Failures to Food, Energy, and Water Systems: Mapping and Simulating Components to Improve Resilience

Alex Modarresi
Department of Computer Science
California State University, Northridge
Northridge CA, USA
alex.modarresi@csun.edu

John Symons
Department of Philosophy
Center for Cyber-Social Dynamics
The University of Kansas
Lawrence KS, USA
johnsymons@ku.edu

The 2023 International Conference on Smart Applications, Communications and Networking (SmartNets)

July 25-27, 2023
Istanbul, Turkey

DOI: 10.1109/SmartNets58706.2023.10215865

Failures to Food, Energy, and Water Systems: Mapping and Simulating Components to Improve Resilience

Alex Modarresi
Department of Computer Science
California State University, Northridge
Northridge CA, USA
alex.modarresi@csun.edu

John Symons
Department of Philosophy
Center for Cyber-Social Dynamics
The University of Kansas
Lawrence KS, USA
johnsymons@ku.edu

***Abstract*—This paper identifies common varieties of threats and perturbations in contemporary food, energy, and water (FEW) systems in order to improve system resilience. We categorize perturbations and challenges faced by subsystems and then concentrate on the structural topology of the project's components. We provide a graph model to represent this topology as an essential tool to improve system resilience. The model is then converted to a system dynamic model for further simulation.**

***Index Terms*—Fault taxonomy; smart agriculture, Future Internet of Things (IoT) resilience, technology interdependence graph**

I. Introduction

Efficient and sustainable agricultural systems are highly complex and interdependent. In the past, agriculture depended mostly on water resources, fertilized soil, weather, and local or regional market dynamics. Contemporary agriculture is increasingly dependent on electricity, fertilizer, transportation, national and international markets, and, in recent years, on emerging technologies such as the Internet of Things (IoT). Thus, modern agriculture must be understood as a system of integrated systems. The novel challenges that contemporary agriculture faces include, for example, cyber-physical attacks, the effects of climate change, legislative and regulatory control, and global financial conditions [1]. The overall food-energy-water system is increasingly vulnerable to cascading failures due to its highly integrated and multi-level nature [2]. This paper describes our approach to modeling the complex set of interacting natural and engineered systems in order to understand potential vulnerabilities and increase overall resilience.

We structure this paper as follows. We review some concepts and background from the relevant literature in the next section. In Section III, we model a general FEW system with graph theory and analyze its structure and important perturbations. The modeling and simulation strategy we describe here is part of ongoing work to develop practical decision-support tools for agricultural communities. Finally, we conclude our paper in Section IV.

This work was supported in part by the National Science Foundation [grant number 1856084] FEWtures project.

II. Background and Concepts

Contemporary agriculture combines traditional practices with water management, water treatment, energy, fertilizer, and cyber systems. As part of a response to climate change contemporary agriculture increasingly integrates renewable energy sources, microgrids, and green ammonia production into agricultural operations. Each of these systems has its own structure, features, and challenges. Perturbations to subsystems can potentially affect other components of the food-energy-water system. In order to have a resilient FEW system, it is imperative to understand these dependence relations. Before we explain the specific challenges facing smart agriculture systems, we describe some of the relevant concepts from the literature.

A. Concepts

System resilience can be characterized in terms of two fundamental factors, perturbations and the dynamics of the system [3]. Recognizing the most important faults and perturbations, and preventing, masking, or avoiding them is the first step to improving system resilience. There is comprehensive literature regarding the taxonomy of faults, perturbations, reliability, availability, and dependability in engineering systems [4] [5] [6] [7] [8] [9] [10] [11]; however, the type, occurrence, frequency, and severity of faults and perturbations are not the same in each system. We identify those groups of faults and vulnerabilities that are more common in FEW systems below.

A *system* is defined loosely as an entity that maintains some degree of individuality as it interacts with other entities [12]. In the case of FEW systems, interactions occur at their boundaries where the system services are offered. System *services* are the results of integration and working several components

inside the system at a specific state. A service is part of the system functionality defined in the system specification. Relations among components define the structure of the system. In order to offer a service at the system boundary, each of the system components should have a specific state. The overall states of these components define the state of the system. When these components deviate from their defined states, an anomaly in the service is observed at the boundary of the system. A *service failure* takes place when the service provided deviates from the correct service. Deviations from correct services can result from *errors*. The cause of an error is called a *fault* [4]. A sudden and discontinuous state change is called *disruption* while a *disturbance* is a continuous state change for a limited time that leads to service failure. The manifested result of disruption at the system boundary is a service outage, while the result of disturbance is a service deviation that may lead to a service outage. A *perturbation* is any unintended changes in the service level resulting in disturbance and disruption caused by internal or external faults [13], which is an equivalent term for service failure.

The system definition given here is recursive and can be extended to the system's components unless the components are atomic. Hence, an error may occur in the system components. If the error is handled internally, the service deviation does not manifest at the system level. An unhandled failure at the component level typically appears as a service error at the system boundary.

Services provided to another system help the receiving system fulfill and offer its own services. Therefore, a faulty service disturbs the receiving system's normal behavior, leading to cascading errors in the systems and a service failure by the receiving system. If failures happen at the component level and do not dismantle the whole service, then the system may offer its services in a degraded mode. The system specification identifies whether the system is in a degraded mode or failure mode. The difference between these two states identifies system resilience. If a system can return from degraded mode within an appropriate time frame while offering basic services to its correct service mode, it is called resilient, while a failed system does not return to its correct service.

The resilience of an engineered system is the system's capacity to return to its correct service level after some fault within an acceptable time frame. Resilient systems usually have some capacity to adapt, such that the next occurrence of similar failure is less disruptive and the recovery time for the system is faster. Figure 1 illustrates two systems, A and B, that are resilient and vulnerable to a particular fault. As observed, System A adapts to the fault and recovers from failures, while System B fails. The adaptive system has a shorter degradation service time and a faster recovery time against consequence faults. However, the ratio of fault frequency to recovery time is a factor that should be considered. If this ratio is greater than 1, it means that faults happen before the system recovers and may cause a system failure.

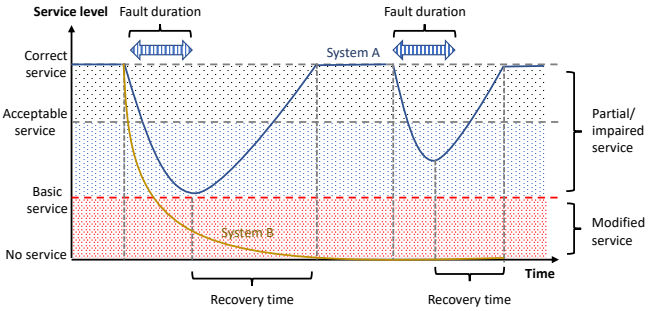

Fig. 1. Resilient and non-resilient system behavior during fault

III. System Modeling

Smart agriculture envisions the integration of green energy, green ammonia production, water supply, and water treatment. The energy system contains microgrids and, for example, in the United States mid-West, renewable energy resources such as wind turbines and solar panels. In the cases we are modeling here, the microgrid supplies energy for other systems, including ammonia production and water treatment systems. The microgrid permits the sale of excess energy or the purchasing of energy from the grid when the cost of production is lower than the system's own energy production capacity offers. Ammonia is produced for two purposes: as fertilizer and as a fuel source for electricity production when profitable. The water supply and treatment system contains components to receive water from various sources in order to treat it for the purpose of irrigation and ammonia production. The communication and information technology components collect data from different parts of the system through sensors, process data, provide feedback to other components, and enforce actions accordingly.

We first modeled an abstract smart agriculture system with graph theory. Graph theory has been one of the common formal approaches for modeling resilience in engineered systems [14] [15] [16]. A graph model expresses the corresponding system with a graph $G(V, E)$ such that V is a set of nodes or vertices representing the system entities or components and E is a set of links or edges representing the connections between nodes. When a system contains similar nodes with bidirectional connections, an undirected graph can model the system; however, other solutions, such as weighted graphs and directed graphs, are used when there is a weight/priority on a link or links are not bi-directional.

To show the abstract model, each node in the graph represents a system; and links among nodes show connectivity among systems. Since different types of links are involved we use an edge-colored graph $\mathbb{G}_{\text{conn}} = (V_c, E_c, C, \chi)$, such that $v_i \in V_c$ is a system and $e_n \in E_c$ is a link between two adjacent systems v_i and v_j. Furthermore, C is a set of colors equivalent to the different types of flows in the graph and $\chi : E_c \to C$ is a function to assign a color to each edge. Precisely, we can define E_c as $E_c = \{((v_i, c_i), (v_j, c_i)) \in V_c \times V_c | \chi(v_i, v_j) = c_i\}$. This representation allows us to integrate multiple networks

together and let us understand how a system can affect others. We can recognize three networks: electricity, ammonia, farm and irrigation, and water supply and treatment.

We use a directed graph to represent interdependency among systems since some of these systems are not mutually dependent on each other. Furthermore, the link directions show the flow of items/objects/energy. In addition, there are partial dependencies among subsystems. To present partial dependencies, we use a weighted graph. Each weight shows the extent of dependency. Weights can also be mapped to the amount of the flow on the link. For instance, w_i on a like that shows the flow of water can be mapped to the minimum amount of water necessary on the link to satisfy dependability. Figure 2 shows the graph model representing a FEW essential systems and their dependencies.

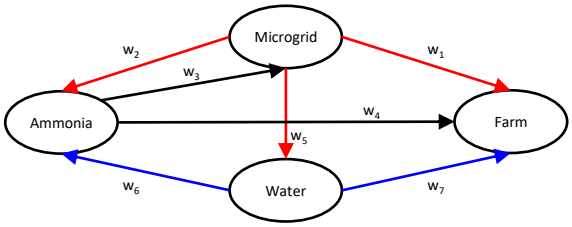

Fig. 2. A FEW abstract model

In a directed graph, a node's in-degree is the number of links points to the node. As observed in the figure, *Farm* has the highest in-degree value followed by *Ammonia*. Higher in-degree values show the higher dependency of a system on others. *Water* and *Micogrid* nodes have the same in-degree values; however, w_i is different for them, which identifies the extent of dependency on the rest of the system. In contrast, a node with a high out-degree value is more independent of other systems.

Since links are different, calculating other graph metrics is not straightforward. For instance, calculating the shortest paths among nodes is not feasible. Similarly, though the clustering coefficient values vary between 0.5 to 0.67, the clustering coefficient values in each separate network are zero, implying weak resilience in each network.

IoT permits control and monitoring of each individual agriculture system. This turns the agriculture system into a cyber-physical system. It also provides control feedback to other systems and increases the overall interconnectedness of the entire system. The cybersystem is represented as an overlay on top of the physical system. Figure 3 illustrates the complete cyber-physical system and its interdependency. All of these systems connect to the Internet and can be accessible via a cloud service. In addition, all communication links are bidirectional. Double-headed arrows in the figure represent these links instead of one arrow for each direction.

A. Faults and Failures in FEW Systems

A FEW system combines engineered and biological systems. Typically, the farm's biological systems are capable

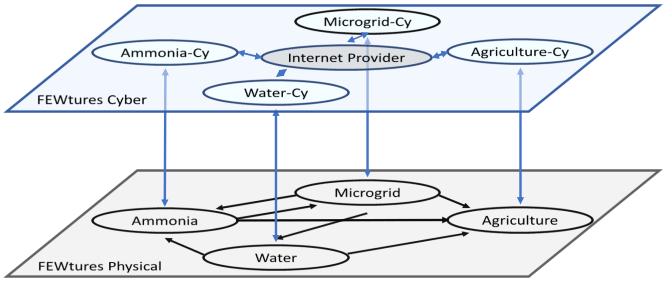

Fig. 3. FEWtures cyberphysical abstract model

of withstanding a range of faults and perturbations, like weather changes or short delays in irrigation. Consequently, the system's overall state changes more slowly during such minor faults (Figure 1). However, the recovery time of many natural systems will be slower once significant perturbations have ceased. In engineered systems, especially those without fault tolerance mechanisms, the system state changes quickly when the fault presents itself as shown in Figure 1, and the system may lose its normal functionality suddenly. Most of the time, in engineered systems, there is no service in the modified service area in Figure 1, while in biological systems, because of adaptation there, a new service can appear in the modified service area. This type of change is common in social and ecological systems.

Faults are inevitable [17]. A fault in a system reduces availability, reliability, and, consequently, dependability. When multiple faults happen at the same time, the results can be catastrophic. In interdependent systems, such as FEW, faults propagate across sub-systems. The effect of failed/degraded service is often amplified in the receiving subsystem/system, resulting in cascading failures of systems. This is where the overall system structure matters.

When systems are more interconnected, their structural topology and interdependency usually become more complex. This complexity can affect the system's overall function. For instance, the topology of a power grid can impact the stability of power transmission, which can affect water treatment systems' operation and the production of ammonia fertilizer for agriculture. These, in turn, can impact profitability.

Previous studies have demonstrated the ways that a local failure in an interdependent system can propagate from one system to another, escalating failures [18], [19]. In addition, many systems are designed with a focus on local considerations only. When such systems are aggregated, they tend to encounter unanticipated demands and transformation of services in order to adapt to novel conditions in the overall system. These changes can also lead to reduced robustness and resilience [20]. Therefore, it is generally unwise to analyze the system resilience of aggregated systems in terms of their separate components.

The life cycle of an engineered system has two phases: design and operation. Design faults can happen during the development process, while failures can occur from aging

components during the operation phase. Additionally, faults that arise during development can lay dormant and trigger service failure during operation. However, for the purposes of this paper, we will not address design faults.

Faults fall into two categories: internal and external. During the operational phase, systems interact with several elements in their physical environment. These elements include the natural environment, users (operators/farmers), intruders (animals, pests, and people), and the infrastructure that the system is part of (e.g., power grids and the economy). These external elements can pose a threat to the system, referred to as external faults, and can trigger challenges for the system. They can activate dormant faults, escalate internal faults, and ultimately lead to errors and service failure.

Within the physical environment of a FEW system, there are a range of potential challenges, including weather patterns, climate change, water resources, and contamination. In terms of infrastructure, the system may rely on the economy, transportation, power grid, and communication systems/Internet. Any potential sources of harm, such as disease, pests, wild animals, and adversaries, are categorized as intruders.

Internal faults arise within the system and are typically manifested at the system boundary as service failures. Any failures due to aging components are also considered to be internal faults.

Faults can be classified based on their origin as either human-made or natural. Natural faults occur without human intervention, such as environmental factors like weather events. On the other hand, environmental contamination is predominantly caused by human activity. Faults in infrastructure can be either natural or human-made as well. For instance, a wind-induced collapse of an antenna is a natural fault, whereas the same antenna being destroyed in a terrorist attack is a human-made fault.

Human-made faults can be categorized further based on the objective of disruptions into malicious and non-malicious. Each group can also be subdivided into deliberate or accidental according to the intent of the actor. However, faults with malicious objectives are deliberate [4]. Human-made, malicious, deliberate perturbations are also known as an attack.

Faults can be classified based on their consequences, duration, and frequency, and each attribute affects the system's proper functioning differently. While these attributes have subjective values and require clear definitions in the system specification, their impact on the system varies significantly. Fault consequences can be minor, severe, or catastrophic and can be quantified in terms of costs, time, number of casualties, and other factors. Ultimately, a fault's consequence will result in a degradation of the service level provided by the system. A catastrophic consequence pushes the system service level to the modified service area in Figure 1. In addition, fault severity can change the slope of the graph in the degradation period of Figure 1. Faults can have severe consequences, causing service disruption and pushing the system into a state of no or modified service in a short time. The duration of a fault determines the length of the degradation period, while the frequency of faults affects the recovery period. A catastrophic fault, such as a strong earthquake, can rapidly push the system into a modified or no-service state. In contrast, a plant disease or drought may have the same effect but over an extended period. The difference between the expected correct service level and the actual system performance indicates the extent of the harm.

Identifying all faults in a system is not feasible. The resources to prevent and mitigate error and faults are limited [21]. Therefore, resources are usually assigned to avoid faults that have severe consequences or a high frequency of occurrence. This is calculated as a risk to the system. Though calculating risks is applicable in simple systems, it is challenging in complex systems since the logical combination of various faults should be considered.

B. Simulation Model and Analysis

We use Stella [22] to model relationships among the component sub-systems of a FEW system. Stella supports system dynamics, discrete event, and agent-based simulation techniques. We use system dynamic and discrete event techniques to study the effect of external faults on the overall performance of the system. The high-level model of the system is illustrated in Figure 4. As mentioned in III, our system has four main components. We assume that the microgrid is a separate component designed only for the FEW operation and it is not connected to the wider electric grid. This assumption simplifies the study of faults in the system resilience.

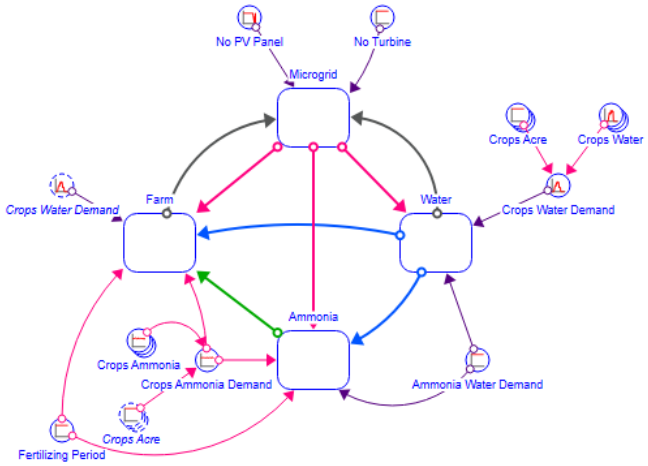

Fig. 4. Simulation model

We consider operational cost as 1 cent per KWh [23]. In this simulation, we do not account for ramp time, which is the time it takes for a plant to start producing electricity when it is turned on. We also do not consider the minimum run time, which is the shortest duration that a plant can generate electricity. The time unit used for wind turbines and solar panels is days. We assume that wind turbines can generate electricity for 24 hours a day, and solar panels can generate electricity for 12 hours a day. The amount of electricity generated during the production time follows a

normal distribution that can be adjusted. The farm module requires electricity to pump water for irrigation. The system determines the amount of water required for irrigation and ammonia production, which serves as fertilizer for different types of crops per acre. The system allows for the adjustment of these parameters. However, the model has a maximum capacity limit for irrigation, which is also adjustable. The module simulates the use of ammonia for fertilization and its consumption in the module. Additionally, the module also simulates the deficiency of water and ammonia, which can be considered a fault in the system. This deficiency can decrease the yield of crops.

The water module in the system simulates pumping water from wells or surface water. The pumping capacity is limited and can be adjusted in the model. Additionally, the cost of pumping water increases linearly with the water level in the wells. When water is pumped from the well, the water level decreases. If the rate of incoming water flow is less than the outflow, it can lead to a decrease in water level, resulting in higher pumping costs until it is no longer economically viable. This scenario can create an external fault in the system. Moreover, if the water level falls to a critical level, it can damage the aquifer, requiring pumping to be stopped for a period, which is adjustable in the model. The pumped water is used for irrigation and ammonia production.

While pumps are typically maintained and repaired before planting season, it is possible that pumps fail during the growing season. Pump failure at the beginning of the season is costly since it damages to the crop yield compared to later in the season when crops are fully grown. This type of fault is also implemented in the model.

The ammonia module simulates a solid oxide electrolysis cell (SOEC) with an exothermal Harber-Bosch Reactor. This technology has about 20% energy saving compared to the conventional Harber-Bosch technology [24]. We consider the same energy consumption pattern in this simulation. We also assume producing one tone of ammonia per day requires 334 kW electricity [25].

The ammonia module consists of four key components: an air separation unit, a solid oxide electrolysis module, a gas compressor, and a Harber-Bosch reactor. The solid oxide electrolysis module is the most electricity-intensive component, requiring approximately 315 KW per day, as well as water. The availability of electricity and water directly influences ammonia production. Insufficient ammonia supply can lead to decreased crop yield. The ammonia demand for crops is adjusted according to the crop type. Any surplus ammonia can be sold or used to generate electricity.

We run a simulation of the model for a sample farm under normal operating conditions where resources are sufficient. However, we also impose some external faults on the model during the simulation. These faults include failures of microgrid components such as PV panels and wind turbines, as well as the water component where the water level in wells is reduced, and the irrigation requirements are changed. As a result of these failures, the amount of pumping water and ammonia production changes. Table I shows the simulation parameters and the type of failure distribution imposed on the model. The *external* values in the table show that faults happen outside the module, but they affect the module. The *fixed* values show constant values during the simulation, but they are adjustable.

Simulation Parameter	Value	Failure distribution
Wind turbine	10	linear
PV Panel	25	linear
Ammonia production	1 tone/day	external
Water (ammonia)	1588 liters	external
Fertilization period	180	fixed
Crops ammonia demand	Normal dist.	external
No. Crops	3 (adjustable)	fixed
Unit of planting	Acre	fixed
Irrigation	Normal dist.	external
Pumping capacity	5000 liters/day	linear/external
Irrigation capacity	2000 liters/day	fixed
Duration	365 days	NA
No. of Runs	10	NA

TABLE I
SIMULATION PARAMETERS

Figure 5 shows electricity production with %95 confidence interval. Ammonia production requires a fixed amount of water and electricity, while irrigation is a periodic process, and the amount of water may change in each period due to any failure in the microgrid and water modules, including the water level in wells. It explains more variability on top of the graph.

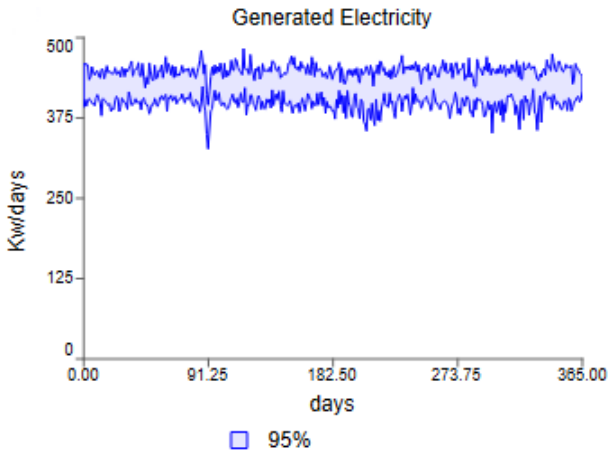

Fig. 5. Electricity Production

Figure 6 illustrates the draining water process with %95 confidence interval. To produce ammonia, a fixed amount of water is required, so the graph shows a straight line for the period when water is pumped for ammonia production. Values below 1588 liters per day indicate a fault, which could be due to pumping failure, electricity production failure, or changes in the water level. Furthermore, irrigation takes priority over ammonia production, meaning that if there is not enough water for both irrigation and ammonia production, the water will be redirected to the irrigation system instead. This policy explains changes in Figure 7. The ammonia system needs a certain amount of water, but sometimes it receives less water due to

faults or failures in the system. The mean value on the graph is close to the top, but there are notable differences between the mean and the bottom of the graph. These differences demonstrate how faults and failures in the system impact the amount of water needed for ammonia production. Additionally, even with sufficient resources for normal operation, ammonia production may not be consistent in the face of minor faults or failures.

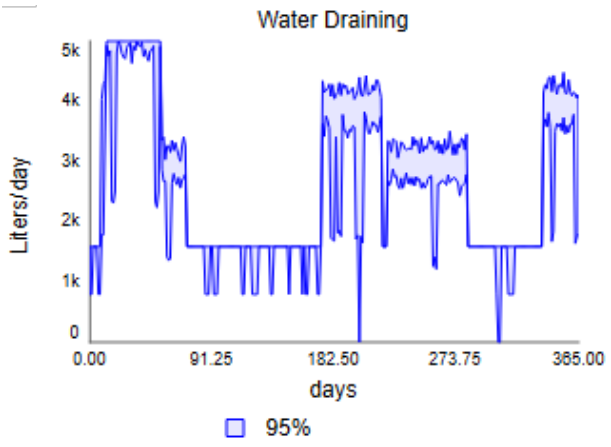

Fig. 6. Water draining from surface and ground resources

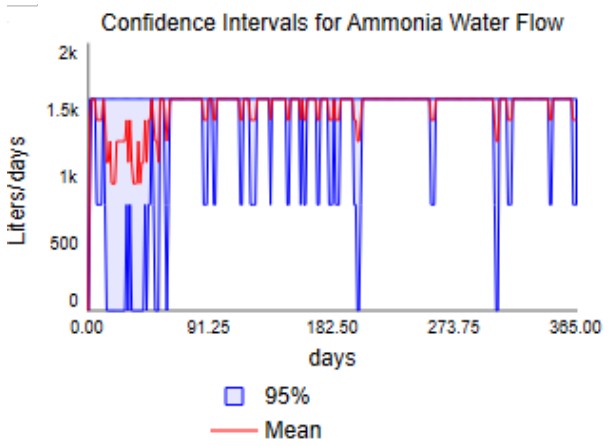

Fig. 7. Water flow for Ammonia production

IV. Conclusion

In this paper, we explain common faults and perturbations for a food-energy-water system and present a model that permits simulation of some of these challenges. This framework will allow for more policies and scenarios to be added to future iterations of the simulator. Our next step, is to use graph theory to consider the ways that failures of internal components affect the resilience of the overall system. The overarching goal of the project is to provide decision-support tools that allow communities and policymakers to simulate possible courses of action for designing resilience into complex smart agriculture systems.

References

[1] A. Modarresi and J. Symons, "Modeling resilience for sustainable food, energy, and water systems," in *AGU Fall Meeting Abstracts*, vol. 2021, pp. IN45F–0501, 2021.

[2] A. Modarresi and J. Symons, "Technological heterogeneity and path diversity in smart home resilience: A simulation approach," *Procedia Computer Science*, vol. 170, pp. 177–186, 2020.

[3] F. Pipa and J. Symons, "Towards an understanding of resilience with complex networks," in *Proceedings of the 6th annual symposium on hot topics in the science of security*, pp. 1–1, 2019.

[4] A. Avizienis, J.-C. Laprie, B. Randell, and C. Landwehr, "Basic concepts and taxonomy of dependable and secure computing," *IEEE transactions on dependable and secure computing*, vol. 1, no. 1, pp. 11–33, 2004.

[5] A. Avižienis, J.-C. Laprie, and B. Randell, "Dependability and its threats: a taxonomy," in *Building the Information Society*, pp. 91–120, Springer, 2004.

[6] A. Avizienis and J. P. Kelly, "Fault tolerance by design diversity: Concepts and experiments," *Computer*, vol. 17, no. 08, pp. 67–80, 1984.

[7] J. von Knop *et al.*, "Basic concepts and taxonomy of dependable and secure computing1," *A Process for Developing a Common Vocabulary in the Information Security Area*, vol. 23, p. 10, 2007.

[8] J. E. Hosford, "Measures of dependability," *Operations Research*, vol. 8, no. 1, pp. 53–64, 1960.

[9] J.-C. Laprie, "Dependable computing and fault-tolerance," *Digest of Papers FTCS-15*, vol. 10, no. 2, p. 124, 1985.

[10] J.-C. Laprie, "Dependability: Basic concepts and terminology," in *Dependability: Basic Concepts and Terminology*, pp. 3–245, Springer, 1992.

[11] F. Cristian, "Understanding fault-tolerant distributed systems," *Communications of the ACM*, vol. 34, no. 2, pp. 56–78, 1991.

[12] J. Symons, "The individuality of artifacts and organisms," *History and philosophy of the life sciences*, pp. 233–246, 2010.

[13] H. Wang, Y. Mitake, Y. Tsutsui, S. Alfarisi, and Y. Shimomura, "A taxonomy of product–service system perturbations through a systematic literature review," *Journal of Risk and Financial Management*, vol. 15, no. 10, p. 443, 2022.

[14] A. Modarresi and J. Symons, "Modeling technological interdependency in IoT - a multidimensional and multilayer network model for smart environments," in *2019 11th International Workshop on Resilient Networks Design and Modeling (RNDM) (RNDM 2019)*, (Nicosia, Cyprus), Oct. 2019.

[15] A. Modarresi and J. P. G. Sterbenz, "Towards a model and graph representation for smart homes in the IoT," in *2018 IEEE International Smart Cities Conference (ISC2) (ISC2 2018)*, (Kansas City, USA), Sept. 2018.

[16] Y. Li, S. Xie, Z. Wan, H. Lv, H. Song, and Z. Lv, "Graph-powered learning methods in the internet of things: A survey," *Machine Learning with Applications*, vol. 11, p. 100441, 2023.

[17] C. Perrow, *Normal accidents: Living with high risk technologies*. Princeton university press, 1999.

[18] S. M. Rinaldi, J. P. Peerenboom, and T. K. Kelly, "Identifying, understanding, and analyzing critical infrastructure interdependencies," *IEEE control systems magazine*, vol. 21, no. 6, pp. 11–25, 2001.

[19] S. E. Chang, "Infrastructure resilience to disasters," *The Bridge*, vol. 39, no. 4, pp. 36–41, 2009.

[20] A. Vespignani, "Complex networks: The fragility of interdependency," *Nature*, vol. 464, no. 7291, p. 984, 2010.

[21] J. K. Horner and J. Symons, "Understanding error rates in software engineering: Conceptual, empirical, and experimental approaches," *Philosophy & Technology*, vol. 32, pp. 363–378, 2019.

[22] "Isee systems website." https://www.e-education.psu.edu/eme801, 2023. Accessed on April 2023.

[23] P. EME 801: Energy Markets and Regulation, "Energy markets, policy, and regulation." https://www.e-education.psu.edu/eme801, 2020. Accessed on April 2023.

[24] C. O. Ofori-Bah and V. Amanor-Boadu, "Directing the wind: Techno-economic feasibility of green ammonia for farmers and community economic viability," *Frontiers in Environmental Science*, vol. 10, p. 2530, 2023.

[25] "Nel hydrogen website." https://nelhydrogen.com/, 2023. Accessed on April 2023.