Deep Learning Exploration of Agent-Based Social Network Model Parameters

Yohsuke Murase[1], Hang-Hyun Jo[2]*, János Török[3,4,5], János Kertész[4,6] and Kimmo Kaski[6,7]

[1]RIKEN Center for Computational Science, Kobe, Japan, [2]Department of Physics, The Catholic University of Korea, Bucheon, South Korea, [3]Department of Theoretical Physics, Budapest University of Technology and Economics, Budapest, Hungary, [4]Department of Network and Data Science, Central European University, Vienna, Austria, [5]MTA-BME Morphodynamics Research Group, Budapest University of Technology and Economics, Budapest, Hungary, [6]Department of Computer Science, Aalto University, Espoo, Finland, [7]The Alan Turing Institute, British Library, London, United Kingdom

Interactions between humans give rise to complex social networks that are characterized by heterogeneous degree distribution, weight-topology relation, overlapping community structure, and dynamics of links. Understanding these characteristics of social networks is the primary goal of their research as they constitute scaffolds for various emergent social phenomena from disease spreading to political movements. An appropriate tool for studying them is agent-based modeling, in which nodes, representing individuals, make decisions about creating and deleting links, thus yielding various macroscopic behavioral patterns. Here we focus on studying a generalization of the weighted social network model, being one of the most fundamental agent-based models for describing the formation of social ties and social networks. This generalized weighted social network (GWSN) model incorporates triadic closure, homophilic interactions, and various link termination mechanisms, which have been studied separately in the previous works. Accordingly, the GWSN model has an increased number of input parameters and the model behavior gets excessively complex, making it challenging to clarify the model behavior. We have executed massive simulations with a supercomputer and used the results as the training data for deep neural networks to conduct regression analysis for predicting the properties of the generated networks from the input parameters. The obtained regression model was also used for global sensitivity analysis to identify which parameters are influential or insignificant. We believe that this methodology is applicable for a large class of complex network models, thus opening the way for more realistic quantitative agent-based modeling.

Keywords: social networks, agent-based modeling, high-performance computing, metamodeling, deep learning, sensitivity analysis

1 INTRODUCTION

When analyzing the structural patterns of real social networks, the synthetic model-generated networks have served as references for comparison and enhanced our insight into their properties (Kertész et al., 2021). Here we classify the models of social networks into two categories: static models and dynamic models. The static models constitute a family of models, in which the network links are randomly generated with certain constraints. The most fundamental one is the Erdös-Rényi model

(Erdös and Rényi, 1960), which generates a random network under the constraint on the average degree. Other examples include the configuration model, the exponential random graph model, and the stochastic block models (Barabási and Pósfai, 2016; Newman, 2018; Menczer et al., 2020). Often these models enable us to compute their properties analytically. After inferring the model parameters for a real social network of interest, these models serve as useful null models. One can then judge, for instance, whether an observed quantity is significantly different from the expected null model value or not, thus serving as hypothesis testing.

On the other hand, dynamic models are models, in which the network evolves as a function of time. One of the major objectives of the dynamic models is to find the mechanisms that lead to certain structures observed in empirical networks. Here the models are defined by the rules on how nodes and links are created or deleted, in order to incorporate the perceived mechanisms of the evolutionary processes of social networks. However, in the light of vast complexities of social dynamics, these mechanisms should, at best, be considered as plausible ones. This class includes Barabási-Albert scale-free network model and its generalizations (Barabási and Pósfai, 2016) as well as Kumpula et al.'s weighted social network model (Kumpula et al., 2007). The latter will be in the focus of the present paper. These models allow us to get insight into how and why the observed networks have been generated and, more importantly, to predict the possible evolution of real networks.

While the static models are often analytically solvable, analytical tractability of dynamic models is limited to some basic cases. These models are usually designed to be as simple as possible, in order to identify the most important mechanisms, yet they should not be considered for quantitative comparison and prediction. When models are extended to incorporate aspects of reality, understanding their behavior becomes a formidable task since the parameter space is high-dimensional and non-trivial relationships between the parameters may occur. This makes the choice of appropriate parameters very difficult.

In this paper, we are going to overcome this difficulty by high performance computing (HPC) approach and by the development of metamodels in order to investigate the parameter space of an agent-based model of social networks. To achieve this, a massive number of simulations will be performed, the results of which are then used as training data for a neural network model to either infer or analyze the properties of the generated networks. Such a regression model is called a metamodel or surrogate model as it is a model of a simulation model (Wang and Shan, 2007; Zhao and Xue, 2010; Ghiasi et al., 2018). Metamodels are developed as approximations of the expensive simulation process in order to improve the overall computation efficiency and they are found to be a valuable tool to support activities in modern engineering design, especially design optimization. Once a good metamodel is obtained, it is useful for various purposes including parameter tuning, understanding of the model behavior, and sensitivity analysis. The metamodel is effective especially when the simulations are computationally demanding, which is the case for agent-based models with many parameters. To the best of our knowledge, this study is the first attempt to apply the metamodeling approach to an agent-based model of social networks. Here we will also carry out a sensitivity analysis to demonstrate that this approach is useful to distinguish between influential and negligible parameters.

This paper is organized as follows. In **Section 2**, we first introduce Kumpula et al.'s weighted social network model and its various extensions, then we formulate a generalized model to incorporate most of the previous extensions. In **Section 3**, we conduct the regression analysis and the sensitivity analysis using neural networks. **Section 4** is devoted to the summary and discussion.

2 GENERALIZED WEIGHTED SOCIAL NETWORK MODEL

2.1 Background

The original weighted social network (WSN) model was introduced in Kumpula et al. (2007), which is a dynamic, undirected, and weighted network model leading to a stationary state. The model starts with an empty network of N nodes. Links are created, updated, and eliminated using three essential mechanisms, which are called a global attachment (GA), a local attachment (LA), and a node deletion (ND). The GA mechanism represents the focal closure in sociology (Kossinets and Watts, 2006), meaning that ties between people are formed due to the shared activities. In the WSN model it is implemented by creating links between randomly chosen nodes. The LA mechanism represents the cyclic or triadic closure (Kossinets and Watts, 2006), implying that friends of friends get to know each other. The created new links by GA or LA mechanisms are assigned link weights of the same positive value. Whenever the LA mechanism is applied, weights of involved links increase by some fixed non-negative value, implying the link reinforcement (note that the triadic closure is not always successful in the LA mechanism, which will be discussed later). Finally, the ND mechanism represents a turnover of nodes, which can be implemented by removing all the links incident to a randomly chosen node. Without the ND mechanism, the model would end up with a complete graph, while with it a stationary state sets in for long times.

The WSN model turns out to generate several stylized facts of social networks (Jo et al., 2018), among which the Granovetterian community structure is the most important one as it connects the global structure of the network with local features (Granovetter, 1973). The Granovetterian community structure means that communities of nodes that are tightly and strongly connected to each other are connected via weak ties, as shown in **Figure 1** (top center); nodes connected by strong links within communities have many common neighbors while the opposite is true for weak links. If one of the above mechanisms is missing from the model, the model is not able to reproduce the Granovetterian community structure, indicating that the model contains the minimal set of the essential ingredients for this structural feature. Among several parameters in the model, link reinforcement is the most crucial
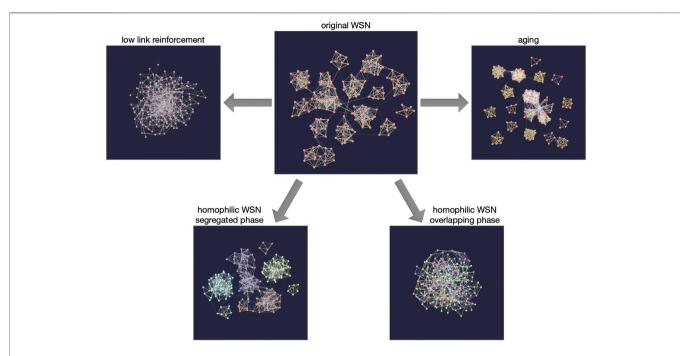
FIGURE 1 | Snapshots of the networks generated by the GWSN model for various combinations of parameter values in the model. The top center panel shows the network generated by the original WSN model showing the Granovetterian community structure when the links are reinforced sufficiently. The Granovetterian community structure is absent when the link reinforcement is not high enough, as shown in the top left panel. The top right panel presents the network with higher clustering than the original WSN model when only the link aging mechanism is applied for the link termination. The bottom panels show the effect of homophilic interaction on the generated networks; depending on the parameter values, it shows the structural transition from the segregated, non-overlapping communities **(bottom left)** to the overlapping communities **(bottom right)**.

one to get the desired community structure (see **Figure 1** (top left)).

Let us briefly sketch how the Granovetterian community structure emerges in the model: As the nodes are initially isolated, they make connections with other nodes by means of the GA mechanism. By the LA mechanism, created links are gradually strengthened and occasionally new links are also formed by the triadic closure. Such clusters formed around triangles are seeds for communities. These seeds develop as their sizes increase by the GA mechanism, and the links inside the communities get more dense due to the triadic closure and their weights become larger by the link reinforcement, thus leading to the Granovetterian community structure. By the ND mechanism, all the links adjacent to a randomly chosen node are removed in order to reach the nontrivial stationary state of network dynamics.

Although the WSN model has remarkably reproduced some salient features of social networks, it lacks other realistic considerations such as the geographical dimension (Barthélemy, 2011) and multiplexity (Boccaletti et al., 2014; Kivelä et al., 2014). In order to consider such factors, Murase et al. (2014a) extended the WSN model by assuming that each node can have connections in different layers for representing different types of connections. These nodes are also embedded into a two-dimensional geographical space, which enables to define the geographical distance between nodes as used in cases when the nodes choose other nodes that are

geographically close to them for the GA mechanism. The locality of interaction naturally leads to the correlation between different layers and to a multilayer Granovetterian community structure. In the absence of interlayer correlations, the generated network does not have a community structure when merged into a single-layer network.

Since the ND mechanism, as the only mechanism eliminating links, could be too abrupt for representing the real dynamics of social networks, alternative mechanisms focusing on the links, namely, link deletion (LD) (Marsili et al., 2004) and link aging, have been studied within the framework of the WSN model (Murase et al., 2015). The LD mechanism is implemented by removing randomly chosen links irrespective of their weights, while for the link aging each link weight gradually decays and is removed when the link weight gets smaller than some threshold value. Different link termination mechanisms lead to different network structures, e.g., see **Figure 1** (top right) for the case of link aging. Three link termination mechanisms (ND, LD, and aging) are not exclusive to each other but may work simultaneously in reality. Note that the WSN model with the link deletion mechanism has been used to study the effect of the so-called channel selection on the network sampling (Török et al., 2016).

Finally, Murase et al. (2019) extended the WSN model to incorporate the effect of homophily or the tendency of individuals to associate and bond with similar others (McPherson et al., 2001). For this, each node is assigned a

set of features, similarly to the Axelrod model and its variants (Axelrod, 1997; Centola et al., 2007; Vazquez et al., 2007; Vazquez and Redner, 2007; Gandica et al., 2011; Tilles and Fontanari, 2015; Min and Miguel, 2017). Precisely, each node is represented by a feature vector that is a collection of traits for each feature. These features may relate to gender, ethnicity, language, religion, etc. The GA and LA mechanisms apply only to the nodes sharing the same trait for a randomly chosen feature per interaction. The number of features represents the social complexity of the population, while the number of traits per feature represents the heterogeneity of the population. Depending on the number of features and the number of traits per feature, the structural transition of the network from an overlapping community structure to a segregated, non-overlapping community structure is observed [see **Figure 1** (bottom)].

In addition, we remark that the temporal network version of the WSN model was studied by incorporating the timings of interactions by the GA and LA mechanisms (Jo et al., 2011). This temporal WSN model generates not only the Granovetterian community structure but also bursty interaction patterns (Karsai et al., 2018).

In this paper we treat the WSN model by considering all generalizations including the geographical dimension, various link termination mechanisms, and the homophilic interaction at the same time, which is called a generalized weighted social network (GWSN) model hereafter.

2.2 Model Definition

We define a generalized weighted social network (GWSN) model for generating realistic social networks. Let us consider a network of N nodes. Each individual node $i = 1,\ldots, N$ is located in a fixed position (x_i, y_i) that is randomly selected in a unit square $[0, 1] \times [0, 1]$ with periodic boundary conditions. The Euclidean distance between two nodes i and j is denoted by r_{ij}. The node i is characterized by a feature vector of F components, i.e., $\vec{\sigma}_i = (\sigma_i^1, \ldots, \sigma_i^F)$ with $\sigma_i^f \in \{1, \ldots, q\}$ for each $f \in \{1, \ldots, F\}$. The value of σ_i^f is called a trait. The value of each σ_i^f is uniformly randomly selected in the beginning of the simulation and then fixed throughout. In reality the different feature may have different numbers of traits, whereas we assume for simplicity that all features have the same number of traits. By the homophilic interaction rule for a randomly chosen feature f, only a pair of nodes with the same trait, i.e., $\sigma_i^f = \sigma_j^f$, are allowed to interact with each other. A link weight between nodes i and j is denoted by w_{ij}, and $w_{ij} = 0$ indicates the absence of the link between them.

The implementation of the GWSN model is summarized as a pseudocode in **Algorithm 1**. A simulation starts with an empty network of size N. For each Monte Carlo time step t, every node updates its neighborhood by sequentially applying the GA and LA mechanisms as well as three link termination mechanisms, i.e., ND, LD, and aging.

For the GA mechanism, let us consider a focal node i and a randomly chosen feature f from $\{1,\ldots, F\}$. We obtain a set of nodes whose fth feature has the same value as σ_i^f and that are not connected with the node i, precisely,

$$\Gamma_i^f = \{j | \sigma_j^f = \sigma_i^f \ \& \ w_{ij} = 0\}. \tag{1}$$

One of nodes in Γ_i^f, say j, is randomly chosen with the probability given by

$$p_{ij}^{\text{GA}} = \frac{r_{ij}^{-\alpha}}{\sum_{j' \in \Gamma_i^f} r_{ij'}^{-\alpha}}. \tag{2}$$

Here the non-negative parameter α controls the locality of interaction. When $\alpha = 0$, there is no geographical constraint. The larger the value of α is, the more likely nodes tend to make connections with geographically closer nodes. If the focal node i has no neighbors, equivalently, if the node i's degree is zero ($k_i = 0$), it makes connection to the node j. Otherwise, if $k_i > 0$, the link between nodes i and j is created with a probability p_r. The initial weight of the new link is set to w_0. See **Figure 2A** for an example of the GA mechanism.

For the LA mechanism, if the focal node i has neighbors, it randomly chooses one of its neighbors j with $\sigma_j^f = \sigma_i^f$ with the probability proportional to the link weight w_{ij}, precisely,

$$p_{ij}^{\text{LA}} = \frac{w_{ij}}{\sum_{j' \in \Lambda_i^f} w_{ij'}}. \tag{3}$$

Here Λ_i^f denotes a set of i's neighbors whose fth feature has the same value as σ_i^f. Then the node j randomly chooses one of its neighbors but i, say l, with $\sigma_l^f = \sigma_j^f$, with the probability proportional to the link weigth w_{jl}.

$$p_{jl}^{\text{LA}} = \frac{w_{jl}}{\sum_{l' \in \Lambda_{j,\sim i}^f} w_{jl'}}. \tag{4}$$

Here $\Lambda_{j,\sim i}^f$ denotes the set of j's neighbors, but i, whose fth feature has the same value as σ_j^f. If nodes i and l are not connected to each other, the link between them is created with the probability p_Δ and the initial weight of the new link is set to w_0. In addition, the weights of all involved links are increased by the reinforcement parameter w_r, irrespective of whether a new link is created or not. **Figure 2B** shows an example of the LA mechanism.

For the link termination, we sequentially apply all of ND, LD, and aging mechanisms. At each time step t, each node is replaced by an isolated node with the probability p_{nd}. In addition, each link in the network is deleted with the probability p_{ld}. For all links that are not deleted, the weight of each link is multiplied by a factor $1 - A$ and the links whose weights are below a threshold value, denoted by w_{th}, are deleted. Here the parameter $A \in [0, 1]$ controls the speed of aging.

We remark that previously studied modifications of the WSN model correspond to this GWSN model with some parameters set to certain values. For example, if each feature in the feature vector can have only one value, i.e., if $q = 1$ for all $f \in \{1,\ldots, F\}$, every node can interact with every other node as $\sigma_i^f = 1$ for all i and f. It means that the GWSN model reduces to the model without the homophilic constraint. In addition, setting p_{nd}, p_{ld}, or A to zero nullifies the ND, LD, or aging mechanism, respectively.

Algorithm 1 Pseudocode for the GWSN model

---

for $i = 1, \ldots, N$ **do**   ▷ Initialization
    Initialize the node i's location (x_i, y_i) and its feature vector $(\sigma_i^1, \ldots, \sigma_i^F)$.
    for $j = 1, \ldots, i-1, i+1, \ldots, N$ **do**
        $w_{ij} \leftarrow 0$.
for $t = 1, \ldots, t_{\max}$ **do**
    for $i = 1, \ldots, N$ **do**   ▷ Global Attachment
        if The node i has no neighbor OR $\text{rand}() < p_r$ **then**
            Randomly choose a feature f from $\{1, \ldots, F\}$.
            Obtain a set Γ_i^f of nodes whose fth feature is the same as σ_i^f and that are not connected to i.
            if Γ_i^f is not empty **then**
                Randomly choose a node j from Γ_i^f with the probability proportional to $r_{ij}^{-\alpha}$ (Eq. (2)).
                Create a link between i and j and $w_{ij} \leftarrow w_0$.
    for $i = 1, \ldots, N$ **do**   ▷ Local Attachment
        Randomly choose a feature f from $\{1, \ldots, F\}$.
        Obtain a set Λ_i^f of i's neighbors whose fth feature is the same as σ_i^f.
        if Λ_i^f is not empty **then**
            Randomly choose a node j from Λ_i^f with the probability proportional to w_{ij} (Eq. (3)).
            $w_{ij} \leftarrow w_{ij} + w_r$.
            Obtain a set $\Lambda_{j,\sim i}^f$ of j's neighbors but i whose fth feature is the same as σ_j^f.
            if $\Lambda_{j,\sim i}^f$ is not empty **then**
                Randomly choose a node l from $\Lambda_{j,\sim i}^f$ with the probability proportional to w_{jl} (Eq. (4)).
                $w_{jl} \leftarrow w_{jl} + w_r$.
                if Nodes i and l are connected **then**
                    $w_{il} \leftarrow w_{il} + w_r$.
                else if $\text{rand}() < p_\triangle$ **then**
                    Create a link between i and l and $w_{il} \leftarrow w_0$.
    for $i = 1, \ldots, N$ **do**   ▷ Node Deletion
        if $\text{rand}() < p_{nd}$ **then**
            Remove all links adjacent to the node i (i.e., $w_{ij} \leftarrow 0$ for every neighbor j).
    for ij in $\{ij | w_{ij} > 0\}$ **do**
        if $\text{rand}() < p_{ld}$ **then**   ▷ Link Deletion
            Remove the link ij (i.e., $w_{ij} \leftarrow 0$).
        else
            $w_{ij} \leftarrow w_{ij} \times (1 - A)$ ▷ Link Aging
            if $w_{ij} < w_{th}$ **then**
                Remove the link ij (i.e., $w_{ij} \leftarrow 0$).

The input parameters of the GWSN model are summarized in **Table 1**, together with the sampling ranges of those parameters for the simulation. Here we respectively fix the values of w_0 and w_{th} to 1 and 0.5, as they set scales only.

2.3 Network Properties

For the systematic comparison between the generated networks by the GWSN model and the empirical networks, we adopt several quantities characterizing network properties as described below.

- Average degree $\langle k \rangle$. It is defined as $\sum_{i=1}^N k_i/N$.
- Degree assortativity coefficient ρ_k. It is defined as $\rho_k = \sum_{kk'} kk' (e_{kk'} - q_k q_{k'})/\sigma_q^2$, where $e_{kk'}$ is the fraction of links connecting nodes with degrees k and k', q_k is the distribution of the excess degree k, and σ_q^2 is the variance of q_k (Newman, 2002, 2003). It characterizes the preference for nodes to attach to others with similar degree.
- Average link weight $\langle w \rangle$. It is defined as $\sum_{ij} w_{ij}/L$, where L is the number of links having positive weights.
- Average clustering coefficient C. It is defined as $\sum_{i=1}^N c_i/N$, where c_i is the local clustering coefficient for the node i. c_i is
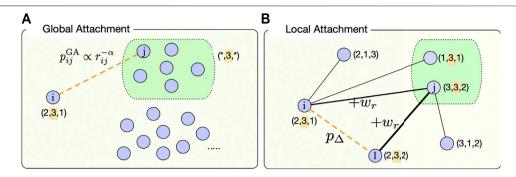
FIGURE 2 | Schematic diagrams of **(A)** the global attachment (GA) and **(B)** the local attachment (LA) of the GWSN model. Each node is described by a feature vector, e.g., $\vec{\sigma}_i = (2, 3, 1)$ for the node i. For each update by the GA and LA mechanisms, a feature is randomly chosen, e.g., $f = 2$, then the node i interacts only with other nodes, say j, with $\sigma_j^2 = \sigma_i^2 = 3$. For the GA mechanism, the dependence on the geographical distance between nodes, r_{ij}, is controlled by the parameter α (see **Eq. 2**). For the LA mechanism, the triadic closure occurs with the probability p_Δ and the involved links increase their weights by w_r.

the number of links between i's neighbors divided by $k_i (k_i - 1)/2$ (Watts and Strogatz, 1998).

- Average link overlap O. It is defined as $\sum_{ij} o_{ij}/L$, where o_{ij} is the link overlap for the link ij. o_{ij} is defined as $\frac{n_{ij}}{(k_i - 1) + (k_j - 1) - n_{ij}}$ with n_{ij} being the number of neighbors common to both i and j (Onnela et al., 2007a,b).

- Pearson correlation coefficient between local clustering coefficient and degree ρ_{ck}. This quantity measures correlation between c_i and k_i for nodes.

- Pearson correlation coefficient between link overlap and weight ρ_{ow}. This quantity measures correlation between o_{ij} and w_{ij} for links.

- Percolation transition points rescaled by the average degree. The Granovetterian community structure can be analyzed by means of the link percolation method (Onnela et al., 2007a,b). For this, links are sorted according to their weights. For the ascending percolation, the weakest links are removed one by one from the network to see how many links need to be removed to disintegrate the global structure of the network. f_c^a denotes the fraction of links removed when the disintegration occurs. Thus, it is called "transition point". f_c^d is similarly defined but by removing the strongest links one by one from the network. Significant difference between f_c^a and f_c^d implies the existence of the Granovetterian community structure. In our work we have recorded the fraction of remaining links at the percolation transition multiplied by the average degree for both ascending and descending link percolation, denoted by $(1 - f_c^a)\langle k \rangle$, and $(1 - f_c^d)\langle k \rangle$, respectively. Generally speaking, the fraction of remaining links multiplied by the average degree indicates the average number of remaining neighbors per node.

2.4 Remarks

The GWSN model has a large number of input parameters as listed in **Table 1**, so that understanding the consequences of the model mechanisms and their effects on the statistical properties of the generated networks becomes a formidable task. So far each of the above mechanisms has been studied independently for

TABLE 1 | Notations and sampling ranges of the input parameters of the GWSN model. Parameters indicated with the asterisk (*) take the values sampled uniformly on the logarithmic scale, while the others take the values sampled uniformly on the linear scale. Parameters indicated with the dagger (†) take only integer values. Values of w_0 and w_{th} are fixed.

Symbol	Input parameter	Sampling range
N	Number of nodes	$[2,000, 5,000]^\dagger$
F	Number of features	$[1,10]^\dagger$
q	Number of traits per feature	$[1,10]^\dagger$
α	Dependence on the geographical distance	$[0, 4]$
p_r	Probability of global attachment	$[10^{-5}, 10^{-2}]^*$
p_Δ	Probability of triadic closure	$[10^{-3}, 10^{-1}]^*$
w_r	Link reinforcement in the local attachment	$[0, 2]$
p_{nd}	Probability of node deletion	$[10^{-5}, 10^{-2}]^*$
p_{ld}	Probability of link deletion	$[10^{-6}, 10^{-3}]^*$
A	Speed of link aging	$[10^{-5}, 10^{-1}]^*$
w_0	Initial weight of new links	1
w_{th}	Threshold weight for removing aged links	0.5

simplicity and theoretical tractability (Murase et al., 2014a, 2015, 2019), however, these mechanisms might coexist and interfere in reality. The non-trivial interplay between such mechanisms makes the analysis of the model highly complex, hence the behavior of the GWSN model cannot be simply predicted based on the previous studies.

In order to overcome such difficulties we will use machine learning technique to perform regression analysis, as a new, general method to handle models with a large number of parameters. By using deep learning we will investigate the input-output relationships, where the inputs are model parameters (**Table 1**) and outputs are network properties (**Table 2**), respectively.

3 PARAMETER-SPACE EXPLORATION WITH DEEP LEARNING METAMODELS

3.1 Regression Analysis

Since the GWSN model has a fairly large number of parameters, it is not easy to understand the model behavior when parameter values change simultaneously, which hinders tuning the parameter values

TABLE 2 | Notations of network properties considered in this work.

Symbol	Network property
$\langle k \rangle$	Average degree
ρ_k	Degree assortativity coefficient
$\langle w \rangle$	Average link weight
C	Average clustering coefficient
O	Average link overlap
ρ_{ck}	Pearson correlation coefficient between local clustering coefficient and degree
ρ_{ow}	Pearson correlation coefficient between link overlap and weight
$(1 - f_c^a)\langle k \rangle$	Percolation transition point rescaled by the average degree: The critical fraction f_c^a of links are removed in ascending order of weights
$(1 - f_c^d)\langle k \rangle$	Percolation transition point rescaled by the average degree: The critical fraction f_c^d of links are removed in descending order of weights

TABLE 3 | Mean squared error (MSE) of the metamodels obtained for each output feature. The variances of the test dataset are also shown as a reference.

Network property	MSE	Variance
$\langle k \rangle$	3.22	676.4
ρ_k	0.000607	0.0162
$\log_{10}\langle w \rangle$	0.000414	0.423
C	0.000363	0.0309
O	0.000247	0.0175
ρ_{ck}	0.00170	0.0730
ρ_{ow}	0.000647	0.0379
$(1 - f_c^a)\langle k \rangle$	0.122	1.23
$(1 - f_c^d)\langle k \rangle$	0.076	0.208

to empirical data. In order to overcome this difficulty, we conduct a regression analysis using machine learning as follows. We first run the simulation of the model a large number of times for sampling the parameter space defined in **Table 1** uniformly randomly. We execute these simulations on the supercomputer Fugaku using four thousand nodes (192 k CPU cores) for 8 h and used CARAVAN framework (Murase et al., 2018) for the job scheduling. Each simulation is implemented as an OpenMP[1] parallelized function running on four threads while the whole program runs as an MPI[2] job. For each set of parameter values, we conduct five independent Monte Carlo runs with different random number seeds. For each generated network, we measure the network properties, such as the average degree and the average link weight, as listed in **Table 2**. The network quantities are measured every 5,000 steps until the simulation finishes at $t = 50,000$, thus yielding 10 entries for each run. Most simulations typically finish in 10 min, however, we find very rare cases that are exceptionally computer time demanding. To avoid such edge cases, we abort the simulation if the average degree exceeds a threshold $k_{th} = 150$ [inspired by Hill and Dunbar (2003)] since the network is no longer regarded as a sparse network for the network sizes used (see **Table 1**). The number of features are 11 since there are 10 input parameters in addition to the simulation time step t. As a result, we obtained training data of about 25 million entries (0.5 million parameter sets).

The sets of input parameters and output network properties are then used to train an artificial neural network (ANN) for the regression analysis. We use an ANN with three hidden layers each of which consists of n_{unit} (=30) fully connected rectifier linear unit (ReLU) with the dense output layer with the linear activation function. We chose ANN because it is one of the standard ways for metamodeling and it indeed showed a significantly better performance for our case compared to the classical regression methods including Ridge, Lasso, and ElasticNet regression using up to third-order polynomial features. All input data are pre-processed by scaling each of the input ranges into the unit interval. The input parameters indicated by the asterisk (*) in **Table 1** are scaled after taking the logarithm. ANNs are trained to minimize the mean squared error for 2,000 epochs using the Adam optimization algorithm with a batch size of 200. We obtained these hyper-parameters by the grid search over the number of hidden layers in [2, 3], the number of units in each layer in [15, 30], and the learning rate in [0.001, 0.01]. The mean squared errors for the test set that are obtained independently from the training set are summarized in **Table 3**. The accuracy of the regression is good enough for our purpose although a better performance could be obtained by a more extensive tuning of hyper-parameters or other regression methods such as support vector regression, Gaussian process, or Kriging (Wang and Shan, 2007; Zhao and Xue, 2010; Ghiasi et al., 2018). We used OACIS for the tuning of hyper-parameters (Murase et al., 2014b, 2017). We conducted regression for each output feature in **Table 2**. For the analysis of the link weights, we take the logarithm of $\langle w \rangle$ since its scale differs by orders of magnitude. Keras (Chollet, 2015), TensorFlow (Abadi et al., 2015), and scikit-learn (Pedregosa et al., 2011) were used for the implementation. The source code for both the simulation and the analysis is available online[3].

An example of the results of the regression analysis is shown in **Figure 3**. In **Figure 3A**, the average degree $\langle k \rangle$ predicted by the machine learning as a function of p_Δ is shown as the dashed curve when $N = 2,000$, $F = 2$, $q = 2$, $\alpha = 0$, $p_r = 0.001$, $w_r = 1$, $p_{nd} = 0.001$, $p_{ld} = 0.01$, $A = 0.003$, and $t = 50,000$ are used. In the same figure, the simulation results are depicted by filled circles. Note that the prediction by the machine learning is not just an interpolation of these simulation results since these simulation results were not included in the training data but conducted separately. **Figure 3**

---

[1]https://www.openmp.org/
[2]https://www.mpi-forum.org/
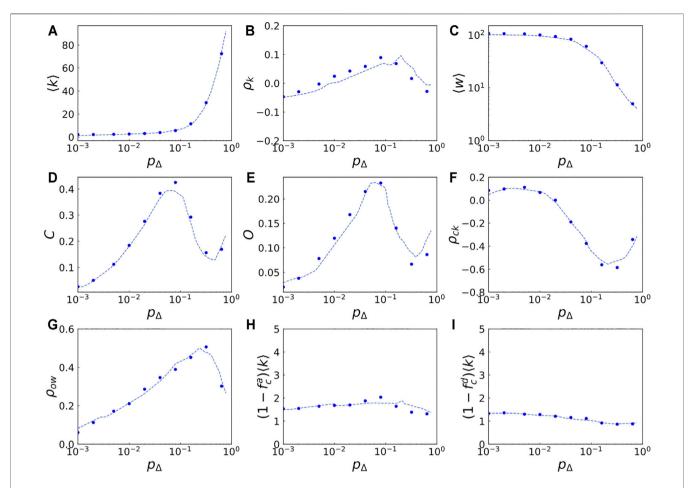
[3]https://github.com/yohm/GWSN_metamodeling

FIGURE 3 | Comparison between the predictions by machine learning and the simulations. Each panel shows how each network property of the generated networks changes with respect to the value of the parameter for the triadic closure, p_Δ. In each panel the dashed curve shows the predictions by machine learning while the filled circles indicate the simulation results averaged over five independent runs, showing that the machine learning predicts the model outputs quite well. Other parameter values are fixed at $N = 2,000$, $F = 2$, $q = 2$, $\alpha = 0$, $p_r = 0.001$, $w_r = 1$, $p_{nd} = 0.001$, $p_{ld} = 0.01$, $A = 0.003$, and $t = 50,000$.

also includes the results for other network properties such as ρ_k, $\langle w \rangle$, C, O, ρ_{ck}, ρ_{ow}, $(1 - f_c^a)\langle k \rangle$, and $(1 - f_c^d)\langle k \rangle$. While the behavior of the average degree $\langle k \rangle$ as a function of p_Δ is easy to interpret, the other quantities show complex non-monotonic dependence on p_Δ. For instance, as shown in **Figure 3D**, the average clustering coefficient C increases with p_Δ for the ranges of $p_\Delta < 0.1$ and $p_\Delta > 0.3$, while it decreases with p_Δ for the range of $0.1 < p_\Delta < 0.3$. Other non-monotonic dependencies on p_Δ are also observed for other network properties. As this example demonstrates, the metamodels reproduce the model outputs, which is useful not only for quantitative parameter tuning but also for understanding the properties of the model. We provide an online interactive chart that allows users to observe the responses of the network properties to the input parameters[4].

We remark as mentioned in **Subsection 2.2** that by setting some parameters to certain values the GWSN model falls back to

the simpler versions in previous studies (Kumpula et al., 2007; Murase et al., 2014a, 2015, 2019). We note that the results obtained by our metamodeling approach are found to be consistent with the previous findings.

3.2 Sensitivity Analysis

The input parameters of the model are not equally important. While some of them have a major impact on the output, others do not alter the results significantly. Sensitivity analysis is a method to evaluate how the uncertainty in the output of a model or system can be divided and attributed to the uncertainty in its inputs, namely, it tells which input parameters are important and which are not. While several methods have been proposed for sensitivity analysis (Saltelli et al., 2007), here we adopt the variance-based sensitivity analysis proposed by Sobol′ (2001) and Saltelli et al. (2010). As we have seen in **Figure 3**, the GWSN model shows highly non-linear dependency on the input parameters. Furthermore, simultaneous perturbations of two or more input parameters often cause greater variations in the output than the sum of the variations caused by each of the

---

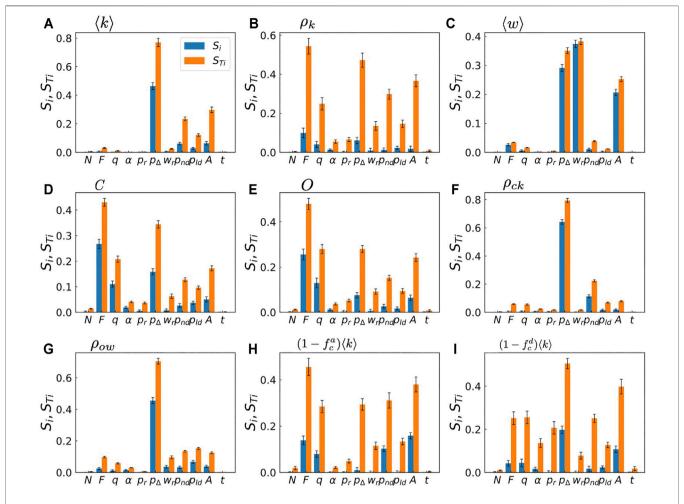[4]https://yohm.github.io/GWSN_metamodeling

FIGURE 4 | Sensitivity analysis of the metamodel. The first-order index S_i and the total-effect index S_{Ti} of each input parameter (**Table 1**) against each output (**Table 2**) are shown with error bars indicating the 95% confidence intervals. See **Eqs. 5, 6** for definitions of S_i and S_{Ti}, respectively.

perturbations alone. The variance-based method is effective for such models because it measures sensitivity across the entire input space (i.e., it is a global method), it deals with nonlinear responses, and it takes into account the effect of interactions in non-additive systems. While the variance-based methods generally require a larger number of sampling than other methods, they are easily calculated once we developed a metamodel as is done in **Section 2**.

In variance-based sensitivity analysis, the sensitivity of the output to an input variable is quantified by the amount of variance caused by the fluctuation of the input. Consider a generic model $Y = f(X_1, X_2, \ldots, X_n)$, i.e., the output Y is a function of n input variables $\mathbf{X}$. Due to the uncertainties of the input parameters, each of which is considered as an independent random variable following the probability distribution $p_i(X_i)$, the output Y has also some uncertainty. Let us denote the variance of Y as $V[Y] = E[Y^2] - E[Y]^2$, where the operator $E[\cdot] \equiv \int \cdot p(\mathbf{X}) d\mathbf{X}$ indicates the expected value. We calculate the two indices representing the sensitivity of Y to X_i, namely, the first-order

sensitivity index S_i and the total-effect sensitivity index S_{Ti}. These indices are defined as

$$S_i = \frac{V_i[E_{\sim i}[Y]]}{V(Y)}, \tag{5}$$

$$S_{Ti} = \frac{E_{\sim i}[V_i[Y]]}{V(Y)}, \tag{6}$$

where $E_i[\cdot]$ and $E_{\sim i}[\cdot]$ denote the expected values averaged over X_i and over all input variables but X_i, respectively. The operator $V_i[\cdot] \equiv E_i[.^2] - E_i[\cdot]^2$ denotes the variance taken over X_i. The first-order index S_i indicates the expected reduction of variance when the input X_i could be fixed. The interactions between the other inputs are not taken into account. On the other hand, the total-effect index S_{Ti} tells us the importance of ith input taking into account all the higher-order interactions in addition to the first-order effect. We used SAlib python package to calculate the sensitivity indices (Herman and Usher, 2017).

Figure 4 shows the results of the sensitivity analysis. In **Figure 4A** we show the first-order and the total-effect indices for the average degree $\langle k \rangle$. The figure indicates that the average degree is the most sensitive against the change in p_Δ whether p_Δ changes independently or with other parameters at the same time. The next important factors for $\langle k \rangle$ are the parameters for link termination, i.e., p_{nd}, p_{ld}, and A. Since they show large values of S_{Ti}, these parameters seem to strongly interact with other parameters. The average link weight $\langle w \rangle$ displays a different tendency as shown in **Figure 4C**. It shows strong dependency mainly on w_r and A in addition to p_Δ, which is reasonable since w_r and A control the link reinforcement and the link aging, respectively.

If we look at the indices for clustering coefficient shown in **Figure 4D**, we find a different behavior. The parameters for the homophilic interaction, i.e., F and q, have high values of S_i and S_{Ti}, indicating that these parameters affect the clustering properties of the emergent networks, while they play minor role in determining the average degree as shown in **Figure 4A**. A similar tendency is observed for the average link overlap O in **Figure 4E**; this is reasonable since both the clustering coefficient and the link overlap essentially quantify the frequency of closed triads.

Finally, some input parameters such as N, t, α, and p_r lead to relatively small values of S_{Ti} for most outputs, indicating that these input parameters rarely affect the outputs even when they change with other parameters. The insignificance of N and t indicates that the finite-size effect is negligibly small and the simulations reach statistically stationary state for most cases. Consequently, the model can be simplified by fixing these parameters to arbitrary values within their sampling ranges.

4 SUMMARY AND DISCUSSION

In this paper we have focused on studying the formation of social network using an agent-based GWSN model, which incorporates several realistic extensions, such as homophilic interactions and link aging, to the original WSN model (Kumpula et al., 2007). The effects of these additional mechanisms were studied independently in our previous papers (Murase et al., 2014a, 2015, 2019) for simplicity and analytical tractability. However, in real world social networks these mechanisms coexist, which, if incorporated within one model, poses a challenge, because they often interact in non-trivial ways. In the framework of modeling this means that with the number of model parameters the number of possible combination effects increases super-linearly. Such difficulties are common for agent-based models (Bonabeau, 2002; Sayama, 2015). Thus, the use of agent-based models has been limited for a long time to either research for qualitative understanding using relatively simplistic models, or research that uses complex models but does not fully explore the parameter space.

Instead of the traditional approach to build first a simple model and add elements one by one to study in sequence their consequences, we started from a generalized model with all the elements incorporated and then studied its behavior using large-scale computation and metamodeling. Hence a massive number of simulations were performed using a supercomputer to sample a high-dimensional input space, and the outputs of the model were then used as training data for machine learning. We demonstrated that the metamodel, obtained by a deep multilayer perceptron, accurately reproduces the behavior of the model over a wide range of input parameters. Although massive computation is needed for the preparation of the training data, these simulations were efficiently executed using a supercomputer since they can be calculated independently and in parallel. Once the learning phase is completed, it is computationally quite cheap to calculate a prediction, which is useful for various purposes including the understanding of model behavior, tuning of parameter values, and sensitivity analysis.

As a demonstration, we conducted sensitivity analysis of our metamodels and identified which input parameters influence the output most, i.e., the properties of the generated networks. In addition, the sensitivity analysis tells us which parameters have negligible effect on the output, indicating that these parameters can be fixed to certain values in their sampling ranges without the loss of generality and for simplifying the model. Therefore, our computational approach enables us to study a model with a larger number of parameters in a systematic and quantitative way. Taking the full GWSN model with the traditional approach of parameter fitting would cause severe, if not unsolvable, difficulties, while our metamodeling approach allowed us to study it with reasonable computational effort.

A good metamodel will also help us finding appropriate parameters to generate a network with some desired properties. The problem would be formulated as a mathematical optimization, in which the objective function represents a certain distance between the desired and the generated networks. We leave such an inverse problem for future studies, since there is another challenge, namely how to define such an objective function appropriately.

There are still a couple of open issues for future research. First of all, the GWSN model should be extended further. While we have so far moved forward from the simplest plausible model to more realistic ones, there is still a long way to go, considering the complexity of real social systems. A possible future direction is the extension of the output network characteristics. We quantified the network properties only by scalar values such as the average degree, yet we did not pay attention to more informative quantities, such as the degree distribution. This is mainly for the simplicity in developing metamodels, however the heterogeneity of the network, often characterized by the broad distributions, is of crucial importance in investigating not only the network properties but also the dynamical processes taking place on those networks such as epidemic spreading (Pastor-Satorras et al., 2015) and random walks (Masuda et al., 2017). It is expected that metamodels approximating the distributions obtained from the agent-based models will be developed in the near future. Another promising future direction would be the extension of models toward temporal, multiplex, and higher-order networks

(Holme and Saramäki, 2012; Kivelä et al., 2014; Battiston et al., 2020) as there has been increasing demand for representing social Big Data in these terms. It would be straightforward to extend the metamodeling approach of this paper beyond the simple pairwise interactions between the nodes of the network.

DATA AVAILABILITY STATEMENT

The datasets presented in this study can be found in online repositories. The names of the repository/repositories and accession number(s) can be found below: https://github.com/yohm/GWSN_metamodeling.

AUTHOR CONTRIBUTIONS

YM, H-HJ, JT, JK, and KK developed the model, conceived and designed the experiments. YM conducted the experiments and analyzed the results. YM, H-HJ, JT, JK, and KK wrote and reviewed the article.

FUNDING

ACKNOWLEDGMENTS

REFERENCES

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., et al. (2015). *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org

Axelrod, R. (1997). The Dissemination of Culture: A Model with Local Convergence and Global Polarization. *J. Conflict Resolution* 41, 203–226. doi:10.1177/0022002797041002001

Barabási, A.-L., and Pósfai, M. (2016). *Network Science*. Cambridge: Cambridge University Press.

Barthélemy, M. (2011). Spatial Networks. *Phys. Rep.* 499, 1–101. doi:10.1016/j.physrep.2010.11.002

Battiston, F., Cencetti, G., Iacopini, I., Latora, V., Lucas, M., Patania, A., et al. (2020). Networks beyond Pairwise Interactions: Structure and Dynamics. *Phys. Rep.* 874, 1–92. doi:10.1016/j.physrep.2020.05.004

Boccaletti, S., Bianconi, G., Criado, R., del Genio, C. I., Gómez-Gardeñes, J., Romance, M., et al. (2014). The Structure and Dynamics of Multilayer Networks. *Phys. Rep.* 544, 1–122. doi:10.1016/j.physrep.2014.07.001

Bonabeau, E. (2002). Agent-based Modeling: Methods and Techniques for Simulating Human Systems. *Proc. Natl. Acad. Sci.* 99, 7280–7287. doi:10.1073/pnas.082080899

Centola, D., González-Avella, J. C., Eguíluz, V. M., and San Miguel, M. (2007). Homophily, Cultural Drift, and the Co-evolution of Cultural Groups. *J. Conflict Resolution* 51, 905–929. doi:10.1177/0022002707307632

Chollet, F. (2015). Keras. Available at: https://keras.io.

Erdös, P., and Rényi, A. (1960). *On the Evolution of Random Graphs*. Publication of the Mathematical Institute of the Hungarian Academy of Sciences, 5, 17–61.

Gandica, Y., Charmell, A., Villegas-Febres, J., and Bonalde, I. (2011). Cluster-size Entropy in the Axelrod Model of Social Influence: Small-World Networks and Mass media. *Phys. Rev. E* 84, 046109. doi:10.1103/PhysRevE.84.046109

Ghiasi, R., Ghasemi, M. R., and Noori, M. (2018). Comparative Studies of Metamodeling and Ai-Based Techniques in Damage Detection of Structures. *Adv. Eng. Softw.* 125, 101–112. doi:10.1016/j.advengsoft.2018.02.006

Granovetter, M. S. (1973). The Strength of Weak Ties. *Am. J. Sociol.* 78, 1360–1380. doi:10.2307/2776392

Herman, J., and Usher, W. (2017). SALib: An Open-Source python Library for Sensitivity Analysis. *J. Open Source Softw.* 2. doi:10.21105/joss.00097

Hill, R. A., and Dunbar, R. I. M. (2003). Social Network Size in Humans. *Hum. Nat.* 14, 53–72. doi:10.1007/s12110-003-1016-y

Holme, P., and Saramäki, J. (2012). Temporal Networks. *Phys. Rep.* 519, 97–125. doi:10.1016/j.physrep.2012.03.001

Jo, H.-H., Murase, Y., Török, J., Kertész, J., and Kaski, K. (2018). Stylized Facts in Social Networks: Community-Based Static Modeling. *Physica A: Stat. Mech. its Appl.* 500, 23–39. doi:10.1016/j.physa.2018.02.023

Jo, H.-H., Pan, R. K., and Kaski, K. (2011). Emergence of Bursts and Communities in Evolving Weighted Networks. *PLoS ONE* 6, e22687. doi:10.1371/journal.pone.0022687

Karsai, M., Jo, H.-H., and Kaski, K. (2018). *Bursty Human Dynamics*. Cham: Springer International Publishing.

Kertész, J., Török, J., Murase, Y., Jo, H.-H., and Kaski, K. (2021). "Modeling the Complex Network of Social Interactions." in *Pathways between Social Science and Computational Social Science*. Berlin: Springer, 3–19.

Kivelä, M., Arenas, A., Barthelemy, M., Gleeson, J. P., Moreno, Y., and Porter, M. A. (2014). Multilayer Networks. *J. Complex Networks* 2, 203–271. doi:10.1093/comnet/cnu016

Kossinets, G., and Watts, D. J. (2006). Empirical Analysis of an Evolving Social Network. *Science* 311, 88–90. doi:10.1126/science.1116869

Kumpula, J. M., Onnela, J.-P., Saramäki, J., Kaski, K., and Kertész, J. (2007). Emergence of Communities in Weighted Networks. *Phys. Rev. Lett.* 99, 228701. doi:10.1103/physrevlett.99.228701

Marsili, M., Vega-Redondo, F., and Slanina, F. (2004). The Rise and Fall of a Networked Society: A Formal Model. *Proc. Natl. Acad. Sci.* 101, 1439–1442. doi:10.1073/pnas.0305684101

Masuda, N., Porter, M. A., and Lambiotte, R. (2017). Random Walks and Diffusion on Networks. *Phys. Rep.* 716-717, 1–58. doi:10.1016/j.physrep.2017.07.007

McPherson, M., Smith-Lovin, L., and Cook, J. M. (2001). Birds of a Feather: Homophily in Social Networks. *Annu. Rev. Sociol.* 27, 415–444.

Menczer, F., Fortunato, S., and Davis, C. A. (2020). *A First Course in Network Science*. Cambridge: Cambridge University Press.

Min, B., and Miguel, M. S. (2017). Fragmentation Transitions in a Coevolving Nonlinear Voter Model. *Scientific Rep.* 7, 12864. doi:10.1038/s41598-017-13047-2

Murase, Y., Jo, H.-H., Török, J., Kertész, J., and Kaski, K. (2015). Modeling the Role of Relationship Fading and Breakup in Social Network Formation. *PLoS ONE* 10, e0133005. doi:10.1371/journal.pone.0133005

Murase, Y., Jo, H.-H., Török, J., Kertész, J., and Kaski, K. (2019). Structural Transition in Social Networks: The Role of Homophily. *Scientific Rep.* 9, 4310. doi:10.1038/s41598-019-40990-z

Murase, Y., Matsushima, H., Noda, I., and Kamada, T. (2018). Caravan: a Framework for Comprehensive Simulations on Massive Parallel Machines." in *International Workshop on Massively Multiagent Systems*. Cham: Springer, 130–143.

Murase, Y., Török, J., Jo, H.-H., Kaski, K., and Kertész, J. (2014a). Multilayer Weighted Social Network Model. *Phys. Rev. E* 90, 052810. doi:10.1103/physreve.90.052810

Murase, Y., Uchitane, T., and Ito, N. (2014b). A Tool for Parameter-Space Explorations. *Phys. Proced.* 57, 73–76. doi:10.1016/j.phpro.2014.08.134

Murase, Y., Uchitane, T., and Ito, N. (2017). An Open-Source Job Management Framework for Parameter-Space Exploration: OACIS. *J. Phys. Conf. Ser.* 921, 012001. doi:10.1088/1742-6596/921/1/012001

Newman, M. E. J. (2002). Assortative Mixing in Networks. *Phys. Rev. Lett.* 89, 208701. doi:10.1103/physrevlett.89.208701

Newman, M. E. J. (2003). Mixing Patterns in Networks. *Phys. Rev. E* 67, 026126. doi:10.1103/physreve.67.026126

Newman, M. E. J. (2018). *Networks*. second edition edn. Oxford, United Kingdom ; New York, NY: United States of America: Oxford University Press.

Onnela, J.-P., Saramäki, J., Hyvönen, J., Szabó, G., Argollo de Menezes, M., Kaski, K., et al. (2007a). Analysis of a Large-Scale Weighted Network of One-To-One Human Communication. *New J. Phys.* 9, 179. doi:10.1088/1367-2630/9/6/179

Onnela, J. P., Saramäki, J., Hyvönen, J., Szabó, G., Lazer, D., Kaski, K., et al. (2007b). Structure and Tie Strengths in mobile Communication Networks. *Proc. Natl. Acad. Sci.* 104, 7332–7336. doi:10.1073/pnas.0610245104

Pastor-Satorras, R., Castellano, C., Van Mieghem, P., and Vespignani, A. (2015). Epidemic Processes in Complex Networks. *Rev. Mod. Phys.* 87, 925–979. doi:10.1103/RevModPhys.87.925

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., et al. (2011). Scikit-learn: Machine Learning in Python. *J. Machine Learn. Res.* 12, 2825–2830.

Saltelli, A., Annoni, P., Azzini, I., Campolongo, F., Ratto, M., and Tarantola, S. (2010). Variance Based Sensitivity Analysis of Model Output. Design and Estimator for the Total Sensitivity index. *Comput. Phys. Commun.* 181, 259–270.

Saltelli, A., Ratto, M., Andres, T., Campolongo, F., Cariboni, J., Gatelli, D., et al. (2007). *Global Sensitivity Analysis. The Primer*. Chichester, United Kingdom: John Wiley and Sons, Ltd. doi:10.1002/9780470725184

Sayama, H. (2015). *Introduction To the Modeling And Analysis Of Complex Systems*. deluxe color ed edn. Geneseo, NY: Milne Library.

Sobol', I. (2001). Global Sensitivity Indices for Nonlinear Mathematical Models and Their Monte Carlo Estimates. *Math. Comput. Simul.* 55, 271–280. doi:10.1016/S0378-4754(00)00270-6

Tilles, P. F. C., and Fontanari, J. F. (2015). Diffusion of Innovations in Axelrod's Model. *J. Stat. Mech. Theor. Exp.*, P11026. doi:10.1088/1742-5468/2015/11/P11026

Török, J., Murase, Y., Jo, H.-H., Kertész, J., and Kaski, K. (2016). What Big Data Tells: Sampling the Social Network by Communication Channels. *Phys. Rev. E* 94, 052319. doi:10.1103/physreve.94.052319

Vazquez, F., González-Avella, J. C., Eguíluz, V. M., and San Miguel, M. (2007). Time-scale Competition Leading to Fragmentation and Recombination Transitions in the Coevolution of Network and States. *Phys. Rev. E* 76, 046120. doi:10.1103/PhysRevE.76.046120

Vazquez, F., and Redner, S. (2007). Non-monotonicity and Divergent Time Scale in Axelrod Model Dynamics. *Europhysics Lett. (Epl)* 78, 18002. doi:10.1209/0295-5075/78/18002

Wang, G. G., and Shan, S. (2007). Review of Metamodeling Techniques in Support of Engineering Design Optimization *ASME. J. Mech. Des.* 129 (4), 370–380. doi:10.1115/1.2429697

Watts, D. J., and Strogatz, S. H. (1998). Collective Dynamics of 'small-World' Networks. *Nature* 393, 440–442. doi:10.1038/30918

Zhao, D., and Xue, D. (2010). A Comparative Study of Metamodeling Methods Considering Sample Quality Merits. *Struct. Multidisciplinary Optimization* 42, 923–938. doi:10.1007/s00158-010-0529-3