A Multi-step and Resilient Predictive Q-learning Algorithm for IoT: A Case Study in Water Supply Networks

Maria Grammatopoulou
Grado Department of Industrial and Systems Engineering, Virginia Tech, Blacksburg, VA, USA
mariagr@vt.edu

Aris Kanellopoulos
Daniel Guggenheim School of Aerospace Engineering, Georgia Tech, Atlanta, GA, USA
ariskan@gatech.edu

Kyriakos G. Vamvoudakis
Daniel Guggenheim School of Aerospace Engineering, Georgia Tech, Atlanta, GA, USA
kyriakos@gatech.edu

ABSTRACT

In this paper, we consider the problem of deriving recommended resilient and predictive actions for an IoT network in the presence of faulty components and malicious agents. The IoT, combining physical and cyber devices, is formulated as a directed graph with a known topology whose objective is to maintain a constant and resilient flow between a source node and a destination node. The optimal route through this network is evaluated via a predictive and resilient Q-learning algorithm which takes into account historical data about irregular operation, including faults and attacks. To showcase the efficacy of our approach, we utilize anonymized data from Arlington County, Virginia to obtain predictive and resilient scheduling policies for a smart water supply system while avoiding neighborhoods with leaks and other faults.

Author Keywords

IoT; big data; predictive Q-learning; resilient scheduling; water supply networks.

INTRODUCTION

According to the World Health Organization, urban residents account for 54% of the total global population and that figure is projected to grow by 2% each year through 2020. That growth means that cities will face increasing challenges to meet demands of growing populations for resources such as energy and water whose availability depends on other factors including climate change, weather patterns and natural and man-made hazards. With this recognition, municipal governments around the globe have started to recognize that big data and Internet-of-Things (IoT) can or will play a major role in developing sustainable connected communities while improving many aspects of the daily life of their citizens. Many cities around the world are already overcrowded leading to transport and traffic congestion, and a strain on resources such as water, energy and safe housing. Meeting sustainability goals in cities requires completely new concepts for urban mobility and the sustainable use of resources in other words, the evolution of a city into a *smart city*. Using sensors, the smart city concept is based on connected technology that has connected "things" communicating "live" information with each other.

Due to the exponential increase of devices equipped with networking capabilities, researchers have introduced the concept of the IoT [4]. The IoT consists of both physical and cyber devices communicating via standard TCP/IP protocols. The key characteristics of the IoT are the unprecedented amount of connected devices, with experts projecting the number to 24 billion by 2020 [5], as well as the heterogeneous nature of those devices. As an example, we can consider a single network containing smart meters deployed in public areas [23], individual wearable devices collecting health data [11] as well as large-scale systems like heating, ventilation, and air conditioning (HVAC) [2], all exchanging data with a single user's mobile phone.

It is expected that not only will the IoT protocols have to support the different big data structures, but in many cases they will also have a direct effect on the physical world as well. Consequently, it is important to ensure safe and robust operation of the IoT networks in case of random or malicious faults.

The aforementioned complexity of the IoT networks, coupled with the need for front-end user interfaces that allow for asymmetric access to the various subsystems only from certain authorized users, has led to concerns about the security/resilience of IoT. Research has been conducted on those issues both in experimental and real-world scenarios.

The authors in [13] investigated security issues in smart-lock systems, showing that the successful attack angles relied heavily on the interconnections between the physical component - the lock itself - and the cyber component, the user's mobile phone. Furthermore, in [24] the authors reported on the effects of Sybil attacks which leverage fake identity manipulation to gain access to different nodes of the IoT network. Sinkhole attacks have also been shown to have the ability to compromise IoT networks [24].

Current research directions pursue the integration of existing urban critical infrastructure with the IoT network via sensing, communicating and actuating devices in order to establish a

IoT'18, Oct 15–18, 2018, Santa Barbara, CA, USA

DOI: https://doi.org/10.1145/3277593.3277605

smart city [12]. Eventually, smart cities will be a large scale extension of the IoT network. Several public services will benefit from IoT-enabled smart cities, in their accessibility as well as their ability to make better use of their resources [23]. One of the most important parts of a city's infrastructure is its water supply network. Owing to the rapid expansion of urban areas, those networks tend to become more complex [10]. However, the need to keep them operational at all times persists, even in the presence of faults. It is known that water supply systems are prone to a number of catastrophic failures like pipe busts and leakage problems [1]. Thus, continuous monitoring of the network's status must be prioritized. However, human analysts can process only a limited amount of the data collected from the infrastructure's sensors. The IoT can ensure safe operation of smart city networks by autonomously collecting raw data from the sensor-integrated infrastructure, processing the information and providing action recommendations to the human analyst.

Machine learning methods can be employed to provide the IoT with adaptive decision making capabilities. While the amount of data exchanged in the IoT act as a hindrance to the smooth operation of the network when it has to be handled by humans, big data solutions can be leveraged to facilitate the training of automated intelligent assistants. Reinforcement learning is a machine learning approach in which the decision makers learn their optimal policies by interacting with the environment and evaluating their actions [19]. Borrowing ideas from Dynamic Programming [6], many reinforcement learning techniques derive the optimal policies via learning value functions. Q-learning was the first provably convergent reinforcement learning algorithm, initially developed for Markov Decision Processes (MDP) with unknown transition probabilities. In Q-learning frameworks, the decision maker evaluates an action-dependent value function, through non-optimal policies [21]. Application of the principles of Q-learning in routing problems in networks has led to the formulation of Q-routing algorithms [7].

Contributions

The contributions of this work are as follows. First, we model the IoT network as a graph and consider the problem of deriving recommended and resilient policies for transferring data and supplies between nodes. Subsequently, we employ a method based on Q-learning to derive the optimal policies online in a dynamic fashion that mitigates congestion in the optimal path while learning to avoid nodes that presented faults, either due to component failure, or due to malicious attacks. Furthermore, we introduce performance metrics to evaluate the resilience of the proposed path to faults. Finally, we apply the proposed framework to a smart pipeline network, including real anonymized data provided by the Arlington County, Virginia, USA.

Related Work

Researchers seek to categorize the different threats and attack angles that are expected to endanger the successful integration of the IoT. The authors in [3] differentiate between the attack vectors taking place in the various layers of the IoT network; the physical, digital, and communication layers. On the other hand, the work of [18] suggests to securely encrypt the data exchanged between the subsystems of the network.

The authors in [9] designed a framework to detect faults in IoT networks by employing various network tools and protocols alongside fault-sensing physical devices, whereas in [17], resource reconfiguration in the IoT was investigated as a solution to service failure. Considerable research has also been conducted for faults due to malicious attacks and mitigating approaches. Several threat models in the IoT, such as authentication spoofing and man-in-the-middle attacks as well as attacks that compromise the network's privacy, were reported in [22]. To defend against such threats, they tested different machine learning algorithms, both supervised and unsupervised, to derive the defending parameters. In [24], the authors leveraged graph-theoretic methods to detect Sybil attacks on IoT networks taking into account the social behavior of regular and malicious users.

Optimization techniques alongside historical data to predict cyber attacks in computer networks were presented in [20]. Reinforcement learning approaches have mostly been employed for cyber security in conjunction with game theory. For example, in [16], the authors proposed a general framework that formulates cyber security as a partial information stochastic game. In [15], semi-supervised learning in support of the IoT networks was employed. Q-routing was introduced in [7], while the authors in [8] extended the framework by introducing prediction elements enabling the decision maker to adapt to dynamic network traffic patterns.

Structure

The paper is organized as follows. The second section formulates the IoT subnetwork and the corresponding scheduling problem we wish to solve, followed by a general description of the predictive and resilient Q-learning algorithm. In the third section, we apply the proposed algorithm to a water supply network utilizing anonymized real data of observed leaks and we define appropriate performance metrics. Finally, the fourth section concludes the paper and discusses future research directions.

PROBLEM FORMULATION AND PROPOSED FRAMEWORK

Network Model

Initially, we model the IoT subnetwork under consideration as a directed graph. With this approach, a variety of different services can be described. For example, one may describe the data transfer between computational components and sensing/actuating devices, or even physical networks monitored and controlled through the IoT, such as a smart city pipeline network. Expert analysts are expected to supervise the smooth and safe operation of different sub-networks in future IoT and smart city scenarios. As a result, algorithms that facilitate the decision making process, by pre-processing the raw data collected by the network and feeding higher level suggestions to the human-in-the-loop operator, must be developed. Thus, analysts will maintain high-speed situational awareness of the network status. This can be achieved by utilizing reinforcement learning techniques that render the network a black box

by predicting future behavior based on previous data, and extracting recommended policies, attack trends and vulnerability assessments, that can be used by the "watch-standers."

Scheduling Problem

The objective of the decision maker is to derive the optimal flow from the source to the destination node of the network. In order to define the scheduling problem as an optimization problem, we associate each edge of the graph with a positive number r_{ij}^k, indicating the cost to transfer a single packet from node i to node j at time k. Thus, we define the cost matrix,

$$R^k = [r_{ij}^k],\ r_{ij}^k \geq 0\ \forall i, j, k.$$

To select the optimal scheduling policy, we seek to minimize the accumulated cost of the flow from the source to the destination.

However, in a safe and robust IoT scenario, we have to account for prior random faults and malicious attacks in certain nodes of the network. For example, if the IoT sub-network in question consists of computers connected to a smart home, then each node is vulnerable to Denial of Service (DoS) attacks, whereas in smart water supply systems, the history of leaks should be considered when evaluating the flow of critical supplies. Consequently, the cost matrix is dynamically updated to reflect data collected during the run of the system, thus containing different values of at each time k.

Predictive Q-learning

We utilize a Q-learning algorithm to derive the optimal policies that will be used as recommended and resilient policies for the network operator. Similar to other Q-learning approaches, in predictive Q-learning we define an action-dependent value function $Q^k(s,a)$ for state s and action a. This function should also contain information about the measured past faults. Therefore, the Q-function has the form,

$$Q^k(s,a) = \sum_{t=k-M}^{k} R^t + \sum_{t=1}^{k-M} b^t R^t, \tag{1}$$

where $b \in (0,1)$ is the discount factor and $M > 0$ is the size of the window. Equation (1) implies that, during each time-step, we take into account the faults and attacks observed throughout a predefined time window of length M, while knowledge observed farther in the past affects the scheduling less.

Finally, our objective is to derive the optimal Q-function,

$$Q_{\text{opt}}^k = \min_a Q^k(s,a).$$

The dynamic nature of the environment, caused by faults and attacks on the network, leads to the shifting structure of the reward matrix and the Q-function as described above. It is important for the decision maker to have the ability to adapt the scheduling policy as fast as possible while being resilient. Therefore, the Q-learning problem in this work is inspired by the structure of Predictive Q-routing (PQ-routing) [8]. The novelty of PQ-routing, is in its consideration of the congestion created by the optimal routing policy itself. Specifically, it is argued that when we need to statically use the optimal path, the increase in traffic from specific nodes would decrease the efficiency of the path.

While simple Q-learning utilizes the principles of reinforcement learning by exploring the state and action spaces and gradually converging to the optimal policy, in predictive Q-learning, even after the algorithm has converged, we use probing packets to test the status (resilience) of different routes. This way, if the optimal path was congested, and we had switched to a different path, the learning scheme still tests the, already learned, optimal path, expecting it to eventually revert back to normal traffic. Special care has to be taken when the frequency of probing is selected. High frequency will increase the resilience levels of the optimal path even more, while low frequency will decrease the response of the network to traffic changes. In predictive Q-learning schemes, the speed with which the packets go through a specific node - the node's recovery rate - is assumed to be unknown and it is estimated online.

Algorithm 1, given below, is the pseudocode for the proposed predictive and resilient Q-learning. Specifically, we define $Q^k(s_i,a_j)$ as the estimated Q-value of the state-action pair s_i and a_j where i and j are the nodes, $B^k(s_i,a_j)$ is the minimum cost incurring when in state s_i, action a_j is taken. Also, $\text{RR}^k(s_i,a_j)$ and $U(s_i,a_j)$ are the recovery rate and the last update time, respectively, when action a_j is chosen from state s_i. We use three learning parameters in the predictive and resilient Q-learning framework, α, β, and γ. As in the classic Q-learning algorithm, α is the Q-function learning parameter, which should be equal to 1 or the accuracy of the recovery rate might be affected. The recovery rate learning parameter, i.e., β, needs to obey $\beta < \gamma$, in order to regulate the decay of the recovery rate, i.e., γ, that has a direct effect on the probing frequency of a non-resilient path.

EXPERIMENTAL ANALYSIS

This section, demonstrates an application of the proposed predictive and resilient Q-learning algorithm on a water network, for the prediction of the location of future leaks and the formation of a path, with determined start (source) and end (destination) points, in which as many locations having leaks as possible are avoided. In order to do so, we used data of the leaks that occurred over the last five years in the Arlington County, Virginia. In this experiment, we want to make sure that no matter what happens (possible attacks or leaks on the pipelines of the water network of the County), water from the assumed source, namely neighborhood 1, reaches the Ronald Reagan Washington Regional Airport, which we assume that is the destination and corresponds to neighborhood 119. The dataset, the assumed network topology, the training, as well as the results, will be shown next.

Dataset

The dataset used contained 1816 instances of leaks in the water network of the Arlington County, over the last five years. Each instance involves information about the location of the leak, the time period between the identification of the leak and the recovery duration, and the occurred cost.

Algorithm 1: Predictive and resilient IoT Q-learning

procedure
Set the α, β, γ parameters.
for every time window k
Set environment rewards matrix R^k
Initialize matrices Q^k_{opt} and B^k with sufficiently large numbers
Initialize matrices Q^k and U^k to zero
Set the matrix RR^k appropriately
for each epoch
Select a random initial state s_0
while the goal state has not been reached
Select action a_i among all possible actions for the current state
Using this action a_i, consider going to the, next state, s_j
$\Delta Q = r^k_{ij} + \min_{a_k} Q^k(s_j, a_k) - Q^k(s_i, a_j)$
$Q^k(s_i, a_j) \leftarrow Q^k(s_i, a_j) + \alpha \Delta Q$
$B^k(s_i, a_j) \leftarrow \min(B^k(s_i, a_j), Q^k(s_i, a_j))$
if $\Delta Q < 0$
$\Delta\text{RR} \leftarrow \Delta Q / \big(\text{current time} - U^k(s_i, a_j)\big)$
$\text{RR}^k(s_i, a_j) \leftarrow \text{RR}^k(s_i, a_j) + \beta \Delta R$
else $\Delta Q > 0$
$\text{RR}^k(s_i, a_j) \leftarrow \gamma \text{RR}^k(s_i, a_j)$
end if
$U^k(s_i, a_j) \leftarrow$ current time
$\Delta t =$ current time $- U^k(s_i, a_j)$
$Q^k_{\text{opt}}(s_i, a_j) = \max\big(Q^k(s_i, a_j) + \Delta t \text{RR}(s_i, a_j), B^k(s_i, a_j)\big)$
Set the next state j as the current state
end while
end for
$y \leftarrow \arg\min\{Q^k_{\text{opt}}(s_i, a_j)\}$
end for
end procedure

For the purposes of this work, the geographic area of Arlington County was divided into 119 neighborhoods as shown in Figure 3 and Figure 4. As mentioned before, we are assuming that neighborhood 1 corresponds to the source of the water network of the County, as it is by the bank of river Potomac, and that the Ronald Reagan Washington Regional Airport, which corresponds to neighborhood 119, is the destination of the water network. These two neighborhoods (1 and 119) are considered to have no leaks.

The data of the location instances were classified into 117 sets depending on their location ID, each set corresponding to a geographic area of the Arlington County, as considered for the purposes of this work. The total number of leaks appearing in each neighborhood are as shown in Figure 1. We can see that on neighborhoods 5 and 86 are the most vulnerable, as they have the most leaks, i.e. 30, followed by neighborhood 49, which has 29 leaks. On the contrary, the neighborhoods 115, 116, 117 and 118 have just one leak each, making them the least vulnerable of all neighborhoods. All the other information, but the location of the leaks, is used to determine the rewards assigned to the edges of the network connecting the neighborhoods.

Specifically, in order to determine the rewards of the edges of the network, first we define the rewards for each one of its nodes, NR^k_i, as

$$\text{NR}^k_i = a\text{NL}^k_i + b\text{OC}^k_i + c\text{TTFL}^k_i,$$

where NL^k_i denotes the number of leaks at node i at time k, OC^k_i the occurred cost because of the leaks at node i at time k, and TTFL^k_i the time to fix the leaks at node i at time k. The values of a, b, and c are taken as 0.2, 0.3, and 0.5, respectively, since the time to fix the leaks of a node is considered to be the most significant parameter among the three.

Having illustrated how the rewards of the nodes of the network are computed, we can now specify the rewards of its edges. Considering nodes i and j, there is an edge connecting these two nodes if and only if they share a border, and we can define the reward of the edge ij at time k, r^k_{ij}, as

$$r^k_{ij} = d\max\big(\text{NR}^k_i, \text{NR}^k_j\big) + (1-d)\min\big(\text{NR}^k_i, \text{NR}^k_j\big),$$

where we consider d to be the rate of the minimum value between NR^k_i and NR^k_j over the maximum value of them, and we use it in order to add more weight to the reward of the most vulnerable node.

As mentioned above, we consider two neighborhoods sharing a border to be connected with a direct pipeline. This is used to define the states and actions of the proposed predictive and resilient Q-learning algorithm. The state-action pair (s_i, a_j) denotes that while being on neighborhood i, we decide to move to neighborhood j, which is directly connected to neighborhood i. Therefore the $Q(s_i, a_j)$ is the cost of channelling the water from neighborhood i to neighborhood j. Undoubtedly, the more the neighborhoods a specific neighborhood shares a border with, the largest the space of the actions corresponding to that state, and the more computationally complex the problem is.

Training Details

The proposed framework of predictive and resilient Q-learning uses time windows, each time window handling $M = 30$ data instances. First the reward matrix for the time window is computed, accounting not only for the current number, the time-to-fix, and the cost produced by the leaks, but also for the past values of them in the neighborhood, in an exponentially decreasing way. Following that, the training phase starts, in which the system is trained for 100 epochs.

Once the training for the time window is completed, the location of the possible future leaks is predicted, and a path, for connecting the source with the destination, involving if not none, as few as possible, neighborhoods with leaks is proposed. The rationale behind the selection of the neighborhoods used in the proposed path is that we want to reach our destination point with a minimum cost. To achieve that, the selection of the nodes is based on the Q_{opt} matrix of the time window, which contains the cost for transitioning from one neighborhood to another. For the neighborhoods having leaks,

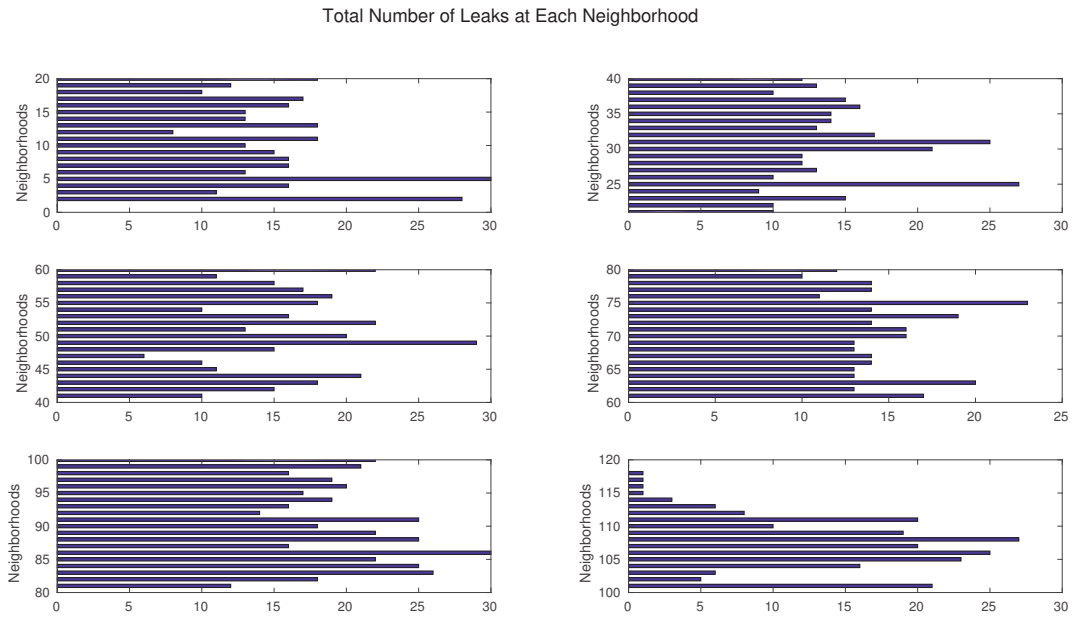

Figure 1. Total number of leaks appearing in each neighborhood in the Arlington County, Virginia.

and not being part of the proposed path, we suggest isolating them from the water network, in order to facilitate the repair of the pipelines and reduce the expenditures.

With a significant amount of training data, the absolute difference between the values of two consecutive Q_{opt} matrices should be approaching zero. As we can see in Figure 2, the absolute value of the difference of the Q_{opt} matrices between two consecutive time windows in our system converges to zero, and after the 50th time window remains 0. The fluctuations appearing before the 50th time window are caused by the fact that leaks in time window $(k+1)$ are appearing in different neighborhoods than in time window k, since our system is not trained on the specific scenarios, and thus the predictions of the locations of the future leaks differ from the actual ones.

At this point, we should note that the forgetting factor is taken equal to $b = 0.1$ since we want to penalize more recent data.

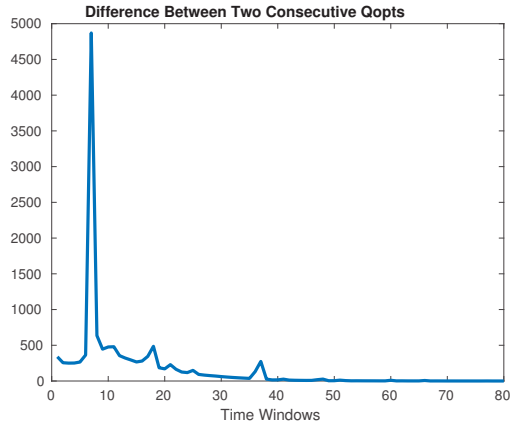

Figure 2. Difference between two consecutive Q_{opt} matrices over time.

Results

The more the training windows, the better the predictions of our system are. To have a better understanding of how the system evolves and learns from the past time windows, we can take a look on the results of the first and last time window.

Figure 3 shows the proposed path, as well as the nodes having leaks at the end of the first time window. The proposed path starts from the source (neighborhood 1), and passing through neighborhoods 4, 10, 17, 18, 19, 27, 42, 44, 54, 55, 76, 87, 91, 97, 107, 111 and 118, reaches the destination (neighborhood 119). One might be confused as for why going from neighborhood A to neighborhood C via neighborhood B (as it happens for neighborhoods 44, 54 and 55). The reason is that it will be more effective than using the direct connection (pipeline) between the first two neighborhoods.

Similarly, on Figure 4 we can see the proposed path, as well as the nodes having leaks at the end of the last time window. The proposed path, following the logic analyzed before, starts at the source (neighborhood 1), passes through neighborhoods 3, 8, 22, 48, 51, 64, 74, 83, 87, 91, 97, 107, 111, 118, and get to the destination (neighborhood 119). Once the last time window has passed, our system has finished the training for all the available data, it has deeper knowledge of the network and the vulnerability of each neighborhood, and can give better predictions. As a consequence, the proposed path consists of less neighborhoods than the one after the first time window.

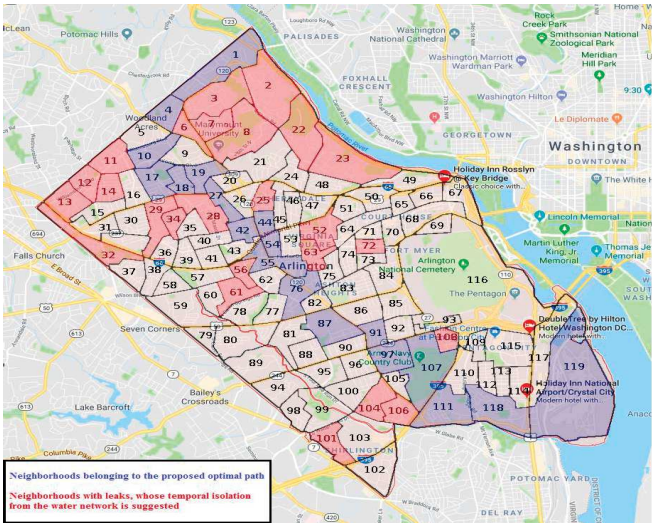

Figure 3. Optimal proposed path and neighborhoods having leaks after the first time window.

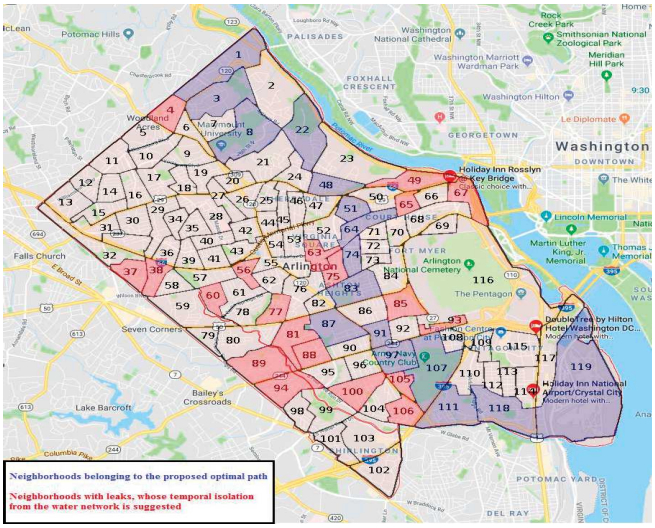

Figure 4. Optimal proposed path and neighborhoods with leaks after the last time window.

Looking at both Figures 3 and 4, one can observe that both after the first and the last time window, the neighborhoods 56 and 106 appear to have leaks and their isolation from the rest of the network is suggested. A neighborhood may appear between the ones suggested for isolation in multiple time windows in the following two cases: (a) the leak in the pipelines has not been fixed yet, or (b) it is of high vulnerability and leaks keep appearing in the network. In the first case the neighborhood will be among the isolated ones in two consecutive time windows, while in the second case this is not necessarily true.

Even though the optimal paths proposed after the first and last time windows do not involve neighborhoods with leaks, this is not always the case. As we can see in Figure 5, which shows the cost of the optimal path when neighborhoods having leaks are part of it, sometimes it is more cost effective to employ a neighborhood with leaks in the optimal path.

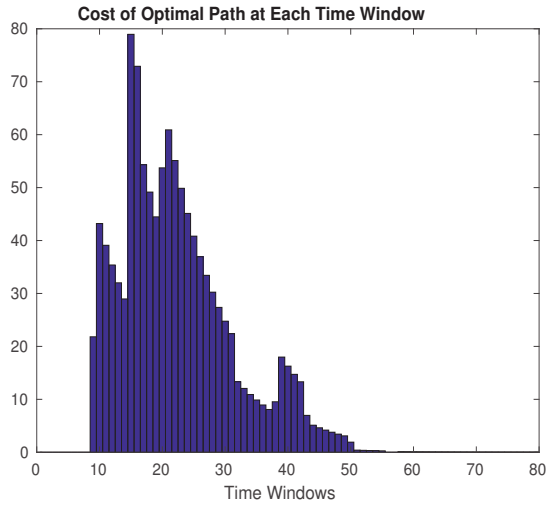

Figure 5. Cost of the optimal path due to the appearance of leaks.

Performance Evaluation

We introduce the metric of blocking probability, as described in [14], which can be used to evaluate the ability of the proposed approach to adapt to node failures. We define BPW_k, the blocking probability of time window k, as,

$$\mathrm{BPW}_k = \frac{\mathrm{BNW}_k}{\mathrm{BNW}_k + \mathrm{OPN}_k},$$

where BNW_k denotes the number of blocked neighborhoods in the network, and OPN_k the number of neighborhoods without leaks that belong to the optimal path. The value of BPW_k is evaluated at each time window k.

Figure 6 shows the blocking probability of our system over time. Although our objective is to propose a path that involves as few nodes with leaks as possible, in every time window we have leaks appearing on multiple neighborhoods. This causes the blocking probability to be high.

In addition, we introduce the metric of the performance of the whole network, NP, as,

$$\mathrm{NP} = \frac{\sum_k \mathrm{BNW}_k}{\sum_k \mathrm{BNW}_k + \sum_k \mathrm{OPN}_k}.$$

In our simulation, the aforementioned performance metric is found to be $\mathrm{NP} = 0.9621$.

CONCLUSION

This work presents an algorithm that enables large-scale IoT networks to provide human operators with recommended and resilient policies for scheduling problems. We model the IoT as a graph on which we formulate a scheduling problem. We integrate past data from recorded node failures in real-time by updating the cost matrix of the graph and thus dynamically shifting the optimal path choice. We utilize predictive and resilient Q-learning to consider the change in the environment

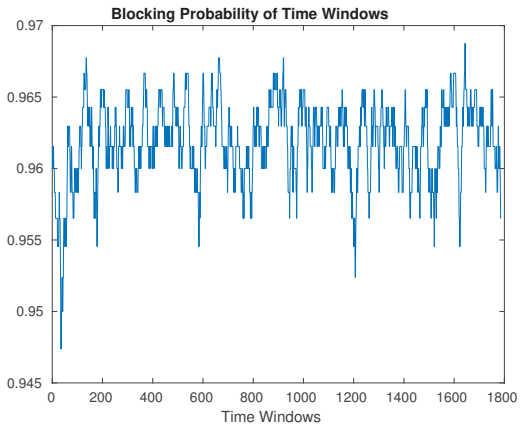

Figure 6. Blocking probability per window.

due to prior inconsistencies, as well as the effect of congestion by the optimal policy itself. The experimental results show the efficacy of the proposed method in a smart water supply system. Real anonymized data provided by the Arlington County, Virginia, USA, were utilized to highlight the effects of malicious, random, and non-random water network failures.

Future work will be focused on applying the developed framework to cyber defense scenarios with human analysts in the loop.

ACKNOWLEDGMENTS

This work was supported in part by ONR Minerva under grant No. N00014-18-1-2160, and by an NSF CAREER under grant No. CPS-1750789.

REFERENCES

1. Maroua Abdelhafidh, Mohamed Fourati, Lamia Chaari Fourati, and Amor Abidi. 2017. Remote Water Pipeline Monitoring System IoT-Based Architecture for New Industrial Era 4.0. In *Computer Systems and Applications (AICCSA), 2017 IEEE/ACS 14th International Conference on*. IEEE, 1184–1191.
2. Ala Al-Fuqaha, Mohsen Guizani, Mehdi Mohammadi, Mohammed Aledhari, and Moussa Ayyash. 2015. Internet of things: A survey on enabling technologies, protocols, and applications. *IEEE Communications Surveys & Tutorials* 17, 4 (2015), 2347–2376.
3. Ioannis Andrea, Chrysostomos Chrysostomou, and George Hadjichristofi. 2015. Internet of Things: Security vulnerabilities and challenges. In *Computers and Communication (ISCC), 2015 IEEE Symposium on*. IEEE, 180–187.
4. Luigi Atzori, Antonio Iera, and Giacomo Morabito. 2010. The internet of things: A survey. *Computer networks* 54, 15 (2010), 2787–2805.
5. Alessandro Bassi and Geir Horn. 2008. Internet of Things in 2020: A Roadmap for the Future. *European Commission: Information Society and Media* 22 (2008), 97–114.
6. Dimitri P Bertsekas, Dimitri P Bertsekas, Dimitri P Bertsekas, and Dimitri P Bertsekas. 1995. *Dynamic programming and optimal control*. Vol. 1. Athena scientific Belmont, MA.
7. Justin A Boyan and Michael L Littman. 1994. Packet routing in dynamically changing networks: A reinforcement learning approach. In *Advances in neural information processing systems*. 671–678.
8. Samuel PM Choi and Dit-Yan Yeung. 1996. Predictive Q-routing: A memory-based reinforcement learning approach to adaptive traffic control. In *Advances in Neural Information Processing Systems*. 945–951.
9. Bishnu Prasad Gautam, Katsumi Wasaki, and Narayan Sharma. 2016. A novel approach of fault management and restoration of network services in IoT cluster to ensure disaster readiness. In *Networking and Network Applications (NaNA), 2016 International Conference on*. IEEE, 423–428.
10. Daniel Granlund and Robert Brännström. 2012. Smart city: the smart sewerage. In *Local Computer Networks Workshops (LCN Workshops), 2012 IEEE 37th Conference on*. IEEE, 856–859.
11. Jayavardhana Gubbi, Rajkumar Buyya, Slaven Marusic, and Marimuthu Palaniswami. 2013. Internet of Things (IoT): A vision, architectural elements, and future directions. *Future generation computer systems* 29, 7 (2013), 1645–1660.
12. José M Hernández-Muñoz, Jesús Bernat Vercher, Luis Muñoz, José A Galache, Mirko Presser, Luis A Hernández Gómez, and Jan Pettersson. 2011. Smart cities at the forefront of the future internet. In *The Future Internet Assembly*. Springer, 447–462.
13. Grant Ho, Derek Leung, Pratyush Mishra, Ashkan Hosseini, Dawn Song, and David Wagner. 2016. Smart locks: Lessons for securing commodity internet of things devices. In *Proceedings of the 11th ACM on Asia conference on computer and communications security*. ACM, 461–472.
14. Shuo Li, Moshe Zukerman, Meiqian Wang, and Eric WM Wong. 2015. Improving throughput and effective utilization in OBS networks. *Optical Switching and Networking* 18 (2015), 222–234.
15. Mehdi Mohammadi, Ala Al-Fuqaha, Mohsen Guizani, and Jun-Seok Oh. 2018. Semisupervised deep reinforcement learning in support of IoT and smart city services. *IEEE Internet of Things Journal* 5, 2 (2018), 624–635.
16. Sajjan Shiva, Sankardas Roy, and Dipankar Dasgupta. 2010. Game theory for cyber security. In *Proceedings of the Sixth Annual Workshop on Cyber Security and Information Intelligence Research*. ACM, 34.

17. Penn H Su, Chi-Sheng Shih, Jane Yung-Jen Hsu, Kwei-Jay Lin, and Yu-Chung Wang. 2014. Decentralized fault tolerance mechanism for intelligent iot/m2m middleware. In *Internet of Things (WF-IoT), 2014 IEEE World Forum on*. IEEE, 45–50.
18. Hui Suo, Jiafu Wan, Caifeng Zou, and Jianqi Liu. 2012. Security in the internet of things: a review. In *Computer Science and Electronics Engineering (ICCSEE), 2012 international conference on*, Vol. 3. IEEE, 648–651.
19. Richard S Sutton and Andrew G Barto. 1998. *Reinforcement learning: An introduction*. Vol. 1. MIT press Cambridge.
20. Kyriakos G Vamvoudakis, Joao P Hespanha, Richard A Kemmerer, and Giovanni Vigna. 2013. Formulating cyber-security as convex optimization problems. In *Control of Cyber-Physical Systems*. Springer, 85–100.
21. Christopher JCH Watkins and Peter Dayan. 1992. Q-learning. *Machine learning* 8, 3-4 (1992), 279–292.
22. Liang Xiao, Xiaoyue Wan, Xiaozhen Lu, Yanyong Zhang, and Di Wu. 2018. IoT Security Techniques Based on Machine Learning. *arXiv preprint arXiv:1801.06275* (2018).
23. Andrea Zanella, Nicola Bui, Angelo Castellani, Lorenzo Vangelista, and Michele Zorzi. 2014. Internet of things for smart cities. *IEEE Internet of Things journal* 1, 1 (2014), 22–32.
24. Kuan Zhang, Xiaohui Liang, Rongxing Lu, and Xuemin Shen. 2014. Sybil attacks and their defenses in the internet of things. *IEEE Internet of Things Journal* 1, 5 (2014), 372–383.