}$ can be defined by the tuple

$$\mathcal{R} = \langle \mathcal{T}, S, s_0, \phi \rangle \, .$$

The tessellation $\mathcal{T}$ of a CA is represented by the network topology, so each vertex will be represented by a cell c_i which is connected with its k_i neighbors. S is the set of states, which in our case is restricted to binary states $s_i = 1$ (alive) and $s_i = 0$ (dead). The function $s(c_i, t)$ indicates the state of a cell c_i

at time t, thus s_0 represents the initial configuration of all the cells within the automaton. The transition function ϕ is the rule governing the dynamics of the network-automaton, which defines how the cell states are updated [28].

The number of neighbors of each cell is restricted to the neighborhood of each vertex of the network. Therefore, the transition function ϕ is given as a function of the neighborhood density $\sigma(c_i, t)$, defined as the proportion of alive neighbors at time t.

Regarding the transition function $\phi : s(c_i, t) \to s(c_i, t+1)$, it can be implemented in different manners. The LLNA was presented in a more generalized form of density function which can be interval, for example, $22\% \leq \sigma(v_i, t) < 33\%$. The neighborhood density function $\sigma(v_i, t)$ of the vertex v_i is the proportion of alive neighbors, described by Equation (1)

$$\sigma(v_i, t) = \frac{1}{k_i} \sum_{j=1}^{N} A_{ij} s(v_j, t)\,, \tag{1}$$

in which A_{ij} is the adjacency matrix of the network, N the number of vertices and k_i is the number of neighbors of vertex v_i defined by Equation (2)

$$k_i = \sum_{j=1}^{N} A_{ij}\,. \tag{2}$$

As the LLNA traces a correspondence between the transition functions and the rules of the 2D CA Life-Like family, therefore the LLNA rules are also described by the notation Bx-Sy, in the form B$x_0x_1 \ldots x_8$-S$y_0y_1 \ldots y_8$, where B and S represent the "born" and "survive" conditions, respectively; and, x_x and y_y stands for the combination of digits ranging from 0 to 8, such as B23-S3, B1357-S2468, B0345-S36, B25-S8, etc. Considering the outer-totalistic 2D CA version, where each cell can have a maximum of 9 neighbors, including itself (Moore neighborhood $r = 9$). Therefore, the LLNA has also $2^{9+9} = 262,144$ transition rules due to the combination of the conditions of birth (B) and survival (S).

The LLNA transition function is defined as:

$$s(c_i, t+1) = \begin{cases} 1, & \text{if } s(c_i,t) = 0 \text{ and } x_x/r \leq \sigma(c_i,t) < (x_x+1)/r \Rightarrow \text{born (B)} \\ 1, & \text{if } s(c_i,t) = 1 \text{ and } y_y/r \leq \sigma(c_i,t) < (y_y+1)/r \Rightarrow \text{survive (S)} \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

For the sake of illustration, we show in Fig. 2 the LLNA dynamics of a given network containing $N = 21$ vertices with rule B25-S8 evolved during three-time steps. In the upper left of Fig. 2a shows the initial configuration s_0 of the network. In Fig. 2b is detailed how the cells c_i and c_j (shaded region) are evolved from $t = 0$ to $t = 1$. First, considering the dead state of cell c_i, i.e. $s(c_i, t) = 0$, then the two birth conditions from the given rule B25 (B2 or B5) can be applied in Equation (3). Since c_i has 5 neighbors, from which 3 are alive

and 2 dead, then it corresponds to a neighborhood density $\sigma(c_i, t) = \frac{3}{5} = 60\%$ which satisfies the second birth condition $\frac{5}{9} \leq \sigma(c_i, t) < \frac{6}{9}$, therefore in the following state it will born $s(c_i, t+1) = 1$. On the other hand, cell c_j, with state $s(c_j, t) = 1$, enables the survival condition of the given rule S8, thus we check the condition $\frac{8}{9} \leq \sigma(c_j, t) \leq 1$, however as $\sigma(c_j, t) = \frac{3}{4} = 75\%$, then it does not satisfies the survival condition, and therefore in the following state this cell will die $s(c_j, t+1) = 0$. In a similar manner, all the states of the automaton are also calculated synchronously. In Fig. 2c, we can observe the corresponding time-evolution pattern generated by the automaton, where the evolution of each cell is stacked from top to bottom. It should be noticed that, for the sake of visualization, the vertices of the network were placed from left to right according to a meaningless criterion, for example, labels of vertices, ordering of their degree connectivity, among others.

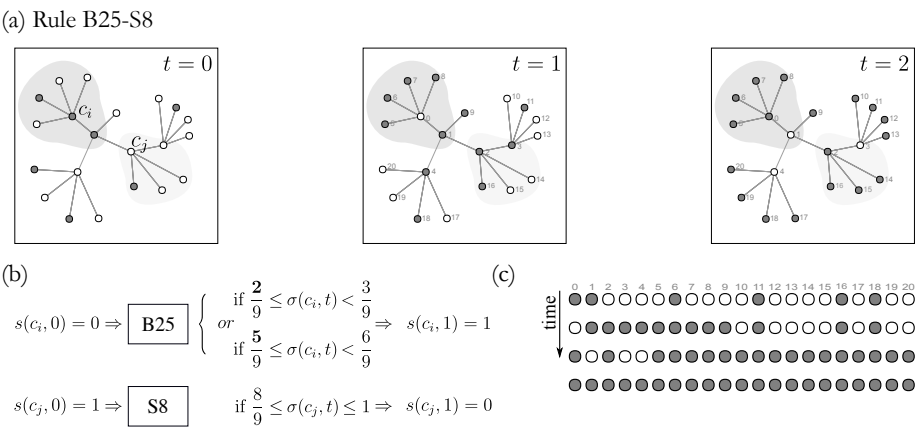

Figure 2: Construction of a Life-Like network automaton using rule B25-S8 for a network with $N = 21$ vertices evolved during three iterations. (a) From top to bottom, the LLNA is evolved, where the states are represented by white (alive) and black (dead). (b) A detailed step-by-step of the transition function B25-S8 for two given cells c_i and c_j satisfying the birth and survival conditions. (c) The time-evolution pattern obtained by the given network, which is formed by stacking from top to bottom the states of all the cells as the automaton evolves.

2.2. LLNA rule selection

Miranda et al. [28] discussed the choice of the best LLNA rules for a specific classification problem. They proposed a rule selection procedure consisting of the search for a set of rules (solutions) that maximizes the accuracy rates for an underlying problem. This is an important process since the same rule can present different behaviors for networks of different classes, so it is not trivial to predict what would be the best rule for a specific problem. In that work, it was also emphasized that, since the rule selection is done through an optimization procedure, it must be assumed that the set of solutions (selected rules) also may bring some rules that do not comply the expectation when presenting a new database. For that reason, each of the 262,144 rules of the Life-Like family

must be evaluated. That is, a network-automaton is built for each rule, then a pre-classification process is made, i.e. feature vectors are extracted and then it is obtained a pre-classification rate. Once a set of best rules is selected then the LLNA can be used for the main classification tasks.

Furthermore, we emphasize that the choice of the best rules should be performed on a representative subset of the network database, separated exclusively for the rule selection, in order to obtain unbiased results, while the rest of networks dataset would be used to evaluate the performance of the classifiers. In this manner, this procedure ensures that an unseen database is used for the rule selection and for the validation of the method aiming to unbiased the cross-validation scheme for the training and testing steps.

2.3. LLNA descriptor: previous work

Once a TEP is obtained from the LLNA dynamics (see Fig. 1c and Fig. 2c), then each vertex of the network can be analyzed as a sequence of ones and zeros, from which several characteristics can be measured. Miranda et al. [28] proposed the histogram distribution combination of three measures: Shannon's entropy ($\vec{\mu}_S$), word length ($\vec{\mu}_W$), and Lempel-Ziv complexity ($\vec{\mu}_L$) as feature vectors. More specifically, the Shannon entropy of a binary sequence is defined as $\mu_{Si} = -(p_i^{(0)} \log_2 p_i^{(0)} + p_i^{(1)} \log_2 p_i^{(1)})$, where $p_i^{(0)}$ e $p_i^{(1)}$ is the probability of having zeros and ones, respectively. The word length describes the homogeneity of the sequence with respect to the length of the words (subsequence of ones limited by zeros). Finally, the Lempel-Ziv complexity is a measure based on the count of all possible combinations of sequences of ones and zeros (blocks) contained in the sequence and it is calculated as $\mu_L = g \log l/l$, where l is the length of the sequence and g the number of blocks found within the sequence. However, Miranda et al. approach [28] account for the combination of the distribution histograms of these measurements, which leads to the lack of an optimal number of bins.

3. Binary pattern descriptor (LLNA-BP)

In this section, we describe the proposed approach to extract information from the time-evolution patterns generated by the LLNA based on binary patterns histograms. Basically, our method can be divided into two parts: (i) to extract binary patterns from TEPs and (ii) to build the binary pattern's signature. The next sections describe these steps.

3.1. Binary patterns from TEP

In this proposed approach, we use binary patterns dictionaries of different lengths D in order to compound a feature vector (signature) extracted from the LLNA time-evolution patterns. A dictionary β_D is a discrete set $\beta \in [0, 2^D - 1]$ containing all possible combinations of zeros and ones of length D, i.e. β_D is the alphabet containing binary patterns such as $\beta_D = \{00,\ 01,\ 10,\ 11\}$ for $D = 2$, $\beta_D = \{000,\ 001,\ 010,\ 011,\ 100,\ 101,\ 110,\ 111\}$ for $D = 3$, and so on. In this

manner, a binary pattern is a statistical descriptor computed by considering the states of a cell (vertex) through time.

First, given a cell c_i (network vertex) at time t, we build a binary pattern containing D bits. This binary pattern is composed by the states of a cell c_i, starting in $s(c_i, t)$ until $s(c_i, t+(D-1))$. For a better representation, we coding this binary patterns in a decimal format $\Phi_D(c_i, t)$, as follows:

$$\Phi_D(c_i, t) = \sum_{j=t}^{(D-1)+t} s(c_i, j) 2^{(j-t)}. \tag{4}$$

Then, we produce a set of binary patterns for each cell c_i via a sliding window, resulting in the following set $\{\Phi_D(c_i, t), \Phi_D(c_i, t+1), \ldots, \Phi_D(c_i, t+(D-1))\}$. In this way, considering a maximum time T and a dictionary size D, for each vertex (cell) from the network it is obtained $(T - D + 1)$ binary patterns, as illustrated in Fig. 1(d).

In this work, two manners to compound the feature vectors using binary patterns dictionary were explored. The first one is called as the global histogram H_g^D and the second one as the degree histogram H_k^D, which are detailed hereinafter.

3.1.1. Global Histogram

Since the number of possible binary patterns is equal to 2^D, from all binary patterns generated for a network, a global histogram H_g^D is computed to characterize it, as can be seen in Fig. 1(e). This global histogram computes the occurrence of the binary patterns in the TEPs of all network, according to

$$H_g^D(p) = \sum_{i=1}^{N} \sum_{t=0}^{(T-D+1)} \delta(\Phi_D(i, t), p)\,, \tag{5}$$

where $p \in [0, 2^D - 1]$ and $\delta_k(x, y, k_i) = \begin{cases} 1, x = y\,, \\ 0, \text{otherwise}\,. \end{cases}$

In Equation 5, observe that p is the decimal representation of a binary pattern output by the proposed approach. It should be noticed that this histogram $H_g^D(p)$ is then computed as a probability density function, i.e. divided by the total number of patterns, in order to turn the histogram invariant to the network size, as follows:

$$H_g^D(p) = \frac{H_g^D(p)}{\sum_{i=0}^{D-1} H_g^D(i)}. \tag{6}$$

3.1.2. Degree Histogram

Regarding the degree k values, we can also consider histograms for the vertices with the same degree k. This histogram aims to obtain information of the

occurrence of binary patterns in vertices that are similar in the network. In this way, a histogram H_k^D is computed for each degree k of the network. These histograms can be defined as follows:

$$H_k^D(p) = \sum_{i=1}^{N} \sum_{t=0}^{(T-D+1)} \delta_k(\Phi_D(i,t), p, k_i)\,, \tag{7}$$

where $p \in [0, 2^D - 1]$ and $\delta_k(x, y, k_i) = \begin{cases} 1, x = y \text{ and } k_i = k\,, \\ 0, \text{otherwise}\,. \end{cases}$

Similarly to the global histogram, the degree histogram $H_k^D(p)$ is computed as a probability density function, dividing by the total number of patterns of each degree k, according to

$$H_k^D(p) = \frac{H_k^D(p)}{\sum_{i=0}^{D-1} H_k^D(i)}. \tag{8}$$

Fig. 3(a) shows the corresponding TEPs obtained for random, small-world, scale-free and geographical networks with $N = 500$ vertices. These TEPs were generated using the LLNA rule B135678-S03456 and were evolved for $T = 500$ time steps. Three examples of histogram $H_g^D(p)$, using different values of D, are given in Fig. 3(b). For each network model, we plotted the corresponding histograms for each TEP sample (Fig. 3(a)). The samples from different network models (classes) are hard to distinguish because they are visually very similar. However, the corresponding histograms of the same classes are similar while the histograms of different classes are different, thus minimizing intra-class variance and maximizing inter-class variance, which corroborates the robustness of our approach.

3.2. Signature

The feature vectors (i.e, the signature) are constructed based on the histograms H_g^D and H_k^D. The first feature vector considered is composed by the global histograms H_g^D with different D values. This strategy aims to combine different sizes of binary patterns for the network representation, according to

$$\vec{\Upsilon^g}_{D_1,D_2,...,D_n} = [H_g^{D_1}, H_g^{D_2}, ..., H_g^{D_n}]. \tag{9}$$

Then, once there is a histogram H_k^D for each degree of the network, we propose the average μ and the standard deviation σ of all these histograms for network characterization. Thus, it is possible to analyze the frequency of the binary patterns in vertices with different degree. For instance, the frequency of a given binary pattern can be high for vertices with a low degree and the opposite for vertices with high degree. In this way, the average feature vector $F(D)_\mu$ and the standard deviation feature vector $F(D)_\sigma$ are obtained, according to

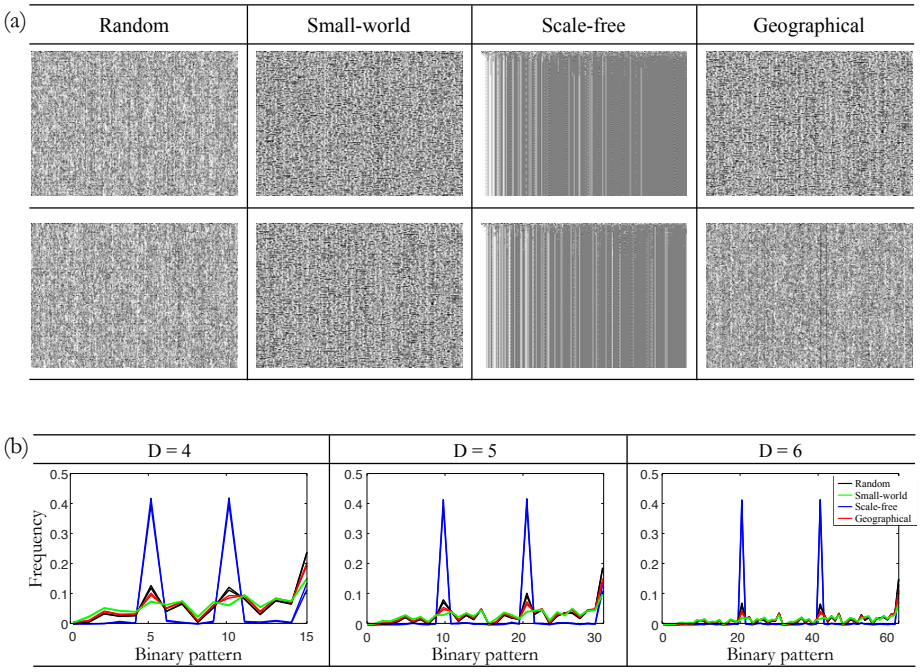

Figure 3: Two samples of time-evolution patterns for four different network models: random, small-world, scale-free and geographical and their histograms. (a) The TEPs correspond to networks with $N = 500$ vertices and average degree $\langle k \rangle = 10$ that were evolved with rule B135678-S03456 during $T = 350$ time steps. Each column is a different network model and each line is a different network sample. (b) Histograms corresponding to the global histogram $H_g^D(p)$ considering different values of binary pattern length D. Each curve corresponds to each of the samples in (a).

$$F(D)_u = \mu\left(\begin{bmatrix} H^D_{k_1} \\ H^D_{k_2} \\ \vdots \\ H^D_{k_K} \end{bmatrix}\right), \tag{10}$$

$$F(D)_\sigma = \sigma\left(\begin{bmatrix} H^D_{k_1} \\ H^D_{k_2} \\ \vdots \\ H^D_{k_K} \end{bmatrix}\right), \tag{11}$$

where K is the maximum number of degrees in the network. To describe the network, we combined the average feature vector $F(D_1)_\mu$ considering different D values. Thus, it is obtained the features vector $\vec{\Upsilon^\mu}_{D_1,D_2,...,D_n}$ as follows:

$$\vec{\Upsilon^\mu}_{D_1,D_2,...,D_n} = [F(D_1)_\mu, F(D_2)_\mu, ..., F(D_n)_\mu]. \tag{12}$$

The standard deviation feature vector $F(D_1)_\sigma$ also is combined using different D values to compose the feature vector $\vec{\Upsilon^\sigma}_{D_1,D_2,...,D_n}$, according to

$$\vec{\Upsilon^\sigma}_{D_1,D_2,...,D_n} = [F(D_1)_\sigma, F(D_2)_\sigma, ..., F(D_n)_\sigma]. \tag{13}$$

Finally, we propose a robust feature vector $\vec{\Psi}_{D_1,D_2,...,D_n}$, as follows,

$$\vec{\Psi}_{D_1,D_2,...,D_n} = [\vec{\Upsilon^g}_{D_1,D_2,...,D_n}, \vec{\Upsilon^\mu}_{D_1,D_2,...,D_n}, \vec{\Upsilon^\sigma}_{D_1,D_2,...,D_n}], \tag{14}$$

which is composed by concatenating the feature vectors $\vec{\Upsilon^g}_{D_1,D_2,...,D_n}$, $\vec{\Upsilon^\mu}_{D_1,D_2,...,D_n}$ and $\vec{\Upsilon^\sigma}_{D_1,D_2,...,D_n}$. This final feature vector combines different information about the TEPs and, consequently, about the network structure, improving the classification performance. In the experiments, this feature vector obtained the best classification results.

3.3. Computational Complexity

Consider a network with N vertices and average degree $\langle k \rangle$. To calculate the computational complexity of our approach LLNA-BP, let first analyze the complexity of the LLNA, which provides the TEP. Given a rule, the LLNA approach computes the next state of each vertex based on the neighboring vertex states. In this way, for each LLNA iteration, $N \times \langle k \rangle$ operations are required. Once the LLNA is iterated by T time steps, to obtain a TEP, $N \times \langle k \rangle \times T$ operations are needed. Therefore, the computational complexity of the LLNA approach to obtain a TEP is given by $O(N \times \langle k \rangle \times T)$.

The proposed approach computes the binary pattern for each network vertex considering the dictionary size D. Thus, $N \times T \times D$ operations are performed in order to obtain the binary patterns histograms. Therefore, considering all the steps of our proposed method LLBP-BP, the computational complexity is $O((N \times T) \times (\langle k \rangle + D))$.

4. Experiments

In this section, we detail the network databases, comparison methods, and validation strategy and classifier used in the experiments. We also describe the experimental setup used by the proposed method.

4.1. Network databases

In this section, we describe the synthetic networks database (*4-models*, *4-models* + $\langle k \rangle$, *noisy* and *scalefree*) and the real networks database (*social* and *metabolic*) used in the experiments. These databases can be downloaded at http://scg.ifsc.usp.br/LLNA. The proposed method and others are assessed over eight different networks databases. Detailed information about the databases is presented below.

- *4-models synthetic-database*: this database is composed of synthetic networks built according to four models: i) random (Erdös & Rényi [13]), with connection probability between two vertices of $p = \langle k \rangle / n$; ii) small-world (Watts & Strogatz [41]), with rewiring probability of $p = 0.1$; iii) scale-free (Barabási & Albert [4]), with both linear and non-linear preferential attachments, and, iv) geographical (Waxman [42]). Each network model was generated following the parameters: $\langle k \rangle$: 4, 6, 8, 10, 12, 14, 16; and, N: 500, 1000, 1500 and 2000. Each of the 28 combinations of $\langle k \rangle - N$ contains 100 networks. Thus, the number of networks for each model is 2800, resulting in a total of 11200 networks in the database;

- *4-models* + $\langle k \rangle$ *synthetic-database*: a second experiment was performed to classify the 28 classes consisting of the combinations $\langle k \rangle - N$, therefore, in this database there are 28 classes with 100 samples each;

- *Noisy-synthetic-database*: this database is composed of the same networks of the *synthetic-database* using three different ρ values: 10%, 20% and 30%, where the noise rate ρ was applied into the networks aiming to modify the network topology by the removal and the addition of edges. Thus, $\frac{\rho}{2}$ of edges are added, regarding the total number of edges, and, $\frac{\rho}{2}$ are removed. Therefore, as the ρ increases, more structural changes are performed on the network topology [28]. This database was used to evaluate the robustness of the method regarding noise tolerance;

- *Scalefree-synthetic-database*: this database contains scale-free networks generated using two different models: Barabási & Albert [4] and Dorogovtsev & Mendes [12]. For the former model, it contains networks with both linear and non-linear preferential attachments (α): 0.5, 1.0, 1.5 and 2.0. In this way, this database has five classes with 100 networks ($N = 1000$ vertices and $\langle k \rangle = 8$) per class;

- *Social-networks-database*: composed of networks from the SNAP (Stanford Network Analysis Project) platform [19]. Each social network (or

"ego-networks") corresponds to the social relationships or friendships of a specific user that is not represented in the network. This database contains 100 samples divided into two classes (Google+ and Twitter) with 50 network samples for each;

- *Metabolic-networks-database*: this database is composed of metabolic networks that were constructed using the substrate-product network model [47]. Thus, metabolites (vertices) from substrates are linked to metabolites from products per each reaction. The biochemical reactions of several organisms were obtained from the Kyoto Encyclopedia of Genes and Genomes database [18] (KEGG). This database was subdivided into seven sets (see Tables S1-S7 from the Supplementary Material), as follows:
  - *kingdom-database*: this database comprises species from the *eukaryota* domain of life, which consists of four kingdoms: *animals*, *plants*, *fungi* and *protist*, each of them containing 40 networks.
  - *Animal-database*: this database contains four classes: *mammals*, *birds*, *fishes* and *insects*, containing 14 samples per class.
  - *Fungi-database*: this database contains four classes: *saccharomycetes*, *sordariomycetes*, *eurotiomycetes* and *basidiomycetes*, each of them containing 15 networks.
  - *Plant-database*: contains three classes *Monocots*, *Green algae* and *Eudicots*, containing 9 organism per each class.
  - *Protist-database*: this database comprises four classes *Amoebozoa*, *Alveolates*, *Stramenopiles* and *Euglenozoa*, each of them containing 5 organisms.
  - *Firmicutes-Bacillis-database*: this database presents four classes *Bacillus*, *Staphylococcus*, *Streptococcus* and *Lactobacillus*, containing unbalanced number of species, 122, 76, 133 and 83 respectively.
  - *Actinobacteria-database*: This database is also unbalanced, presenting three classes *Mycobacterium*, *Corynebacterium* and *Streptomyces* with 60, 86, and 53 species, respectively.
- *Metabolic-rule-selection-database*: Since the choice of the best LLNA rules for each classification problem should be performed on a exclusive database, therefore we generated seven databases containing metabolic networks from other species of the same classes, to be used for the LLNA rule selection (see Section 2.2). Thus, the *kingdom-selection-database* comprises 9 samples per each class. The *animal-selection-database*, *fungi-selection-database*, *plant-selection-database* and *protist-selection-database* comprises 2 samples per each class, while the *firmicutes-Bacillis-selection-database* and *actinobacteria-selection-database* contains 10 networks per class. The complete list of organisms and their respective network databases are found in Tables S8-S14 from the Supplementary Material.

4.2. LLNA-BP setup

In this section, we detail the network-automata configuration to generate the TEPs used in the experiments. Since the purpose of our approach is to improve the TEPs characterization, thus we used the same parameters regarding the specifications of the LLNA from the original work [28]. Therefore, the initial configuration s_0 was setup randomly using probabilities of 50% of cells alive, and the network-automata were evolved during $T = 350$ iterations. Furthermore, we used the same optimal rules selected for each of the databases used in Ref. [28].

The rule B135678-S03456 was used for the *4-models synthetic-database* and the *noisy-synthetic-database*. For the *scale-free synthetic-database* was considered the rule B0157-S457, while rule B01678-S0457 was considered for the *4-models + $\langle k \rangle$ synthetic-database*. Regarding the real-world applications, for the *social-database* was used the rule B0167-S248. Thus, it is possible to compare in a fair way the proposed approach and the previous method.

Since, in this work, we introduced new metabolic networks databases, therefore we also performed the rule selection procedure (see Section 2.2) using the *metabolic-rule-selection-database*, which contains a small number of other metabolic networks for each classification problem. We computed the rule selection following the same criteria established by Miranda et al. [28]. As a result, we selected the following optimal rules: B02345678-S123468, B023468-S01468, B04-S1468, B0468-S0467, B0236-S123567, B0468-S0458, B1237-S267 for the *Kingdom*, *Animal*, *Fungi*, *Plant*, *Protist*, *Firmicutes-Bacillis* and *Actibacteria-databases* respectively.

4.3. Classification setup

In the classification process, it was used a 10-fold cross-validation strategy, which is a generalized way to evaluate the prediction capacity of a model. This strategy divides the dataset into 10 equal or almost equal subsets, one subset is used for testing and the rest of the 9 for training. All subsets are used as test set. Since this validation strategy is not deterministic, the cross-validation procedure was applied 100 times and, the mean accuracy and standard deviation are considered as the performance of the method. The SVM (Support Vector Machines) classifier was used in the experiments, in order to follow the setup experiments of Miranda et al [28]. This classifier uses hyperplanes as decision boundaries and the classification is performed by finding the hyperplane that yields the maximal separation between two classes [16].

4.4. Comparison methods

In this work, we compared the performance of the LLNA-BP with the previous LLNA [28]. Therefore, we reproduced the results obtained in Ref. [28] using the Shannon entropy, word length, and Lempel-Ziv complexity histogram distribution given by $[\vec{\mu}_S, \vec{\mu}_W, \vec{\mu}_L]$ as the feature vector. Besides the comparison with the LLNA method, the LLNA-BP was also compared to the classical topological networks measurements. As suggested by Costa *et al.* [7], a set of

network measurements from different categories, such as mean degree, degree distributions, correlations, distances, and path lengths, hierarchical, spectral measures, transitivity, and clustering coefficient, among others, are commonly used to characterize the topology of networks. Therefore, we used the combination of the average degree ($\langle k \rangle$), average hierarchical degree of level 1 ($\langle H_{k_1} \rangle$) and level 2 ($\langle H_{k_2} \rangle$), average clustering coefficient ($\langle cc \rangle$), average path length (l) and degree Pearson correlation (ρ_P) [8, 7]. More specifically their equations are stated as follows:

- $\langle k \rangle = \frac{1}{N} \sum_{i=1}^{N} k_i$;
- $\langle H_{k_h} \rangle = \frac{1}{N} \sum_i k_i^h$, when $h = 2$ it represents the sum of the degrees of the neighbors of vertex v_i. Similarly, when $h = 3$ represents the sum of the degrees of the neighbors of the neighbors of vertex v_i;
- $\langle cc \rangle = \frac{1}{N} \sum_i cc_i$, where the clustering coefficient is defined by $cc_i = \frac{e_i}{k_i(k_i-1)}$ and e_i is the number of pairs connected to each of all the neighbors of v_i;
- $\langle l \rangle$, is the average length of the shortest paths between any two vertices of the network.
- ρ_P is a measure of the network assortativity.

5. Results

In this section, we report the experimental results obtained in the classification task applied to all of the network databases studied here. We also analyzed the performance of the proposed method using different parameters. Finally, we also compare the classification performance between our proposed method, traditional network measurements and the previous approach [28].

5.1. Parameter evaluation

In order to efficiently apply our proposed approach for network classification, we first must define the ideal feature vector and binary pattern length of D. Table 1 shows the results achieved on the *social-database* when using different values of binary pattern length D to compute the proposed feature vectors $\vec{\Upsilon^g}_{D_1,D_2,\ldots,D_n}$, $\vec{\Upsilon^\mu}_{D_1,D_2,\ldots,D_n}$, $\vec{\Upsilon^\sigma}_{D_1,D_2,\ldots,D_n}$ and $\vec{\Psi}_{D_1,D_2,\ldots,D_n}$. The results refer to the percentage of network samples correctly classified using each parameter. The results show that the proposed method is robust, achieving good results for all feature vectors and combinations of D. Concerning the different feature vectors, the best results were obtained with the feature vector $\vec{\Psi}_{D_1,D_2,\ldots,D_n}$. This feature vector contains more information about the network patterns because it combine the global features and specific degree characteristics ($\vec{\Upsilon^\mu}_{D_1,D_2,\ldots,D_n}$ and $\vec{\Upsilon^\sigma}_{D_1,D_2,\ldots,D_n}$).

Considering only the feature vector $\vec{\Psi}_{D_1,D_2,\ldots,D_n}$, the Table 2 and Table 3 show the results for the *synthetic-database* and the *metabolic-database* using

Table 1: Accuracy (%) and standard deviation for the *social-network-database* using different feature vectors (columns) and various combinations of D values (rows).

$\{D_1, D_2, ..., D_n\}$	$\vec{\Upsilon^g}_{D_1,D_2,...,D_n}$	$\vec{\Upsilon^\mu}_{D_1,D_2,...,D_n}$	$\vec{\Upsilon^\sigma}_{D_1,D_2,...,D_n}$	$\vec{\Psi}_{D_1,D_2,...,D_n}$
{4}	93.00 (± 0.00)	93.00 (± 0.00)	91.00 (± 0.00)	92.80 (± 0.40)
{5}	92.70 (± 0.64)	93.00 (± 0.00)	90.90 (± 0.30)	92.40 (± 0.49)
{6}	93.10 (± 0.30)	93.20 (± 0.75)	90.20 (± 0.40)	92.70 (± 0.64)
{7}	93.00 (± 0.00)	93.20 (± 0.40)	90.20 (± 0.60)	92.50 (± 1.02)
{8}	92.90 (± 0.70)	92.00 (± 0.63)	89.30 (± 0.90)	92.50 (± 1.02)
{4, 5}	93.00 (± 0.63)	93.00 (± 0.00)	90.90 (± 0.30)	92.30 (± 0.46)
{4, 6}	93.00 (± 0.00)	93.10 (± 0.70)	90.10 (± 0.54)	92.40 (± 0.66)
{4, 7}	93.00 (± 0.00)	93.10 (± 0.30)	90.30 (± 0.46)	92.30 (± 0.78)
{4, 8}	92.80 (± 0.87)	91.70 (± 0.78)	89.30 (± 1.19)	92.80 (± 0.75)
{5, 6}	93.20 (± 0.40)	93.10 (± 0.54)	90.00 (± 0.45)	92.50 (± 0.67)
{5, 7}	**93.10 (± 0.30)**	**93.10 (± 0.30)**	**90.00 (± 0.63)**	**93.40 (± 0.92)**
{5, 8}	92.80 (± 0.87)	91.70 (± 0.78)	89.40 (± 1.02)	92.90 (± 0.70)
{6, 7}	93.10 (± 0.30)	93.00 (± 0.45)	89.90 (± 0.54)	93.40 (± 1.02)
{6, 8}	92.80 (± 0.87)	91.60 (± 0.66)	89.80 (± 0.87)	92.90 (± 1.04)
{7, 8}	92.00 (± 0.63)	91.40 (± 0.66)	89.20 (± 1.08)	92.70 (± 1.00)
{4, 5, 6}	93.20 (± 0.40)	93.10 (± 0.30)	89.90 (± 0.54)	92.60 (± 0.49)
{4, 5, 7}	93.10 (± 0.30)	93.00 (± 0.45)	90.10 (± 0.70)	93.60 (± 0.92)
{4, 5, 8}	92.80 (± 0.87)	91.70 (± 0.78)	89.40 (± 1.02)	93.10 (± 0.70)
{4, 6, 7}	93.10 (± 0.30)	93.00 (± 0.45)	90.00 (± 0.45)	93.70 (± 0.78)
{4, 6, 8}	92.70 (± 0.90)	91.60 (± 0.66)	89.40 (± 1.02)	93.20 (± 0.87)
{4, 7, 8}	92.00 (± 0.63)	91.40 (± 0.66)	89.10 (± 0.83)	92.60 (± 1.02)
{5, 6, 7}	93.10 (± 0.30)	92.90 (± 0.54)	89.90 (± 0.54)	93.90 (± 0.83)
{5, 6, 8}	92.60 (± 0.80)	91.60 (± 0.66)	89.60 (± 0.80)	93.30 (± 0.78)
{5, 7, 8}	91.90 (± 0.70)	91.40 (± 0.66)	89.10 (± 0.83)	93.00 (± 1.18)
{6, 7, 8}	91.70 (± 0.78)	91.30 (± 0.64)	88.70 (± 0.78)	93.20 (± 1.08)

different values and combinations of D. On the *animal-database* and the *fungi-database* we can note that the accuracy tends to increase as we increase the value of D. On the other hand, the results obtained on the *synthetic-database*, *social-database* and *kingdom-database* show a high correct classification rate for any combination of D values. In this way, for these tests, we did not combine more than three D values to compose the feature vectors, in order to avoid a large number of descriptors. Also, the results show a small improvement in the classification rate as we combine more than two D values. Therefore, we suggest the feature vector $\vec{\Psi}_{5,7}$, as it represents a good tradeoff between performance and number of features of all databases, to be used as a default parameter of our approach. This feature vector has 480 descriptors that are much smaller as compared to the size of other proposed feature vectors such as $\vec{\Psi}_{6,7,8}$, which has 1344 features.

Table 2: Results for different combinations of D values with the feature vector $\vec{\Psi}_{D_1,D_2,...,D_n}$ on the *synthetic-databases*.[a].

$\{D_1, D_2, ..., D_n\}$	*4-models*	*4-models* + $\langle k \rangle$	*Scale-free*	*Noisy Networks*		
				$\rho = 10\%$	$\rho = 20\%$	$\rho = 30\%$
{4}	99.53	94.39	99.36 (± 0.23)	99.44	99.09	98.65
{5}	99.98	96.71	99.46 (± 0.21)	99.96	99.94	99.77
{6}	100.00	97.60	99.54 (± 0.19)	100.00	99.99	99.97
{7}	100.00	98.23	99.58 (± 0.22)	100.00	99.98	99.99
{8}	100.00	98.58	99.70 (± 0.19)	100.00	99.99	99.97
{4, 5}	99.98	96.99	99.32 (± 0.21)	99.95	99.94	99.75
{4, 6}	100.00	97.75	99.32 (± 0.25)	100.00	99.99	99.96
{4, 7}	100.00	98.27	99.36 (± 0.23)	100.00	99.99	99.99
{4, 8}	100.00	98.59	99.38 (± 0.26)	100.00	99.99	99.97
{5, 6}	100.00	97.90	99.50 (± 0.22)	100.00	99.99	99.96
{5, 7}	**100.00**	**98.31**	**99.52 (± 0.19)**	**100.00**	**99.98**	**99.99**
{5, 8}	100.00	98.61	99.52 (± 0.19)	100.00	99.99	99.97
{6, 7}	100.00	98.40	99.56 (± 0.21)	100.00	99.99	99.99
{6, 8}	100.00	98.63	99.56 (± 0.21)	100.00	99.99	99.98
{7, 8}	100.00	98.68	99.66 (± 0.25)	100.00	99.99	99.98
{4, 5, 6}	100.00	97.96	99.46 (± 0.27)	100.00	99.99	99.96
{4, 5, 7}	100.00	98.39	99.48 (± 0.19)	100.00	99.98	99.99
{4, 5, 8}	100.00	98.65	99.46 (± 0.25)	100.00	99.99	99.97
{4, 6, 7}	100.00	98.46	99.42 (± 0.24)	100.00	99.98	99.98
{4, 6, 8}	100.00	98.66	99.42 (± 0.24)	100.00	99.99	99.97
{4, 7, 8}	100.00	98.70	99.46 (± 0.19)	100.00	99.99	99.98
{5, 6, 7}	100.00	98.48	99.54 (± 0.19)	100.00	99.98	99.99
{5, 6, 8}	100.00	98.68	99.52 (± 0.19)	100.00	99.99	99.98
{5, 7, 8}	100.00	98.72	99.52 (± 0.19)	100.00	99.99	99.98
{6, 7, 8}	100.00	98.73	99.58 (± 0.22)	100.00	99.99	99.98

[a] Smallest standard deviations between ±0.0 and ±0.09 are not shown.

Table 3: Accuracy (%) and standard deviation for the classification of three *metabolic-networks-database* using different combinations of D values with the feature vector $\tilde{\Psi}_{D_1,D_2,...,D_n}$.

$\{D_1, D_2, ..., D_n\}$	*Kingdom*	*Animal*	*Fungi*
{4}	94.12 (± 6.08)	68.03 (± 20.30)	62.00 (± 17.90)
{5}	96.19 (± 4.69)	78.00 (± 16.65)	71.50 (± 17.62)
{6}	97.06 (± 4.57)	78.43 (± 16.31)	75.00 (± 17.33)
{7}	97.25 (± 4.20)	83.63 (± 15.52)	77.17 (± 16.69)
{8}	96.56 (± 4.20)	81.53 (± 16.39)	77.50 (± 17.14)
{4, 5}	96.19 (± 4.69)	74.80 (± 17.36)	70.67 (± 15.91)
{4, 6}	97.12 (± 4.21)	78.53 (± 17.35)	74.67 (± 17.80)
{4, 7}	97.50 (± 4.07)	82.57 (± 16.02)	77.67 (± 17.12)
{4, 8}	96.56 (± 4.20)	81.77 (± 15.97)	78.83 (± 16.56)
{5, 6}	96.69 (± 4.65)	78.03 (± 17.29)	74.50 (± 17.32)
{5, 7}	**97.44 (± 3.98)**	**84.87 (± 15.25)**	**76.17 (± 17.45)**
{5, 8}	96.69 (± 4.11)	82.27 (± 15.75)	78.33 (± 16.33)
{6, 7}	97.19 (± 4.29)	82.93 (± 14.61)	76.67 (± 17.08)
{6, 8}	96.62 (± 4.30)	82.47 (± 15.59)	78.17 (± 16.87)
{7, 8}	96.56 (± 4.20)	82.60 (± 15.33)	77.50 (± 16.65)
{4, 5, 6}	96.69 (± 4.65)	77.60 (± 17.44)	73.83 (± 17.29)
{4, 5, 7}	97.50 (± 3.87)	82.73 (± 15.84)	76.33 (± 16.95)
{4, 5, 8}	96.75 (± 4.12)	81.20 (± 15.98)	79.17 (± 16.30)
{4, 6, 7}	97.44 (± 4.08)	83.10 (± 15.51)	76.50 (± 18.06)
{4, 6, 8}	96.81 (± 4.21)	82.27 (± 15.57)	78.83 (± 16.73)
{4, 7, 8}	96.62 (± 4.11)	82.60 (± 15.85)	78.33 (± 16.50)
{5, 6, 7}	97.44 (± 3.98)	82.80 (± 15.16)	75.67 (± 18.11)
{5, 6, 8}	96.94 (± 4.12)	82.63 (± 15.41)	78.50 (± 16.29)
{5, 7, 8}	96.75 (± 4.12)	82.77 (± 16.12)	77.83 (± 16.42)
{6, 7, 8}	96.69 (± 4.30)	83.37 (± 15.87)	78.50 (± 16.46)

5.2. Comparison and discussion

In order to scrutinize more the evaluation of our proposed approach, we compare the results using classical network measurements and the previous LLNA. The classical network measurements used in this experiment are described in Section 4.4. For the previous LLNA, we replicated the results obtained from Miranda et al. [28], as detailed in Section 4.4. For our proposed approach, we considered the feature vector $\vec{\Psi}_{5,7}$ in all databases, since it has obtained a good performance as reported in Section 5.1.

Table 4 presents the results obtained for each approach evaluated in different network databases. The *4-models synthetic-database* and the *noisy-synthetic-database* showed that the proposed method obtained very similar accuracies compared to the other approaches. On the other hand, regarding the *4-model* + $\langle k \rangle$ database, the LLNA-BP (98.31%) outperformed the previous LLNA (90.76%) and the classical network measurements (65.20%). Also, on the *scalefree-synthetic-database*, our method presents a slight improvement of 1.22% and 3.32% compared to the previous LLNA and the classical network measurements, respectively.

The proposed method was also compared with other real-world networks databases (see Table 4). The classification tasks from these databases is a challenging problem. Each application has different properties allowing to evaluate the generalization ability of the method. Regarding the task of identifying structural patterns in social networks, the proposed method obtained the highest accuracy when compared to the other approaches. Our method provided an accuracy improvement of 1.4% for the previous LLNA, and 5.4% for the classical measures, demonstrating to be a better discrimination method.

Regarding the task of identifying organisms using metabolic networks, the proposed method also presented higher accuracy than the other approaches. On the *fungi-database* the experimental results indicate that our method significantly improves accuracy, e.g. from 54.58% (± 19.38) to 76.17% (± 17.45) over the previous LLNA and from 54.90% (± 15.39) to 76.17% (± 17.45) over the classical network measures. On the *animal-database* an accuracy of 84.87% (± 15.25) is achieved by the proposed method, which is followed by the accuracy of 83.71% (± 15.29) and 77.25% (± 16.29) obtained by the classical network measures and previous LLNA, respectively. For the *kingdom-database*, the maximum accuracy obtained with proposed method was 97.44% (± 3.98) in contrast to 96.61% (± 4.33) using the classical network measures as a feature vector.

In the experiment using the *Plant-database*, the proposed method obtained the highest accuracy, 74.81%(±5.64). On the *Protist-database*, the proposed method LLNA-BP outperformed the previous LLNA and the structural measures by 18.45% and 41.90%, respectively. In the other two databases (*Firmicutes-Bacillis* and *Actinobacteria*), the LLNA-BP also obtained a better performance when compared to the others. Note that on some databases it is evident that structural measures cannot deal with the complex patterns and thus, it is not a good option for the network characterization. Besides that, the results demonstrate that the binary pattern's signature can better characterize the complex-

ity of the TEPs generated by the LLNA. In general, the experimental results showed that our approach improved the accuracy in all the studied databases when compared to the original LLNA.

Table 4: Comparison of the accuracy (%) and standard deviation obtained by the proposed approach, the original LLNA and the structural network measures applied in the classification of eight network databases. The LLNA-BP used the feature vector $\vec{\Psi}_{5,7}$. The LLNA used $[\vec{\mu}_S,\ \vec{\mu}_W,\ \vec{\mu}_L]$ as a feature vector. The structural measurements feature vector was $[\langle k\rangle, \langle H_{k_1}\rangle, \langle H_{k_2}\rangle, \langle cc\rangle, l, \rho_P]$.

	Databases	**Approaches**		
		LLNA-BP	**LLNA**	**Structural measures**
Synthetic	*4-models*	100.0 (± 0.00)	99.992 (± 0.002)	100.0 (± 0.00)
	4-model + $\langle k\rangle$	98.31 (± 0.02)	90.76 (± 0.07)	65.20 (± 0.20)
	Scale-free	99.52 (± 0.19)	98.30 (± 0.2)	96.20 (± 0.04)
	Noise $\rho = 10\%$	100.0 (± 0.00)	99.983 (± 0.004)	100.0 (± 0.00)
	Noise $\rho = 20\%$	99.98 (± 0.00)	99.97 (± 0.01)	100.0 (± 0.00)
	Noise $\rho = 30\%$	99.99 (± 0.00)	99.95 (± 0.01)	100.0 (± 0.00)
Real	*Social*	93.40 (± 0.92)	92.00 (± 1.00)	88.00 (± 2.00)
	Kingdom	97.44 (± 3.98)	93.10 (± 5.38)	96.61 (± 4.33)
	Animal	84.87 (± 15.25)	77.25 (± 16.29)	83.71 (± 15.29)
	Fungi	76.17 (± 17.45)	54.58 (± 19.38)	54.90 (± 15.39)
	Plant	74.81 (± 5.64)	69.70 (± 4.67)	54.19 (± 9.17)
	Protist	87.00 (± 5.29)	68.55 (± 6.30)	45.10 (± 10.02)
	Firmicutes-Bacillis	98.30 (± 1.17)	84.63 (± 2.00)	95.67 (± 0.59)
	Actinobacteria	95.13 (± 1.22)	91.48 (± 1.60)	93.16 (± 0.70)

5.3. Processing Time

In order to evaluate the processing time, we consider networks from the *scalefree-synthetic-database* with $\langle k\rangle = 16$. The proposed method LLNA-BP took on average 0.10s, 0.21s, 0.32s, and 0.45s to compute the signatures from the networks with $N = 500$, 1000, 1500 and 2000 nodes, respectively. On the other hand, the traditional LLNA method took on average, 0.04s, 0.09s, 0.14s, and 0.19s to obtain the signatures from the same network samples, respectively. In these tests, we used a 3.60GHz Intel(R) Core i7, 64GB RAM and 64-bit Operating System. Although the LLNA-BP has spent a slightly higher running time compared to the traditional LLNA, the results indicate that both approaches have very competitive running times for real-time application.

6. Conclusions

In this paper, we presented an innovative manner to improve the feature extraction of LLNA time-evolution patterns in the context of pattern recognition. Thus, we proposed to represent their sequences of zeros and ones into frequencies of binary patterns, from which various descriptors were used for network

characterization. We have demonstrated the robustness of our method with respect to two types of databases: synthetic and real-world networks. First, we investigated the effect of the involved parameters on the performance of classification on the databases. From this study, it was possible to define a set of default parameters for all databases that represents a good tradeoff between performance and number of features. Then, experimental results confirmed the effectiveness of the proposed method against the original LLNA and structural network measurements obtained directly from the network topology when used as feature vectors. In particular, the proposed method significantly improves the accuracy compared to the original LLNA in real-world applications. This clearly demonstrates that even in challenging tasks, the proposed method leads to an effective interpretation of the complex patterns present in the TEPs. Finally, besides we showed an interesting case where the binary patterns descriptors improve the performance, certainly, this same proposal can be extended to other cellular automata TEPs.

Acknowledgments

Lucas C. Ribas gratefully acknowledges the financial support grant #2016/23763-8, São Paulo Research Foundation (FAPESP). Jeaneth Machicao acknowledges the scholarship from the National Council for Scientific and Technological Development (CNPq grant #405503/2017-2 and #155957/2018-0). Odemir M. Bruno acknowledges support from CNPq (grant #307897/2018-4) and FAPESP (Grant #s 14/08026-1 and 16/18809-9). All of the authors gratefully acknowledge the support of NVIDIA Corporation with the donation of the Quadro P6000 GPU and the GTX XP GPU used for this research.

References

[1] C. Akimushkin, D. R. Amancio, O. N. Oliveira, Jr., Text Authorship Identified Using the Dynamics of Word Co-Occurrence Networks, PLOS ONE 12 (1) (2017) 1–15.

[2] K. Arii, L. Parrott, Examining the colonization process of exotic species varying in competitive abilities using a cellular automaton model, Ecological Modelling 199 (3) (2006) 219–228.

[3] A. Banerjee, J. Jost, Spectral plot properties: Towards a qualitative classification of networks, NHM 3 (2) (2008) 395–411.

[4] A.-L. Barabási, R. Albert, Emergence of scaling in random networks, Science 286 (5439) (1999) 509–512.

[5] E. Bullmore, O. Sporns, Complex brain networks: graph theoretical analysis of structural and functional systems, Nature Reviews Neuroscience 10 (3) (2009) 186–198.

[6] B. Chopard, P. Luthi, M. Droz, Reaction-diffusion cellular automata model for the formation of Leisegang patterns, Physical Review Letters 72 (9) (1994) 1384–1387.

[7] L. d. F. Costa, P. R. V. Boas, F. Silva, F. Rodrigues, A pattern recognition approach to complex networks, Journal of Statistical Mechanics: theory and experiment 2010 (11) (2010) P11015.

[8] L. d. F. Costa, F. Rodrigues, G. Travieso, P. R. V. Boas, Characterization of complex networks: A survey of measurements, Advances in Physics 56 (1) (2007) 167–242.

[9] T. Czaran, S. Bartha, Spatiotemporal dynamic models of plant populations and communities, Trends in Ecology & Evolution 7 (2) (1992) 38–42.

[10] N. R. da Silva, J. Baetens, M. Oliveira, B. d. Baets, O. Bruno, Classification of cellular automata through texture analysis, Information Sciences 370–371 (2016) 33–49.

[11] P. S. Dodds, R. Muhamad, D. J. Watts, An experimental study of search in global social networks, science 301 (5634) (2003) 827–829.

[12] S. N. Dorogovtsev, J. F. Mendes, Evolution of networks, Advances in physics 51 (4) (2002) 1079–1187.

[13] P. Erdös, A. Rényi, On random graphs, Publicationes Mathematicae Debrecen 6 (1959) 290–297.

[14] H. A. Filho, J. Machicao, O. M. Bruno, A hierarchical model of metabolic machinery based on the kcore decomposition of plant metabolic networks, PLOS ONE 13 (5) (2018) e0195843.

[15] J. M. Gomez, M. Verdu, F. Perfectti, Ecological interactions are evolutionarily conserved across the entire tree of life, Nature 465 (2010) 918—921.

[16] M. Hearst, S. Dumais, E. Osuna, J. Platt, B. Scholkopf, Support vector machines, IEEE Intelligent Systems and their Applications 13 (4) (1998) 18–28.

[17] H. Jeong, B. Tombor, R. Albert, Z. N. Oltvai, A.-L. Barabási, The large-scale organization of metabolic networks, Nature 407 (6804) (2000) 651–654.

[18] M. Kanehisa, Y. Sato, M. Kawashima, M. Furumichi, M. Tanabe, KEGG as a reference resource for gene and protein annotation, Nucleic Acids Research 44 (D1) (2015) D457–D462.

[19] J. Leskovec, A. Krevl, SNAP Datasets: Stanford Large Network Dataset Collection, `http://snap.stanford.edu/data`, 2014.

[20] J. Machicao, J. M. Baetens, A. G. Marco, B. D. Baets, O. M. Bruno, A dynamical systems approach to the discrimination of the modes of operation of cryptographic systems, Communications in Nonlinear Science and Numerical Simulation 29 (1-3) (2015) 102–115.

[21] J. Machicao, E. A. Corrêa, G. H. B. Miranda, D. R. Amancio, O. M. Bruno, Authorship attribution based on Life-Like Network Automata, PLOS ONE 13 (3) (2018) e0193703.

[22] J. Machicao, H. A. Filho, D. J. G. Lahr, M. Buckeridge, O. M. Bruno, Topological assessment of metabolic networks reveals evolutionary information, Scientific Reports 8 (1).

[23] J. Machicao, A. G. Marco, O. M. Bruno, Chaotic encryption method based on life-like cellular automata, Expert Systems with Applications 39 (16) (2012) 12626–12635.

[24] J. Machicao, L. C. Ribas, L. F. Scabini, O. M. Bruno, Cellular automata rule characterization and classification using texture descriptors, Physica A: Statistical Mechanics and its Applications 497 (2018) 109–117.

[25] C. Marr, M. Hütt, Outer-totalistic cellular automata on graphs, Physics Letters A 373 (5) (2009) 546–549.

[26] A. Mehri, A. H. Darooneh, A. Shariati, The complex networks approach for authorship attribution of books, Physica A: Statistical Mechanics and its Applications 391 (7) (2012) 2429 – 2437.

[27] G. H. Miranda, J. Machicao, O. M. Bruno, An optimized shape descriptor based on structural properties of networks, Digital Signal Processing 82 (2018) 216–229.

[28] G. H. B. Miranda, J. Machicao, O. M. Bruno, Exploring spatio-temporal dynamics of cellular automata for pattern recognition in networks, Scientific Reports 6 (37329).

[29] M. E. Newman, The structure of scientific collaboration networks, Proceedings of the National Academy of Sciences 98 (2) (2001) 404–409.

[30] G. Palla, I. Derényi, I. Farkas, T. Vicsek, Uncovering the overlapping community structure of complex networks in nature and society, Nature 435 (7043) (2005) 814–818.

[31] J. Poleszczuk, H. Enderling, A High-Performance Cellular Automaton Model of Tumor Growth with Dynamically Growing Domains, Applied Mathematics 05 (01) (2014) 144–152.

[32] J.-C. Rain, L. Selig, H. De Reuse, V. Battaglia, C. Reverdy, S. Simon, G. Lenzen, F. Petel, J. Wojcik, V. Schächter, et al., The protein–protein interaction map of Helicobacter pylori, Nature 409 (6817) (2001) 211–215.

[33] L. C. Ribas, J. J. Junior, L. F. Scabini, O. M. Bruno, Fusion of complex networks and randomized neural networks for texture analysis, arXiv preprint arXiv:1806.09170 .

[34] L. C. Ribas, M. B. Neiva, O. M. Bruno, Distance transform network for shape analysis, Information Sciences 470 (2019) 28 – 42.

[35] T. P. Ribeiro, L. N. Couto, A. R. Backes, C. A. Zorzo Barcelos, Texture Characterization via Automatic Threshold Selection on Image-Generated Complex Network, in: A. Pardo, J. Kittler (Eds.), Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications, Springer International Publishing, Cham, ISBN 978-3-319-25751-8, 468–476, 2015.

[36] P. Rosin, Training cellular automata for image processing, IEEE Transactions on Image Processing 15 (7) (2006) 2076–2087.

[37] L. F. Scabini, D. O. Fistarol, S. V. Cantero, W. N. Gonçalves, B. B. Machado, J. Jose F. Rodrigues, Angular descriptors of complex networks: A novel approach for boundary shape analysis, Expert Systems with Applications 89 (2017) 362 – 373.

[38] G. Sirakoulis, I. Karafyllidis, A. Thanailakis, A cellular automaton model for the effects of population movement and vaccination on epidemic propagation, Ecological Modelling 133 (3) (2000) 209–223.

[39] D. Smith, J. Onnela, C. Lee, M. Fricker, N. Johnson, Network automata: coupling structure and function in dynamic networks, Advances in Complex Systems 14 (3) (2011) 317–339.

[40] U. Stelzl, U. Worm, M. Lalowski, C. Haenig, F. H. Brembeck, H. Goehler, M. Stroedicke, M. Zenkner, A. Schoenherr, S. Koeppen, et al., A human protein-protein interaction network: a resource for annotating the proteome, Cell 122 (6) (2005) 957–968.

[41] D. J. Watts, S. H. Strogatz, Collective dynamics of 'small-world' networks, nature 393 (6684) (1998) 440–442.

[42] B. M. Waxman, Routing of multipoint connections, IEEE Journal on Selected Areas in Communications 6 (9) (1988) 1617–1622.

[43] S. Wolfram, Statistical mechanics of cellular automata, Reviews of Modern Physics 55 (3) (1983) 601–644.

[44] S. Wolfram, Cellular automata as models of complexity, Nature 311 (5985) (1984) 419–424.

[45] A.-C. Wu, X.-J. Xu, Y.-H. Wang, Excitable greenberg-hastings cellular automaton model on scale-free networks, Physical Review E 75 (3) (2007) 032901.

[46] R. Xin, J. Zhang, Y. Shao, Complex Network Classification with Convolutional Neural Network, CoRR abs/1802.00539, URL http://arxiv.org/abs/1802.00539.

[47] J. Zhao, H. Yu, J. Luo, Z. W. Cao, Y. Li, Complex networks theory for analyzing metabolic networks, Chinese Science Bulletin 51 (13) (2006) 1529–1537.

[48] Y. Zhao, S. Billings, D. Coca, Cellular automata modelling of dendritic crystal growth based on Moore and von Neumann neighbourhoods, International Journal of Modelling, Identification and Control 6 (2) (2009) 119.

[49] H. Zhou, R. Lipowsky, Dynamic pattern evolution on scale-free networks, Proceedings of the National Academy of Sciences of the United States of America 102 (29) (2005) 10052–10057.

[50] O. Zinovieva, A. Zinoviev, V. Ploshikhin, V. Romanova, R. Balokhonov, A solution to the problem of the mesh anisotropy in cellular automata simulations of grain growth, Computational Materials Science 108 (2015) 168–176.