

Topology-Informed Information Dynamics Modeling in Cyber-Physical-Social System Networks

Yan Wang
Georgia Institute of Technology, Atlanta, GA 30332, USA
yan-wang@gatech.edu

Abstract

Cyber-physical-social systems (CPSS) are physical devices that are embedded in human society and possess highly integrated functionalities of sensing, computing, communication, and control. CPSS rely on their intense collaboration and information sharing through networks to be functioning. In this paper, topology-informed network information dynamics models are proposed to characterize the evolution of information processing capabilities of CPSS nodes in networks. The models are based on a mesoscale probabilistic graph model, where the sensing and computing capabilities of the nodes are captured as the probabilities of correct predictions. A topology-informed vector autoregression model and a latent variable vector autoregression model are proposed to model the correlations between prediction capabilities of nodes as linear functional relationships. A hybrid Gaussian process regression model is also developed to capture both the nonlinear spatial and temporal correlations between nodes. The new information dynamics models are demonstrated and tested with a simulator of CPSS networks. The results show that the topological information of networks can improve the efficiency in constructing the time series models. The network topology also has influences on the prediction capabilities of CPSS.

Keywords:

Complex systems, Gaussian process regression, Information diffusion, Probabilistic graph model, Vector autoregression

1 Introduction

Cyber-physical-social systems (CPSS) are physical devices that are embedded in the human society and possess highly integrated functionalities of sensing, computing, communication, and actuation. They are the essential elements for smart home, smart office, smart manufacturing, personalized medicine, autonomous transportation, and many other applications. Although human is regarded as a component of the complex sociotechnical systems at large, our focus of discourse is engineering design of physical systems that interface with human beings.

In CPSS networks, decision makings are done at the individual node level and enhanced by intensive information sharing, as more high-quality information leads to better decisions. The quality of information can be measured by accuracy, consistency, completeness, timeliness, precision, and interpretability [1]. CPSS nodes also interact with and change the surrounding environment via their control function, which in turn affects their neighboring nodes. As a result, the level of interdependencies among CPSS nodes for their collected information is very high. There is a strong need to understand the deep information dependency between CPSS devices in a network. The knowledge about the behaviors of CPSS networks can help us to design more reliable and dependable systems.

Existing research on information diffusion focused on traditional computer networks and social networks. There is a lack of studies of information dynamics in CPSS networks which have unique characteristics. First, the topology of a CPSS network is complex with numerous weak and dynamically changing connections due to its ad hoc nature, e.g., in vehicle networks and manufacturing plants. Second, the nodes are heterogeneous CPSS devices with various functions. In particular, they have the sensing and control capabilities which do not exist in traditional computer networks. Information sharing is intensive and decisions of what and when to share are made locally by individual nodes. Thus strong correlations can exist in the shared information. Third, the types of information generated and shared among CPSS nodes can vary greatly. In contrast, existing studies of information diffusion in traditional computer networks and social networks focus on the passive propagation or simple forwarding of specific information, such as virus, keywords, and new concepts. To properly design the architecture of CPSS networks with good adaptability and scalability, it is important to understand the evolution of information in

CPSS networks as a system of systems, as well as the sensitivity of the system and device behaviors with respect to the information propagation.

In a dynamically evolving CPSS network, the effects of information generation and sharing need to be quantified and analyzed so that the long-term behaviors of the network can be predicted, which is useful in systems design. To model the effects of information exchange, recently a data-driven information dynamics modeling approach [2] was proposed to analyze the information interdependency among CPSS nodes and subnetworks. The model keeps track of the probability that each node detects and predicts the true state of its environment along time with its sensing, computing, and communication functions. The interdependency between the probabilities of different nodes was explicitly modeled with copulas and vector autoregression (VAR) models. The proposed linear models provide the insight of how nodes have an influence on each other on the prediction capabilities when information is exchanged and how the behavior of networks evolves dynamically. This dynamics analysis is useful for the design of an open system with good adaptability, where the topology of ad hoc networks changes with nodes continuously added or removed. These models predict the global trend of information dynamics well. Nevertheless, the black-box approaches for model fitting require a large amount of training data as the number of nodes and thus the number of model parameters increase. Furthermore, nonlinear relationships between state variables associated with the nodes need to be captured with more complex models. In this paper, a gray-box modeling approach is taken. A topology-informed VAR model is proposed so that the prior knowledge of correlation between the predictions as a result of direct connections between nodes can be applied. This simplifies the modeling process such that the number of coefficients or model parameters to be fitted is reduced. The amount of training data can be reduced. The reduction is significant when the network connection is sparse. To further capture the correlation among the nodes' prediction capabilities, a latent variable VAR model is proposed where hidden state variables are introduced. The observed changes in prediction capabilities are caused by the interactions among hidden state variables. In addition, a topology-informed Gaussian process regression (GPR) modeling approach is developed to predict the time series of information dynamics, where nonlinear temporal and spatial correlations are captured. A hybrid kernel function is developed with continuous time and discrete node labels as the inputs. The node labels are specifically designed based on network topology such that the adjacency

information is applied to infer the similarity of information sources between nodes to improve the accuracy of the kernel function.

The proposed information dynamics modeling approaches are based on a generic probabilistic graph model of networks ([4], [5]), where information exchange and processing at nodes are modeled at the mesoscale. In the probabilistic graph model, the sensing and computing capabilities of each node are characterized by a prediction probability, whereas the communication capabilities between nodes are characterized by pairwise reliance probabilities. The prediction probability measures how well a node can gather information and make sound judgment. The reliance probabilities capture the extent of influences for one node to another via information exchange. In the proposed information dynamics models, the dynamics is modeled based on the evolution of prediction probabilities as a result of information exchange, instead of explicitly modeling the information that is being exchanged as in other information diffusion models. The evolution of prediction capabilities of nodes is captured in the time series models, where the prediction accuracies of nodes are influenced by each other, given that the decision of each node is made based on the information gathered from itself as well as its neighboring nodes. To train the model parameters, simulations are performed based on the probabilistic graph model for demonstration.

In the remainder of this paper, the background of CPSS design, existing models of information diffusion in networks, and the probabilistic graph model are introduced in Section 2. The proposed information dynamics models are described in Section 3. The models are demonstrated and tested with a CPSS simulator in Section 4.

2 Background

In this section, an overview of relevant work on CPSS design at the system-of-systems level is given. The models of information diffusion in networks are reviewed. The background of the probabilistic graph model which the proposed information dynamics models are based on is also provided.

2.1 CPSS System-of-Systems Level Design

The design of CPSS architecture at the system-of-systems level needs to incorporate several factors. First, given the evolution nature of cyber and physical technologies, adaptability that

enables the capabilities of self-learning, self-organization, and context awareness is important to design open systems that can evolve along technology advancement [6][7]. Using new technologies as the augmentation to existing products can effectively enhance adaptability [8]. Second, the complexity of the CPSS has significantly increased from traditional products and devices. The CPSS products are connected through Internet of Things and heavily rely on data exchange from each other to realize their functions. Communication between devices plays a major role. Therefore, how to design systems of CPSS which have dependable communication is important. Reliable large-scale networked systems that do not fail are impossible to achieve. Resilient systems that can recover automatically from partial failures are more likely to be realized [4][5]. Third, the high-dimensional design space of CPSS includes not only the cyber and physical subspaces, but also the social subspace. Examples of the emerging research issues are how to design the modalities for human-system interaction [9], how to enable context awareness and personalized communication between CPSS and humans [10], and how to quantify trustworthy strategic relationships for information sharing [11]-[15].

Network connectivity is essential for CPSS. A standalone CPSS device cannot perform the functions which it is designed for. Compared to traditional products, the design of CPSS devices requires engineers to have better understanding of the system level behaviors, as well as the new methodology for the optimization at the network scale. Systems level modeling methods and tools have been developed for CPSS design and analysis, such as hybrid discrete-event and continuous simulations [16]-[18], inductive constraint logic programming [19] abductive reasoning [20], hybrid timed automaton [21], ontologies [22], information schema [23], UML [24], and SysML [25].

2.2 *Information Diffusion in CPSS Networks*

The information flow in computer networks and social networks has been studied by researchers. The propagation of information can be modeled in different ways. The most used approach is the epidemic model of networks, where transmission probabilities of virus between nodes are mainly used to model the speed of infection and the dynamics of outbreak and decay is captured with ordinary differential equations [26][27]. The epidemic model has been widely applied to study the propagation of keywords or phrases among blogs [28] and within social networks [29]. In the linear influence model [30], the propagation of information is modeled and

parameterized by the influences of individual nodes in the network. In the event-driven modeling approaches, the adoption of new information by nodes is characterized by discrete Poisson processes [31][32] or continuous hazard function [33].

Very limited work has been done in studying information diffusion in CPSS networks. Yagan et al. [34] studied the information transmission in the coupled social and physical networks according to the epidemic model. Lu et al. [35] investigated how to maximize the information diffusion in CPSS networks when nodes are connected probabilistically. Wang et al. [36] used a game theoretic approach in combination with the epidemic model at the system level to study the effects when nodes make local decisions of forwarding information to others. Yi et al. [37] defined the states of each node in both physical and social spaces for the epidemic model so that the mutual influence between social behavior spread and multimedia data transmission can be revealed. Wang [2][3] developed statistical approaches where copula and VAR models were used to model the information correlations among CPSS nodes.

Given that CPSS nodes possess local functions of sensing, computing, and decision making, we need to study not only how information propagates but more importantly how the information propagation directly affects these functions. In the proposed models, the propagation of information elements is not modeled directly. Rather, the dynamic effect of information diffusion, which is the change of the sensing and computing capabilities of CPSS nodes, is used to quantify the information dynamics. That is, the dynamic changes of those capabilities along time are directly captured in the proposed information dynamics models. The sensing and computing capabilities of CPSS are quantified based on a probabilistic graph model, as introduced in the next section.

2.3 Probabilistic graph model

A recently developed probabilistic graph model [5] for CPSS networks is the foundation of the proposed information dynamics models. In the probabilistic graph model, each node has its own sensing, reasoning, and communication units. As illustrated in Figure 1, there are probabilities associated with information gathering and exchange between nodes. For each node, there is a prediction probability indicating the capabilities of information gathering and reasoning. For each directed edge indicating information exchange, there are reliance probabilities associated with it. The two types of probabilities are defined as follows.

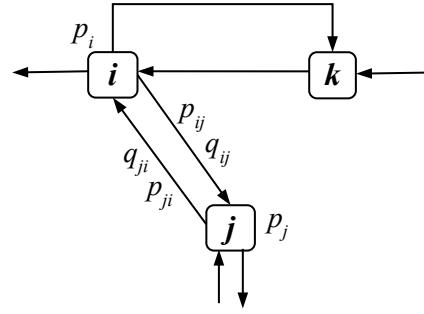


Figure 1. Probabilistic graph model of CPSS systems

The *prediction probability* that the k -th node detects the true state of world θ is

$$\mathbb{P}(x_k = \theta) = p_k \quad (1)$$

where x_k is the state variable. For convenience, we denote $q_k = 1 - p_k$. The information dependency between nodes i and j is modeled with *P-reliance probability*

$$\mathbb{P}(x_j = \theta | x_i = \theta) = p_{ij} \quad (2)$$

which is the probability that the j -th node predicts the true state of world given that the i -th node predicts correctly. Similarly, we also have *Q-reliance probability*

$$\mathbb{P}(x_j = \theta | x_i \neq \theta) = q_{ij} \quad (3)$$

because nodes could be negatively correlated, or miscommunication between nodes could exist. The reliance probability can be used to model reliability of communication between nodes, e.g. in moving vehicles' ad hoc wireless networks, data packet loss is not uncommon.

Therefore, different from the adjacency matrix in traditional graph model with binary “yes-or-no” edge connection topology, there are reliance probabilities associated with each pair of nodes in the new probabilistic graph model. If the communication channel from node i and node j is disrupted, both p_{ij} and q_{ij} are zeros.

The random state variables with binary values ($= \theta$ or $\neq \theta$) can be extended to multiple values or continuous. For instance, one sensor measures a value (e.g. temperature or flow speed) which follows some distribution, as in prediction probability. If there are a finite set of possible values $\{\theta_1, \dots, \theta_N\}$ for state variables. The prediction probability $\mathbb{P}(x_k = \theta_n)$ and reliance probability $\mathbb{P}(x_j = \theta_n | x_i = \theta_m)$, where $1 \leq m, n \leq N$, can be enumerated similarly. As a

simplification without loss of generality, in this paper we assume that the state variables take binary values.

The edges in the probabilistic graph are directional. The neighbors of each node can be further differentiated as *source* nodes or *destination* nodes, as illustrated in Figure 2. For one node, its source nodes are those sending information to this node, whereas the destination nodes are those receiving information from it. When receiving different cues from source nodes, a CPSS node can update its prediction probability to reflect its perception of the world. The aggregation of prediction probabilities sensitively depends on the rules of information fusion during the prediction update.

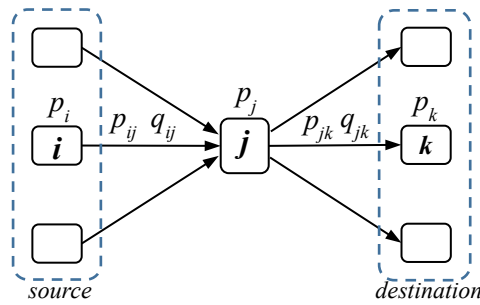


Figure 2. Source and destination nodes with respect to node j

If $P(x_k)$ and $P(x_k^c)$ denote the probabilities of a positive and a negative prediction from node k respectively, we define the *best-case* fusion rule as

$$P'(x_k) = 1 - (1 - P(x_k)) \prod_{i=1}^{M_P} P(x_i)(1 - P(x_k|x_i)) \prod_{j=1}^{M_N} P(x_j^c)(1 - P(x_k|x_j^c)) \quad (4)$$

where node k updates its prediction based on its own current prediction and those cues from its $M_P + M_N$ source nodes, out of which M_P of the source nodes provide positive predictions whereas M_N of them provide negative predictions, $P(x_k|x_i)$ indicates the probability that a positive message from node i leads to a positive prediction of node k , and $P(x_k|x_j^c)$ is the probability that a negative message from node j leads to a positive prediction of node k . Therefore, if any of the cues from the source nodes is positive, the prediction of the node is positive. Some variations of this fusion rules exist. For instance, the previous prediction from itself can be either included or excluded during the update.

Similarly, the *worst-case* fusion rule can be defined as

$$P'(x_k) = P(x_k) \prod_{i=1}^{M_P} P(x_i)P(x_k|x_i) \prod_{j=1}^{M_N} P(x_j^c)P(x_k|x_j^c) \quad (5)$$

That is, if any of the cues from the source nodes is negative, the prediction of the node is negative. The Bayesian fusion rule is defined as

$$P'(x_k) = \frac{P(x_k) \max_p \{ (P(x_k))^r (1-P(x_k))^{S-r} \}}{\int (P(x_k))^r (1-P(x_k))^{S-r} dP} \quad (6)$$

where the prediction of the node is updated to P' from prior prediction P , and out of S cues that the neighboring nodes provide, r of them provide are positive, if the maximum likelihood principle is taken.

The probabilistic graph model provides a system level abstraction and a mesoscale description of CPSS networks, where information exchange and aggregation are captured with the overall probabilistic measures instead of the detailed level of information elements. More details about the probabilistic graph model can be found in Ref.[5].

3 The Proposed Information Dynamics Models

The information dynamics models are to characterize and predict how information is consumed and affects the decision making of individual nodes in a networked CPSS environment. In CPSS networks, each node produces information by sensing and processing. Information is exchanged between nodes. When a node receives some information from others, the received information is combined and digested, which is then used to update the prediction of the node. Thus the prediction probabilities of CPSS nodes are dynamically updated with the mutual influence from each other. As a result, strong dependencies exist among the prediction probabilities from different nodes.

As an extension of previous work [2][3], the dynamic changes of network topology are considered in the proposed VAR models, where the connections between nodes can vary instead of being static. The proposed data-driven models can be updated periodically. When the network topology is changed, the VAR models can be re-parameterized with the new training dataset. Furthermore, a new latent variable VAR model is proposed so that the hidden correlations can be parameterized with latent variables. This enables the linear VAR models to capture more complex dependency relations.

In our information dynamics models, the effects of information diffusion are quantified by the prediction probabilities associated with nodes. The influences and interdependency among

information producers and consumers are captured as functions of these probabilities. Generally speaking, the dynamics of prediction probabilities can be modelled by

$$\begin{aligned}\frac{dP(x_1, t)}{dt} &= g_1(P(x_1, t), \dots, P(x_n, t)) + \epsilon_1 \\ &\vdots \\ \frac{dP(x_n, t)}{dt} &= g_n(P(x_1, t), \dots, P(x_n, t)) + \epsilon_n\end{aligned}\tag{7}$$

where $g_1, \dots, g_n$ can be linear or nonlinear functions to capture the interdependency between prediction probabilities $P(x_k)$'s in the probabilistic graph model as introduced in Section 2.3. The high correlation between nodes is incorporated in modeling the dynamics, where the evolution of the prediction probability for each node is characterized. With the data-driven approach, the dynamics model can be simplified as time series in

$$\mathbf{p}(\mathbf{x}, t) = \mathbf{f}(\mathbf{x}, t) + \boldsymbol{\epsilon}(\mathbf{x})\tag{8}$$

where $\mathbf{x} = (x_1, \dots, x_n)$ denotes the vector of state variables for all nodes, $\mathbf{p}(\mathbf{x}, t) = (P(x_1, t), \dots, P(x_n, t))$ is the vector of all prediction probabilities, and $\boldsymbol{\epsilon} = (\epsilon_1, \dots, \epsilon_n)$ characterizes the uncertainty.

Three information dynamics models are introduced here to capture the information diffusion in CPSS networks based on the prediction probabilities that the nodes produce meaningful information. The first model is the latent variable VAR where hidden state variables are introduced. The second one is the topology-informed VAR model where the dynamic network topology is applied for more flexible and efficient model training. The third one is the topology-informed GPR model where the connectivity provides additional information about the similarity between nodes. In the VAR modeling approach, the interactions of the individual prediction probabilities are directly modeled with the parameters in VAR models. The parameters or coefficients of the linear models capture the coupling explicitly as functional relationships. That is, the prediction probability of one node is a function of the probabilities from its neighbors. The prior knowledge of network topology provides us a more compact VAR model. In the GPR modeling approach, the time series with inputs of time and node labels is developed based on both spatial and temporal correlations. The correlations are captured implicitly with the covariance or kernel functions in the model.

The effects of information accumulation are manifested by how well CPSS nodes make sound decisions. The accuracy and effectiveness of decisions are quantified by the probabilistic measures of their prediction capabilities, i.e. prediction probabilities. Based on new information, the CPSS nodes update their prediction probabilities as their perceptions of the world based on certain information fusion rules. The information dynamics models, discretized as time series models, elucidate how the prediction probabilities of nodes in the network are correlated. The understanding of correlation helps us to answer design questions, such as how to design network topology to control variability or fluctuation of prediction capabilities, how to assign key influencers in the network to improve the effectiveness of information propagation, and others.

To train the proposed data-driven models, training data are generated from simulations of information propagation in the probabilistic graph model. The coefficients in the VAR models and hyperparameters in the GPR models are first trained based on the simulation data before they can be applied for future prediction. During simulations, predictions about the state of the world are randomly generated based on the prediction probability for each node. The information exchanged between nodes is also generated based on the reliance probabilities. Based on information fusion rules introduced in Section 2.3, the prediction of each node is then updated. The new predictions are recorded and the statistics are used to update the prediction probability. The above simulation procedures are repeated iteratively for each time epoch, and the evolutions of the prediction probabilities are simulated.

In the remainder of this section, the VAR modeling approach is introduced in Section 3.1, where the latent variable VAR model and the topology-informed VAR model are introduced. In Section 3.2, the new spatial-temporal GPR model is described, where the correlation along time for each node and correlation between nodes are captured based on a new kernel function.

3.1 *Vector Autoregression Models*

VAR models provide a direct approach to represent time-dependency and mutual influences between variables in time series problems. In traditional VAR models, it is assumed that all variables are linearly dependent on each other for each intermediate step. Here, a latent variable VAR and a topology-informed VAR model are proposed. The prior knowledge of network topology is incorporated in model training, and functional dependency between nodes is directly modeled with the adjacency information. That is, two prediction probabilities are functionally

related to each other at two immediate time steps only if there is a direct connection between the two nodes. This approach can reduce the number of parameters to be trained and improve the training efficiency.

3.1.1 Topology-Informed VAR

The traditional VAR time series model is

$$\mathbf{P}(s) = \mathbf{A}_0 + \sum_{l=1}^L \mathbf{A}_l \mathbf{P}(s-l) + \epsilon \quad (9)$$

where the values of prediction probabilities for all n nodes at the s -th time step form the column vector $\mathbf{P}(s) = (P_1(s), \dots, P_n(s))$. The prediction probabilities at the s -th time step are dependent on the values of $\mathbf{P}(s-1), \dots, \mathbf{P}(s-L)$ at all the previous L steps. $\mathbf{A}_0 \in \mathbb{R}^n$ is the column vector of intercepts that indicate the bias. The noise $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_\epsilon)$ is modeled as the multi-variant normal random variable, and the $n \times n$ coefficient matrices $\mathbf{A}_l$'s capture the interdependency between prediction probabilities. The VAR model in Eq. (9) captures the time and location dependencies of nodes simultaneously as the linear relationships. The time series model captures the memory effect where the data sharing history in the previous L steps affects the current values. It also implicitly captures the potential effect of possible delays in communication, where data arriving at a node could be sent by other nodes several time steps before. The positive or negative signs of the elements in $\mathbf{A}_l$'s indicate the positive or negative correlations of the pairs. The correlations will cause fluctuations of the $\mathbf{P}$ values between 0 and 1.

One issue of VAR for network modeling is that the number of parameters increases quadratically as the number of nodes increases. In the topology-informed VAR model, not all parameters in Eq. (9) are equally important. If one node is connected with and shares information directly to another node, the information dependency is more evident than those that are not directly connected. Therefore, the topology of the network provides indications of dependency. This prior knowledge can be applied to the VAR model as the constraints to reduce the effective coefficients and simplify the model training process. In a sparsely connected network, dependency between nodes is loose, and the number of effective coefficients becomes small. This will improve the efficiency of the training process where the required training data size can be reduced accordingly.

The topology-informed VAR model is defined as

$$\mathbf{P}(s) = \mathbf{A}_0 + \sum_{l=1}^L (\mathbf{D}^T \circ \mathbf{A}_l) \mathbf{P}(s-l) + \epsilon \quad (10)$$

where $\mathbf{D} = (d_{ij})_{n \times n}$ is the adjacency matrix of the network with n nodes. Its elements are $d_{ij} = 1$ if a directed edge exists from node i to node j , and $d_{ij} = 0$ otherwise. The diagonal elements $d_{ii} = 1$ for all i 's. In addition, T is the matrix transpose operator, and $\circ$ is the element-wise product or Hadamard product between two matrices.

In the original VAR model in Eq. (9), the number of parameters or coefficients that need to be calibrated through training is $n + n^2L$. When the topology is considered as the constraint, the number of parameters is reduced to $n + eL$ where e is the number of directed edges in the graph. The reduction is significant if the nodes are sparsely connected. With a smaller number of parameters, the training or data fitting process can be more efficient where the required number of training data points for good fitting is also reduced. Note that the topology-informed VAR is reduced to the original VAR when the network is fully connected.

The training of topology-informed VAR model in Eq. (10) can be implemented as a constrained optimization problem, where the loss or error of model prediction is minimized subject to constraint

$$\sum_{l=1}^L \|(\mathbf{\Pi} - \mathbf{D}^T) \circ \mathbf{A}_l\| = 0 \quad (11)$$

where $\mathbf{\Pi}$ is a matrix with the same size of $\mathbf{D}$ and all elements are 1's. $\|\cdot\|$ is the matrix norm which quantifies the distance to the origin and can be calculated with L_2 norm. Constrained nonlinear optimization algorithms can be applied to solve this training problem. Because the elements in coefficient matrices $\mathbf{A}_l$'s contain the physical meanings of adjacency between nodes, they are also related to the reliance probabilities as described in Section 2.3. Therefore, the coefficients can also be treated as probabilities. The values of the coefficients can be further constrained to between 0 and 1.

Notice that all coefficients and parameters of the VAR models can be time-dependent. When the network topology dynamically changes, the model parameters need to be re-calibrated to reflect the changes. One advantage of data-driven modeling is that the model parameters can be updated frequently once new datasets are available. The proposed VAR models can be readily applied to the dynamic CPSS networks where the topology is not fixed and evolves along time.

3.1.2 Latent Variable VAR

More complex models can be applied to capture the functional interdependency between prediction probabilities. Here, latent variables or hidden state variables are introduced to capture the inherent correlations between prediction capabilities. The latent variable VAR model is defined as

$$\mathbf{P}(s) = \mathbf{B}\mathbf{Y}(s) + \boldsymbol{\eta} \quad (12)$$

$$\mathbf{Y}(s) = \sum_{l=1}^L \mathbf{T}_l \mathbf{Y}(s-l) + \boldsymbol{\varepsilon} \quad (13)$$

where Eq. (12) captures the relation between observable $\mathbf{P} \in \mathbb{R}^n$ and hidden state or latent variables $\mathbf{Y} \in \mathbb{R}^K$, and the evolution of hidden state variables as state transition is modeled in Eq.(13). Here, $\mathbf{T}_l \in \mathbb{R}^{K \times K}$ ($l = 1, \dots, L$) is the coefficient matrix that captures the dependency between the hidden state variables, and $\mathbf{B} \in \mathbb{R}^{n \times K}$ is the observation matrix. Notice that the dimension of state vector $\mathbf{Y}$ is not necessarily the same as the dimension of observable vector $\mathbf{P}$. The number of hidden state variables is typically less than that of the observable in a correlated system, i.e. $K \leq n$. $\boldsymbol{\eta} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_\eta)$ and $\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_\varepsilon)$ are associated with the noises of observation and state transition, respectively, and modeled as multi-variate normal variables. In the hidden state model, the interdependency among nodes is captured through the hidden state variables. The direct correlations between state variables cause the inherent correlations between the observables so that more complex dependency relations are captured.

In the VAR model in Eq. (9), the parameters to be calibrated are vector $\mathbf{A}_0$, matrices $\mathbf{A}_l$'s, and covariance matrix $\boldsymbol{\Sigma}_\varepsilon$. For the latent variable VAR model in Eqs. (12) and (13), the parameters to be calibrated or trained include $\mathbf{T}_l$'s, $\mathbf{B}$, as well as $\boldsymbol{\Sigma}_\eta$ and $\boldsymbol{\Sigma}_\varepsilon$. The Expectation-Maximization (EM) algorithm [38] can be applied to train the model with a large number of parameters. The EM algorithm consists iterations of two alternating steps. In the E-step, the expectation of the log-likelihood for hidden state variables and the available training dataset conditional on the parameters $\Lambda = \{\mathbf{T}_l, \mathbf{B}, \boldsymbol{\Sigma}_\eta, \boldsymbol{\Sigma}_\varepsilon\}$ is calculated. The available training dataset $\mathcal{P}$ is the time series of prediction probabilities $\mathbf{P}$'s. The hidden state variables $\mathcal{V}$ include all $\mathbf{Y}$'s. The expectation of the log-likelihood at the k -th iteration is calculated as $\mathbb{E}(\log f(\mathcal{P}, \mathcal{V} | \Lambda') | \Lambda^{(k-1)})$. In the M-step, the parameters are updated by solving the maximization problem $\Lambda^{(k)} =$

$\text{argmax}_{\Lambda'} \mathbb{E}(\log f(\mathcal{P}, \mathcal{V} | \Lambda') | \Lambda^{(k-1)})$. The E-step and M-step are applied iteratively until the parameters converge.

When the number of latent variables K is the same as the number of the observable n , the latent variables can be interpreted as the state variables associated with the individual nodes. The coefficients $\mathbf{T}_l$'s then can be regarded as the associations between nodes in information exchange, similar to the topology-informed VAR model in Section 3.1.1. Similarly, the network topology can be applied as constraints to simplify the model training process. In the topology-informed latent variable VAR model, the state transition in Eq.(13) is replaced by

$$\mathbf{Y}(s) = \sum_{l=1}^L (\mathbf{D}^T \circ \mathbf{T}_l) \mathbf{Y}(s-l) + \boldsymbol{\varepsilon} \quad (14)$$

where $\mathbf{D}$ is similarly the adjacency matrix of the network. The training of the constrained latent variable VAR model can be done with constrained nonlinear optimization methods.

3.2 Topology-Informed Gaussian Process Regression Model

Different from the VAR models where the linear response relationships between variables are assumed, GPR is a more generic regression method that can capture the nonlinearity. GPR also provides uncertainty predictions of responses in addition to the mean values locally at different locations in the input space. In the information dynamics model, the function in Eq.(8) is now a Gaussian process, with the noise level $\epsilon \sim \mathcal{N}(0, \sigma_0^2)$.

3.2.1 GPR Model

In the GPR model $y(x) \sim \mathcal{GP}(\mu(x), k(x, x'))$ with mean function $\mu(x)$ and covariance or kernel function $k(x, x')$, given M samples $\mathbf{x} = (x_1, \dots, x_M)$ and the functional values $\mathbf{y}(\mathbf{x}) = (y(x_1), \dots, y(x_M))$ as the training dataset, the joint distribution between the training data set and test data set $\mathbf{y}(\mathbf{x}_*)$ follows Gaussian distribution

$$\begin{bmatrix} \mathbf{y}(\mathbf{x}) \\ \mathbf{y}(\mathbf{x}_*) \end{bmatrix} \sim \mathcal{N} \left(\mathbf{0}, \begin{bmatrix} \mathbf{K}(\mathbf{x}, \mathbf{x}) & \mathbf{K}(\mathbf{x}, \mathbf{x}_*) \\ \mathbf{K}^T(\mathbf{x}, \mathbf{x}_*) & \mathbf{K}(\mathbf{x}_*, \mathbf{x}_*) \end{bmatrix} \right) \quad (15)$$

where $\mathbf{K}(\mathbf{x}, \mathbf{x})$ is the covariance matrix of training data points, $\mathbf{K}(\mathbf{x}, \mathbf{x}_*)$ is the covariance matrix between the training data and test data. The posterior predictions for new samples $\mathbf{x}_*$ given the existing ones from $\mathbf{x}$ is $p(\mathbf{y}_* | \mathbf{y}) \sim \mathcal{N}(\boldsymbol{\mu}_*, \boldsymbol{\Sigma}_*)$, where

$$\boldsymbol{\mu}_* = \mathbf{K}^T(\mathbf{x}, \mathbf{x}_*) (\mathbf{K}(\mathbf{x}, \mathbf{x}) + \sigma_0^2 \mathbf{I})^{-1} \mathbf{y} \quad (16)$$

$$\Sigma_* = \mathbf{K}(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{K}^T(\mathbf{x}, \mathbf{x}_*)(\mathbf{K}(\mathbf{x}, \mathbf{x}) + \sigma_0^2 \mathbf{I})^{-1} \mathbf{K}(\mathbf{x}, \mathbf{x}_*) + \sigma_0^2 \mathbf{I} \quad (17)$$

with identity matrix $\mathbf{I}$. GPR models predict not only mean values but also uncertainty associated with the predictions locally. This provides more flexibility than traditional regression models.

3.2.2 Spatial-Temporal GPR Time Series Model

Here a new GPR model is proposed to explicitly capture the information correlation between nodes as a nonlinear time series model. If the time series of prediction probability for each node is modeled by the traditional GPR model with time as the input, the correlations between different time series cannot be explicitly captured. The proposed GPR model uses two dimensions of inputs, which are time and node label. The node label indicates which time series or which node the output is associated with. With discretized time index s and node label m , the GPR model is

$$\begin{bmatrix} \mathbf{y}(s, m) \\ \mathbf{y}(s_*, m_*) \end{bmatrix} \sim \mathcal{N} \left(\mathbf{0}, \begin{bmatrix} \mathbf{K}((s, m), (s', m')) & \mathbf{K}((s, m), (s_*, m_*)) \\ \mathbf{K}^T((s, m), (s_*, m_*)) & k((s_*, m_*), (s_*, m_*)) \end{bmatrix} \right) \quad (18)$$

where $\mathbf{K}((s, m), (s', m'))$ is the covariance matrix of existing observations, $\mathbf{K}((s, m), (s_*, m_*))$ is the column vector of covariance between existing observations and the new prediction (s_*, m_*) . The core idea of this GPR model is the new kernel function with hybrid inputs of continuous time and discrete node label, defined as

$$k((s, m), (s', m')) = k_1(s, s')k_2(m, m') \quad (19)$$

which is composed with kernel functions $k_1(s, s')$ for the time dimension and $k_2(m, m')$ for the spatial dimension indicated by node labels. Kernel function $k_1(s, s')$ needs to capture the temporal correlation which causes fluctuations of prediction probabilities. That is, one node's prediction probability is not static and it fluctuates between 0 and 1. The fluctuation is a result of positive or negative influence from other nodes as well as the potential time delay factor. In GPR modeling, the fluctuation pattern is usually quantified with a periodic kernel function. Therefore, the periodic kernel here is defined as

$$k_1(s, s') = \exp(-2 \sin^2(\pi d_E(s, s')/p)/l^2) \quad (20)$$

with hyperparameters of period p and length scale l , and Euclidean distance function $d_E(\cdot, \cdot)$.

Kernel function $k_2(m, m')$ needs to be designed properly to capture the spatial correlation between nodes. The kernel needs to incorporate the similarity or difference between input variable values, which is quantified as the distance between them. A properly defined distance function allows the kernel to distinguish different input values, which in turn helps build an accurate surrogate. In this work, the input variables for $k_2(m, m')$ are node labels. Node labels need to include enough information so that the distance or difference between two nodes can be intuitively calculated. We propose to use

$$k_2(m, m') = \exp(-0.5d_H(m, m')/z^2) \quad (21)$$

with the hyperparameter of length scale z , and Hamming distance $d_H(\cdot, \cdot)$ between node labels. The node labels are based on the adjacency information of the nodes, because two directly connected nodes have stronger interdependency or correlation with information sharing. If two nodes have similar connectivity in the neighbourhood, i.e. they have similar information sources, then the values of their prediction probabilities should be similar. Thus the labels of the nodes need to reflect the topology of the network. In combination with the Hamming distance, the node labels are encoded as binary strings. For a network with n nodes, the node label is an n -bit binary number, where each bit corresponds to a node. In the label of node i , if node j is directly connected to node i , the bit corresponding to node j is “1” in the label of node i . Otherwise, it is “0”. For example, the labels for a four-node network are four-bit strings as “ $b_3b_2b_1b_0$ ”, where b_3, b_2, b_1, b_0 correspond to nodes 3, 2, 1, and 0, respectively. If node 0 is connected to nodes 1 and 3, and node 1 is also connected to node 2, the labels for nodes 3, 2, 1, and 0 will be “1001”, “0110”, “0111”, and “1011”, respectively. The bit for the node itself is always set to be “1”. The difference between two node labels measured by the Hamming distance indicates to some extent how strong the correlation is between the two nodes. A small difference indicates that the two nodes are connected to similar neighbours and have similar information sources. For instance, in the above four-node example, the distance between node 3 and node 2 is $d_H(m_3, m_2) = 4$. Similarly, $d_H(m_3, m_1) = 3$, $d_H(m_3, m_0) = 1$. The prediction of node 3 is more similar to the one of node 0 than the other two nodes, given that node 3 is directly connected to node 0. The Hamming distance based on the labels can provide some estimations of differences that are meaningful for the GPR surrogate model. Therefore, correlations of information between nodes can be quantified based on the labels.

4 Demonstrative Examples

In this section, several examples are used to demonstrate the proposed information dynamics models. Some simple networks are constructed with the associated prediction and reliance probabilities randomly generated. A CPSS network simulator ProbNet is developed to simulate the information update based on Monte Carlo sampling. The proposed information dynamics models were developed and compared with the Monte Carlo simulation results. Both the simulator and the information dynamics models were implemented in python programming language.

Monte Carlo sampling is applied to simulate the process of prediction probability updates. In each time step, random samples of observations are generated for each node based on its current prediction probability. Then the observations are shared with the neighboring nodes, and the shared information is sampled based on the reliance probabilities. When a node receives the information from its source nodes, a fusion rule (e.g. worst-case, best-case, Bayesian) is applied to update its prediction. The predictions are compared with the randomly generated ground truth state value and the correct instances are recorded. The above sampling procedure repeats many times, and the probability of correct prediction for each node is obtained and updated for this time step. The simulation clock advances, and the next iteration of update is done in the same way.

4.1 Demonstration of VAR Models

The VAR models are first demonstrated with a four-node-four-edge example in Figure 3 (a). The simulation data are collected to train the VAR model in Eq. (9) with lag order $L=2$. The training is done with the simulation data from the first 50 time steps. The forecast in the next 30 time steps is generated from the VAR model. The same data are used to train the constrained VAR model in Eq. (10). The results are shown in Figure 3 (b) and (c) respectively, where the solid lines are the simulated probabilities and the dash lines indicate the predictions or forecasts from the models. The shaded regions indicate the predicted error bounds with one standard deviation ($\pm\sigma$). Here the worst-case fusion rule is applied in the simulation. It is seen that the models predict the general trend well. The predictions from the two models are similar. Similar to any other time series models, VAR models focus more on the near-term forecasts. The forecasts of the distant future are more of the average trend. The calibrated coefficients of the two VAR models after training are compared in Table 1. If there is no directed edge connection between nodes, the corresponding coefficients in the constrained VAR model are zeros. Notice that some fitted

coefficients in the traditional VAR model have negative values, whereas the ones in the constrained VAR model are all positive with values between zero and one. Since the values of both independent and dependent variables here are probabilities, it is difficult to interpret the negative coefficients. In contrast, the fitted coefficients in the constrained VAR model are associated with the reliance probabilities. They can be interpreted as the conditional probability of positive prediction for a node given its information source. Therefore, the sparser set of parameters in the constrained VAR model provides both efficient and physically meaningful representations. To quantitatively compare the forecast accuracy, mean squared error (MSE), which is the average squared difference between the forecasts and the original data during the forecast period, is calculated. The MSE for the VAR model is 0.02083, whereas the one for the constrained VAR model is 0.02116. Although the coefficients in the constrained VAR model are much sparser and there are only 13 out of 36 coefficients are non-zero, the model still can have the similar performance as in the traditional VAR model.

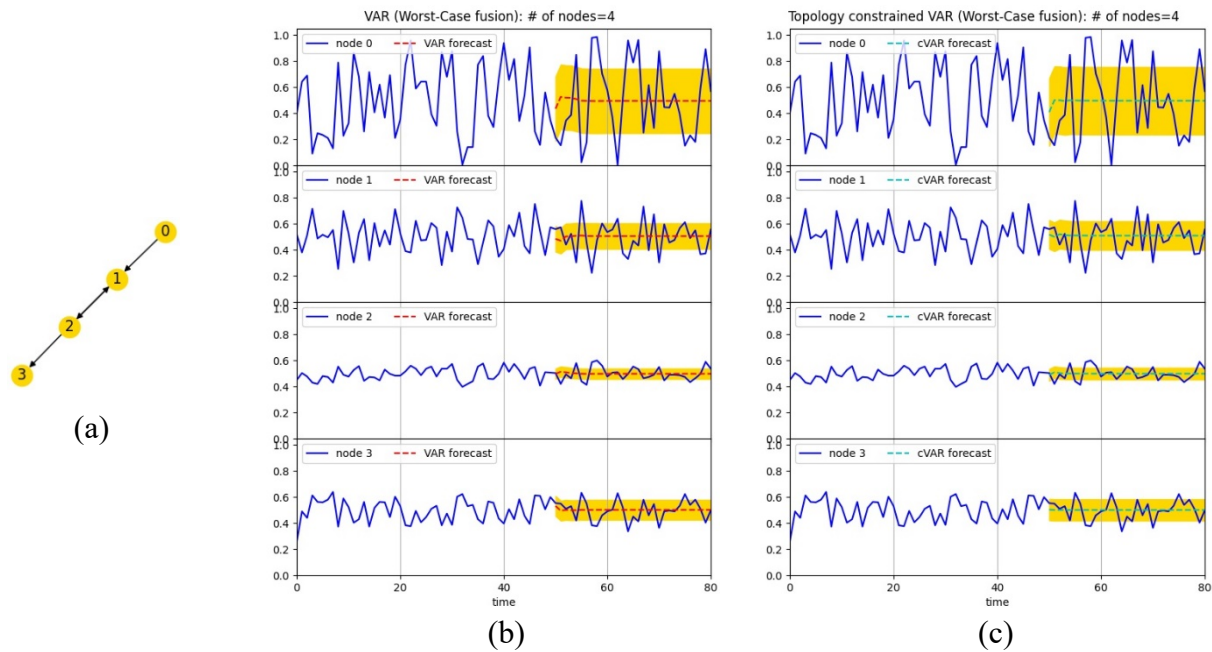


Figure 3. The simulated prediction probabilities and forecast with 50 time steps of training. (a) the four-node-four-edge example; (b) result of the VAR model (MSE=0.02083); and (c) result of the constrained VAR model (MSE=0.02116).

Table 1. Comparison of the calibrated parameters between the VAR model and constrained VAR model in Figure 3 for the four-node-four-edge network.

	VAR	Constrained VAR
A_0	$[-0.69957 \quad 0.86059 \quad -0.14549 \quad -0.13852]^T$	$[0 \quad 0.34793 \quad 0 \quad 0.46340]^T$
A_1	$\begin{bmatrix} -0.30989 & 0.37412 & -0.18996 & -0.04054 \\ 1.13367 & -0.35650 & 0.67226 & 0.24123 \\ 0.80635 & -0.29293 & 0.33254 & 1.07381 \\ 1.96312 & -1.30183 & 1.37625 & 0.26925 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0.34452 & 0 & 0.50053 & 0 \\ 0 & 0.32673 & 0.07335 & 0 \\ 0 & 0 & 0.14887 & 0.00713 \end{bmatrix}$
A_2	$\begin{bmatrix} 1.32495 & -0.51906 & 0.85481 & 0.29281 \\ 0.84116 & -0.31187 & 0.53460 & 0.09564 \\ -3.13919 & 1.63936 & -2.11252 & -0.60259 \\ -0.25101 & 0.05738 & -0.18614 & -0.05667 \end{bmatrix}$	$\begin{bmatrix} 0.18678 & 0 & 0 & 0 \\ 0.42275 & 0 & 0.25119 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0.00582 & 0.03474 \end{bmatrix}$

To further demonstrate the training efficiency, fewer training data are applied to the four-node-four-edge example. Only the first 10 steps of simulation data are used to train the VAR models. The results are shown in Figure 4. The traditional VAR model is not fully trained and the predictions become very unstable with the limited training data, whereas the constrained VAR model still predicts reasonably well. The MSEs for the VAR model and constrained VAR model are 0.16667 and 0.03118, respectively.

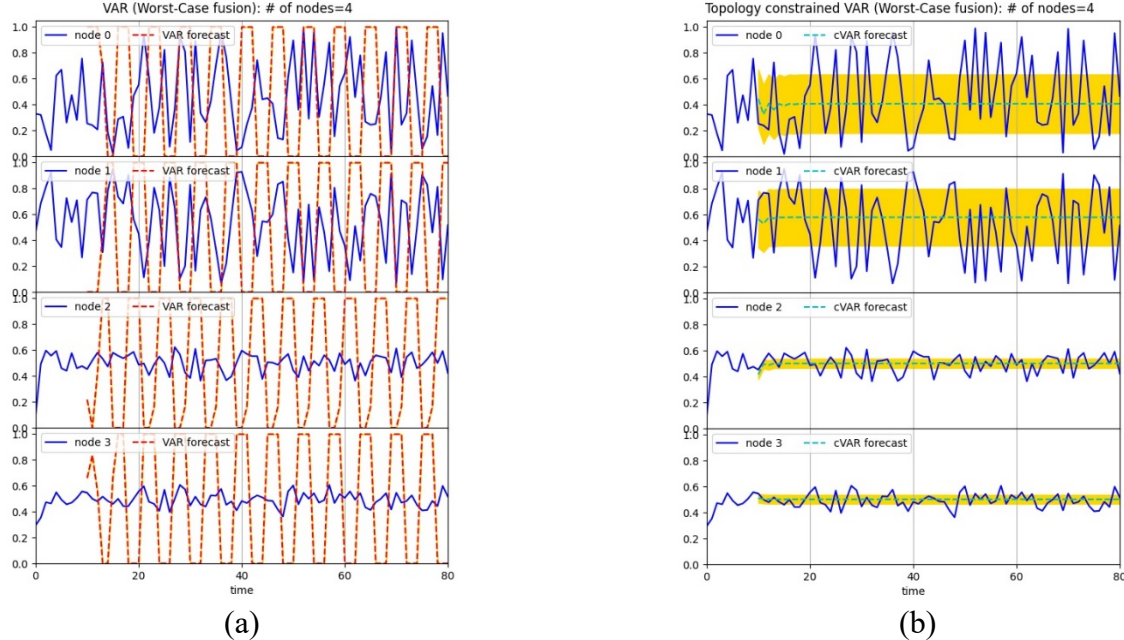


Figure 4. The simulated prediction probabilities and forecast with only 10 time steps of training from (a) the VAR model (MSE=0.16667) and (b) the constrained VAR model (MSE=0.03118).

The sensitivity of the model performance with respect to the lag order is tested. The traditional and constrained VAR models with lag orders of $L = 6$ and $L = 12$ are constructed. The results are compared in Figure 5. It is seen that higher-order models can provide more details of longer-term predictions instead of average values only. However, the number of model parameters to be fitted also increases. As seen in Figure 5 (b), the number of coefficients in the traditional VAR model with lag order $L = 12$ has increased so much that 50 time steps of training data become not enough for model training. In contrast, the constrained VAR model has the reduced number of effective coefficients. As seen in Figure 5 (d), the model with $L = 12$ shows good prediction performance.

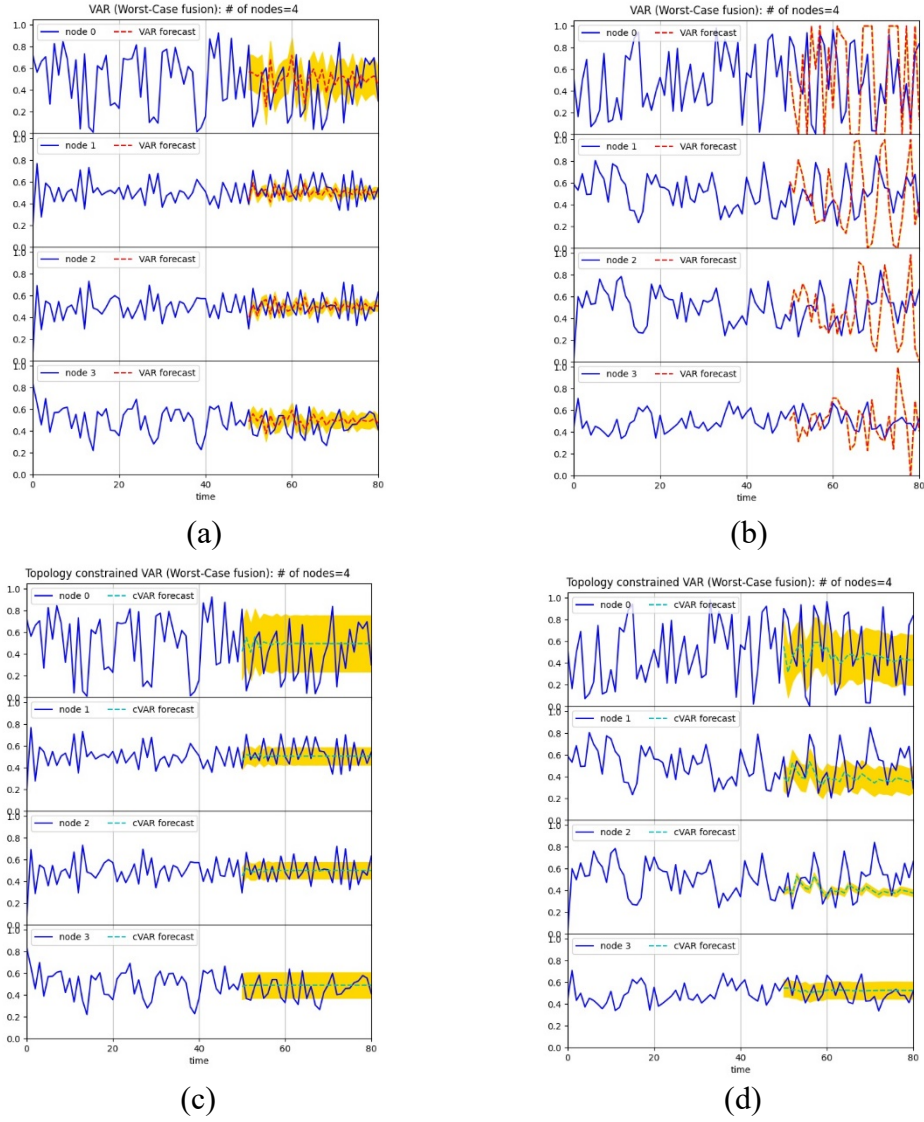
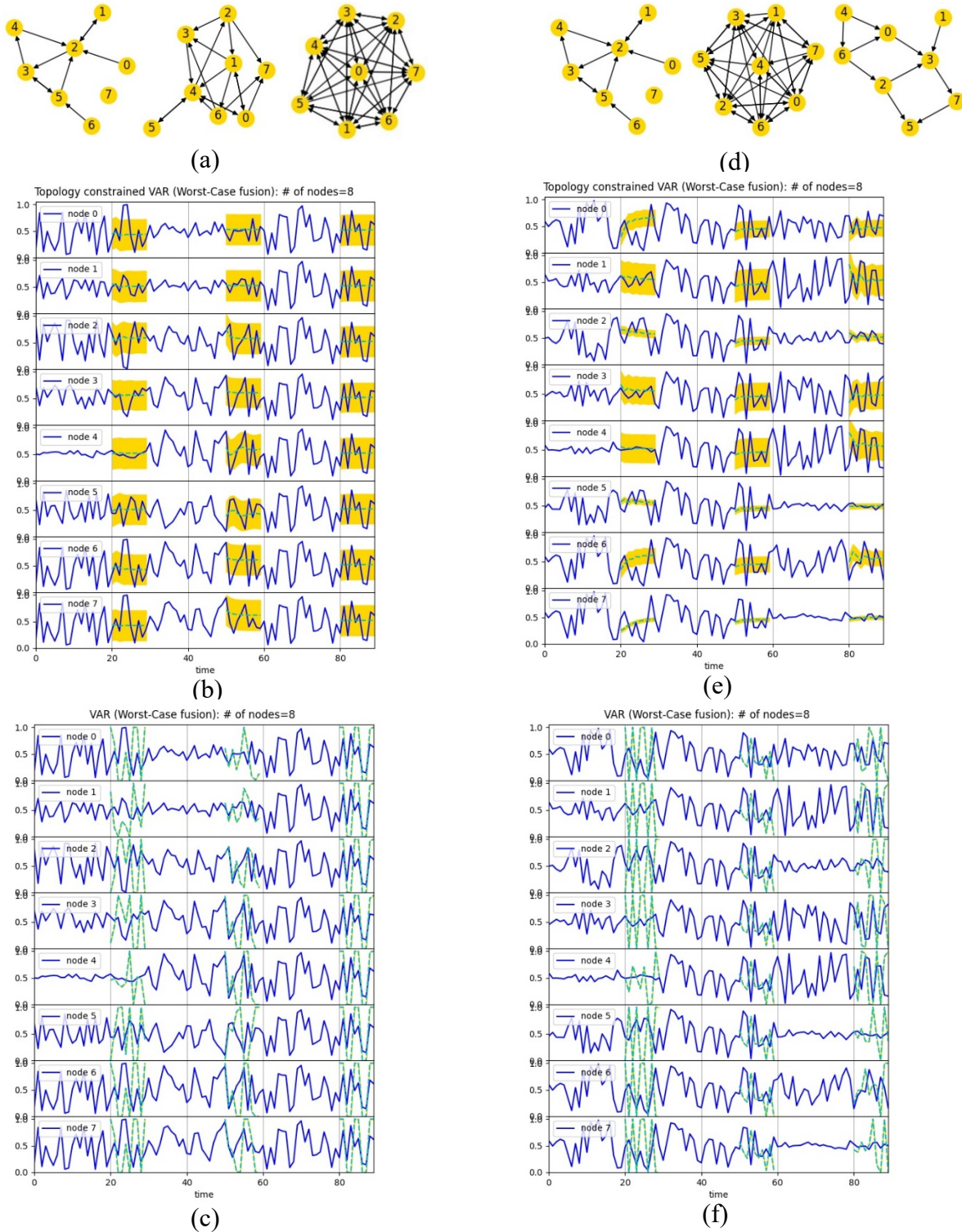


Figure 5. The sensitivity studies of different lag orders. (a) VAR model with $L = 6$ (MSE=0.03522); (b) VAR model with $L = 12$ (MSE=0.15196); (c) constrained VAR model with $L = 6$ (MSE=0.02697); (d) constrained VAR model with $L = 12$ (MSE=0.04394).

A second example is to demonstrate how data-driven modeling can be applied to dynamic networks where topology changes along time. As shown in Figure 6 (a), an 8-node network with 10 edges is initially constructed. After 30 time steps, the network topology is changed to 18 edges at the second epoch. The network is further changed to 54 edges during the third epoch. The simulated and forecasted prediction probabilities with the constrained VAR model are shown in Figure 6 (b), whereas the forecasts by the traditional VAR model are shown in Figure 6 (c). For each epoch of 30 time steps, the data of the first 20 time steps are used to train the VAR models.

The forecasts of the next 10 steps are compared with the simulation data. A different case is shown in Figure 6 (d), where the numbers of edges change to 43 and 10 during the next two epochs from the same initial 8-node-10-edge network as in Figure 6 (a). The simulation and forecast results are shown in Figure 6 (e) and (f). It is seen that the constrained VAR model predicts the trends reasonably well even with the dynamic topological changes and small training datasets. Dynamically evolving networks may not allow us to collect a large amount of data for training.

As also seen in Figure 6, the fluctuation patterns of prediction probabilities vary when the network topology changes, because information sharing patterns are different. When there are more connections, the coupling between nodes becomes stronger. The prediction probabilities tend to fluctuate more and be more synchronized. When a node does not receive information from others, such as nodes 0, 6, and 7 during the first epoch and nodes 1 and 4 during the third epoch in Figure 6 (d), the prediction can still fluctuate as a result of pure random effects. Some levels of information sharing suppress the fluctuations. Yet fully connected network can amplify the fluctuation with synchronization.



4.2 Demonstration of Latent Variable VAR Model

Here the latent variable VAR model is demonstrated. The model is constructed to predict the previous 4-node-4-edge network example in Figure 3 (a). Different values of factor number K and lag order L in Eqs. (12) and (13) are tested. The results are shown in Figure 7. The MSEs of forecasts from the latent variable VAR, topology constrained VAR, and traditional VAR are compared in Table 2. The forecasts by latent variable VAR and constrained VAR are more accurate than those by traditional VAR. Similar to other VAR models, when the lag order increases, the longer-term fluctuation can be predicted. In comparison with Figure 5 (a) for traditional VAR model of lag order $L = 6$, the results in Figure 7 (b), (d), and (f) with the same lag order are more stable and accurate. Therefore, introducing latent variables helps identify the intrinsic interdependency between observable variables.

Table 2. Comparison of forecast MSEs by the latent variable VAR, constrained VAR, and traditional VAR models

		Latent variable VAR	Constrained VAR	Traditional VAR
$K = 1$	$L = 2$	0.04138	0.04117	0.04401
	$L = 6$	0.04187	0.03486	0.05372
$K = 3$	$L = 2$	0.01790	0.01844	0.02223
	$L = 6$	0.02634	0.02696	0.05726
$K = 4$	$L = 2$	0.03167	0.03108	0.03481
	$L = 6$	0.02573	0.02238	0.03994

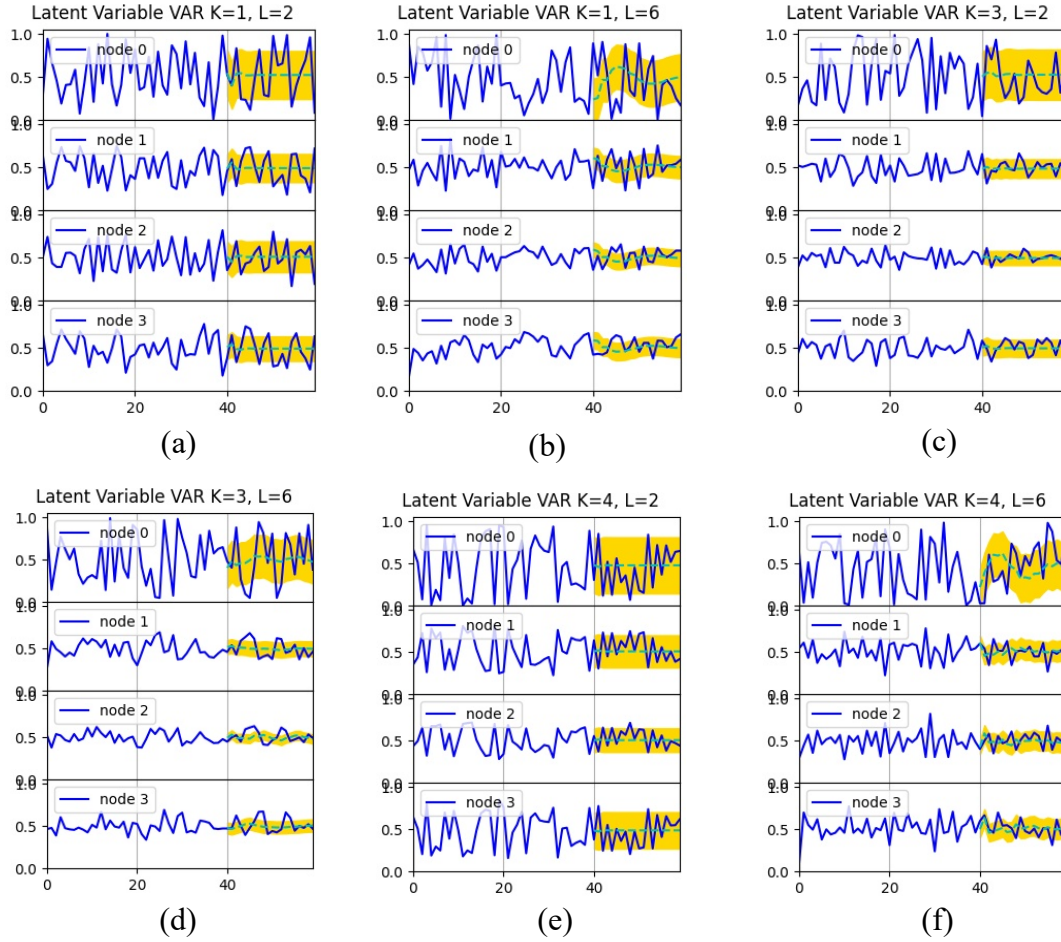


Figure 7. The latent variable VAR model forecasts for 4-node-4-edge network with different latent variable number K and lag order L . (a) $K=1, L=2$; (b) $K=1, L=6$; (c) $K=3, L=2$; (d) $K=3, L=6$; (e) $K=4, L=2$; (f) $K=4, L=6$.

The topology constrained latent variable VAR is further demonstrated with the 4-node-4-edge network example. With the factor number fixed as $K = 4$, three cases with different lag orders are compared. The results of MSEs from the constrained latent variable VAR model, in comparison with the latent variable VAR, constrained VAR, and traditional VAR, are shown in Table 3. It is seen that the topology constraints can help improve the forecast accuracy for both VAR and latent variable VAR models. The accuracy levels are similar for these two constrained models. The forecasts of the latent variable VAR models with and without topology constraints are also shown in Figure 8. As the lag order increases, the number of coefficients to be trained also increases, which requires more training data. The topology constraints can effectively reduce the

number of coefficients and improve the training efficiency with less training data, such as the case when $L = 8$.

Table 3. Comparison of forecast MSEs by constrained latent variable VAR, latent variable VAR, constrained VAR, and traditional VAR with $K = 4$

	Constrained latent variable VAR	Latent variable VAR	Constrained VAR	Traditional VAR
$L = 3$	0.06912	0.08165	0.07330	0.08594
$L = 4$	0.05405	0.07160	0.04918	0.06606
$L = 8$	0.04178	0.26609	0.04116	0.28155

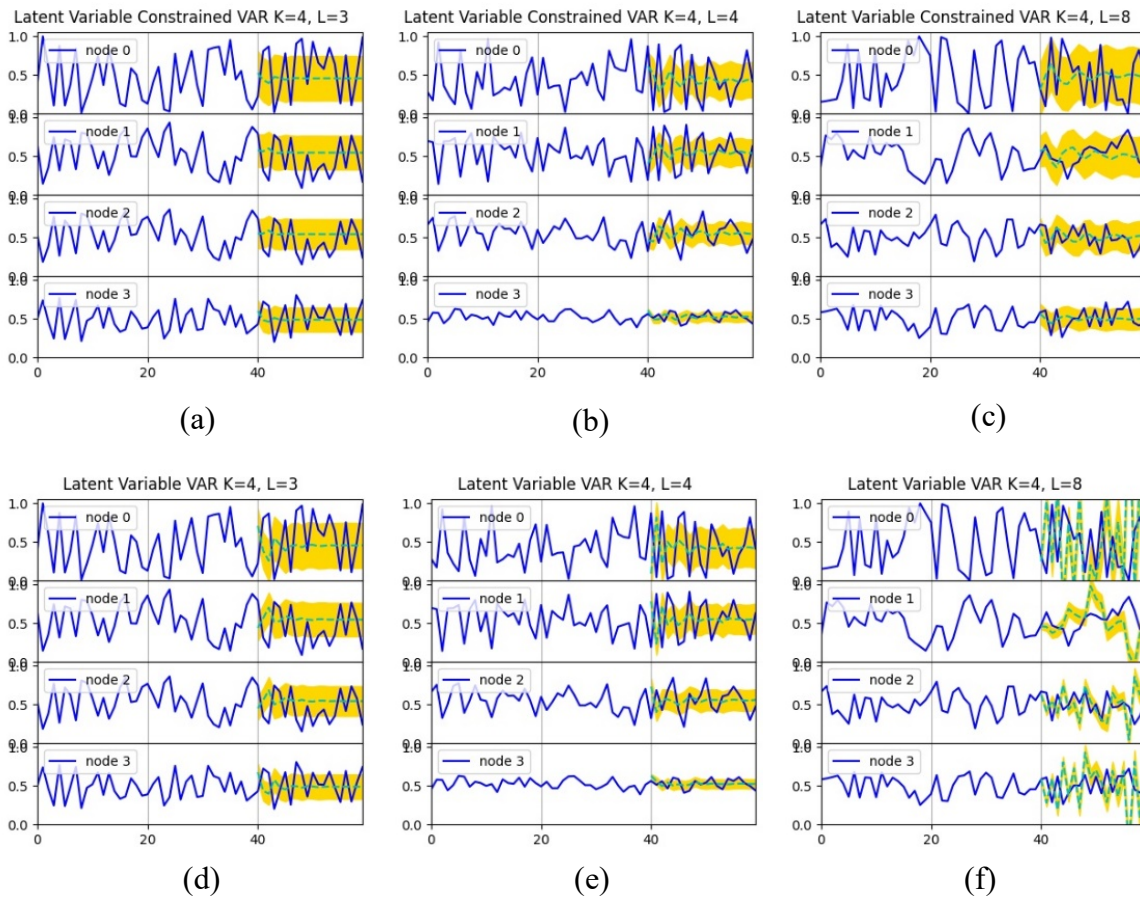


Figure 8. The latent variable VAR model forecasts for the 4-node-4-edge network with different lag orders where $K=4$. (a) $L=3$ with constraints; (b) $L=4$ with constraints; (c) $L=8$ with constraints; (d) $L=3$ without constraints; (e) $L=4$ without constraints; (f) $L=8$ without constraints.

4.3 Demonstration of Spatial-Temporal GPR Model

We also implemented and tested the proposed spatial-temporal hybrid GPR time series model to predict the dynamics of prediction probabilities. The results for the 3-node-3-edge example are shown in Figure 9. Two different data fusion rules, worst-case and best-case scenarios, are applied here. The results for the 8-node-10-edge example are shown in Figure 10. Compared to the previous linear VAR models which only predict the general trend in long term, the GPR model predicts the nonlinear dynamics of probabilities better. More fluctuations for both means and variances are observed in the forecasts. This is because of the kernel functions in GPR models which are able to capture differences locally based on the distance metrics. Thus the GPR model provides more details about the long-term dynamics in the networks. The proposed GPR model with the considerations of both spatial and temporal correlations between nodes captures the information dependency between nodes.

When all 8 nodes in the network are fully connected, the simulation and forecast results of the 8-node-56-edge example are shown in Figure 11. Because each node is connected with all other nodes, very strong correlations exist in the fully connected network. It is seen that the prediction probabilities of all nodes are synchronously fluctuating with the same values after a short period. The GPR model also predicts the synchronized fluctuations with error bounds.

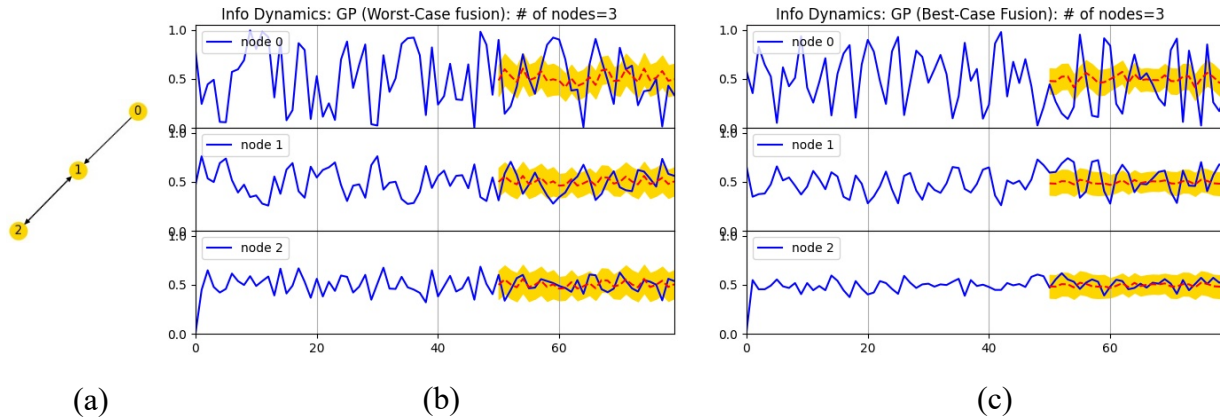


Figure 9. The simulated prediction probabilities and forecast by the GPR model in the 3-node-3-edge example, with (a) the worst-case fusion rule (MSE=0.03203), and (b) the best-case fusion rule (MSE=0.03795).

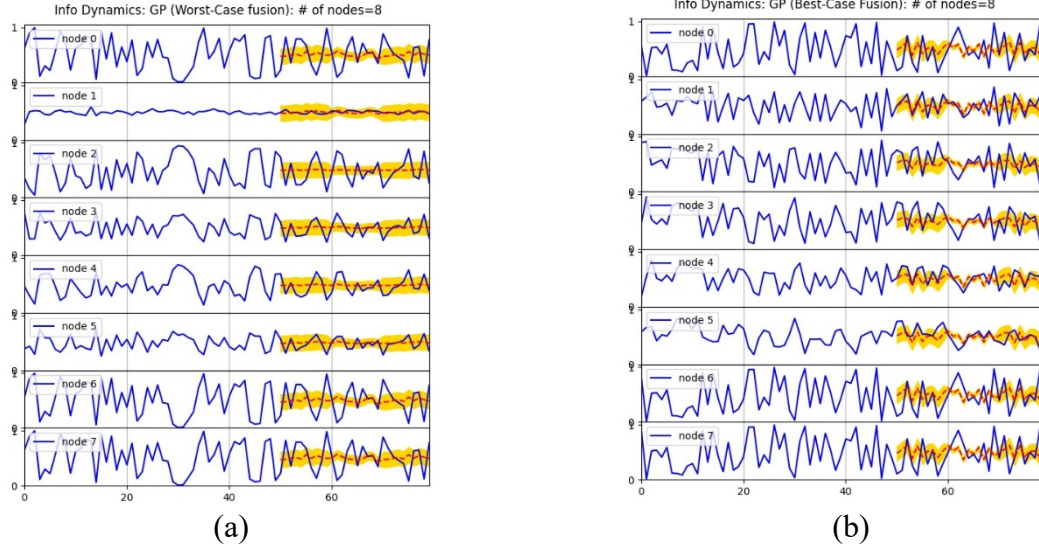


Figure 10. The simulated prediction probabilities and forecast by the GPR model in the 8-node-10-edge example, with (a) the worst-case fusion rule (MSE=0.03924), and (b) the best-case fusion rule (MSE=0.05461).

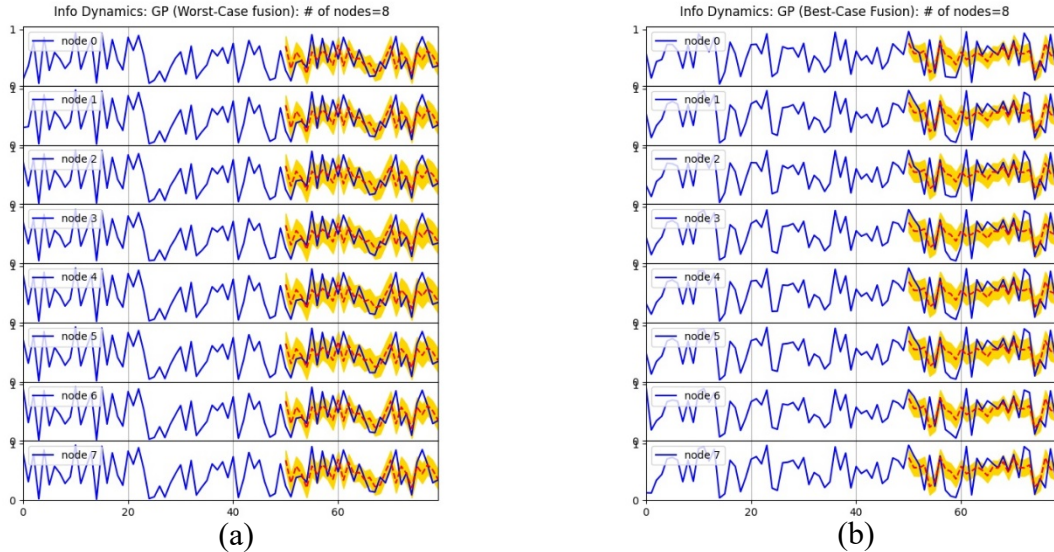


Figure 11. The simulated prediction probabilities and forecasts by the GPR model in the 8-node-56-edge example, with (a) the worst-case fusion rule (MSE=0.06174), and (b) the best-case fusion rule (MSE=0.08055).

5 Concluding Remarks

The analyses of the systems level behavior of CPSS networks enable us to design better systems. How to design a system of CPSS which promotes effective information sharing or

prevents misinformation propagation is one of the major aspects of design. Therefore, we need models that can characterize and predict the effects of information sharing and system behaviors.

In this paper, three information dynamics models are proposed to predict the information propagation within a CPSS network. Based on a recently developed mesoscale probabilistic graph model, the dynamics models are introduced to capture the mutual influences between nodes during their reasoning processes. The representation of prediction correlations between nodes is the central theme in both types of models. The results show that the topological information of networks can improve the efficiency in constructing the time series models. The network topology also has influences on the prediction capabilities of CPSS. Compared to other information diffusion models for CPSS, the proposed two types of models focus on the effects of information propagation on reasoning and prediction, instead of only on the diffusion speed and patterns in the network as in other models such as the epidemic model.

The first type of models represent the interdependency between nodes for their prediction capabilities explicitly as linear functions. It is demonstrated that the VAR linear model and latent variable VAR model can predict the general trend and the error bounds well. By introducing prior knowledge of network topology, the proposed topology-informed VAR models can significantly improve the training efficiency by reducing the number of effective coefficients or model parameters. The results show that the information correlations between nodes can be reasonably assumed to be directly related to the network connectivity. The adjacency information is useful to make the VAR models more compact and efficient. This is particularly important for large-scale networks where the number of nodes increases. If the connections remain sparse, the constrained VAR models are scalable. Training efficiency with reduced amount of training data is also important for dynamically evolving networks.

The proposed two-dimensional GPR model captures the correlated time series patterns, where a new kernel function is developed to consider both the temporal and spatial correlations of the data collected by CPSS nodes. The composite kernel function models the discrete spatial correlation with the topological adjacency relationship between nodes, in addition to temporal correlation. The GPR model has shown the advantage of revealing longer-term nonlinear dynamics in comparison with the linear models. The extent of local fluctuations can be predicted by the GPR model. This is because GPR models predict based on the similarity between inputs which is quantified with the kernel functions.

The prediction accuracy from the data-driven models sensitively relies on the training datasets. In general, larger datasets are always better for model training and calibration. The training effectiveness of parameters or hyperparameters also depends on the optimization algorithms applied in the training. For GPR models, the optimization of hyperparameters can affect the accuracy of model predictions. For situations where there is a lack of training data, the proposed modeling approach will not be feasible. Alternative modeling approaches that are based more on the detailed knowledge about the systems will be needed. Another limitation of the data-driven models is that the determination of model-form parameters such as lag orders and number of latent variables relies on empirical sensitivity studies. The choice of the best model form remains problem-specific and requires additional efforts. Nevertheless, it is seen that the physical knowledge of interdependency between nodes can enhance the data-driven approaches in modeling correlations, as demonstrated with the topology-informed VAR models and GPR model. The spatial correlation captured by the adjacency relationships between nodes deserve further studies.

Declaration of Competing Interest

The author declares no known competing interests that bias this work.

Acknowledgment

This work was supported in part by National Science Foundation under grant CMMI-1663227.

References

- [1] Batini, C. and Scannapieco, M. (2016) *Data and Information Quality: Dimensions, Principles and Techniques*. Springer, Switzerland. DOI: 10.1007/978-3-319-24106-7
- [2] Wang Y. (2020) Information dynamics in the network of cyber-physical systems. In *Proceedings of The 2020 International Symposium on Tools & Methods of Competitive Engineering (TMCE2020)*, May 11-15, 2020, Dublin, Ireland, pp.13-26.

- [3] Wang, Y. (2021) Probabilistic modeling of information dynamics in networked cyber-physical-social systems. *IEEE Internet of Things Journal*, (in press)
- [4] Wang, Y. (2016) System resilience quantification for probabilistic design of Internet-of-Things architecture. In *Proc. 2016 ASME International Design Engineering Technical Conferences & The Computer and Information in Engineering Conference (IDETC/CIE2016)*, Aug. 21-24, 2016, Charlotte, North Carolina, pp. V01BT02A011.
- [5] Wang, Y. (2018). Resilience Quantification for Probabilistic Design of Cyber-Physical System Networks. *ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part B: Mechanical Engineering*. 4(3), 031006.
- [6] Horváth, I., & Gerritsen, B. H. (2012). Cyber-physical systems: Concepts, technologies and implementation principles. In *Proc. 9th International Symposium on Tools and Methods of Competitive Engineering (TMCE2012)*, pp. 19–36.
- [7] Tavčar, J., & Horváth, I. (2018). A review of the principles of designing smart cyber-physical systems for run-time adaptation: Learned lessons and open issues. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 49(1), 145-158.
- [8] Vroom, R. W., & Horváth, I. (2014). Cyber-physical augmentation: An exploration. *Proc. 10th International Symposium on Tools and Methods of Competitive Engineering (TMCE2014)*, pp. 1509-1520.
- [9] Horváth, I., & Wang, J. (2015). Towards a comprehensive theory of multi-aspect interaction with cyber physical systems. In *Proc. ASME 2015 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (IDETC/CIE2015)*, pp. V01BT02A009.
- [10] Li, Y., Horváth, I., & Rusák, Z. (2018). Constructing Personalized Messages for Informing Cyber-Physical Systems based on Dynamic Context Information Processing. In *Proc. 12th International Symposium on Tools and Methods of Competitive Engineering (TMCE2018)*, pp.105–120.
- [11] Wang, Y. (2018). Trustworthiness in designing cyber-physical systems. In *Proc. 12th International Symposium on Tools and Methods of Competitive Engineering (TMCE2018)*, pp.27–40.

- [12] Wang, Y. (2018). Trust Based Cyber-Physical Systems Network Design. In *Proc. ASME 2018 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (IDETC/CIE2018)*, pp. V01AT02A037.
- [13] Wang, Y. (2018). Trust quantification for networked cyber-physical systems. *IEEE Internet of Things Journal*, 5(3), 2055-2070.
- [14] Wang, Y. (2020). Quantifying trust perception to enable design for connectivity in cyber-physical-social systems. In: *Emotional Engineering, Vol. 8: Emotion in the Emerging World* (eds. by S. Fukuda), pp.85-114. Springer, Cham, Switzerland
- [15] Wang, Y. (2021). Design of trustworthy cyber-physical-social systems with discrete Bayesian optimization. *Journal of Mechanical Design*, 143(7): 071702.
- [16] Jeon, J., Chun, I., & Kim, W. (2012). Metamodel-based CPSS modeling tool. In: *Embedded and Multimedia Computing Technology and Service*, Lecture Notes in Electrical Engineering, Vol. 181, pp. 285-291. Springer, Cham, Switzerland.
- [17] Lee, K. H., Hong, J. H., & Kim, T. G. (2015). System of Systems Approach to Formal Modeling of CPSS for Simulation-Based Analysis. *ETRI Journal*, 37(1), 175-185.
- [18] Lee, E. A., Niknami, M., Nouidui, T. S., & Wetter, M. (2015, October). Modeling and simulating cyber-physical systems using CyPhySim. In *Proceedings of the 12th IEEE International Conference on Embedded Software* (pp. 115-124). .
- [19] Saeedloei, N., & Gupta, G. (2011). A logic-based modeling and verification of CPSS. *ACM SIGBED Review*, 8(2), 31-34.
- [20] Horváth, I. (2019). A Computational Framework for Procedural Abduction Done by Smart Cyber-Physical Systems. *Designs*, 3(1), 1.
- [21] Burmester, M., Magkos, E., & Chrissikopoulos, V. (2012). Modeling security in cyber-physical systems. *International Journal of Critical Infrastructure Protection*, 5(3-4), 118-126.

- [22] Petnga, L., & Austin, M. (2016). An ontological framework for knowledge modeling and decision support in cyber-physical systems. *Advanced Engineering Informatics*, 30(1), 77-94
- [23] Pourtalebi, S., & Horváth, I. (2017). Information schema constructs for instantiation and composition of system manifestation features. *Frontiers of Information Technology & Electronic Engineering*, 18(9), 1396-1415.
- [24] Magureanu, G., Gavrilescu, M., Pescaru, D., & Doboli, A. (2010, September). Towards UML modeling of cyber-physical systems: A case study for gas distribution. In *Proc. IEEE 8th International Symposium on Intelligent Systems and Informatics* (pp. 471-476).
- [25] Palachi, E., Cohen, C., & Takashi, S. (2013, April). Simulation of cyber physical models using SysML and numerical solvers. In *Proc. 2013 IEEE International Systems Conference (SysCon)* (pp. 671-675).
- [26] Newman, M. E. (2002). Spread of epidemic disease on networks. *Physical review E*, 66(1), 016128.
- [27] Wu, F., Huberman, B. A., Adamic, L. A., & Tyler, J. R. (2004). Information flow in social groups. *Physica A: Statistical Mechanics and its Applications*, 337(1-2), 327-335.
- [28] Gruhl, D., Guha, R., Liben-Nowell, D., & Tomkins, A. (2004, May). Information diffusion through blogspace. In *Proceedings of the 13th international conference on World Wide Web* (pp. 491-501).
- [29] Kleineberg, K. K., & Boguná, M. (2014). Evolution of the digital society reveals balance between viral and mass media influence. *Physical Review X*, 4(3), 031046.
- [30] Yang, J., & Leskovec, J. (2010, December). Modeling information diffusion in implicit networks. In *2010 IEEE International Conference on Data Mining* (pp. 599-608).
- [31] Simma, A., & Jordan, M. I. (2010, July). Modeling events with cascades of poisson processes. In *Proceedings of the Twenty-Sixth Conference on Uncertainty in Artificial Intelligence* (pp. 546-555).
- [32] Iwata, T., Shah, A., & Ghahramani, Z. (2013, August). Discovering latent influence in online social activities via shared cascade poisson processes. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 266-274).
- [33] Du, N., Song, L., Woo, H., & Zha, H. (2013, April). Uncover topic-sensitive information diffusion networks. In *Proceedings of the 16th International Conference on Artificial Intelligence and Statistics* pp. 229-237.

- [34] Yagan, O., Qian, D., Zhang, J., & Cochran, D. (2013). Conjoining speeds up information diffusion in overlaying social-physical networks. *IEEE Journal on Selected Areas in Communications*, 31(6), 1038-1048.
- [35] Lu, H., Lv, S., Jiao, X., Wang, X., & Liu, J. (2015). Maximizing information diffusion in the cyber-physical integrated network. *Sensors*, 15(11), 28513-28530.
- [36] Wang, J., Jiang, C., Han, Z., Quek, T. Q., & Ren, Y. (2017, July). Private information diffusion control in cyber physical systems: A game theory perspective. In *Proceedings of 2017 IEEE 26th International Conference on Computer Communication and Networks (ICCCN)*, (pp. 1-10).
- [37] Yi, Y., Zhang, Z., & Gan, C. (2019). The outbreak threshold of information diffusion over social-physical networks. *Physica A: Statistical Mechanics and its Applications*, 526, 121128.
- [38] Bańbura, M., & Modugno, M. (2014). Maximum likelihood estimation of factor models on datasets with arbitrary pattern of missing data. *Journal of Applied Econometrics*, 29(1), 133-160.