Graph Classification of Evolutionary Dynamic Networks

Joseph Johnson[0000−1111−2222−3333] and Christophe Giraud-Carrier[0000−0002−4129−3239]

Department of Computer Science, Brigham Young University, Provo, UT, USA
josephjohnson11@gmail.com,cgc@cs.byu.edu

Abstract. We propose a model for graph classification of evolutionary dynamic networks that is based on game theoretic, network theoretic, and chaos theoretic principles. Increasingly complex algorithms are being developed for graph classification in highly-structured domains, such as image processing and climate forecasting. However, these methods tend to break down through over-fitting and running time when applied to human interaction networks. The method proposed in this paper incorporates the invariant principles of evolutionary pressure, adaptation, utility, and properties of chaotic trajectories, allowing explicit exploitation of latent structure while avoiding over-fitting when the system transitions into chaos. The approach is appropriate for any dynamic network in which agents both compete and cooperate, such as social media networks, stock markets, political campaigns, legislation, and geopolitical events. We demonstrate our method's value on two applications.

Keywords: Graph Classification · Dynamic Networks · Supervised Learning.

1 Introduction

Human systems are full of paradoxes, ambiguity, and contextual nuance, which are the product of unique evolutionary pressures wherein agents find themselves competing for scarce resources, yet having to cooperate in order to survive the broader competitive world [25,26], all within the limits of finite knowledge and perception [1]. As a result, human interactions differ fundamentally from interactions of the physical world. Indeed, they create dynamic networks in which some variables are "genuinely 'live' or 'free' variables, the value of which [one] cannot determine... [resulting in] neither an ordinary maximum problem, nor a conditional maximum problem, nor a problem of the calculus of variations" [14]. Hence, as determined by von Neumann and Morgenstern, the analysis of the complexity of human interactions requires novel methods [23].

Easley and Kleinberg's seminal work on networks provides an insightful view into these methods as part of a multidisciplinary framework, where "understanding highly-connected systems...requires a set of ideas for reasoning about *network structure*, *strategic behavior*, and the *feedback effects* they produce across large populations. These are ideas that have traditionally been dispersed across many different disciplines... Each of these fields brings important ideas to the discussion, and a full understanding seems to require a *synthesis of perspectives* from all of them." [9]. Hence, understanding the dynamics of human (highly-connected) systems requires elements of network theory (*network structure*), game theory (*strategic behavior*), and non-linear system dynamics or chaos (*feedback effects*).

Here, we explore how elements of these theories affect the data that is used to quantify them and propose a method for making predictions about human societies where competitive pressures exist (i.e., an evolutionary setting). While such environments are ubiquitous, we anticipate our method to be most useful in domains where actions are clearly quantified and the environment is highly fluid, such as early warning violence prediction, financial markets, and social media networks. The essence of our proposed approach is to transform a graph time-series classification task into a standard graph classification task by encoding the time-series data into each element of the time-series through fast Fourier transforms of relevant graph structural properties.

The remainder of the paper is organized as follows. We first review related work in Section 2 in order to highlight progress and potential improvements in this area. In Section 3, we discuss the implications of network theory, game theory and chaos theory on modeling human systems – specifically, how they influence patterns in the data. Section 4 presents our method, describing how it addresses the issues brought up in Section 3. Section 5 describe experimental results with cryptocurrency networks and the Eurovision context to validate our proposed approach. Finally, Section 6 concludes the paper.

2 Related Work

We briefly review research efforts relevant to the classification of complex human behavior, namely early violence detection and graph neural networks. Our work is built upon critical insights from these fields.

One of the more successful fields of research in human network classification is early violence detection (EVD). The main approach is to identify a few critical factors associated with geopolitical instability and make predictions with logistic regression and Bayesian methods [34], or other machine learning methods [36]. Interestingly, this research has identified network-theoretic power structures as most relevant to conflict [11,37], and has shown outstanding effectiveness: 80% accuracy for two-year forecasts. EVD brings two critical insights to graph classification for human systems. First, good data on the power structures of the systems is the most relevant for global predictions [34]. Second, the dynamics of systems composed of larger organizations tend to change slower and more predictably than systems of individuals [35]. Hence, to analyze a highly-dynamic, decentralized system which does not exhibit explicit power structures, we would want to 1) identify the implicit power structures and 2) make such an identification efficient computationally in order to keep apace with changes. Our method seeks to do precisely that.

One approach to automatically learning structure in graphs so as to classify them with machine learning is the emergent field of graph neural networks (GNN) [32]. Since all permutations of the orderings of nodes and edges produce isomorphic graphs, tabular representations of graphs for consumption by standard machine learning techniques have proven largely unsatisfactory. GNNs relax the notion of isomorphism in favor of smoother metrics associated with graph structure that can become data points for a standard vector representation [15]. The idea is to use convolution and pooling to extract these permutation-invariant properties [5,39,41,18,21,40,38], where convolution creates embeddings for each node through embeddings of their neighbors, and pooling essentially extracts information from subgraphs. These are certainly movements in the right direction towards automatic feature extraction for graph classification. However, there are complications for dynamic networks. For example, a particular pooling operator will wax and wane in significance as different power structures emerge throughout the system. Most methods will be designed to focus on the most significant operator. But for human systems, as we will see, it is not just the existence of a certain topology or feature that leads to certain system characteristics, but their absence, breakdown, or drop below a certain level. Hence, it becomes important to track the degree to which many structures hold throughout a trajectory.

Each of these methods makes valuable contributions to graph classification. The EVD community identified critical graph structures associated with what can be considered improbable events, such as wars. The GNN community has developed automatic feature detecting methods for computing these structures, but they tend to be computationally expensive and to overfit to past data structures in highly dynamic environments. They also fail to take into account *apriori* critical game theoretic principles. Our method seeks to address these shortcomings through an unified approach.

3 Synthesizing Perspectives

Following Easley and Kleinberg's lead, we explore how elements of network/graph theory, game theory and chaos theory affect the data that is used to quantify them and can be brought together into a unified computational model.

3.1 Graph Theory: Recursive Relationships

Graphs are widely accepted as the fundamental data structure for modeling human systems, as they can represent the rich, recursive relationships that characterize strategic interactions. GNNs can learn non-random structures from graphs that correlate with given classification labels. For human systems, learning non-random patterns, or even highly complex structure, is not sufficient, because human societies exhibit a balance of order and diversity over time that is necessary for maximal adaptation [12,19] (sometimes described as the edge of chaos [2], or degrees of randomness [33], or bounded rationality [1]). For dynamic systems the structure of previous graphs may be irrelevant to the future, or misleading. Game theory allows us to understand the structure, diversity, and randomness in the data.

3.2 Game Theory: Cooperation and Conflict

One of the fundamental ways in which human systems differ from physical systems is in terms of interdependence of action. The notion of independence does not hold. Furthermore, the type of interaction within human systems results in different data structures, as follows:

- **Pure Conflict: Randomness**. In pure conflict (sometimes referred to as zero-sum games), the outcomes for two agents i and j are inversely related. In these games, there exists a technical solution. Player i is guaranteed a specific payoff by playing the $max - min$ strategy, leading, ironically, to i being indifferent as to what j does [23,22]. For most non-trivial zero-sum games, this strategy thus involves randomization. While this may first appear somewhat counter-intuitive, consider the thought process of the well-known game of rock-paper-scissors. The best *apriori* strategy is to randomly pick one of the three options. Hence, data from zero-sum games will be random in nature as the past has not effect on i's choice of strategies.
- **Pure Cooperation: Structure**. Games of pure cooperation are such that the interests of all players are totally aligned. That is, agents i and j will either both win or both lose depending on their actions. Higher payoffs induce i to employ any means of communication to coordinate with j's actions. Agent i has no incentive to be deceptive, nor indifferent [33], but to want to send clear, unambiguous signals to j. The data coming from these signals, hence, will be structured. Past states will be good predictors of future or unknown states. While there is no technical solution to these problems, there is a practical one, namely tacit communication. Speech recognition, text recognition, and much of natural language processing deal with games of pure cooperation.
- **Mixed-Motive: Structure and Randomness**. The most interesting games, analytically, are games in which players' interests are partly aligned and partly at odds [30]. These problems have neither technical, nor practical solutions [33]. Since these scenarios involve both cooperation and conflict, it follows that data from these interactions will be both structured (from cooperation) and random (from conflict).

We note that bounded rationality, i.e., the fact that humans have finite perceptive and computational capacities [1], affects some of these interactions and data structures. Indeed, pure conflict will not be perfectly random; pure cooperation will not be perfectly structured; nor will agents behave perfectly rationally in mixed-motive games. This allows for both randomness and structure that is not theoretically justified. Furthermore, actions in human systems are clearly sequential. Agents continually adapt, but have finite ability to capture the implications of their planned actions. Hence, the recent past or unusual events from the distant past may inform their decisions. We note that the concepts of cooperation and conflict are fundamental concepts that appear in various guises in different domains.

3.3 Chaos Theory: Evolutionary Change

An evolutionary system introduces the dimensions of time and natural selection to the game theoretic and network theoretic principles discussed above. The effect is paradoxical. Natural selection induces competition while competition induces cooperation. We begin by exposing a fundamental cycle, then discussing how it might affect our data. The forces at play include natural selection, valuation, and self-organization [29]. The forces of natural selection compel agents to increase their fitness. Hence, there is a natural conflict between agents. However, organized groups of cooperators will be stronger than unorganized individual competitors. Hence, the competition spurs cooperation. Through valuation, agents self-organize into societal structures which appear to increase the fitness of the members with respect to the larger population. At some point in this self-organizing, there will be agents, through valuation, who realize they will be the "losers" in the new structure. Evolutionary pressures will then cause them to break off and organize along possibly novel, but not necessarily more complex, structures they perceive to be more fitness-maximizing. Often this means reconfiguring existing dimensions of value/power [28] that will eventually scale up to oust the current dominant structure (e.g., disruptive technologies [7]). Note that, ultimately, these structures are formed in order to maximize fitness. Hence, there will be a constant rise and fall of network structures that compete with one another for increased fitness, in other words, an ebbing and flowing of cooperation and conflict. This process will be chaotic in the mathematical sense, i.e., partly structured and partly random, emanating from cooperation and defection, respectively.

4 Graph Representation

We propose a method for graph classification of dynamic networks, that attempts to synthesize the aforementioned relevant constructs from graph, game and chaos theories. Network theory allows us to quantify the structural properties of a graph, such as node centrality, overall graph connectivity, node preference for linking with highly connected or less connected nodes, node preference for linking with neighbors of neighboring nodes, etc. For each of these properties there may be high degrees of structure, random structure, and diverse

Property	Level	Description
Fairness (*fair*) [17]	Node	Degree of a node's accuracy in judging goodness
Goodness (good) [17]	Node	Degree to which a node is trusted by fair agents
Eigenvector Centrality (*eigen*) [4,3]	Node	Measure of a node's status proportionate to the status of its neighbors
Clustering Coefficient (*clust*) [31,27,8,10]	Node	Degree to which a node's neighbors are connected to each other
Average Clustering Coefficient (*aclust*)	Graph	Average *clust* of all nodes
Assortativity (*assort*) [24]	Graph	Degree to which highly connected nodes connect to each other
Gini (*gini*) [6]	Graph	Degree to which influence is unequally distributed among a population
s-metric (*s-metric*) [20]	Graph	Sum of the pairwise products of node degrees

Table 1: Graph Structural Properties

structure at various time steps based on the level of trust and distrust, and actions of the past. Adaptation in evolutionary environments shows that the relationship between the graph class label and the set of structural properties may vary through time, such that the values of each feature evolve over time and the prediction for graph G_t at time t is a function of information from all, or a subset of, graphs $G_1, \ldots, G_{t-1}$. Hence, we must capture both static and dynamic properties.

4.1 Static Properties

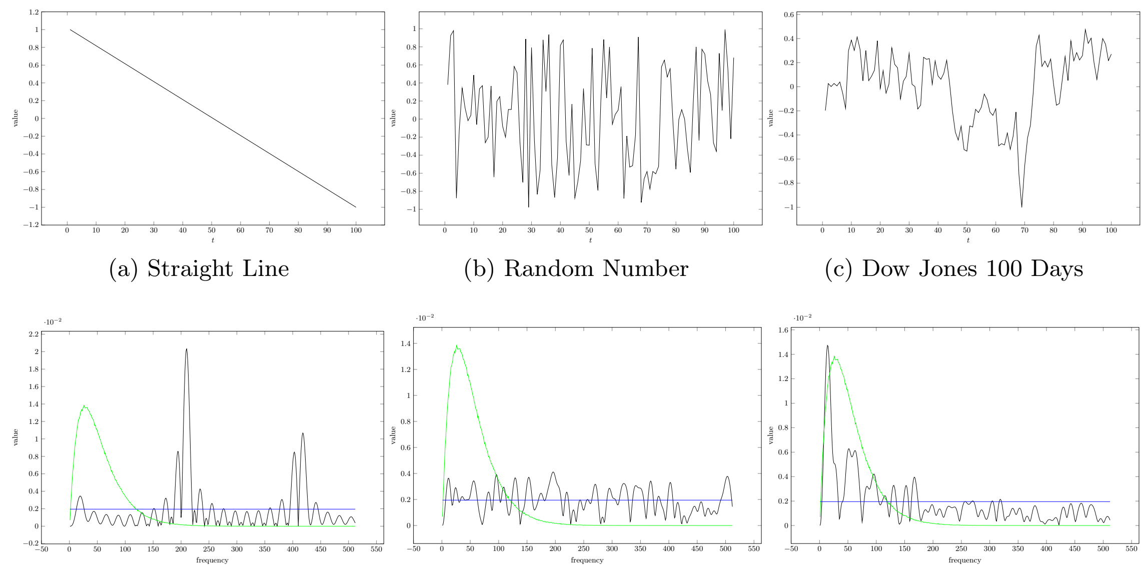
Here, we select a few representative network properties, inspired from the literature, to characterize graphs at each time step, as shown in Table 1. These properties attempt to capture game-theoretic elements of both cooperation and conflict. A node-level property, such as eigenvector centrality, can be transformed into a graph-level property by computing the correlation between the list of values of that property and the corresponding list of values of some application-specific notion of node influence. Intuitively, if the correlation is high, then we may assume that the associated property generates high influence, and *vice versa*. We denote by S the set of structural properties.

4.2 Dynamic Properties

The trajectories of the feature values are chaotic, i.e., aperiodic, noisy, and most correlated with recent time steps. This makes fitting a curve to the corresponding time-series and projecting it forward less effective. We do know, from Section 3.3 that individual trajectories are generally not highly structured, nor highly random for long periods of time. In the former case, those not benefiting from the structure will construct an alternate structure. In the latter case, the disorganized structure will eventually be organized to compete with whatever structure is dominant at the time.

Just as the graph is a higher-order representation of human interaction, the frequency domain becomes a higher-order representation of a dynamic distribution. We propose to measure the structure of a trajectory by mapping the corresponding dynamic feature values to the frequency domain by means of the Fast Fourier Transform (FFT). We then estimate the level of chaos by fitting various distributions to the computed FFTs. To illustrate the value of this approach, consider Figure 1, which depicts three different time series and their corresponding FFTs. In addition to the FFTs, the graph show uniform (blue) and gamma (green) distribution fits. The first distribution (1a) exhibits high structure, monotonically decreasing at a constant rate. Its FFT (1d), has pronounced amplitudes in two narrow frequency ranges, and as expected does not fit either uniform nor gamma distributions very well. The second distribution (1b) is random, with no underlying structure. Its FFT (1e), exhibits similar randomness, and as such fits an uniform distribution. Finally, the third distribution (1c) represents the Dow Jones daily closing levels over a period of 100 days. It exhibits chaotic behavior (i.e., both some structure and some randomness), and its FFT (1f) shows a range of pronounced amplitudes at the higher frequencies, as well as good fit to a gamma distribution, as expected for such a chaotic trajectory.

We denote by $FFT(S_i, t-w:t)$ the FFT of structural property S_i over the time window $[t-w, t]$, where w is an application-dependent hyperparameter. After fitting each FFT for S_i to a gamma distribution, we



(a) Straight Line (b) Random Number (c) Dow Jones 100 Days

(d) Straight Line FFT (e) Random Number (f) Dow Jones 100 Days

Fig. 1: Different Time-Series and their Corresponding FFTs

compute a goodness-of-fit measure D_i. We denote by D the set of all of these measures. In addition, we track the interplay of the structural properties' trajectories by computing a multidimensional Fourier Transform over them, denoted by $MFFT(t-w:t)$. As with the individual trajectories, the resulting function is fit to a gamma distribution. It is also fit to a log normal and an uniform distributions. We expect the fit of $MFFT(t-w:t)$ to not vary from a gamma distribution for long. Goodness-of-fit values are computed and denoted by $MD = \{MD_1, MD_2, MD_3\}$.

4.3 Graph Transformation

We can now transform our graph time-series into tabular form by transforming each graph G_t as follows.

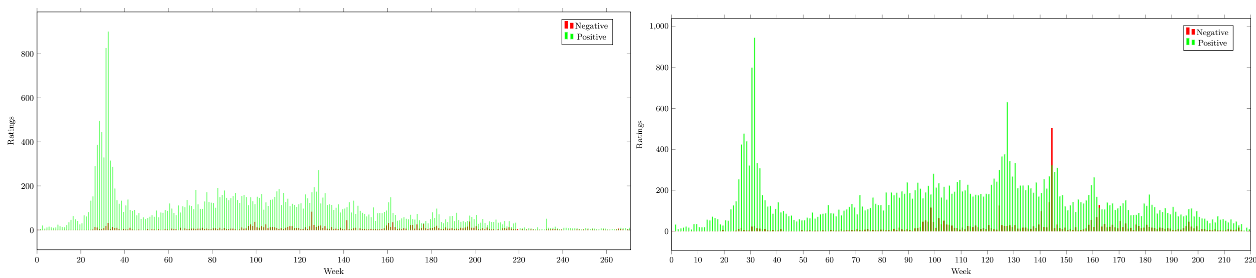
1. Extract the values of the properties in S from G_t
2. For each property S_i in S
   (a) Compute $FFT(S_i, t-w:t)$.
   (b) Extract the values of goodness-of-fit D_i of $FFT(S_i, t-w:t)$ to the Γ distribution
3. Compute $MFFT(t-w:t)$
4. Extract the goodness-of-fit values $MD = MD_1, MD_2, MD_3$ of $MFFT(t-w:t)$ to the gamma, the uniform, and the log normal distributions, respectively

As in all classification tasks, each graph G_t is further labelled with a value from the corresponding application-specific set Y of possible classes.

Note that the above transformation allows our applications to benefit from being framed as *graph* classification problems, as opposed to just using a feature vector of other related data to classify, or make prediction, at each time step. The graph step allows to leverage network structure and interaction. Indeed, the feature vector does not consist of aggregated data (e.g., total attacks, number of attacks, etc.) but of graph centrality measures that represent levels of organization and randomness, which are hard to measure with aggregated (non-graph) data.

5 Experimental Results

We demonstrate the applicability of our approach in two different domains of human interactions, as follows.



(a) BTC Alpha (b) BTC OTC

Fig. 2: Weekly Positive and Negative Ratings

Class	BTC Alpha	BTC OTC
`first`	$[0]$	$[0, 2]$
`second`	$[1, 3]$	$[3, 8]$
`third`	$[4, 7]$	$[9, 16]$
`fourth`	$[8, 82]$	$[17, 503]$

Table 2: Target Classes and Corresponding Ranges of Negative Rating Values

5.1 Cryptocurrency Networks

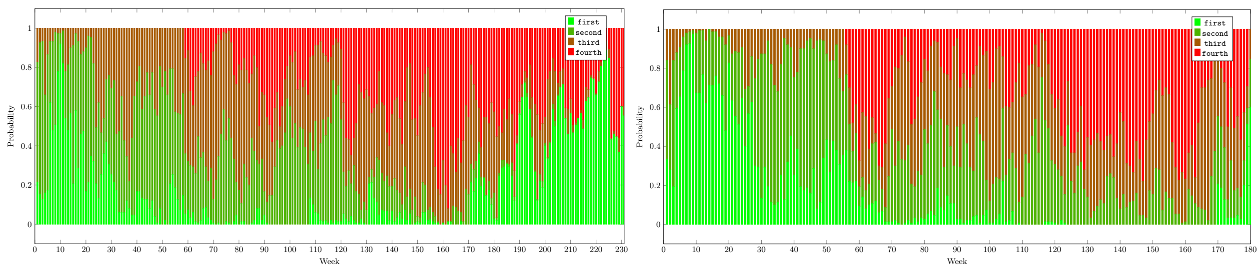
We consider two cryptocurrency networks, namely the Bitcoin Alpha (BTC Alpha) and the Bitcoin OTC (BTC OTC) data sets [17,16]. The BTC Alpha and OTC platforms allow users to rate each other based on level of trust. This is crucial in the Bitcoin community since the appeal to the currency is the decentralized nature of transaction management. No governing figure regulates the cryptocurrency, so users self-regulate potential fraudulent or risky behavior by rating each other in terms of trustworthiness. In this context, the generic game theoretic terms “cooperation” and “conflict” introduced in Section 3.2 are embodied as trust and distrust, respectively, for Bitcoin users.

The BTC Alpha data set contains 24,186 ratings made by 3,783 users over roughly a six-year period. The BTC OTC data set contains 35,592 ratings made by 5,881 users, also over a six-year period. The percentage of negative ratings in each set is 7% and 11%, respectively. In this data set, agent interactions entail rating each other. A user may rate another user only once, and the rating carries through time. Hence, the graph at time t also includes all ratings made prior to t. Ratings normally range in the interval $[-10, 10]$, with positive values corresponding to the level of trust and negative values to the level of distrust. We map ratings to $\{-1, 1\}$ depending on whether they were positive or negative, respectively. In order to obtain a sequence of graphs (i.e., a time-series), we group ratings by week, resulting in 271 graphs and 220 graphs for the BTC Alpha and BTC OTC data sets, respectively. Figure 2 shows the number of positive and negative ratings per week (2a).

The nodes of the graphs are the users of the bitcoin community, and there is an edge between node i and node j in graph G_t if user i has rated user j at some time $t' \leq t$. For the purposes of computing the set S of structural properties, we define a node's influence simply as its aggregated ratings, i.e., the sum of ratings it has received from other nodes in the network. Each graph in the series is labeled based on the percentage of weekly negative ratings, using four discrete classes, $Y = \{\texttt{first}, \texttt{second}, \texttt{third}, \texttt{fourth}\}$ corresponding to the four quartiles, as shown in Table 2. The class distributions are depicted in Figure 3.

5.2 Eurovision Contest

Eurovision is one of the longest-running international songwriting competitions, organized annually since 1956. Each participating country (mainly European) submits an original song that is performed on live television and radio, with competing countries casting votes for the other countries' songs to determine a winner. Outcomes and statistics from annual contests are collected on the Eurovision's official web site (eurovision.tv). However, they are difficult to exploit directly. Borsteinn Adalsteinsson has scraped and cleaned



(a) BTC Alpha

(b) BTC OTC

Fig. 3: Class Probability Distributions

the data using an R script, and made it publicly available for download and analysis from his web site (https://data.world/rhubarbarosa/eurovisionvotingstats). A number of papers about Eurovision have been published mainly focusing on such issues as culture and bias, which our proposed graph-based system is intended to leverage.

The data set contains 20,081 voting actions (i.e., number of points country a assigns to country b for event at time t) made by 52 countries across 81 contests. Voting values range from 1 to 12, and are based on a complicated voting scheme of jury and televoting. The points were normalized to the range $[0, 1]$, scaled by the maximum value. Note that semi-final events were considered to be separate events. For certain years (e.g., 2008), there were three events (first semi-final, second semi-final, grand final); others had two events (semi-final, grand final); and others just had a single grand final.

The nodes of the graphs are the participating countries, and there is an edge between node i and node j when country i assigns points to the song of country j. Since all countries vote for all other countries, all of the graphs are complete. For the purposes of computing the set S of structural properties, we define a node's influence simply as the total number of points it received from all other nodes in the graph. Each graph is labeled by the world region in which the winning country is found, namely Balkan (1), Baltic (2), East (11), West (59), and Independent (8).

5.3 Experimental Setup and Results

For the set D, we compute discrete FFTs, as well as gamma histograms, of 1,024 bins. Both the transforms and the histograms are normalized to sum to 1. We then use the root mean-squared error (RMSE) between each FFT and its fitted gamma distribution as the goodness-of-fit measure, i.e., $D_i = RMSE(FFT(S_i, t-w : t), \Gamma_i)$. The MD_i values are similarly computed, with uniform and log normal fits in addition to gamma.

As the graphs form a time-series, each graph G_t is first used as a test instance, whose classification is predicted based on the current training set. The pair (G_t, Y_t) is then added to the training set, and in turn participates in the prediction of the classification of G_{t+1}. To allow for comparison with other algorithms, we set the training set size to T. Hence, the training set for G_t is $\{G_{t-T}, Y_{t-T}), \ldots, (G_{t-1}, Y_{t-1})\}$. We use the first $T-1$ graph instances exclusively for training, as a kind of warm start. Test results are then reported for the remaining graphs.

Hyperparameters T and w were set through experimentation. For the BTC Alpha and BTC OTC data sets, $T = 40$ and $w = 30$ (for $t < 30$, $w = t$). Test results are thus reported for 231 graphs for the BTC Alpha data set and for 180 graphs for the BTC OTC data set. For the Eurovision dataset, $T = 30$ and w starts at 10 but is then dynamically incremented by 1 at each time step. Test results are reported for 51 graphs.

Since our approach essentially transforms a graph time-series problem into a standard classification task by encoding the time-series data into each element of the series via the feature sets D and MD, it is possible to use any classification learning algorithm with the input G_t to predict Y_t. As the transformed data still exhibits a non-linear relationship with Y, we use a Bagging Ensemble method featuring 10 Random Forest classifiers.

For human systems, it is often the case that the naïve function consisting of simply assigning to G_t the classification of G_{t-1} produces better accuracy than randomly guessing, especially in systems of large bureaucracies where policy changes are difficult [35]. This pattern tends also to arise with more aggregated time steps, e.g., the Dow Jones Industrial Average generally moves in the same direction from year-to-year,

	BTC Alpha	BTC OTC	Eurovision
Bagging	.479	.507	.569
LSTM (all)	.463	.422	.431
LSTM (S)	.429	.388	.373
Naïve	.446	.439	.451

Table 3: Accuracy Results

while its monthly and daily trends exhibit frequent reversals. We seek relatively low-granularity predictions (i.e., weekly for the BTC data sets) that outperform the naïve ones. We also compare performance to a long short-term memory (LSTM) recurrent neural network [13], with cross-entropy loss, adam optimizer, one hidden layer of 100 neurons, and trained with 100 epochs (cryptocurrency networks) or 200 epochs (Eurovision contest). As expected, the inputs to the LSTM is a sequence of T vectors, $G_{t-T}, G_{t-T+1}, \ldots, G_t$ defined over the same input features.

Table 3 reports on the accuracy for all three of our data sets. It compares the results of our approach (Bagging) and the LSTM with the naïve approach. The results for Bagging are averaged over 10 runs to account for the potential effect of random states in the FFTs. For the LSTM, two cases are shown, one where only S is used as input, and one where S, D and MD (all) are used as inputs.

Note that the sets D and MD are akin to derivatives of S. So, the subset S describes position of the graph structure, while D and MD capture its velocity and direction of change. As shown in Table 3, while position at t is a good indicator of position at $t + 1$ (given the performance of the naïve classifier in this environment), a more precise prediction comes of including the velocity and direction of change. Indeed, Bagging performs better than the naïve classifier, for all three data sets.

The LSTM does not perform significantly better than the naïve classifier at any point of the sequence for the Alpha OTC data set. It actually underperforms the naïve classifier. The LSTM is not able to adjust when the class changes frequently from week to week, namely weeks 100-215 for BTC OTC, and the first 150 weeks for BTC Alpha. In these weeks, the sequence of values for D and MD are more chaotic. The LSTM is designed to identify patterns that will repeat. There is no regular cycle here to associate with the class labels, as the LSTM is designed to do. Hence, our graph transformation approach together with Bagging classification outperforms the LSTM and the naïve classifier. Code and data sets are available on GitHub (link will be provided upon acceptance).

6 Conclusions

We have proposed an approach to graph time-series classification based on an encoding of the evolutionary behavior of graph structural properties by mapping to the frequency domain via fast Fourier transforms. The method captures critical power structures in the graph, together with critical features of their trajectories through time. The time-series classification problem is thus transformed into a standard tabular classification task. While we cannot claim that our results generalize to all human graph data, they do corroborate our conjecture that the data sets S, D and MD work in synergistic fashion, and show promise as an effective graph classifier for dynamic networks.

Our results show modest, but consistent, improvement over the naïve classifier. Given the difficulty of predicting chaotic systems, this is to be expected. The comparison with the LSTM is probably most insightful. The rigid time-series structure required by the LSTM does not adapt to chaotic environments. It looks for a very specific sequence of values, something that is practically guaranteed not to be the case in evolutionary environments. In fact, the more epochs, the worse it does. Furthermore, the addition of D and MD to S does not enhance performance enough to bring it on par with the naiïve classifier for BTC OTC. While it is clear that we cannot expect to predict chaotic systems with exactness, our claim is that our approach can predict them to a degree that will be useful. LSTM-type algorithms are geared towards exploiting structure and tend to break down with ambiguity. This limitation also extends to current GNN implementations. While GNN can automatically create features similar to our set S, it is doubtful whether it can extract anything resembling the sets D and MD, unless programmed to do so. Future work will address these issues.

Finally, our comparison of S, and D and MD to position of graph structure, and velocity and direction of change, respectively, does beg the question of what acceleration would contribute. We submit that the same procedure we applied to S in order to derive D and MD may be applied to D and MD to derive a higher-level representation of change. We leave this as a future line of research.

References

1. Arthur, W.B.: Challenges for human, social, cultural, and behavioral modeling. The American economic review **84**(2), 406–411 (1994)
2. Axelrod, R., Cohen, M.D.: Harnessing Complexity: Organizational Implications of a Scientific Frontier. Basic Books, New York (2000)
3. Bonacich, P.: Power and centrality: A family of measures. American Journal of Sociology **92**(5), 1170–1182 (1987)
4. Bonacich, P., Lloyd, P.: Calculating status with negative relations. Social networks **26**(4), 331–338 (2004)
5. Bronstein, M.M., Bruna, J., LeCun, Y., Szlam, A., Vandergheynst, P.: Geometric deep learning: going beyond euclidean data. IEEE Signal Processing Magazine **34**(4), 18–42 (2017)
6. Ceriani, L., Verme, P.: The origins of the gini index: extracts from variabilità e mutabilità (1912) by corrado gini. The Journal of Economic Inequality **10**(3), 421–443 (2012)
7. Christensen, C.M.: The innovator's dilemma: when new technologies cause great firms to fail. Harvard Business Review Press (2013)
8. Costantini, G., Perugini, M.: Generalization of clustering coefficients to signed correlation networks. PloS one **9**(2), e88669 (2014)
9. Easley, D., Kleinberg, J.: Networks, crowds, and markets: reasoning about a highly connected world. Cambridge University Press (2010)
10. Fagiolo, G.: Clustering in complex directed networks. Physical Review E **76**(2), 026107 (2007)
11. Goldstone, J.A., Bates, R.H., Epstein, D.L., Gurr, T.R., Lustik, M.B., Marshall, M.G., Ulfelder, J., Woodward, M.: A global model for forecasting political instability. American Journal of Political Science **54**(1), 190–208 (2010)
12. Heath, B.L., Hill, R.R.: Some insights into the emergence of agent-based modeling. In: Agent-based modeling and simulation, pp. 32–44. Springer (2014)
13. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation **9**(8), 1735–1780 (1997)
14. Kaysen, C.: A revolution in economic theory? The Review of Economic Studies **14**(1), 1–15 (1946)
15. Kriege, N.M., Johansson, F.D., Morris, C.: A survey on graph kernels. Applied Network Science **5**(1), 1–42 (2020)
16. Kumar, S., Hooi, B., Makhija, D., Kumar, M., Faloutsos, C., Subrahmanian, V.: Rev2: Fraudulent user prediction in rating platforms. In: Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining. pp. 333–341. ACM (2018)
17. Kumar, S., Spezzano, F., Subrahmanian, V., Faloutsos, C.: Edge weight prediction in weighted signed networks. In: 2016 IEEE 16th International Conference on Data Mining (ICDM). pp. 221–230. IEEE (2016)
18. Lee, J., Lee, I., Kang, J.: Self-attention graph pooling. In: International conference on machine learning. pp. 3734–3743. PMLR (2019)
19. Levy, S.: Artificial life: A report from the frontier where computers meet biology (1992)
20. Li, L., Alderson, D., Doyle, J.C., Willinger, W.: Towards a theory of scale-free graphs: Definition, properties, and implications. Internet Mathematics **2**(4), 431–523 (2005)
21. Ma, Y., Wang, S., Aggarwal, C.C., Tang, J.: Graph convolutional networks with eigenpooling. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. pp. 723–731 (2019)
22. von Neumann., J.: Zur theorie der gesellschaftsspiele. Mathematische Annalen **100**, 295–320 (1928)
23. von Neumann, J., Morgenstern, O.: Theory of games and economic behavior. Princeton University Press (1953)
24. Newman, M.E.: Assortative mixing in networks. Physical review letters **89**(20), 208701 (2002)
25. Nowak, M., Sigmund, K.: The evolution of stochastic strategies in the prisoner's dilemma. Acta Applicandae Mathematicae **20**(3), 247–265 (1990)
26. Nowak, M.A.: Natural cooperation. In: Neurosciences and the Human Person: New Perspectives on Human Activities. vol. 121, pp. 237–241. Pontifical Academy of Sciences, Vatican (2013)
27. Onnela, J.P., Saramäki, J., Kertész, J., Kaski, K.: Intensity and coherence of motifs in weighted complex networks. Physical Review E **71**(6), 065103 (2005)
28. Porter, M.E., et al.: What is strategy? (1996)
29. Radzicki, M.J.: Institutional dynamics, deterministic chaos, and self-organizing systems. Journal of economic issues **24**(1), 57–102 (1990)
30. Rapoport, A., Chammah, A.M., Orwant, C.J.: Prisoner's dilemma: A study in conflict and cooperation. University of Michigan press (1965)
31. Saramäki, J., Kivelä, M., Onnela, J.P., Kaski, K., Kertesz, J.: Generalizations of the clustering coefficient to weighted complex networks. Physical Review E **75**(2), 027105 (2007)
32. Scarselli, F., Gori, M., Tsoi, A.C., Hagenbuchner, M., Monfardini, G.: The graph neural network model. IEEE transactions on neural networks **20**(1), 61–80 (2008)
33. Schelling, T.C.: The strategy of conflict. prospectus for a reorientation of game theory. Journal of Conflict Resolution **2**(3), 203–264 (1958)

34. Schrodt, P.: Technical forecasting of political conflict. Indiana University Workshop in Methods (2020)
35. Schrodt, P.A.: Early warning of conflict in southern lebanon using hidden markov models. In: American Political Science Association (1997)
36. Schrodt, P.A.: Seven deadly sins of contemporary quantitative political analysis. Journal of peace research **51**(2), 287–300 (2014)
37. Valentino, B.A.: Final solutions. Cornell University Press (2013)
38. Wei, L., Zhao, H., Yao, Q., He, Z.: Pooling architecture search for graph classification. In: Proceedings of the 30th ACM International Conference on Information & Knowledge Management. pp. 2091–2100 (2021)
39. Xu, K., Hu, W., Leskovec, J., Jegelka, S.: How powerful are graph neural networks? arXiv preprint arXiv:1810.00826 (2018)
40. Yao, Y., Joe-Wong, C.: Interpretable clustering on dynamic graphs with recurrent graph neural networks. In: AAAI. pp. 4608–4616 (2021)
41. Ying, Z., You, J., Morris, C., Ren, X., Hamilton, W., Leskovec, J.: Hierarchical graph representation learning with differentiable pooling. Advances in neural information processing systems **31** (2018)