A Network Synthesis and Analytics Pipeline with Applications to Sustainable Energy in Smart Grid

Swapna Thorve
Biocomplexity Institute and Initiative
University of Virginia
Charlottesville, VA, USA
st6ua@virginia.edu

Aparna Kishore
Biocomplexity Institute and Initiative and
Dept. of Computer Science
University of Virginia
Charlottesville, VA, USA
ak8mj@virginia.edu

Dustin Machi
Biocomplexity Institute and Initiative
University of Virginia
Charlottesville, VA, USA
dm8qs@virginia.edu

S. S. Ravi
Biocomplexity Institute and Initiative
University of Virginia
Charlottesville, VA, USA
ssravi0@gmail.com

Madhav V. Marathe
Biocomplexity Institute and Initiative and
Dept. of Computer Science
University of Virginia
Charlottesville, VA, USA
marathe@virginia.edu

2023 IEEE 19th International Conference on e-Science (e-Science) | 979-8-3503-2223-1/23/$31.00 ©2023 IEEE | DOI: 10.1109/E-SCIENCE58273.2023.10254867

***Abstract*—Transitioning to clean and low-carbon energy is becoming a crucial goal for many entities in the energy systems sector such as governments, power utilities, and policymakers. This shift to clean energy is supported by a diverse portfolio of data products such as satellite data, smart meter data, power networks, green energy datasets (e.g., solar installations & electric vehicles), microgrid networks, and building stock data. Among these, network datasets are becoming increasingly common in addressing a wide array of issues in residential energy, especially in applications that focus on social good. Thus, streamlining the process of generating different types of networks will be helpful. In this work, we propose a versatile network synthesis and analytics pipeline developed using software design principles that make it modular, scalable, and extensible. Three case studies are presented to illustrate the significance of network data in sustainable energy applications.**

***Index Terms*—Network generation and analytics, Modeling & simulation, Energy systems, Solar adoption, Synthetic population, Placement of EV charging stations, Scalability**

I. Introduction

A. Energy systems and network data

Entities in the energy sector from local/state governments, power companies (i.e., utilities), policymakers, researchers, to different types of energy end users in the residential (e.g., household) and commercial (e.g., office buildings) energy sectors are focusing on tackling the challenges in the energy systems landscape. Transitioning to sustainable energy systems is one of the primary goals for achieving net zero carbon emissions. Examples of sustainable and renewable energy systems include wind, solar, and hydropower. Interdisciplinary teams are working towards providing access to clean and affordable energy for all through different channels such as improved public policies and innovative economic incentives. Many researchers are turning to machine learning (ML) and artificial intelligence (AI) tools to facilitate the development and operation of energy system analytics and sustainable energy modeling and simulation (M&S) platforms [1].

The growing interest in the application of ML tools is a result of energy-related datasets becoming available at a fast rate. For example, data from smart meters enable energy companies, consumers, and researchers to learn about household energy consumption patterns. Satellite data has played a crucial role in sustainable energy applications such as solar photovoltaic systems, offshore wind projects, hydropower projects, and geothermal energy [2]. This valuable information can be used to design fair energy policies at various spatial and temporal levels. A Time Of Use pricing scheme has been implemented in some geographical regions (e.g., California) in order to incentivize the shifting of load from evening peak times to non-peak times so as to reduce stress on the power grid. Household data attributes such as socio-economic status and affordability can be used to build decision models and optimization frameworks for retrofitting building stock. Synthetic energy datasets such as those discussed in [3]–[5] are becoming popular for building detailed bottom-up modeling frameworks and/or studying the effects of dynamic pricing, renewables, and EV.

Apart from demographic, spatial, and socio-economic data, network/graph datasets have started gaining attention in the field of sustainable energy (see e.g., [4], [6]–[8]). Many M&S efforts have focused on studying the effects of peer/neighborhood networks on energy technology adoption such as solar [9], [10] and electric vehicles (EV) [11], [12]. Integrating details of the power distribution networks is crucial in understanding the stability of the power grid with increasing EV penetration (see e.g., [13]). Networks have also been used in the literature to identify potential microgrids and peer-

to-peer energy trading [14]. Du et al. [8] use a weighted directed social network for modeling the propagation of energy savings. Bale et al. [6] develop a linear model that calculates community influence from peer networks along with demographics to predict whether a household will adopt new energy technologies.

Many of these frameworks employ similar networks in their models. For example, references [15] and [10] both employ similar neighborhood networks in developing solar adoption models. However, the process for generating these networks is not streamlined. This is mainly due to the lack of commonality in the design of these network generation models. To improve human productivity, it is beneficial to separate and automate the network generation operations into a workflow/pipeline for improved efficiency, reproducibility, and scalability using software design principles. In this work, we propose a network generation pipeline named *NetXpipe* for sustainable energy applications using heterogeneous datasets.

B. Background and related work

This section provides relevant background and examples of software architectures and design principles for developing pipelines. Architecting scalable, modular, and robust pipelines (or workflows) using big data, machine learning, and software-defined infrastructure is gaining widespread attention in different domains. Microservices-oriented architecture (MSA) [16]–[18] is one such architectural style that offers extensibility for accommodating big data design and provides flexibility for rapid development and integration of multiple systems/processes. MSA consists of loosely coupled, reusable, specialized, and independent modules/functions (i.e., microservices) that often work independently of one another. This type of design is able to work well with complex simulations and experiment designs by introducing "separation of concerns" through loose coupling and independence. The microservices offer the modularity and extensibility to incorporate new models or behaviors in a large-scale system without having to make big changes to the existing codebase. Similarly, the extensibility feature of MSA makes the addition of new datasets (or data types) to such systems easier in terms of human hours since components are not tightly coupled and have a workflow in place for adding specialized services and/or modifying existing services. In this work, we build a pipeline using MSA principles for graph synthesis and showcase its use in sustainable energy applications.

Pipelines have been designed and deployed in various domains (e.g., bioinformatics, smart grid, IoT, online games) for automation and organization of tasks, improved efficiency, modularity, re-usability, and extensibility. We provide four separate examples below to emphasize the significance of various aspects of pipeline design in different domains. Cedeno-Mieles et al. [19] propose a novel pipeline architecture for networked social science experiments. Their MSA software system consists of pipelines for *data analytics*, *model property inference*, *experiment models and analyses* that allow for human in-loop and repeated network experiments that deal with detailed human behavior data. The MSA design also offers automation of tasks such as analyses of experiments, easy addition of computational models due to a flexible and extensible pipeline format, and a resilient data model. In another application, Asaithambi et al. [20] propose a Microservice-Oriented Big Data Architecture (MOBDA) for analytics in intelligent transportation systems (ITS). There, the MSA architecture is exploited to support real-time processing of massive data. MSA was able to offer better response time, flexibility, scalability, and seamless integration with hybrid ML data architectures as compared to traditional ITS architectures.

Thorve et al. [21] propose a suite of five composable pipeline templates for designing large-scale bottom-up simulations. The templates are designed for data processing, modeling, validation, visual analytics, and a pipeline for parallelizable operations. Each of these encapsulates elements of some of the most important tasks in modern-day complex systems. This work shows the application of MSA design by building a flexible and distributed software tool for large-scale residential energy simulation. Simmhan et al. [22] provides an excellent example of intelligent demand response in smart grid using scalable cloud-based software platforms. The system is designed to have workflows for dynamic data ingestion, deployment of machine learning models for peak time demand forecast, and a visualization portal for exploring consumption patterns. The use of this system is demonstrated for various stakeholders in a university campus smart grid for reducing peak-time energy demand.

Rodrigues et al. [23] employs a graph-based approach for designing extensible pipelines. They show that pipelines are able to encapsulate complex analysis tasks by breaking them down into micro-tasks in tedious experiments, thus making the complicated tasks easier to debug and update. Du et al. [24] introduce 'GraphGT', a machine learning graph generation/modeling pipeline that simplifies the process for data operations, experimental setup, and model evaluation. Their work generated 36 datasets from 9 domains and made them publicly available. Purohit et al. [25] describe a data and graph generation tool for modeling adversary activity. It is a scalable graph generation and modeling tool to produce realistic and diverse sets of graphs that are useful in developing subgraph matching algorithms.

We have described numerous examples of the advantages of MSA designs in different areas. However, it should be noted that this architectural style comes with some caveats. For example, when systems become extremely large and complex, it is hard to test and debug issues related to service coordination. It is also challenging to manage such a large number of microservices. Since microservices are self-contained, they rely heavily on the network for communication which may increase network latency and traffic. As this architectural style is employed in different domains, one should consider potential trade-offs of MSA with other software design styles depending on the use case. For example, some drawbacks of MSA can be addressed by blending MSA principles with other frameworks thereby providing a hybrid architecture (as shown

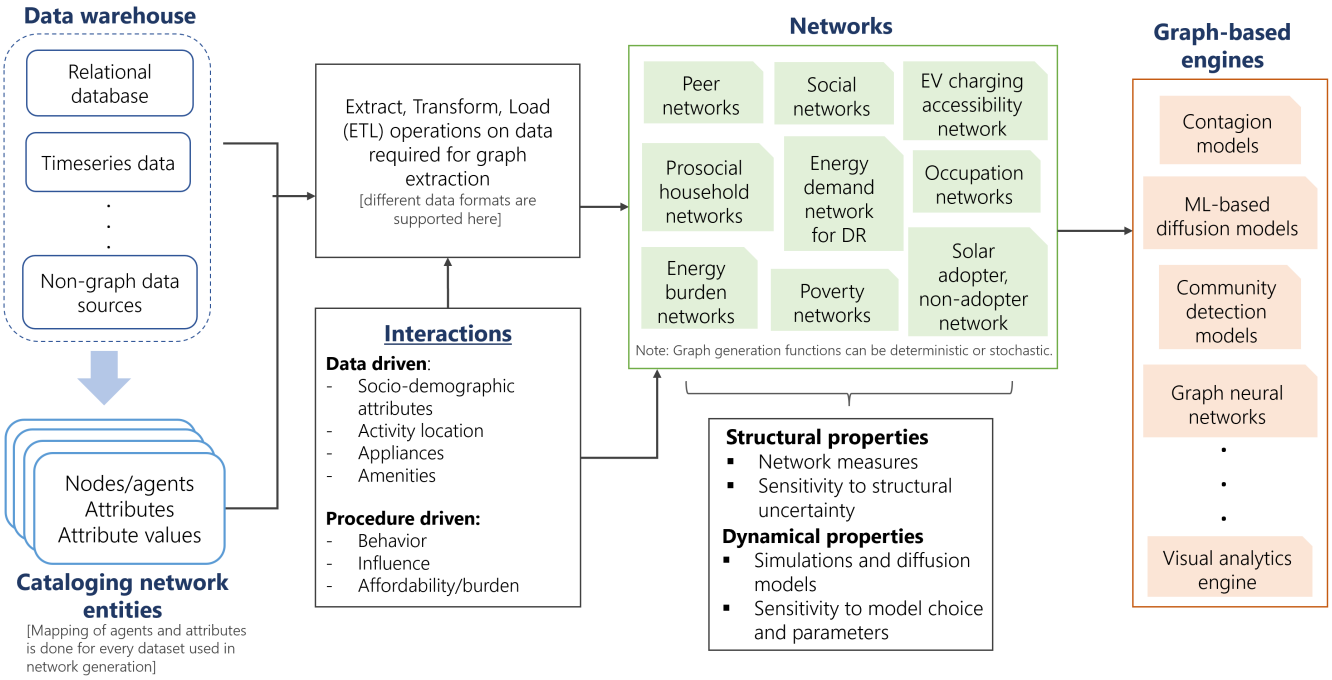


Fig. 1: Abstract view of the proposed graph synthesis pipeline

in Asaithambi et al. [20]).

C. Contributions

In this work, we present a network generation pipeline named *NetXpipe* for sustainable energy applications for streamlining the process of generating networks for sustainable and social good applications in the energy domain. Although we focus on sustainable energy applications, *NetXpipe* can be easily used in other domains as well.

We employ a microservices-oriented architecture (MSA) for developing *NetXpipe*. The network generation pipeline, which is modular, scalable, and extensible, is designed for plugging in network-style data in M&S systems in different applications. We have made 21 networks generated using *NetXpipe* public through the net.science platform.[1].

The value of the network generation pipeline is demonstrated through three case studies. The first case study considers retrofitting as a social contagion and studies the spread of adoption of this energy technology over different types of population networks with varying edge probability generated by our pipeline. This study highlights the extensibility and scalability of the pipeline. The second case study illustrates the use of the pipeline for studying the influence of peer effects on solar adoption in rural regions. This study highlights the scalability and modularity of the pipeline. The third case study builds an innovative network using synthetic populations and data on EV charging stations to study the accessibility of EV chargers for all types of dwellings in a region. This case study shows that the pipeline is able to work with different types of data sources and formats in generating networks.

[1]https://net.science/files/resources/datasets/NetXpipe_graphs_wsc2023/

D. Paper outline

The proposed network generation pipeline is described in Section II for applications in sustainable energy. Next, we present three case studies in Section III to demonstrate the usefulness of the network generation pipeline in different settings and conclude in Section IV.

II. Network Generation Pipeline (NetXpipe)

A. System Overview

Figure 1 shows a logical view of the proposed system. The graph processing system has a data warehouse that stores multiple types of data from different sources that are useful in generating networks. The datasets are analyzed to discover entities and their features. Then, this information is cataloged in the system to create a set of nodes $\mathcal{V}$ and an attribute list A_v for each $v \in \mathcal{V}$. For example, if each node v represents a house in a city, attributes associated with v may include its coordinates (i.e., latitude and longitude values), its address, number of people living in the house, household income, whether the house has solar panels, etc. We have designed several network generation modules (highlighted in green in Figure 1) that are suitable for generating different types of networks with the current datasets. Users can choose a type of network, node(s) and the attributes that are used for edge generation.

The goal of the proposed pipeline *NetXpipe* is to construct a graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ given user preferences. Let v be an object represented by a vertex/node in a network. As mentioned above, for each $v \in \mathcal{V}$, there is an attribute list A_v that can be utilized in generating networks. Let $a_{v,k}$ be the kth node attribute in A_v. Let $f(v_i, v_j)$ be a predicate which is a function

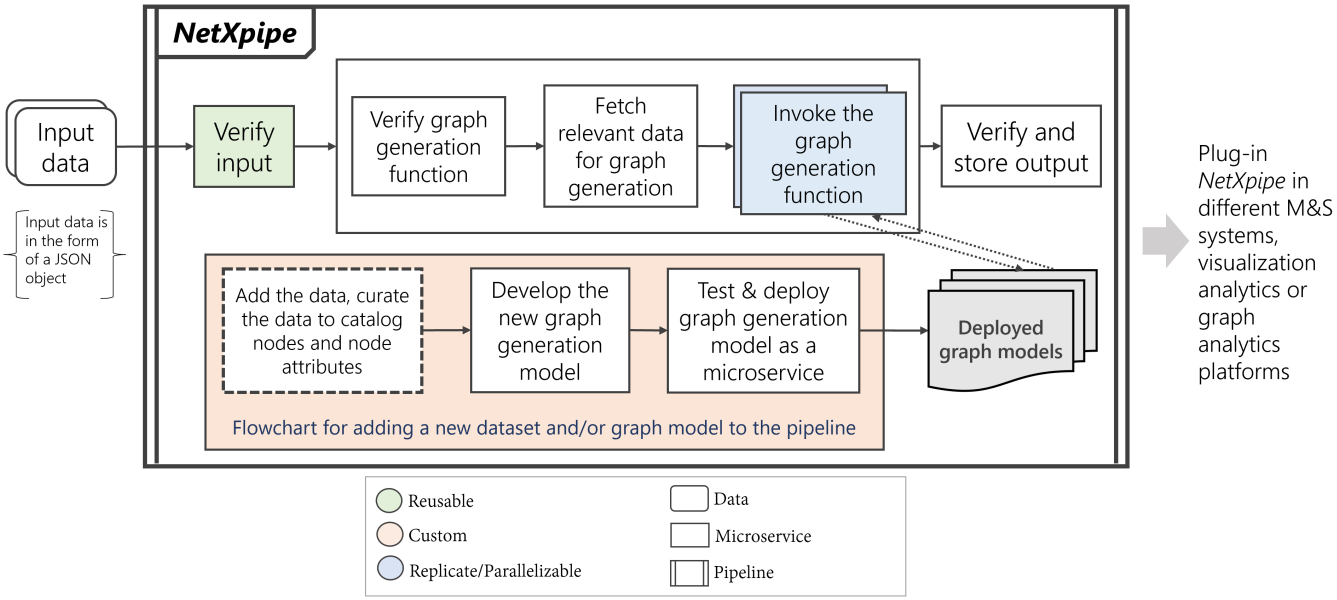


Fig. 2: Network generation pipeline *NetXpipe* overview. The text in each rectangle defines the goal of the microservices bundled in the pipeline. The colors depicted in the legend describe the different types of microservices employed in the pipeline. Note that, not only the pipeline *NetXpipe* itself is extensible, modular, and flexible, but also the individual components can be easily replaced/updated or new graph generation models can be incorporated into the pipeline.

that determines whether there should be an edge between two nodes v_i and v_j. The predicate can be deterministic or probabilistic. In the deterministic case, the edge $\{v_i, v_j\}$ is added to the network if the value $f(v_i, v_j)$ is TRUE. In the probabilistic case, there is an additional input, namely a probability value p. If the function f returns the value TRUE, then the edge $\{v_i, v_j\}$ is added to the network with probability p. We now present two examples of such predicates.

Example 1: Suppose a user wants to generate a peer network of households in Arlington, VA. The pipeline identifies the network type as a 'peer network' where each node corresponds to a house. We first consider the deterministic case. Let the function f denote the following simple condition: add an edge between v_i and v_j if the distance between the houses represented by the two nodes is at most 5 miles. Let 'location' denote the name of the attribute representing the coordinates of a house. Thus, the coordinates of the house are represented by v_i are given by the attribute value $a_{v_i,\text{location}}$. Let D denote the function that returns the distance between two locations. Thus, the function $f(v_i, v_j)$ is represented within the pipeline by the condition "$D(a_{v_i,\text{location}}, a_{v_j,\text{location}}) \leq$ distance_constraint", where the the system variable distance_constraint has the value 5. If a user chooses the probabilistic version and specifies a probability value (say) 0.7, then the edge $\{v_i, v_j\}$ gets added with probability 0.7 when the above condition is TRUE.

Example 2: This example specifies a condition involving two attributes for adding edges. As in Example 1, assume that a user wants to generate a peer network of households. Here, the function $f(v_i, v_j)$ represents the following condition: the annual income of each household is at most \$40,000 and the distance between the houses is at most 2 miles. Using system variables income_constraint to represent the value \$40,000 and distance_constraint to represent the value 2, the conditions specified by f can be represented using node attributes as follows.

$$\begin{aligned} &\text{C1:} \quad D(a_{v_i,\text{location}}, a_{v_j,\text{location}}) \leq \text{distance\_constraint} \\ &\text{C2.1:} \quad a_{v_i,\text{income}} \leq \text{income\_constraint} \\ &\text{C2.2:} \quad a_{v_j,\text{income}} \leq \text{income\_constraint} \end{aligned}$$

$$\text{if (C1 and C2.1 and C2.2) then add edge } \{v_i, v_j\} \qquad (1)$$

The algorithm for generating a peer network is quite straightforward for the given input (Equation (1)). The algorithm selects all households within the distance threshold and adds other conditions enforced by the user input to generate the predicate function. For a pair of nodes v_i and v_j, If the value of $f(v_i, v_j)$ is TRUE, then an edge is generated between the two nodes. In the probabilistic case, when the value of f is TRUE, a simple Bernoulli trial with the specified probability is used to decide whether or not the edge $\{v_i, v_j\}$ is added. While the above examples presented simple predicates for edge generation, one can also use a complex algorithm (e.g., [26]) that determines if two nodes should be connected by an edge. In general, there can be more than one node type in a graph. The above examples describe a simple but popular type of network that has a wide array of applications in different domains. The individual case studies presented in later sections describe how specific datasets are instantiated to generate particular networks.

TABLE I: Examples of networks generated by *NetXpipe* and used in case studies.

Name	Region	Edge prob.	#Nodes	#Edges	Avg. degree	Max. kcore	Edge direction	Size of smallest connected component	Size of largest connected component
Prosocial_1	003;540	1	54,222	70,463,774	2599	3066	undirected	3	54,215
Prosocial_2	003;540	0.75	54,220	52,854,071	1949.6	2265	undirected	2	54,214
Prosocial_3	003;540	0.1	48,975	701,145	28.6	29	undirected	2	43,384
Prosocial_4	003;540	0.01	36,675	69,694	3.8	4	undirected	2	30,148
Prosocial_5	003;540	1	54,203	22,282,078	822.1	1289	undirected	2	51,682
Prosocial_6	003;540	0.1	43,158	218,666	10.1	10	undirected	2	51,494
Peer_1	157	1	3,131	812,591	519.1	416	undirected	3,131	3,131
Peer_2	157	1	3,130	108,922	69.6	152	undirected	3,130	3,130
EVCS home	VA	1	2,115,494	2,458,208	2.3	39	directed	1	1
EVCS work	VA	1	1,267,455	6,464,725	10.2	54	directed	12	1,267,394
EVCS	VA	1	512,317	1,698,959	6.6	53	directed	2	45,994

B. Pipeline architecture

In this section, we focus on MSA design and software-defined infrastructure for developing *NetXpipe*. Recall that the goal of the pipeline is to generate a network based on user selections of network type and constraints on node attributes. The structure of *NetXpipe* is illustrated in Figure 2.

The pipeline accepts the user input in the form of a JSON (Javascript Object Notation) object that has been customized for the pipeline. The JSON object contains the following information: type of network (e.g., peer network, solar network, EV network), geographical region, node(s), node attributes, constraints for edge generation, and an option of a deterministic or probabilistic way of generating the graph. First, the pipeline verifies the input and then proceeds to invoke the specified graph generation model. Examples of graph generation models are shown in Figure 1 (green colored corner snipped rectangles).

The first step in generating the networks is to verify the input. The input consists of a selection of nodes, attributes and conditions, and a graph module that describes the type of network and the method to generate the graph. In the next step, the system processes this input and assembles the datasets required for generating the graph. This step is executed by the "Verify graph generation function" service of Figure 2. The "Fetch relevant data for graph generation" service gathers all the necessary data and then invokes the "Graph generation" module. These models are encapsulated as services that accept the required data as input and generate a network as output. Creating certain types of networks can be time-consuming; hence, these components (blue-colored rectangles in Figure 2) are designed to harness the power of high-performance computing architectures whenever possible. Once the network is generated, the output is verified and written to the file system. The current format for storing networks is an adjacency list, a standard representation for executing many graph algorithms [27]. Currently, we have employed preliminary software-related graph verification steps (E.g., check if correct algorithm is invoked, if network was generated successfully).

The input datasets currently used in the system are in text and relational database formats. These datasets are obtained from trustworthy sources and are validated extensively. When a researcher wants to add a new graph generation algorithm or an additional dataset, a subpipeline is invoked (shown in orange in Figure 2). As shown, once the graph model is tested, it is packaged as a service and added to the deployed models component. Once deployed, it is available for user selection. Note that the graph generation model service can be a sequential or a parallel algorithm. For adding a dataset, the authenticity of the dataset is verified manually. The dataset is cataloged for valid node entities and attributes that can be employed in graph generation.

Two noteworthy advantages provided by *NetXpipe* are flexibility and extensibility. The proposed pipeline works in a standalone setting as well as in a setting where it is plugged into an existing ecosystem. The pipeline is robust and modular since all network generation functions are packaged as services that require specific inputs and generate specific outputs.

C. Examples of networks generated from NetXpipe

Table I lists some of the examples of the types of networks generated using *NetXpipe*. Over 20 networks have been generated, and are available for public use through the net.science platform. The networks described here use a combination of four datasets: synthetic populations, energy-use data for households, EV charging station database, and census data. The 'Region' column in Table I represents the geographic area for which the network is generated. Region code '003;540' represents Albemarle county and Charlottesville city in Virginia, and '157' is Rappahannock county in Virginia.

Recall that the k-core of a graph $\mathcal{G}$ is the subgraph $\mathcal{H}$ of $\mathcal{G}$ with the largest number of nodes such that the degree of every node in $\mathcal{H}$ is $\geq k$. For each graph in Table I, the largest value of k for which the k-core of the graph is nonempty is given in column labeled **Max.kcore**. The networks Prosocial_1 through Prosocial_4 have a distance threshold of 1 mile while the networks Prosocial_5 and Prosocial_6 have a distance threshold of 0.5 miles. Similarly, both peer networks and the three EVCS networks were generated by varying distance thresholds.

III. CASE STUDIES

In this section, we describe three sustainable energy-related applications that employ network data. First, we consider retrofitting, a crucial effort towards sustainable energy goals

since it focuses on refurbishing an existing dwelling to reduce the carbon footprint and environmental impact of the building [28]. In the first case study, graphs generated by *NetXpipe* are used to study retrofitting scenarios by developing a contagion modeling experiment. The next sustainable energy application here is solar adoption in the residential sector. Solar adoption is increasing in households as a green energy tool to tackle extreme heat events, reduce peak time demands, and so on. Thus, in the second case study, the generated peer networks are plugged into an existing solar adoption simulation system to model which households are ideal for solar adoption. The third application considers another green technology, namely electric vehicles (EV). A major hurdle in EV adoption is the availability of an EV charging station at home or within a feasible distance. We study the problem of accessibility of current EV charging infrastructure in different locations such as home and work locations and area types such as urban and rural areas.

A. Retrofitting as a contagion

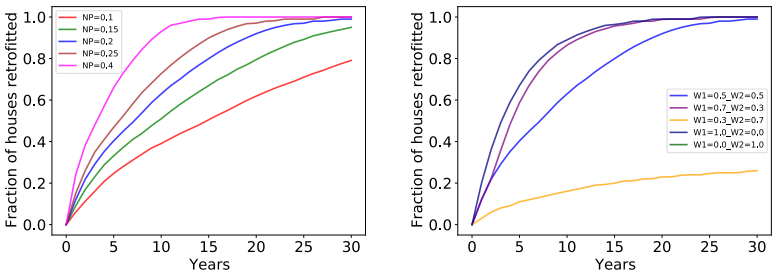
This case study demonstrates the pipeline's extensibility and scalability by using the generated networks in contagion modeling frameworks such as CSonNet [29], [30]. We simulate retrofitting as a social contagion that spreads over 30 years (from 2019 to 2049). We chose 2019 chosen as the starting point since the currently available data is for that year.

To model this scenario, we consider four networks of pro-social households within 1-mile distance threshold in Albemarle County, VA, including the City of Charlottesville. We follow the 'pro-social household' concept defined in Pillai et al. [31] as households favoring retrofitting if it satisfies at least one of the following conditions: (i) low-income household with about 30% of monthly income going towards monthly energy bills, (ii) single detached dwelling, (iii) old dwelling and (iv) the presence of children in the household. The inputs to develop the pro-social network are (i) synthetic population [32] and (ii) residential energy consumption data [5]. The networks differ in the probability of adding an edge between a pair of nodes; probability values 0.01, 0.1, 0.75, and 1.0 were used in this study. The reasoning behind analyzing various probability distributions is that they allow us to capture the degree of influence a household may have on neighboring households in propagating the retrofitting contagion. The structural properties of the generated networks (Prosocial_1, Prosocial_2, Prosocial_3, Prosocial_4) are in Table I. In the table, the difference in the numbers of nodes of these four networks is due to the fact that after generating each network, we retained only the nodes with degree ≥ 1.

We perform the contagion propagation experiment using an agent-based simulator called CSonNet [29], [30]. CSonNet is a lightweight framework that runs simulations using an extensible set of user-defined state transition rules to form contagion models. Serial and parallel execution modes are available in CSonNet. We created a new heterogeneous model in CSonNet for this study, inspired by the model of Bale et al. [6]. The utility function defined in our work is influenced by the personal benefit component and the social component in adopting energy policies. The transition from 0 (non-adopting state) to 1 (adopting state) happens if the utility function value is greater than a predefined threshold. The state transition is progressive; that is once a node changes to state 1, it doesn't change back to 0. These models help to capture the personal advantage of adopting specific policies along with the social influence in contrast to conventional threshold models [33] used to analyze the spread of social contagions. The utility function for household i is expressed as $u_i = w_1 \cdot p_i + w_2 \cdot s_i$. Here, w_1 is the weight for personal benefit, w_2 is the weight for social influence, p_i and s_i are respectively the personal benefit value and the social influence value for a household i. We assign personal benefit value based on how many pro-social conditions household i satisfies and the value s_i is obtained based on the neighbors of household i in state 1. The node threshold is defined based on the ability of the household to adopt retrofitting. Here, it is based on the income level of the households. We also use a node probability parameter, denoted by n_p, to account for the stochastic behavior of nodes. Thus, when the utility function exceeds the node threshold, a household adopts retrofitting with probability n_p.

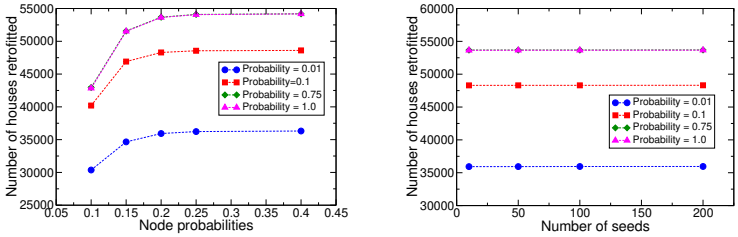
Our experiments vary the node threshold, personal benefit, and social influence values based on household attributes. We keep the weights and the node probability the same for all nodes. We perform sensitivity analysis for different values of node probability and weights. We performed experiments varying node probabilities, keeping the weights constant, varying the weights keeping the node probability value constant, and the fraction of nodes infected for various networks as in Figure 3. We used the data analysis and plotting modules in CSonNet [34] to analyze and visualize the results.

Figure 3(a) shows that the fraction of nodes infected (i.e., the number of households that choose to retrofit) at the end of 30 time steps varies between 0.79 to 1.0 as node probabilities are varied between 0.1 and 0.4. While the cumulative number of infections hasn't reached a plateau by 30 time steps when the node probability is 0.1, all the nodes got infected within 16 time steps (years) when the node probability is 0.4, indicating a complete adoption of retrofitting. Interestingly, the curve did not take off when the personal weight (w_1) was zero and social weight (w_2) was ten, as in Figure 3(b). This phenomenon indicates the necessity of having a personal benefit irrespective of the social influence to adopt sustainable energy policies. When the personal weight was less than the social weight, only 26% of the pro-social population adopted retrofitting by 2049. Our study on different networks obtained by varying the edge probabilities as in Figure 4(a) shows that the number of nodes in a network plays a crucial role in retrofitting more houses. Here, the network with an edge probability of 0.75 has 54220 nodes and about 52 million edges while the network with edge probability 1.0 has 54222 nodes and about 70 million edges. Though the number of edges is higher in the latter network, the number of houses retrofitted is similar. In Figure 4(b), we varied the number of seed nodes (number of houses retrofitted) from 10 to 200 for different edge probabilities in a pro-social



(a) Varying node probabilities (b) Varying weights

Fig. 3: Each simulation was performed for 50 iterations and 30 time steps. The number of initial seed nodes is 10 selected randomly. (a) Cumulative infection curve for a pro-social network with edge probability of 0.75, with the number of years along the X-axis and fraction of nodes that adopted retrofitting along the Y-axis for constant personal weights (w_1) and social weight (w_2) of 0.5 and 0.5 respectively. (b) Cumulative infection curve for a pro-social network with edge probability of 0.75 and node probability .



(a) Varying edge probability (b) Varying seed nodes

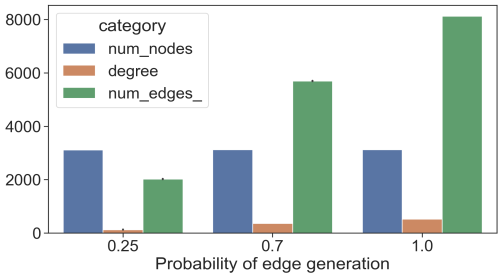
Fig. 4: Average number of houses retrofitted with $\pm$ one standard deviation for different edge probabilities in a pro-social network. Each simulation was performed for 50 iterations and 30 time steps. (a) Different node probabilities along the X-axis and number of houses retrofitted along the Y-axis with personal weight (w_1) and social weight (w_2) of 0.5 and 0.5 respectively. (b) Different number of seed nodes selected at random along the X-axis and number of houses retrofitted along the Y-axis for a node probability (n_p) at 0.2 and personal weight (w_1) and social weight (w_2) of 0.5 and 0.5 respectively.

network. The selection of different numbers of seed nodes had a lesser impact on the number of houses getting retrofitted. The seed nodes are selected at random for each iteration. Thus, for each simulation, we have 50 different sets of randomly selected houses as seed nodes. This accounts for the uncertainty in the number of houses getting retrofitted due to the differences in initial seeding. The standard deviations for each point in Figure 4(b) show a negligible difference in the total number of retrofitted houses.

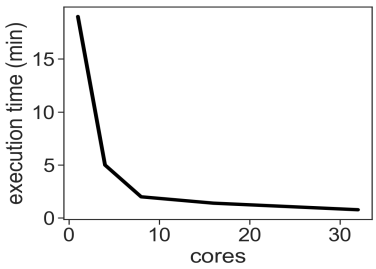
B. Peer networks in solar adoption

This case study highlights the pipeline's modularity and extensibility features. We demonstrate that *NetXpipe* can replace an existing network generation process and be used in an existing M&S system with ease. We specifically focus on peer network generation for modeling household-level solar panel adoption in regions of the state of Virginia. The pipeline generates networks and plugs them into an existing solar adoption diffusion model described in [9], [15]. This model predicts whether a household will adopt solar panels depending on socio-demographic and peer attributes. For the sake of completeness, we summarize the solar adoption model here. The model simulates the diffusion process for 15 time steps and employs three peer networks for 1-mile, 3-mile, and 4-mile radii to examine the propagation of solar adoption in the network(s) across the time horizon. The peer effect (i.e., neighborhood effect) is computed from the peer networks. It is the number of adopters within a 1-mile, 3-mile, and 4-mile radius of each household. At the end of each time step, the peer effects are updated in the model. For further details, we refer the reader to the 'Virginia Model' described in [9].

We ran the above solar adoption model for Rappahannock county in Virginia. It has $\approx$ 3300 households. Previously, the peer networks were generated manually by querying a relational database. Now, we replace the manual process with *NetXpipe*. From the data warehouse shown in Figure 1, we utilize the 'synthetic populations' dataset for generating the networks. First, an input is generated for the pipeline in JSON format. Once the input is verified by the pipeline, the data processing module will assemble the synthetic population



(a) Network measures across replicates



(b) Strong scaling

Fig. 5: (a) The network measures show stability across graph replicates for a given edge probability. (b) The curve shows strong scaling as the number of cores for computing the network increases.

for the given region. Then, the pipeline invokes the specific network generation component depending upon the input. Here, the algorithm for generating the graph is straightforward – households within the enforced distance threshold are connected with an edge. When carried out sequentially, this task is time-consuming; hence, in this case study, we parallelize it. Once the network is generated, it is verified by the pipeline and stored as an adjacency list to be used as an input for the solar adoption model.

Peer networks are commonly used in diverse applications. We show that our pipeline is able to generate such networks with ease. The same network can be plugged into a solar adoption model or into a disaster evacuation model without running the pipeline again. Thus, when complex processing tasks such as network generation can be streamlined using software-defined infrastructure such as MSA, it increases human productivity. Figure 5(a) shows the stability of 4-mile radius peer networks for Rappahannock across replicates. Figure 5(b) shows the strong scaling behavior of the graph generation method as the number of cores is increased.

C. Accessibility of current EV charging infrastructure from synthetic households

The goal of this case study is to demonstrate that *NetXpipe* can work with multiple types of node entities and diverse data sources. Here, we examine the accessibility of the current EV Charging Stations (EVCS) from households in a given region. In this instance, the pipeline assembles data from synthetic populations and an EV charger location database[2] for the given region. As an example, we focus on the state of Virginia (VA) in this case study. The number of synthetic households in VA is $\approx$ 3 million. The number of public EVCS in VA is 1124. Three network generation scenarios are considered in this case study.

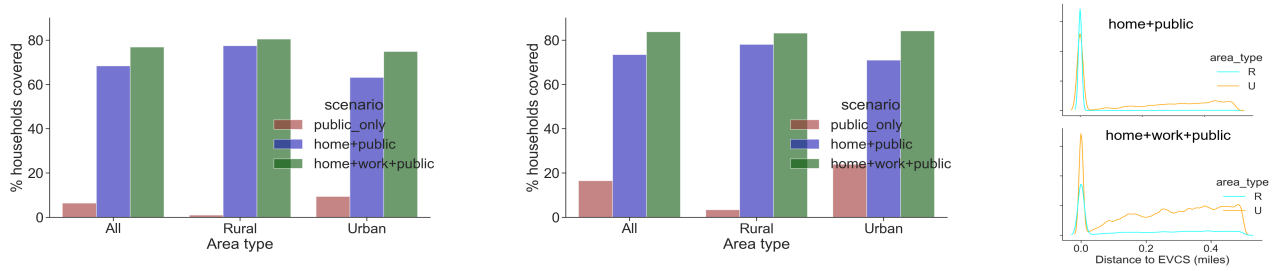
Scenario 1 - Public only : In this scenario, households are assumed to not have the facility to charge their EVs at home and thus rely only on public EV charging infrastructure. Two types of nodes are used in the construction of this network – (i) households (attributes: home location) and (ii) EV charger stations (attributes: location). The resulting network is a bipartite graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ where the node set $\mathcal{V}$ is partitioned into two subsets V_1 (households), and V_2 (EV chargers) and each edge $e \in \mathcal{E}$ connects a node in V_1 to a node in V_2. All the results for this scenario are shown in red in Figure 6c.

Scenario 2 - Home+Public : In this scenario, we allow certain households to have the ability to charge their EVs at home (e.g., detached households with a garage). A recent survey from the International Council on Clean Transportation (Nicholas et al. [35]) shows that 95% of EV adopters from detached and attached households charge their EVs at home in the U.S. Hence, in this case, this network will provide for charging accessibility at home and public EVCS. Thus, nodes in the construction of this network are of two types: (i) household (attributes: home location, dwelling type) and (ii) EVCS (attributes: location). All the results for this scenario are shown in blue in Figure 6c.

Scenario 3 - Home+Work+Public : This scenario provides three options for charging EVs: at home, EVCS near home and EVCS near a person's work location. Thus, this network will be comprised of three types of nodes – (i) household (attributes: home location, dwelling type), (ii) EV charger station (attributes: location), (iii) person (attributes: work location). All the results for this scenario are shown in green in Figure 6c. Note that if detailed data such as a person's work location is unavailable, it is possible to use POI (point of interest) datasets that recognize commercial and industrial buildings/zones. Literature has shown that people have time and distance preferences for accessibility from an EVCS when charging their vehicle [36]. For each scenario, we generated two networks, one for each distance threshold: 0.25 miles and 0.5 miles. These networks have a large number of nodes since we consider the entire population of VA in this case study.

Analyses of the six resulting graphs are shown in Figure 6. The overall household node coverage for access to EV charging increases from 77% to 84% when the distance is increased from 0.25 to 0.5 miles (refer to the green bars in Figures 6(a) and (b)). If only public infrastructure access is considered (*Scenario 1: public_only*), the number of households that will gain access to EVCS increase from 6% to 16% when access

[2]https://afdc.energy.gov/stations/\#/find/nearest



(a) Accessibility within 0.25 miles

(b) Accessibility within 0.5 miles

(c) Edge weight distribution

Fig. 6: Parts (a) and (b) show the number of households (i.e., a node type in the graph) that have access to one or more ways of charging (i.e., at home, at public EVCS, and/or at work) within a 0.25-mile radius and 0.5-mile radius, respectively. Figure (c) shows the histogram for edge weights i.e. distance of household/work location nodes to EVCS under *Scenario 2* and *Scenario 3* for 0.5-mile radius. We observe that many people from urban areas have access to EV charging at work locations. This is also evident in the green urban area bar in (b).

distance increases from 0.25 to 0.5 miles (refer to the red bars in Figures 6(q)a and (b)). It is evident from the bar charts that home charging will play an important role in EV charging. Note the increase from *Scenario 1: public_only* to *Scenario 2: home+public* (compare the heights of red bars and blue bars). It is observed that rural households benefit more from charging at home than at work or in public spaces given the current infrastructure, whereas households in urban areas have access to more work location chargers than their rural counterparts. Figures 6(a) and (b) focused on household accessibility (i.e., node coverage) to EVCS at home, near home, and near work. Figure 6(c) focuses on the edge weights in graphs; i.e., the distance between (i) household location and public EVCS locations and (ii) work locations + public EVCS locations within a 0.5 mile radius. Many households in urban areas have access to EV chargers within 0.2 to 0.5 mile of work locations. The peak close to zero in both histograms indicates the availability of home chargers for a large number of households. We acknowledge that access to detailed data (e.g. household locations) may not be openly available due to customer privacy concerns. In that case, we can conduct a similar analysis using block group or census tract data that is openly available from U.S. Census.

IV. CONCLUSIONS

Network data is becoming increasingly common in addressing a wide variety of topics in energy systems. To support this work, we have developed an extensible, scalable, and modular network synthesis pipeline based on MSA principles. *NetXpipe* can be integrated into an existing system for network synthesis or can be used as a standalone tool for generating networks. The pipeline generates large-scale networks with multiple entity types in a deterministic or stochastic manner. Researchers/developers can add new algorithms for generating graphs as a "service" in *NetXpipe*. We illustrated the benefits of this pipeline through three case studies. Although we have focused on sustainable energy applications in this work, graphs generated from *NetXpipe* can be used in other applications as shown in Qiu et al. [37] and Chen et al. [38].

Future work includes making enhancements to the pipeline and extending the pipeline to support other social impact applications. For example, one enhancement involves adding more complex functions for inserting edges into the network to allow the generation of specific classes of networks such as small-world networks and scale-free networks [39]. Other enhancements would be to provide facilities to generate temporal networks (where the nodes/edges vary over time) and multi-layer networks (where each layer captures a different form of interaction among the nodes) which are useful in many applications (e.g., [40], [41]). Another useful addition is a verification module to test for privacy concerns.

ACKNOWLEDGMENTS

We thank the reviewers for providing helpful suggestions. This work is partially supported by National Science Foundation (NSF) RAPID Grant No. CCF-2142997, University of Virginia Strategic Investment Fund award number SIF160 and NSF Grant No.: OAC-1916805 (CINES).

REFERENCES

[1] P. L. Donti and J. Z. Kolter, "Machine learning for sustainable energy systems," *Annual Review of Environment and Resources*, vol. 46, no. 1, pp. 719–747, 2021. [Online]. Available: https://doi.org/10.1146/annurev-environ-020220-061831

[2] M. R. Edwards, T. Holloway, R. B. Pierce, L. Blank, M. Broddle, E. Choi, B. N. Duncan, A. Esparza, G. Falchetta, M. Fritz, H. K. Gibbs, H. Hundt, T. Lark, A. Leibrand, F. Liu, B. Madsen, T. Maslak, B. Pandey, K. C. Seto, and P. W. Stackhouse, "Satellite data applications for sustainable energy transitions," *Frontiers in Sustainability*, vol. 3, pp. 1–24, 2022. [Online]. Available: https://www.frontiersin.org/articles/10.3389/frsus.2022.910924

[3] C. Klemenjak, C. Kovatsch, M. Herold, and W. Elmenreich, "A synthetic energy dataset for non-intrusive load monitoring in households," *Scientific Data*, vol. 7, no. 1, p. 108, Apr 2020.

[4] R. Meyur, M. Marathe, A. Vullikanti, H. Mortveit, V. Centeno, and A. Phadke, "Creating realistic power distribution networks using interdependent road infrastructure," in *2020 IEEE International Conference on Big Data (Big Data)*. Atlanta, GA, USA: IEEE, Dec 2020, pp. 1226–1235.

[5] S. Thorve, Y. Y. Baek, S. Swarup, H. Mortveit, A. Marathe, A. Vullikanti, and M. Marathe, "High-resolution synthetic residential energy use profiles for the united states," *Sci Data*, vol. 10, p. 76, 2023. [Online]. Available: https://doi.org/10.1038/s41597-022-01914-1

[6] C. S. Bale, N. J. McCullen, T. J. Foxon, A. M. Rucklidge, and W. F. Gale, "Harnessing social networks for promoting adoption of energy technologies in the domestic sector," *Energy Policy*, vol. 63, pp. 833–844, 2013. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0301421513009579

[7] M. U. Haider, F. Mumtaz, H. H. Khan, M. Asif, M. S. Rashid, S. R. Abbas, and M. Zeeshan, "Smart energy meters in renewable-energy-based power networks: An extensive review," *Engineering Proceedings*, vol. 20, no. 1, pp. 1–5, 2022. [Online]. Available: https://www.mdpi.com/2673-4591/20/1/23

[8] F. Du, J. Zhang, H. Li, J. Yan, S. Galloway, and K. L. Lo, "Modelling the impact of social network on energy savings," *Applied Energy*, vol. 178, pp. 56–65, 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0306261916307899

[9] S. Thorve, Z. Hu, K. Lakkaraju, J. Letchford, A. Vullikanti, A. Marathe, and S. Swarup, "An active learning method for the comparison of agent-based models," in *Proceedings of the 19th International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*. IFAAMAS, 2020, pp. 1377–1385.

[10] H. Zhang, Y. Vorobeychik, J. Letchford, and K. Lakkaraju, "Data-driven agent-based modeling, with application to rooftop solar adoption," *Autonous Agents and Multi-Agent Systems*, vol. 30, no. 6, pp. 1023–1049, 2016.

[11] Q. Zhang, J. Liu, K. Yang, B. Liu, and G. Wang, "Market adoption simulation of electric vehicle based on social network model considering nudge policies," *Energy*, vol. 259, p. 124984, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0360544222018825

[12] V. Breschi, C. R. Mara Tanelli, S. C. Strada, and F. Dabbene, "Social network analysis of electric vehicles adoption: a data-based approach," 2020, arXiv 2001.09704, primary class cs.SI.

[13] R. Meyur, S. Thorve, M. Marathe, A. Vullikanti, S. Swarup, and H. Mortveit, "A reliability-aware distributed framework to schedule residential charging of electric vehicles," in *2022 International Joint Conference on Artificial Intelligence (IJCAI)*. IJCAI, 2022, pp. 1–7.

[14] D. Dwivedi, K. V. S. M. Babu, P. K. Yemula, P. Chakraborty, and M. Pal, "Evaluation of energy resilience and cost benefit in microgrid with peer-to-peer energy trading," 2022, arXiv 2212.02318, primary class = eess.SY.

[15] Z. Hu, X. Deng, A. Marathe, S. Swarup, and A. Vullikanti, "Decision-adjusted modeling for imbalanced classification: Predicting rooftop solar panel adoption in rural virginia," in *Proceedings of The Computational Social Science Conference*. Springer, 2019, pp. 381–399.

[16] T. Cerny, M. J. Donahoo, and M. Trnka, "Contextual understanding of microservice architecture: Current and future directions," *ACM Special Interest Group on Applied Computing Review*, vol. 17, no. 4, p. 29–45, Jan 2018. [Online]. Available: https://doi.org/10.1145/3183628.3183631

[17] R. Mark, *Microservices vs. Service-Oriented Architecture*. California: O'Reilly Media, Inc., 2016.

[18] E. Wolff, *Microservices: Flexible Software Architecture*. United States: Leanpub, 2016.

[19] V. Cedeno-Mieles, Z. Hu, Y. Ren, X. Deng, N. Contractor, S. Ekanayake, J. M. Epstein, B. J. Goode, G. Korkmaz, C. J. Kuhlman, D. Machi, M. Macy, M. V. Marathe, N. Ramakrishnan, P. Saraf, and N. Self, "Data analysis and modeling pipelines for controlled networked social science experiments," *PLOS ONE*, vol. 15, no. 11, pp. 1–58, 11 2020. [Online]. Available: https://doi.org/10.1371/journal.pone.0242453

[20] S. P. R. Asaithambi, R. Venkatraman, and S. Venkatraman, "Mobda: Microservice-oriented big data architecture for smart city transport systems," *Big Data and Cognitive Computing*, vol. 4, no. 3, pp. 1–27, 2020. [Online]. Available: https://www.mdpi.com/2504-2289/4/3/17

[21] S. Thorve, A. Vullikanti, S. Swarup, H. Mortveit, and M. Marathe, "Modular and extensible pipelines for residential energy demand modeling and simulation," in *2022 Winter Simulation Conference (WSC)*. INFORMS, 2022, pp. 855–866.

[22] Y. Simmhan, S. Aman, A. Kumbhare, R. Liu, S. Stevens, Q. Zhou, and V. Prasanna, "Cloud-based software platform for big data analytics in smart grids," *Computing in Science Engineering*, vol. 15, no. 4, pp. 38–47, 2013.

[23] M. R. Rodrigues, W. C. Magalhães, M. Machado, and E. Tarazona-Santos, "A graph-based approach for designing extensible pipelines," *BMC Bioinformatics*, vol. 13, no. 1, p. 163, Jul 2012. [Online]. Available: https://doi.org/10.1186/1471-2105-13-163

[24] Y. Du, S. Wang, X. Guo, H. Cao, S. Hu, J. Jiang, A. Varala, A. Angirekula, and L. Zhao, "GraphGT: Machine learning datasets for deep graph generation and transformation," in *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*. Curran Associates, Inc., 2021. [Online]. Available: https://openreview.net/forum?id=NYgt9vcdyjm

[25] S. Purohit, P. S. Mackey, J. A. Cottam, M. P. Dunning, and G. Chin, "Synthetic data and graph generation for modeling adversarial activity (final project report)," https://www.osti.gov/biblio/1871012, 2 2022.

[26] J. Hu, L. Sankar, and D. J. Mir, "Cluster-and-connect: An algorithmic approach to generating synthetic electric power network graphs," in *2015 53rd Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. University of IL Urbana-Champaign, 2015, pp. 223–230.

[27] J. Kleinberg and E. Tardos, *Algorithm Design*. New York, NY: Pearson Education, Inc., 2006.

[28] Department of Energy, "Weatherization Assistance Program," https://www.energy.gov/scep/wap/weatherization-assistance-program, Accessed: Apr, 2023.

[29] J. D. Priest, A. Kishore, L. Machi, C. J. Kuhlman, D. Machi, and S. Ravi, "Csonnet: An agent-based modeling software system for discrete time simulation," in *2021 Winter Simulation Conference (WSC)*. IEEE, 2021, pp. 1–12.

[30] A. Kishore, L. Machi, C. J. Kuhlman, D. Machi, and S. Ravi, "A framework for simulating multiple contagions over multiple networks," in *Complex Networks & Their Applications X: Volume 2, Proceedings of the Tenth International Conference on Complex Networks and Their Applications COMPLEX NETWORKS 2021 10*. Springer, 2022, pp. 241–252.

[31] A. Pillai, M. T. Reaños, and J. Curtis, "An examination of energy efficiency retrofit scheme applications by low-income households in ireland," *Heliyon*, vol. 7, no. 10, p. e08205, 2021.

[32] S. Swarup and M. V. Marathe, "Generating Synthetic Populations for Social Modeling: Second Tutorial," http://staff.vbi.vt.edu/swarup/synthetic_population_tutorial_2/AAMAS_2017_generating_synthetic_populations_for_social_modeling_full_tutorial.pdf, 2017, tutorial at the Sixteenth International Conference on Autonomous Agents and Multiagent Systems (AAMAS).

[33] M. Granovetter, "Threshold models of collective behavior," *American Journal of Sociology*, vol. 83, no. 6, pp. 1420–1443, 1978.

[34] T. Ferdousi, A. Kishore, L. Machi, D. Machi, C. J. Kuhlman, and S. Ravi, "A web-based system for contagion simulations on networked populations," in *2022 IEEE 18th International Conference on e-Science (e-Science)*. IEEE, 2022, pp. 306–315.

[35] M. Nicholas, D. Hall, and N. Lutsey, "Quantifying the electric vehicle charging infrastructure gap across u.s. markets," https://theicct.org/sites/default/files/publications/US_charging_Gap_20190124.pdf, 2019, 39 pages.

[36] J. Dong, C. Liu, and Z. Lin, "Charging infrastructure planning for promoting battery electric vehicles: An activity-based approach using multiday travel data," *Transportation Research Part C: Emerging Technologies*, vol. 38, pp. 44–55, 2014.

[37] Z. Qiu, A. Yuan, C. Chen, M. V. Marathe, S. S. Ravi, D. J. Rosenkrantz, R. E. Stearns, and A. Vullikanti, "Assigning agents to increase network-based neighborhood diversity," in *Proceedings of the 22nd International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*. IFAAMAS, 2023, pp. 600–608.

[38] J. Chen, S. Hoops, A. Marathe, H. Mortveit, B. Lewis, S. Venkatramanan, A. Haddadan, P. Bhattacharya, A. Adiga, A. Vullikanti, A. Srinivasan, M. L. Wilson, G. Ehrlich, M. Fenster, S. Eubank, C. Barrett, and M. Marathe, "Effective social network-based allocation of covid-19 vaccines," in *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, ser. KDD '22. New York, NY, USA: Association for Computing Machinery, 2022, p. 4675–4683. [Online]. Available: https://doi.org/10.1145/3534678.3542673

[39] D. Easley and J. Kleinberg, *Networks, Crowds and Markets: Reasoning About a Highly Connected World*. New York, NY: Cambridge University Press, 2010.

[40] M. M. Hosseinzadeh, M. Cannataro, P. H. Guzzi, and R. Dondi, "Temporal networks in biology and medicine: a survey on models, algorithms, and tools," *Network Modeling Analysis in Health Informatics and Bioinformatics*, vol. 12, no. 1, pp. 1–22, 2022.

[41] M. Kivelä, A. Arenas, M. Barthelemy, J. P. Gleeson, Y. Moreno, and M. A. Porter, "Multilayer networks," *Journal of complex networks*, vol. 2, no. 3, pp. 203–271, 2014.