RESEARCH ARTICLE | JUNE 13 2024

Heterogeneous reinforcement learning for defending power grids against attacks

Mohammadamin Moradi ; Shirin Panahi ; Zheng-Meng Zhai ; Yang Weng; John Dirkman ; Ying-Cheng Lai

APL Mach. Learn. 2, 026121 (2024)
https://doi.org/10.1063/5.0216874

Heterogeneous reinforcement learning for defending power grids against attacks

Cite as: APL Mach. Learn. 2, 026121 (2024); doi: 10.1063/5.0216874
Submitted: 1 May 2024 • Accepted: 22 May 2024 •
Published Online: 13 June 2024

Mohammadamin Moradi,[1] Shirin Panahi,[1] Zheng-Meng Zhai,[1] Yang Weng,[1] John Dirkman,[2] and Ying-Cheng Lai[1,3,a)]

AFFILIATIONS

[1] School of Electrical, Computer and Energy Engineering, Arizona State University, Tempe, Arizona 85287, USA
[2] Resource Innovations, 719 Main Street, Half Moon Bay, California 94019, USA
[3] Department of Physics, Arizona State University, Tempe, Arizona 85287, USA

[a)] Author to whom correspondence should be addressed: Ying-Cheng.Lai@asu.edu

ABSTRACT

Reinforcement learning (RL) has been employed to devise the best course of actions in defending the critical infrastructures, such as power networks against cyberattacks. Nonetheless, even in the case of the smallest power grids, the action space of RL experiences exponential growth, rendering efficient exploration by the RL agent practically unattainable. The current RL algorithms tailored to power grids are generally not suited when the state-action space size becomes large, despite trade-offs. We address the large action-space problem for power grid security by exploiting temporal graph convolutional neural networks (TGCNs) to develop a parallel but heterogeneous RL framework. In particular, we divide the action space into smaller subspaces, each explored by an RL agent. How to efficiently organize the spatiotemporal action sequences then becomes a great challenge. We invoke TGCN to meet this challenge by accurately predicting the performance of each individual RL agent in the event of an attack. The top performing agent is selected, resulting in the optimal sequence of actions. First, we investigate the action-space size comparison for IEEE 5-bus and 14-bus systems. Furthermore, we use IEEE 14-bus and IEEE 118-bus systems coupled with the Grid2Op platform to illustrate the performance and action division influence on training times and grid survival rates using both deep Q-learning and Soft Actor Critic trained agents and Grid2Op default greedy agents. Our TGCN framework provides a computationally reasonable approach for generating the best course of actions to defend cyber physical systems against attacks.

I. INTRODUCTION

The growing needs for dependable energy and technological breakthroughs have driven further development of the power grids. The vital infrastructure of the electric power grids is susceptible to unanticipated random failures and, more worrisome, to hostile physical and/or cyberattacks that can frequently result in widespread, cascade types of breakdowns with significant damages. For example, during the 2003 eastern US/Canada blackout, ~50 × 10^6 people in eight US States and two Canadian provinces were impacted.[1] In December 2015, a synchronized and coordinated cyberattack on three regional electric power distribution firms occurred in Ukraine, causing power disruptions that lasted several hours and affected about 225 000 people.[2] These events have continued to occur in recent times. For example, in May 2021, the Colonial Pipeline,[3] which supplies fuel to the US East Coast, suffered a cyberattack that led to a temporary shutdown of its operations. The attack was attributed to the DarkSide ransomware group, causing disruptions to fuel supplies and highlighting the vulnerability of critical energy infrastructure. In February 2021, a cyberattack[4] targeted the water treatment plant in Oldsmar, FL. An unauthorized individual gained access to the plant's control system and attempted to increase the levels of sodium hydroxide (lye) in the water supply to dangerous levels. The attack was promptly detected and mitigated, preventing any harm to the public.

Reinforcement learning (RL) has been exploited to generate effective defense strategies against cyberattacks on cyber physical systems (see Sec. II for a comprehensive literature review). Despite demonstrated successes, some significant challenges remain. In RL, an agent explores the action space according to certain reward criterion and gradually approaches an optimal solution to deliver the

best course of actions to protect the system. For a conventional power grid, not only will the number of elements in the action space grow exponentially and quickly become unmanageable (e.g., the total number of actions for the IEEE 14-Bus system is 1.46×10^{15}) but also the possible actions are diverse, as shown in Fig. 1. Reducing the size of the action space in RL can be approached through various techniques. Curriculum learning[5] involves gradually increasing the complexity of the learning problem, starting from a simplified version of the environment and gradually introducing additional actions. This allows the agent to learn in a structured manner, preventing being overwhelmed from a large action space. Action pruning[6] focuses on identifying and eliminating irrelevant or suboptimal actions. By removing such actions, the action space is reduced, leading to more efficient learning and decision-making. Action space embedding[7] techniques map the high-dimensional action space to a lower-dimensional space while preserving its essential structure. This embedding is learned using methods such as autoencoders or dimensionality reduction techniques, enabling the RL algorithms to operate in a reduced-dimensional action space, improving efficiency and scalability. These approaches contribute to addressing the challenge of dealing with large action spaces in RL, allowing agents to learn and make decisions effectively. Consider a power-grid network where the human control operator observes the network's dynamics, power flow, voltage magnitude vector, and other states and chooses the best action among the available ones to maintain the functions of the network even in the event of attack. The types of actions can be discrete or continuous. Examples of discrete actions are topology actions that change the topology of certain substations and transmission line switching known as status actions. Continuous actions include redispatch actions that change the operating schedule of power plants, curtailment actions that limit the production of renewable generators, and set-storage actions that change the role of some storage units from loads to generators or vice versa. As a result, the current RL methods are applicable only to systems with a limited set of actions.

In this paper, we articulate a heterogeneous reinforcement learning framework to address the aforementioned two challenges. First, to make RL applicable to power grids with a vast action space, we divide the available actions into a number of subgroups and deploy an equal number of RL agents, each taking the actions from a single subgroup. Since the nature of the actions in different subgroups can be quite distinct, the RL agents are heterogeneous. Given a power grid, when an attack occurs, it is necessary to determine the best type of RL actions to maximize the integrity of the grid. The question is how to select the optimal RL actions based on the available measurements of the current flows on the grid. Our idea is to develop a set of specialized machine-learning-based time-series predictors, each tailored to a specific type of RL actions. For example, if five types of RL actions are possible, we train five special types of machine-learning predictors, as shown in Fig. 1. Since the time-series data of the current flows on the grid are spatiotemporal—spatially they are graph-structured; we choose temporal graph convolutional networks (TGCNs)[8] as the time-series predictors to deal with the spatial dependencies and the temporal features of the measurements. To generate the data for training the specialized TGCNs, we take advantage of the Grid2Op platform, which can accommodate and simulate realistic power grids to generate the current-flow time series from all transmission lines under arbitrary attacks and RL actions. When the adequately trained TGCNs are deployed to the real world, each TGCN is able to predict the time series into a future time window, under the specific RL actions, based on the available current-flow data at the present time. The current-flow patterns predicted by all the specialized TGCNs can then be compared, resulting in the "best" pattern preserving the healthy functioning of the grid and, accordingly, the specific RL actions to take to protect the grid.

The main contribution of our work lies in the innovative integration of heterogeneous RL agents and TGCNs to address the challenges associated with large action spaces in power grid defense. By introducing a diverse set of RL agents, each exploring a distinct subset of the action space, we enhance the scalability and efficiency of our general machine-learning framework, enabling real-time decision-making in the face of cyberattacks. Moreover, TGCNs allow us to effectively model the spatiotemporal dynamics from the power grid data, facilitating accurate prediction of system behavior and informing optimal defense strategies. This interdisciplinary framework leverages the strengths of both RL and machine learning techniques to develop a comprehensive solution for enhancing the grid resilience. Our heterogeneous RL/TGCN framework introduces a strategic RL defensive policy for power grids to significantly enhance their cybersecurity.

II. REVIEW OF THE LITERATURE ON MACHINE LEARNING AS APPLIED TO POWER GRIDS

A. Reinforcement learning

Reinforcement learning enables a machine to learn via interactions with its surroundings or the environment to generate the optimal course of actions. Previous studies demonstrated that RL is especially suited for solving sophisticated cyberdefense problems. Recent studies employing RL in smart grid cyber security systems include attacks on false data injection systems, topological attacks, attack mitigation, attack detection, and persistent attacks.[9–13] Moreover, Q-learning, an important class of RL that aims to optimize discounted reward and make future rewards less prioritized than near-term rewards, was used to investigate the susceptibility of smart grids to sequential topological attacks, where the attacker can utilize the algorithm to exacerbate the effects of such attacks on system failures with the least amount of effort.[14] For small systems, Q-learning can be done using the traditional Q-function. However, for larger systems, the Q-function approach becomes inefficient because of the large number of state-action pairs. Neural networks can be used to approximate the Q-function when the state-action space is large, as in a power grid. To counter the shortcomings of conventional RL, deep RL[15,16] was often employed for solving the power-grid security problem.[17–19] For instance, deep RL has been used in power grids for topological assaults[18] and cyberattack mitigation.[18] There is also a growing interest in RL-based control of power grids in recent years. While the efficiency of the conventional methods and load-flow software are dependent upon the accuracy of the nonlinear model that describes the dynamics of the system, RL is model-free. Indeed, being model-free is a key advantage of RL in that its performance does not depend on knowing the explicit dynamics governing the system. RL methods can learn the internal dynamical interactions and the physics of the systems without any domain knowledge,

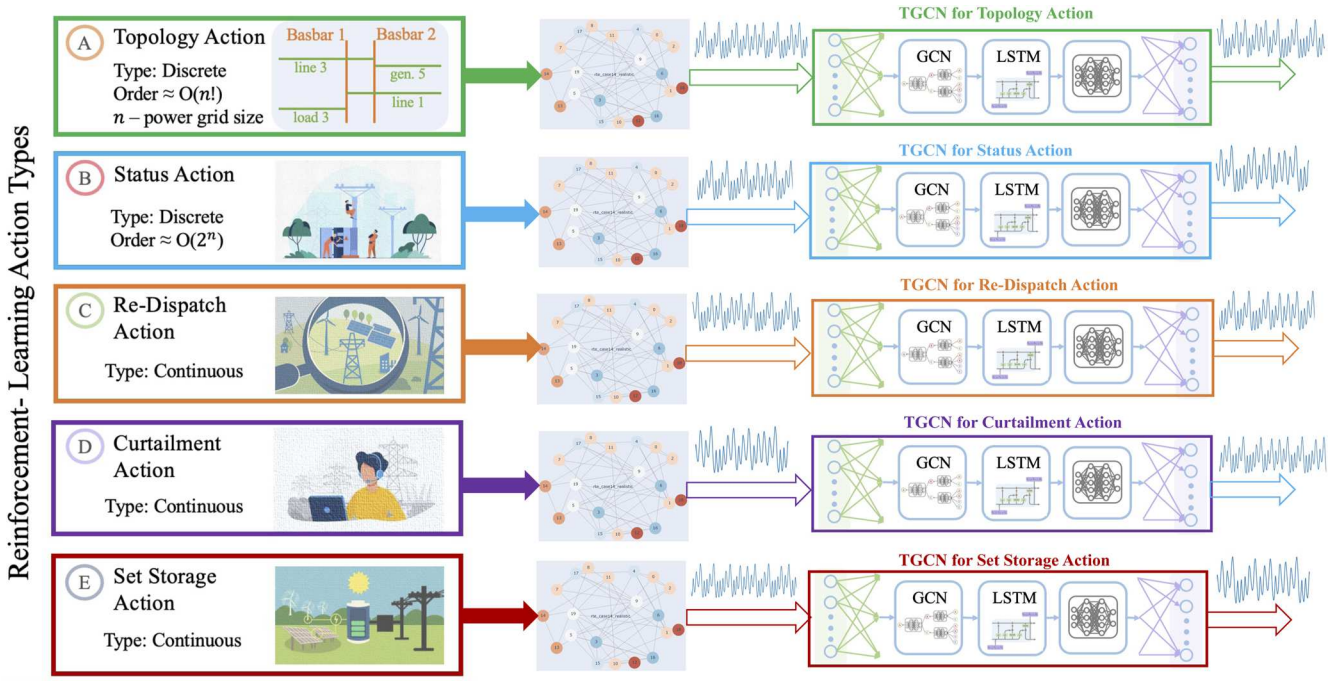

FIG. 1. Proposed TGCN framework for defending power grids against attacks. The goal is to determine the "best" type of RL control actions to mitigate the attack. A power grid entails different types of RL actions, rendering the action space prohibitively high for a single RL agent and necessitating the use of heterogeneous RL agents. The sizes of action spaces of several common benchmark power-grid systems are presented in Table I. Five common types of actions for power grids are topology, status, redispatch, curtailment, and set-storage actions, as illustrated in the left column, which require five different types of RL agents. For a given type of action, the corresponding RL agent is activated to take the actions to ensure that the grid survives under attacks. Computationally, this is done using the Grid2Op platform, where a given benchmark power grid (schematically shown in the middle column) can be simulated subject to attacks and RL actions. The outcomes of the simulations are the time series of the currents in all the transmission lines of the grid, providing the required data for training the TGCN tailored for the particular type of RL action, as illustrated in the right column. During the training, the RL-action-specific TGCN takes in the time series from the power-grid simulation and generates predictions of the current flows in a future time window. The specialized TGCNs so trained can be deployed to advise the best control actions: once an attack occurs, from the actual time series collected from the grid, each TGCN will predict the imminent trends of the current flows. The optimal current pattern (e.g., all currents are well below a threshold) can be chosen to yield the best RL actions.

although learning RL policies model-free for safety-critical applications may be debatable.[20,21] In power systems, RL can be utilized not only to control the power system but also to predict power flows.[22,23]

In applying RL to power systems, an important development is the establishment of the L2RPN (learning to run a power network) challenge by the French national grid operator Reseau de-Transport Electricite (RTE), where a framework named Grid2Op,[24] built on top of open-source libraries, such as pypownet, was introduced. Grid2Op is a Pythonic, easy-to-use modular framework, which can be used to develop, train, and evaluate the performances of an RL agent that acts on a power grid in different ways. A baseline RL method to control power flows on the grid by taking topological switching actions was proposed,[25] where bus bar splitting was the only type of action allowed. A feed-forward neural network was used to model the policy. By using a cross-entropy method (a Monte Carlo technique), the best episode is selected for training. Deep RL was employed to the power grid by changing the architecture or topology of the network,[26] where the agent, as modeled by dueling deep Q network, has both state-value and action-advantage functions. To improve the performance, historical data were added to the architecture as a memory so the control strategy consists of two offline training and an online operating system. The offline system provides the trained agent for the online system, ensuring real-time control of the grid topology. The robustness of the performance of the RL algorithm in the presence of noise or perturbation and their vulnerability under the cyberattack were investigated,[27] where the proposed vulnerability assessment method enables the agent to first identify and reduce the security risks and then practically apply deep RL control models.

Quite recently, a combinational model taking advantage of both deep learning power flow estimation and receding horizon control methods for power grid control was proposed,[28] where a Monte Carlo tree search was constructed using the predictions of the relative line loading by graph neural networks (GNNs). The constructed tree helps the agent to select the optimal action that maximizes both the probabilities of secure operation and of carrying actions to alleviate any potential overloading in the future. Another approach to overcoming the curse of dimensionality is an actor–critic-based agent that uses a curriculum-based approach with reward tuning for training.[29] To reduce the sampling bias, a parallel training procedure was used to stabilize the power grid and make it robust against the natural variability in the grid operations. While it is infeasible to

directly compare our approach to previous methods, we recognize the value in discussing the unique aspects of our work that differentiate it from existing literature. Our focus is on emphasizing the novelty of the TGCN framework and how it specifically addresses the challenges of large action spaces in RL for power grid defense. While there are quality studies done in the field of cybersecurity in power systems,[30–33] and they provide valuable insights into the strategies for improving security in smart grid energy systems, our work offers a unique solution tailored to the spatiotemporal nature of the power grid data and the complexity of decision-making in real-time defense scenarios.

B. Temporal graph convolutional networks

Temporal graph convolutional networks (TGCNs) have been used to capture the topological structure of a network in order to model the spatial dependencies and the temporal features of the grid. In particular, a TGCN is the combination of a graph convolutional network (GCN) and a gated recurrent-unit network[8] to provide a neural network-based forecasting method. TGCN was first used to simulate the urban road network, where the nodes in the graph represent the roads, the edges indicate the connections between the roads, and the nodes' attributes are the traffic data on the roads.[8] In this set up, the GCN was utilized to capture the spatial topological structure of the graph in order to determine the spatial reliance, and the gated recurrent-unit model was for capturing the dynamical change in the node attribute to determine the temporal dependence. It was demonstrated that the TGCN model performs better than a number of traditional models in the spatiotemporal analysis for various prediction horizons when tested on two real-world traffic datasets, such as the historic-average model,[34] the auto-regressive integrated moving average model,[35] the support vector regression model,[36] the GCN model alone,[37] and the gated recurrent-unit model alone.[38] Since a power grid is, in fact, a network or a graph, TGCN can be beneficial in solving various problems. As such, in power grid research, TGCN is gaining increasing attention for problems such as state estimation,[39–41] load forecasting,[42] stability analysis,[43] power optimization,[44] and anomaly detection based on the time series.[45]

III. METHODS AND JUSTIFICATION

Our main method is a combination of parallel but heterogeneous RL agents and TGCN networks. Here, we first describe conventional RL and its shortcomings when applied to power grid cybersecurity, justifying our articulation of the parallel RL scheme. We then introduce TGCNs for spatiotemporal analysis and prediction of time series, e.g., the current flows along different transmission lines in the power grid. Finally, we explain how parallel RL and TGCN networks can be combined to yield a general and powerful framework for defending power grids against cyberattacks. To comprehensively demonstrate the effectiveness of the proposed

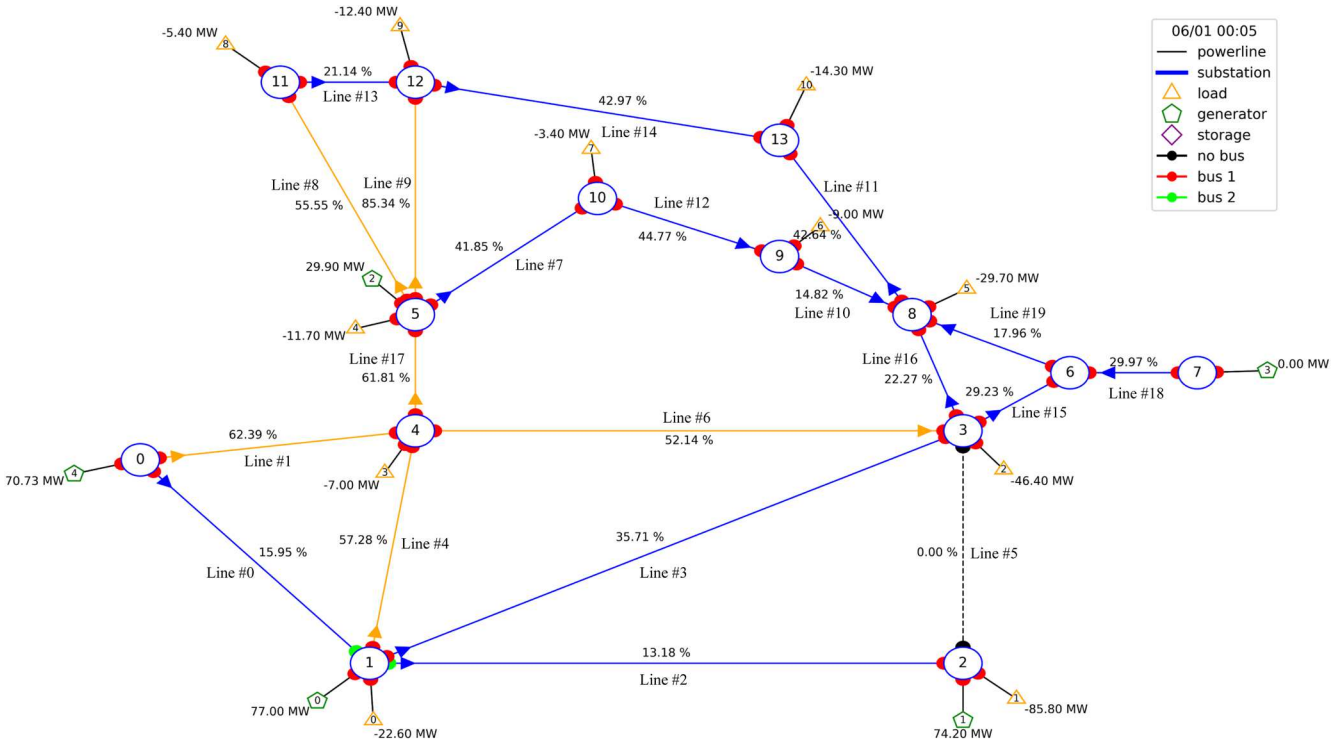

FIG. 2. Benchmark RTE 14-Bus system (rte_case14_realistic). The system has 14 substations, 20 transmission lines, 5 generators, and 11 loads. The percentage numbers shown with the transmission lines are the values of parameter ρ—the current flow in the line divided by the thermal capacity.

TGCN framework, we employ several performance metrics tailored to assessing RL training times of different action spaces and obtain the episodic rewards (mean alive and mean rewards) as well as the causality relations of power lines in the IEEE 14-Bus and 118-Bus systems. This analysis provides insights into the scalability of our framework for real-time applications in large power grids. Furthermore, by comparing the predicted outcomes of the TGCN framework to the ground truth data obtained from the simulations, we quantify its ability to accurately predict the spatiotemporal behavior of the power grid under adverse conditions.

A. Reinforcement learning for power grids

Learning through experience and interaction or experimental learning is one of the human instincts to improve knowledge about the environment and ourselves. Interaction-based learning formed a core idea behind almost all theories of learning and intelligence.[46] RL is a computational manifestation of the interaction-based learning theory in machine learning.[47] Typically, an RL algorithm is characterized by six sub-elements: agent, state, environment, policy, action, and reward.[48] In particular, for a power grid, an *agent* is the person sitting in the power grid control room, who interacts with the environment—the power grid, by taking an action and observing the state evolution. More specifically, the agent observes the network dynamics, date, time, active and reactive power, active and reactive load, power flow, or voltage magnitude vector that define a state. The objective of RL is to learn to identify the optimal action in each state to obtain the maximum reward in the long run, where *state* is a vector of continuous or discrete valued variables, such as the current flow or the status of the transmission lines in a power grid, which reflects the dynamical features of the environment. *Environment* stands for a physical world in which the agent can operate and interact with and *policy* defines the rules for agents to act, which generally is a mapping from the state space to the action space. It is the derived policy that guides the control operator to react under different circumstances in order to protect the grid. An *action* or a control action is an operation the agent can perform, where each action is associated with a *reward* and the maximum total reward over the long run defines the goal in RL.

In mathematical terms, the goal of an RL algorithm is to learn a policy $\pi(s(t), a(t))$ that provides action $a(t)$ operating at $s(t)$, which maximizes the long-run expected reward $r(t)$. In the language

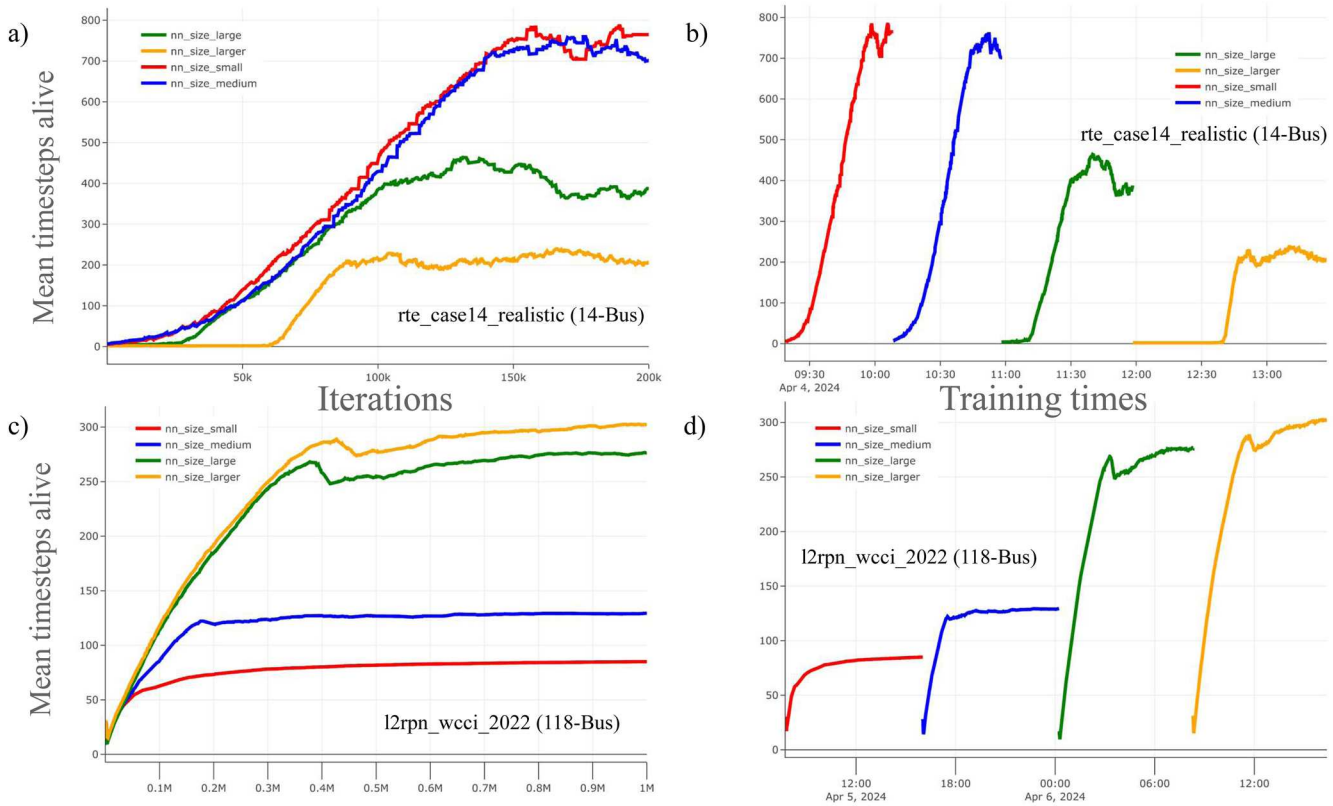

FIG. 3. Comparison of mean time steps alive for different neural network sizes in the *rte_case_14_realistic* and *l2rpn_wcci_2022 (118-bus)* power grid systems. For panels (a) and (c), the x axis denotes the training iterations, while for panels (b) and (d), the x axis shows the training length timings. The y axis indicates the mean time steps alive. Each trajectory corresponds to a specific neural network size, with varying performances observed across different network sizes and grid systems.

of a Markov decision process, the cumulative future return can be written as

$$R_t = \sum_{t=0}^{\infty} \gamma^t r(s(t), a(t)), \tag{1}$$

where $0 < \gamma < 1$ is a discount factor. A well-known RL approach is the Q-learning algorithm that provides the optimal action-value function. The Bellman equation[48] stipulates that the action-value function for policy π is the value of taking action $a(t)$ in the state $s(t)$ under a policy π, formulated as

$$q_\pi(s(t), a(t)) = \mathrm{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | s(t), a(t)\right], \tag{2}$$

where $\mathrm{E}_\pi[*]$ is the expected value function. With the Bellman equation, the Q-learning algorithm can be updated online to control the Q-value $q_\pi(s(t), a(t))$ toward the Q-target $q^*_\pi(s(t), a(t))$,

$$q_\pi(s(t), a(t)) \leftarrow q_\pi(s(t), a(t)) + \alpha\left[r(t) + \gamma \max_{a(t+1)} q_\pi(s(t+1), a(t+1)) - q_\pi(s(t), a(t))\right]. \tag{3}$$

Our development of the TGCN-based parallel heterogeneous RL framework is motivated by the following considerations. Generally, Q table is suitable for discrete states, while deep Q networks (with neural networks as Q-function approximators) are for continuous states (or actions). Q learning becomes slow when the state-action space is large. In a power grid, the agent observes the network's dynamics, power flow, or voltage magnitude vector and other states of the network in order to choose the best action among available actions so that the network can survive as long as possible under both normal and critical conditions.

The performance of RL algorithms for power grid defense is contingent upon various factors, such as the size and complexity of the grid as well as the architecture of the neural network approximators. To elucidate the impact of neural network size on the RL performance, we conduct experiments on two distinct power grid systems: "rte_case_14_realistic (14-bus)" (illustrated in Fig. 2) and "l2rpn_wcci_2022 (118-bus)". Figure 3 compares the mean time steps alive over the past 100 episodes for different neural network sizes for each system. In particular, for the "rte_case_14_realistic" system, which represents a smaller-scale grid, we employ a neural network structure defined by the formula, size_multiplier = 4 × $(i + 1)$, where i denotes the network size index (0 for small, 1 for medium, 2 for large, and 3 for larger). The neural network structure is configured as [size_multiplier × 2, size_multiplier × 1, size_multiplier × 2]. Surprisingly, the performance deteriorated with larger neural networks, indicating that smaller networks are sufficient for effective learning in this system within 200 000 iterations. Conversely, for the "l2rpn_wcci_2022 (118-bus)" system that represents a larger and more complex grid, the performance consistently improves with larger neural networks. Similar to the previous system, we utilize the same neural network structure formula that yields varying network sizes. However, unlike the smaller system, the larger neural networks exhibit an enhanced performance, suggesting that more complex grids require larger networks to achieve optimal rewards, even with longer training duration. While smaller grids may benefit from simpler neural network architectures to avoid overfitting and undertraining, larger and more intricate grids necessitate more complex network structures to capture the underlying dynamics adequately. The results of these numerical experiments highlight the need for continued research into RL methods with more efficient exploration and exploitation strategies to address the challenges posed by large-scale power grid defense scenarios.

The types of actions that the agent can perform are shown in Fig. 1. In more detail, a topology action can be performed to change how different lines coming from loads, generators, or transmission lines are connected to different bus bars in a substation. The agent

TABLE I. Comparison of actions-space size in different benchmark power grids.

Systems	RTE 5-Bus	RTE 14-Bus
Total topology actions (Dis.)	31 320	1 397 519 564
Legal topology actions (Dis.)	117	179
Total status actions (Dis.)	256	1 048 576
Legal status actions (Dis.)	8	20
Total discrete actions	8 017 920	1.46×10^{15}
Total legal discrete actions	936	3580
TGCN aided RL discrete actions	125	199

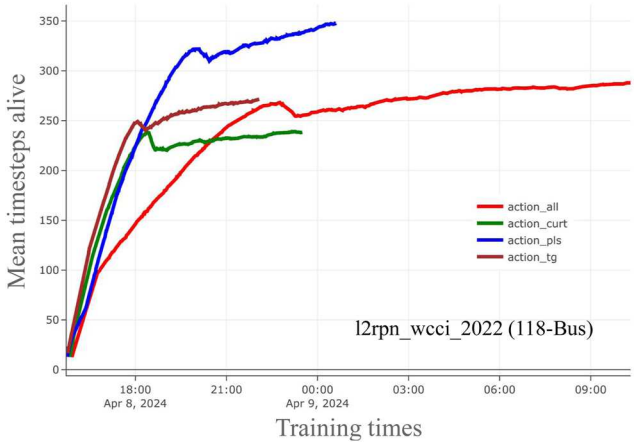

FIG. 4. Comparative analysis of the training efficiency and performance of RL agents deployed on the "l2rpn_wcci_2022 (118-bus)" power grid under attack scenarios using deep Q-learning. Four distinct RL training scenarios are evaluated, each focusing on specific subsets of the action space: (1) comprehensive action exploration, (2) curtailment actions, (3) power line switch actions, and (4) topology actions. The results indicate notable disparities in both training times and mean time steps alive across the various RL agents. Particularly, while the power line switch agent demonstrates extended training duration, it achieves the highest mean time steps alive, signifying its efficacy in maintaining grid stability amid attacks. Conversely, the topology agent exhibits shorter training times yet a comparable performance to the comprehensive action exploration agent, underscoring its efficiency in navigating the action space. On the other hand, the all-action space agent required ~18 h for training while achieving the second worst rewards, highlighting the exponential increase in training time associated with exploring the entire action space. The findings underscore the effectiveness of specialized agents in optimizing training efficiency and performance in RL-based power grid defense systems.

can also impose a status action to reconnect or disconnect the transmission lines. A redispatch action causes the generators in the grid to change their production set point. When the grid contains renewable sources, it is often necessary to limit their production to maintain the grid stability, which can be achieved by a curtailment action. For instance, the windmills should reduce their output if there is excessive wind in a certain area. Some grids also contain storage units of finite capacity behaving as loads or generators, whose role is to generate or absorb power. When the storage units reach their maximum capacity so that they can no longer take power from the grid, they can still function as generators, leading to a set-storage action. Even for a relatively small power grid, all these actions constitute an immensely large action space that makes it hard for the RL algorithm to explore efficiently. Even when not all actions are applicable or "legal" due to the specific structure of the power grid, the action space can still be large, as presented in Table I, where the numbers of topology and status actions are compared for two benchmark power grid systems.

Figure 4 presents a comprehensive analysis of the training efficiency of the RL agents in power grid defense against cyberattacks, which is a comparative overview of the mean time steps "alive" and training times for RL agents trained on the power grid "l2rpn_wcci_2022 (118-bus)" under attack conditions using deep Q learning. The training duration was set to one million iterations, although substantially longer training periods (e.g., on the order of 100×10^6 iterations) are typically required for comprehensive training. This initial analysis serves to highlight the potential benefits of our proposed approach in terms of efficiency and effectiveness. In particular, we compare four different RL training scenarios, each exploring distinct subsets of the action space. One RL agent explores the entire action space, another focuses solely on curtailment actions, while the remaining two agents specialize in power line switch and topology actions, respectively. The results reveal notable differences in both training times and mean time steps alive among the different RL agents. The power line switch agent exhibits the second longest training time, ~8 h, yet achieving the highest mean time steps alive, averaging around 350 time steps. Conversely, the topology agent, with the fastest training time of ~6 h, achieved comparable mean time steps alive to the all-action space agent, indicating its efficiency in navigating the action space. The curtailment agent, with a training time of around 7 h, demonstrates the lowest mean time steps alive among the specialized agents but remains competitive with the all-action space agent.

Of particular significance is the stark contrast in training times between the specialized agents and the all-action space agent. While the specialized agents can achieve a comparable or superior performance in terms of the mean time steps alive, they do so within significantly shorter training times. In contrast, the all-action space agent requires ~18 h for training while achieving the second worst rewards, highlighting the exponential increase in training time associated with exploring the entire action space. The training time is going to increase even more in comparison to specialized agents in longer training. This observation underscores the inherent scalability challenges posed by the expansive action space associated with power grid defense. The findings from this analysis underscore the importance of dividing the action space and utilizing specialized agents to optimize training efficiency in RL-based power grid defense systems. By focusing on specific subsets of actions, specialized agents can navigate the action space more efficiently, thereby achieving a comparable or superior performance with reduced

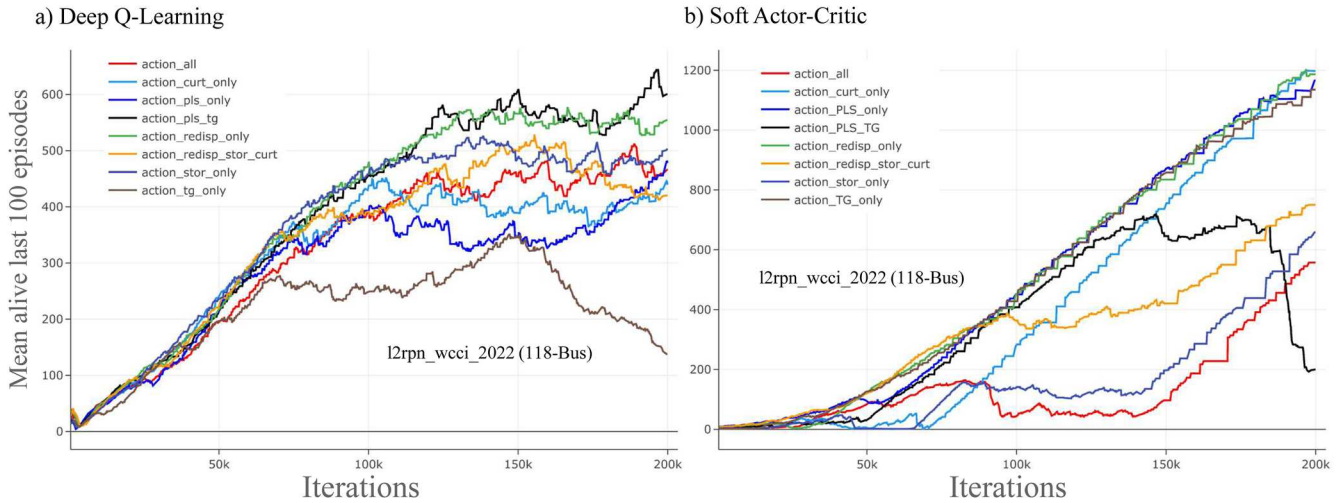

FIG. 5. Illustration of the mean time steps alive over the last 100 episodes for the "l2rpn_wcci_2022 (118-bus)" power grid. The performance of various RL agents trained using (a) deep Q learning and (b) soft actor–critic algorithms area compared. Each agent explores different subsets of the action space, ranging from comprehensive exploration of all actions to specialized focus on individual action types, such as curtailment, power line switch, topology, redispatch, and storage actions. The training duration spans 200 thousand iterations, providing insights into the efficiency and effectiveness of dividing the action space and employing specialized agents in power grid defense. The comparison highlights the scalability challenges associated with comprehensive action exploration and underscores the nuanced performance differences between specialized agents across different RL algorithms.

training times. In addition, the use of specialized agents mitigates the scalability challenges associated with exploring the entire action space, enabling faster training and higher rewards. These insights reinforce the rationale behind our proposed approach and provide empirical evidence of its efficacy in enhancing the efficiency and effectiveness of RL-based power grid defense mechanisms.

We carry out computations to justify using specialized agents. Figure 5 shows a comprehensive comparison of the mean time steps alive over the past 100 episodes for the "l2rpn_wcci_2022 (118-bus)" power grid under attack conditions, utilizing two distinct reinforcement learning (RL) algorithms: deep Q learning and soft actor–critic. Notably, soft actor–critic is distinguished from deep Q learning by its incorporation of an entropy regularization term, facilitating a more exploratory behavior and potentially improving sample efficiency. Surprisingly, despite this added exploration, the all-action-space exploring agent performs even worse relative to other agents in soft actor–critic compared to deep Q learning. The training duration spans 200 thousand iterations, serving as an illustrative example rather than a comprehensive training regimen. This comparison underscores the efficiency and effectiveness of dividing the action space and employing specialized agents. Eight different RL training scenarios are evaluated, each targeting specific subsets of the action space: comprehensive action exploration; curtailment actions; power line switch actions; topology actions; redispatch actions; storage actions; combinations of power line switch and topology actions; and combinations of redispatch, curtailment, and storage actions. Our aim is to obtain an understanding of how different RL agents navigate the complex action space inherent to power grid defense.

The results reveal that the all-action-space agent's performance deteriorates significantly with increasing iterations, underscoring the scalability challenges associated with exploring the entire action space. Conversely, specialized agents, such as those focusing solely on curtailment or power line switch actions, demonstrate a more consistent and efficient performance across both RL algorithms. Interestingly, in soft actor–critic, agents exploring only a single action type, such as curtailment or power line switch, exhibit better performance compared to their counterparts in deep Q learning. This suggests that the inclusion of larger neural networks in soft actor–critic may increase the performance gap between all-action-space and specialized agents. These results emphasize the importance of dividing the action space and utilizing specialized agents in mitigating the scalability challenges inherent to RL-based power grid defense systems. Moreover, they highlight the potential implications of the RL algorithm choice on the performance of specialized agents, offering valuable insights into future research and practical implementation.

B. Temporal graph convolutional neural networks (TGCNs)

TGCNs[8] are a type of machine learning model that extend the traditional graph convolutional networks (GCNs) to handle data on temporal graphs with features evolving over time. TGCNs can be used to analyze the sequences of interactions in social networks to understand the spread of diseases on networks or to model the evolution of physical systems. A key challenge in designing TGCN models is defining how to incorporate the temporal information into the graph convolution operation, which has been addressed through, e.g., using a recursive formulation or introducing a temporal convolutional layer.[39,40,44,45]

Our aim is to analyze the network characteristics of various power grid components, such as the current flows of the transmission lines, their thermal capacity, the load, and generator conditions under the control of multiple agents with varying policies in order to accurately predict how well they will perform in the event of an attack. Due to the intricate spatiotemporal relationships, a number of difficulties can arise. Because of the spatial dependence, the topological structure of the power grid network is a dominant factor determining the change in the current flows on the transmission lines. Analogs with the traffic flow problem,[49] the current flow of the main transmission lines impact the flow on other lines through the transfer effect, and the traffic status at side transmission lines impact the main lines' current flows through feedback. The temporal dependence means the current flows change dynamically over time and are mainly reflected in the periodicity and trend in a power grid. (For example, the load power can change periodically over a week or a day.) In addition, the current flow can be affected by the conditions of the power grid in the previous hours or even longer. Some recent machine-learning based time-series prediction methods[50–53] consider the temporal dependence but tend to ignore the spatial dependence, so the predicted changes in current flows may not be accurate. Our solution is to employ TGCN that was originally designed for traffic forecasting tasks based on urban road networks.

C. TGCN-aided reinforcement learning

Our method consists of two steps. The preparation phase involves training individual RL agents to learn the control of the power grid and also using TGCN to learn the behavior of the trained agents under attack. In the execution phase, we use the trained TGCN to predict the behavior of the trained agents under the attack, and the best individual agent is selected based on a set of predefined criteria. The standard TGCN model consists of a graph convolutional network and a gated recurrent unit (GRU). However, we find that the GRU model does not perform well on power grids, so we replace the GRU by an LSTM network because of its performance-wise superiority. Both LSTM and the GRU are recurrent neural networks and share a similar set of fundamental principles: They use a gated mechanism to memorize as much long-term information as possible and are equally effective for various tasks. In terms of computations, LSTM is more complex and requires a longer training time, while GRU is simpler and faster.

In applications of graph neural networks, it is a common practice to learn the features of the nodes rather than of the edges. Take the various social media learning engines as an example, where users are the nodes and their interactions constitute the edges. Features such as the profile pictures or the user age, are associated with the nodes. The opposite situation arises for the power grids, where the interested features are associated with the transmission lines representing the edges of the original network, thereby requiring converting the original network into a line graph. More generally, a line graph of a graph G is obtained by associating a node with each edge of the graph and connecting two nodes with an edge if and only if the corresponding edges of G have a node in common.[54] For example, for the benchmark RTE 14-Bus system, the original grid network

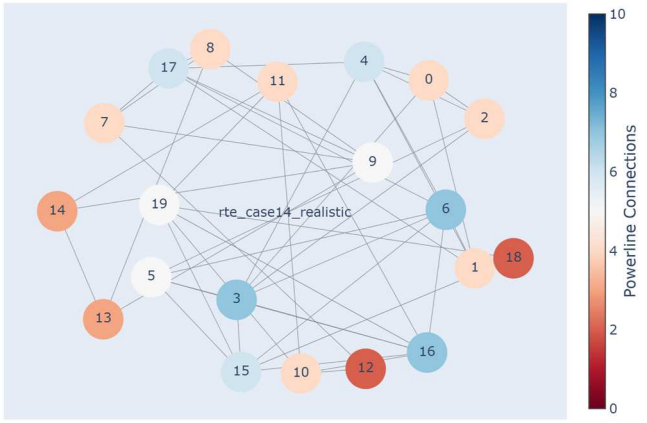

FIG. 6. Corresponding line graph of the benchmark power grid shown in Fig. 2. Given a power-grid network, its "line graph" is obtained by converting the nodes (edges) in the original network into edges (nodes), where the nodes now represent the transmission lines and the edges denote that if two power lines have a mutual connection to each other through a node. The features on the nodes are the various current flows in the original network. The line graph is provided as an input to our method shown in Fig. 1.

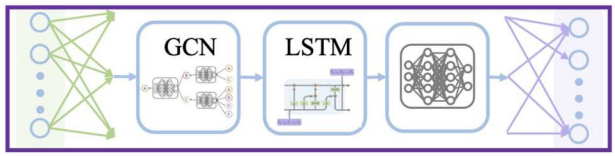

FIG. 7. Proposed TGCN framework. The structure of TGCN, where the input is the current time series from all the nodes in the line graph and the output is the predicted time series from the same set of nodes. The time series are generated from the power grid under distinct RL actions using the Grid2Op platform. Specifically, in the platform simulations, the grid is assumed to be attacked multiple times. Each time an attack occurs, an RL action of a specific type (e.g., topology action, status action, redispatch action, curtailment action, or set storage action) is taken to protect the grid, generating time-series segments before and after the attack. The time series from many attack events are combined to form the training data for TGCN with the goal of predicting the time series under this specific type of RL actions. Because of the complexity of the organization of the time series from the grid network, GCN is used to handle the graph-structured data to deal with the spatial dependence of the time series, and the LSTM is used to capture the temporal dependence. TGCN is action-specific because, for a different type of RL actions, it is necessary to train a new TGCN. The training process yields a set of distinct TGCNs, each corresponding to a specific type of RL actions. (An analogy is medical doctors of different specialties who are trained to deal with different types of diseases.).

shown in Fig. 2 is transformed into the line graph shown in Fig. 6, where the nodes correspond to the transmission lines with current flows as the key features and an edge arises if two power lines have mutual connection to each other.

The structure and training of the TGCN is shown in Fig. 7. Briefly, for training the TGCN, we generate a number of scenarios under the control of the individual RL agents, which include both normal operation and attacks. The input to the TGCN consist of time series and the line graph shown in Fig. 6, and the graph convolution network designed to handle arbitrary graph-structured data are used to capture the topological structure of the power grid, ensuring that the TGCN captures the spatial dependency. The output with the spatial features is sent to the LSTM to capture the temporal dependency of the dynamical information on the network. Additional training is achieved by using a fully connected dense layer.

For the testing phase, during the normal operation of the power grid, from time $t = 0$ to $t = T_A$, the grid is controlled by the *reconnect agent* whose task is to reconnect a line if it gets disconnected. For an attack occurring at time $t = T_A$, the time interval between $t = T_A$ to $t = T_D$ is the *decision interval* in which our TGCN framework is deployed to gather the previous data containing both normal and attack modes to predict the performance of the trained RL agents. After the required data are gathered, the TGCN predicts the performance of the agents in the future time interval from $t = T_D$ to $t = T_{SS}$, the time that the power grid settles into a steady state after the attack, and selects the best performing RL agent based on the grid survival criterion (described below). For instance, as shown in Fig. 8, the performance of the TG agent (*topology greedy agent*) is predicted to be better than that of the PLS agent (*power line switch*) so the TG agent is chosen to perform to mitigate the effects of the attack.

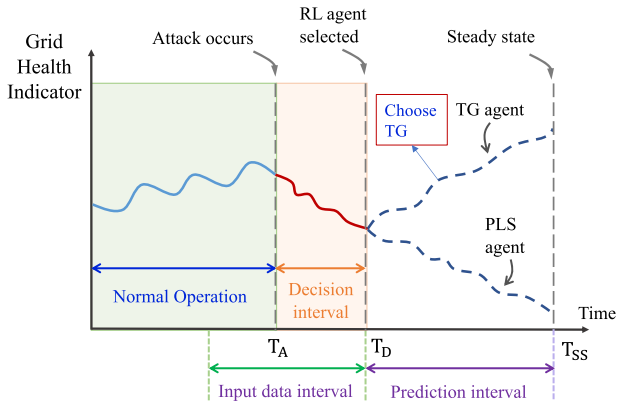

FIG. 8. Deployment of trained TGCNs. A schematic illustration of some grid survival indicator as a function of time. Before an attack occurs at T_A, the grid functions normally. For $t > T_A$, the value of the grid survival indicator decreases with time. A time interval is required for making the decision and taking the appropriate action to protect the grid—the decision interval, where T_D is the present time. This time interval cannot be long as any meaningful action will need to be taken before the grid collapses. The time series of the currents from all nodes in the grid collected during the decision interval are fed into each trained TGCN for it to predict the future time series, from which the time evolution of the grid survival indicator can be calculated. The RL actions associated with the TGCN that yields the fastest and greatest recovery of the indicator value are selected to mitigate the specific attack. (It is assumed that the time required for the TGCNs to carry out the prediction of the time series is small compared to the length of the decision interval and is practically negligible.)

To have a criterion for selecting the best RL agent, we define the following indicator of grid health:

$$I_G \equiv \frac{1}{1 + N_>}, \tag{4}$$

where $N_>$ is the total number of transmission lines with the ratio ρ of their current flows to the thermal capacity above a certain threshold r_{th}, where $0 < r_{\text{th}} < 1$ during the prediction time interval (from time $t = T_D$ to time $t = T_{SS}$). The smaller the number $N_>$, the larger is the value I_G and the "healthier" the grid is. The RL agent predicted to have the largest I_G value is selected. Note that if r_{th} is close to one, fewer control actions are triggered but the safety margin of the grid is reduced. On the contrary, if r_{th} is far less than one, then more control actions will be needed but the grid will be safer against attacks.

IV. RESULTS

A. Simulation Settings

To demonstrate the workings of our TGCN-aided RL method, we use the benchmark RTE 14-Bus system that has 20 transmission lines, 5 generators, and 11 loads and employ the Grid2Op platform (a Python environment), specifically designed for applying reinforcement learning algorithms to power grid control. In particular, Grid2Op is a user-friendly Python tool that enables developing, training, and evaluating agent or controller performance, taking into account many physical and dynamical aspects of the power grid. The package is modular and can be used to test the effectiveness of optimal control methods or to train RL agents. It is adaptable and enables users to employ the algorithm of one's choosing to compute the power flows, e.g., in all the transmission lines of the power grid. The application is compatible with the openAI gym programming interface—the state-of-the-art tool for simulating RL algorithms. In fact, arbitrary controllers can be implemented on Grid2Op even though it was originally developed in the RL community. In our simulations, the power grid dynamics are modeled using the Grid2Op platform, which provides detailed CSV logs containing information on loads, generator powers, and voltages. Even under "normal conditions" without attacks or random failures, the power grid is dynamic due to factors such as fluctuating demand and generator outputs. While the grid may be in a steady state overall, small fluctuations in load demand and generation output can occur, necessitating continuous monitoring and potential intervention by control agents. As a result, the temporal dynamics of the system are simulated to ensure the model accurately reflects real-world scenarios. Trained RL agents are designed to adapt to these dynamics and make decisions accordingly, contributing to the robustness and effectiveness of our proposed approach.

In our work, all the simulations were performed on a desktop PC with an Intel Core i7-6850K CPU and 128 GB of RAM. Table II presents the simulation parameter values. For the RTE 14-Bus system, we simulate its dynamics under attack conditions for 10 000 episodes under the control of two types of default Grid2Op greedy agents: *topology greedy* (denoted as *TG* agents) and *power line switch* (*PLS* agents), where the former performs only topological actions at the buses in the power grid and the latter takes status actions whenever necessary. Greedy agents adopt a brute force strategy by exhaustively testing all the available actions and selecting the one yielding the highest reward. In our implementation, we opt for greedy agents due to their speed advantage. However, our framework is versatile enough to accommodate any reinforcement learning trained agent in practice. Moreover, in a normal operation time interval, the *reconnect* agent (denoted as *Reco*, also a greedy agent) controls the whole power grid by reconnecting a line if it is disconnected at any time.

TABLE II. TGCN simulation parameters used for learning the behavior of individual RL agents.

Parameters	Values
Time step (ts)	5 min
Input sequence length	24 ts
Forecast horizon	12 ts
Multi-horizon	True
Learning rate	0.001
Patience	10 episodes
Training data	70%
Validation data	20%
Test data	10%

B. Justification for TGCN

A basic justification for our TGCN framework is certain degree of correlations (which assures spatial dependency) and causality (that reveals temporal dependency) among the input time series. In particular, the framework is articulated for predicting the current flows in the power grid in a future time window, using the flows in a previous time window up to the present time as the input. These time series are from a power grid so they have a graph or network structure in the sense that they must be inter-related or correlated with each other. The correlations of various pairs of time series can neither be too small nor too large, as the TGCN needs to be trained with time series from the power grid to learn its inherent network structure. Figures 9 and 10 show the correlations and causality relations among the normalized currents in various transmission lines, where for any line, the normalized current (ρ) is the actual current divided by the thermal capacity of the line. The correlation heat map shows the strength and direction of the linear relationships between different features at each time point. High correlation (positive or negative) indicates that the features tend to move together or in opposite directions over time. Anti-correlation suggests that when one feature increases, the other tends to decrease and vice versa. Granger causality analysis goes beyond the correlation by assessing whether one time series can predict another. Values greater than zero indicate that the past values of one feature contain information that helps predict the future values of another better than just using the past values of that feature alone. The presence of nontrivial correlation and Granger causality suggests that there are dynamic relationships between the current flows in the transmission lines. The features that are correlated but do not show significant Granger causality (such as lines 7–14 under the TG agent) imply a synchronous behavior but not necessarily any causal relationship. Conversely, features with significant Granger causality but low correlation (such as lines 2–6 and lines 7–11 under the TG agent) suggest directional dependencies that are not apparent from linear correlation alone. The results from this correlation and Granger causality analyses indicate that our dataset exhibits both spatial

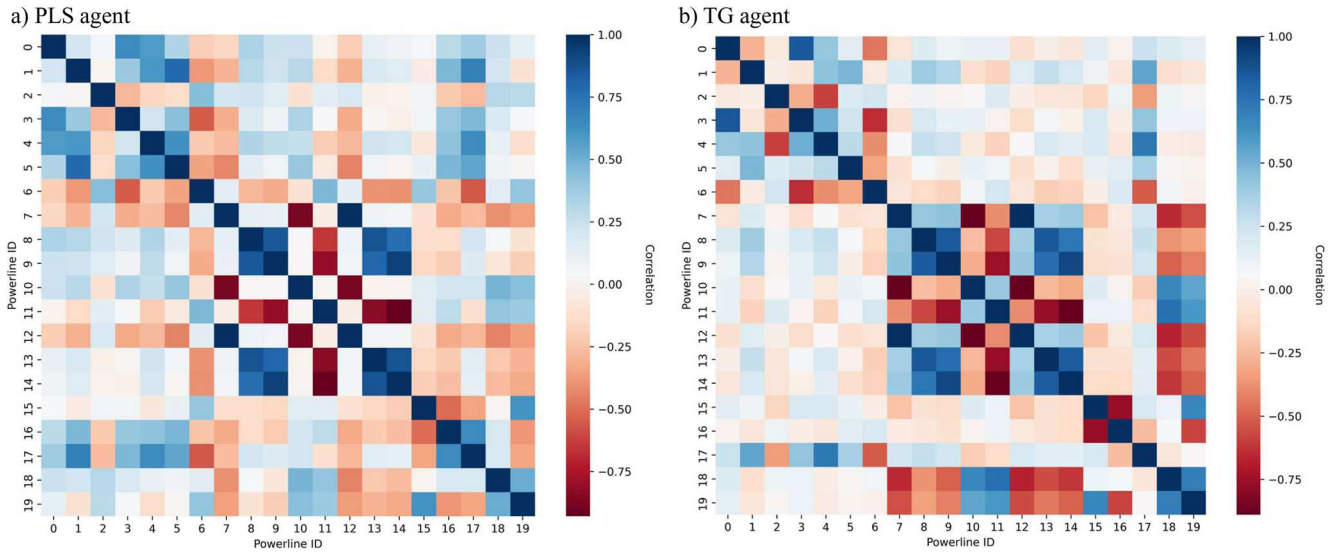

FIG. 9. Correlations among current flows in the RTE 14-Bus power grid (under attack scenario). The correlations associated with the normalized currents, all pairs of transmission lines in the power grid. The networked system is simulated under attack conditions for 10^4 episodes under the actions of (a) PLS and (b) TG agents. Correlation solely captures the linear relationships among current flows and falls short in depicting the complete dependence among transmission lines. TGCN takes into account the spatial interdependence of closely situated lines and thus emerges as a dependable approach.

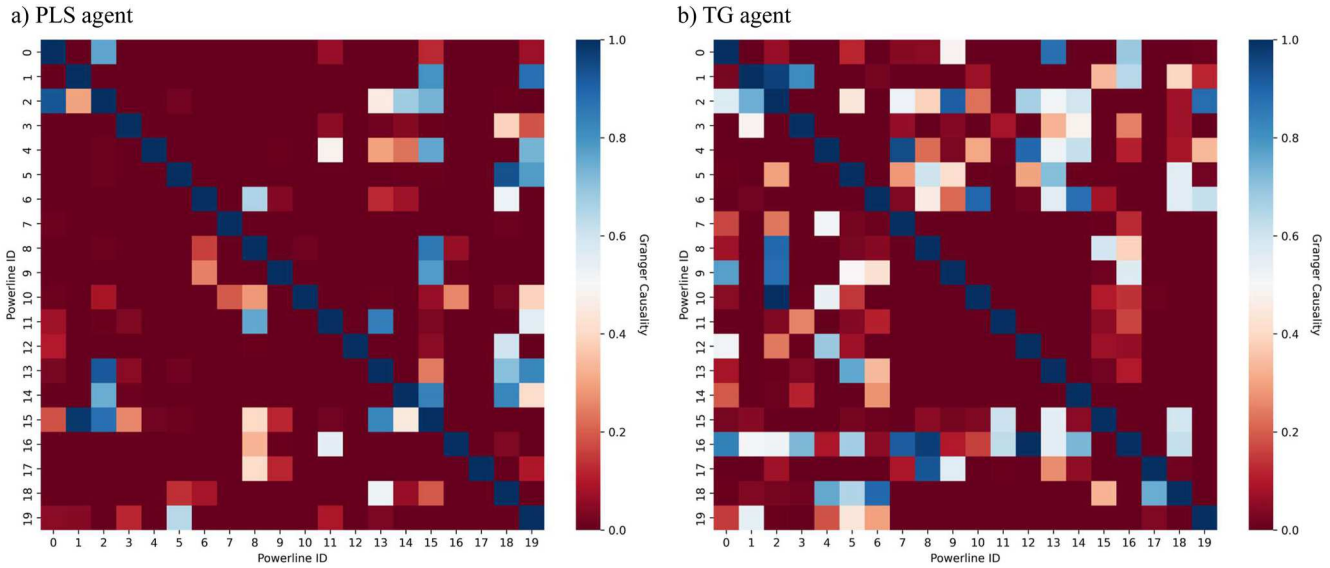

FIG. 10. Granger causality within the RTE 14-Bus power grid's current flows. It demonstrates how the past signal history shapes the future of others, unveiling the temporal interconnection of power lines' current flow, further justifying TGCN. The networked system is simulated under attack conditions for 10^4 episodes under the actions of (a) PLS and (b) TG agents.

(correlation) and temporal (Granger causality) dependencies. Traditional machine-learning models tend to struggle to capture these complex dependencies effectively. TGCN is specifically designed to handle data with both spatial and temporal dependencies, making it a suitable choice for modeling the dynamic interactions between features over time in the dataset.

C. TGCN Implementation

We now simulate an attack scenario that disconnects a transmission line. Concretely, we assume that the set of transmission lines $\{3, 4, 15, 12, 13, 14\}$ is susceptible to the attack. In our study, for simplicity, we use selective targeting of transmission lines in the RTE 14-bus power grid, which is commonly adopted in the emerging

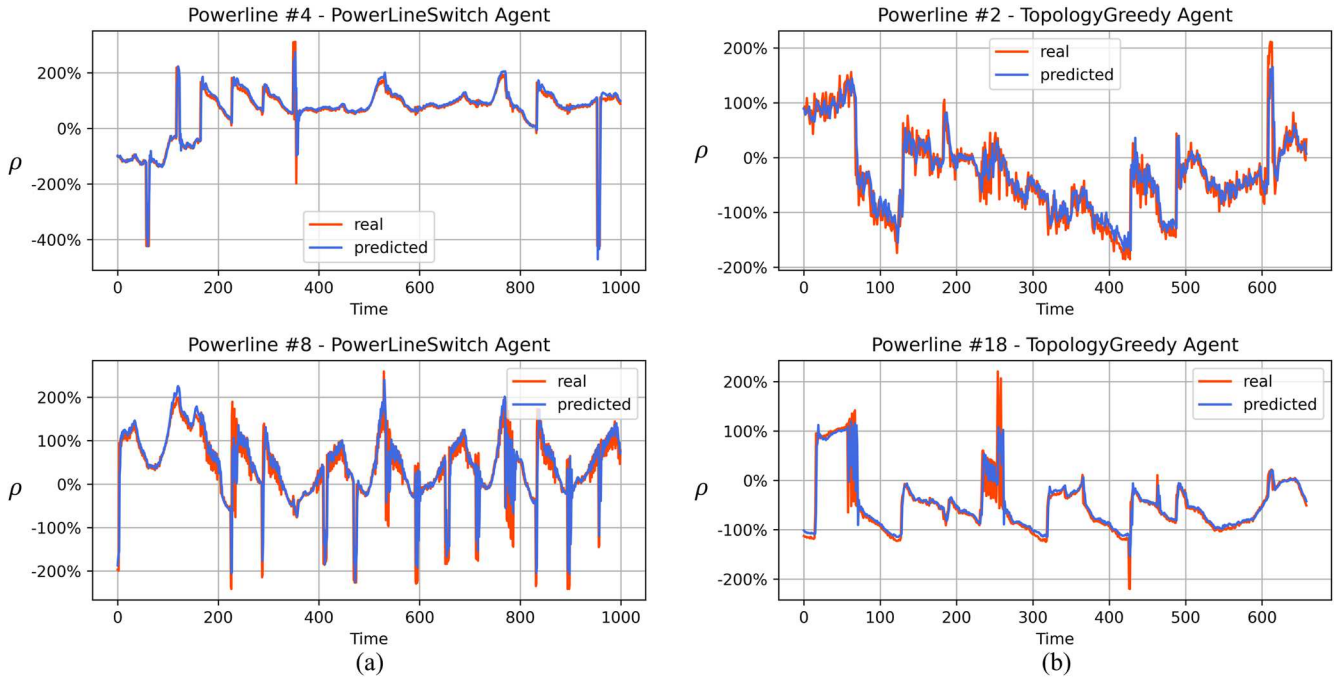

FIG. 11. Performance of TGCN for predicting the current flows in the RTE 14-Bus power grid. The normalized currents under the control actions of (a) TG agent for transmission lines (Nos. 2 and 18) and (b) PLS agent transmission lines (Nos. 4 and 8). The input time series contains a phase of normal operation, followed by an attack phase.

field of RL for power grid security. In fact, RL applications in this domain are relatively recent, where simplified scenarios are assumed to facilitate successful training. Naturally, the simplified approach may not fully capture the diverse range of vulnerabilities present in real-world power grids, but future studies will incorporate more comprehensive and realistic representations of power grid systems, leading to improved training outcomes and more robust defense strategies. When a line is attacked, we simulate the dynamics of the power grid at the 5-min step. The attack duration is 30 min (five time steps) during which the defender is unable to reconnect the attacked line. At the end of this attack interval, the line is reconnected and we continue to simulate the power grid for 3 h during which no further attack can occur. This is to keep the attacker–defender interaction as a fair game. The attacker has the initial budget of 0 units at the start of the simulation, which increases by 0.1 units/time step. Each attack costs 1 unit of resources. The attack budget assures that the attacker does not have infinite resources where it can attack indefinitely and recklessly. For the training phase, 10 000 episodes of such attacks are simulated with the maximum length of 24 h (not all episodes take 24 h).

Our heterogeneous RL agents are trained in the divided action spaces and we create a TGCN for each type of RL agents. To demonstrate the predictive power of the TGCNs, we simulate a large number of scenarios of normal operation and attacks, controlled by two types of RL agents: TG and PLS. The input time series length is 2 h (24 time steps) and the prediction horizon is 1 h (12 time steps). The input time series contains the data from the normal interval and the attack interval, as shown in Fig. 8. Figures 11(a) and 11(b) show the predicted normalized currents in the transmission lines Nos. 4 and 8 under the control actions of TG and PLS agents, respectively, where the spikes indicate the attack events. It can be seen that the specialized TGCNs are able to accurately predict the time evolution of the currents, even during the attack.

The grid health indicated, as defined in Eq. (4), makes use of the number $N_>$ of the transmission lines in which the current exceeds a threshold value r_{th}. How does this threshold affect the choice of the RL agents? To answer this question, we vary the threshold systematically from zero to one. For each value of r_{th}, we simulate 50 different attack scenarios and calculate the fractions of different RL agents, selected to best protect the power grid. Figure 12 shows the fractions of three RL agents being chosen vs r_{th}. It can be seen that for $r_{th} \leq 0.5$, the selected RL agent is almost exclusively PLS. For $r_{th} = 0.7$, the TG agent should be used. For $r_{th} > 0.8$, the best RL agent is Reco. This process is expandable to any other set of user-defined agents. The results can be intuitively understood by considering the different functionalities and costs associated with the RL agents used to protect the power grid under attacks. When the threshold parameter is set below 50%, the dominant choice for protecting the power grid is the power line switch agent. This agent is responsible for disconnecting or reconnecting power lines based on the situation. In this case, it is likely that a significant level of attack or disruption is present in the system. To mitigate the effects of the attack, the RL system chooses to disconnect specific power lines that are vulnerable or compromised. However, when the threshold parameter

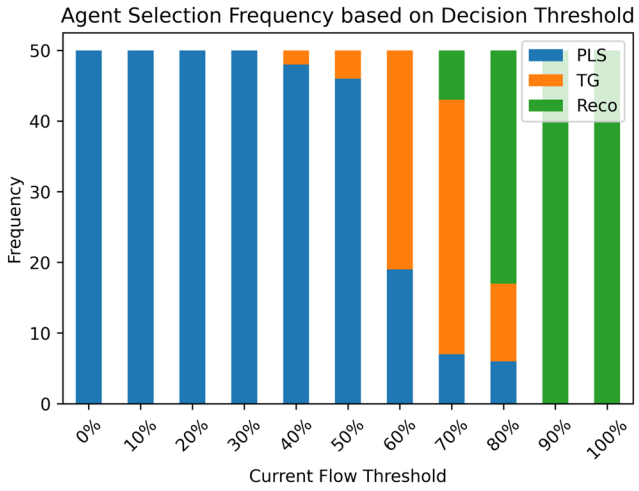

FIG. 12. Frequencies at which different RL agents are selected to best protect the power grid under attacks. The fractions of three types of RL control actions selected vs the threshold parameter r_{th}. The results show that for a threshold below 50%, the chosen agent is the power line switch, indicating significant disruption. Above 90%, the preferred agent is the reconnect agent, focusing on restoring disconnected lines. For thresholds between 50% and 90%, a mix of agents is used, including power line switch, topology greedy, and reconnect agents.

is set above 90%, the preferred agent becomes the reconnect agent switch agent. This agent's primary function is to try and reconnect a power line that has been previously disconnected. At this threshold level, it is likely that the system has already undergone significant restoration efforts and most power lines are functioning properly. Therefore, the RL system focuses on re-establishing any remaining disconnected lines to optimize the overall power grid's stability and functionality. For threshold values between 50% and 90%, the RL system employs a mixture of power line switch, topology greedy, and reconnect agents. This suggests that in situations where the power grid is moderately affected by attacks, a combination of strategies can be utilized. The power line switch agent may still be used to isolate vulnerable or compromised lines, while the topology greedy agent, although more expensive, can be employed to ensure that the overall structure of the power grid remains intact. The presence of the reconnect agent indicates that there are still some disconnected lines that need to be addressed, albeit in a lesser proportion than the higher threshold case. It is important to note that these results were obtained using the RTE 14-bus system, but similar outcomes can be expected for other benchmark systems. The choice of the RL agents depends on the specific characteristics of the power grid and the severity of attacks. Furthermore, the higher cost associated with the topology greedy agent suggests that it should only be employed when necessary, potentially due to its resource-intensive nature.

D. Real-world implementation

Real-world implementation of the proposed TGCN framework comes with several challenges and considerations that must be addressed to ensure its practical relevance and applicability. One key consideration is scalability, as real power grids often consist of thousands of transmission lines, generators, and loads, leading to significantly larger network sizes compared to the benchmark systems used in the simulations. Scalability challenges arise not only in terms of model training and inference but also in data collection and preprocessing. Addressing scalability requires efficient algorithms and distributed computing frameworks capable of handling large-scale spatiotemporal data. Another consideration is computational resource requirements, particularly in real-time applications where decisions must be made rapidly to mitigate cyberattacks and maintain grid stability. The computational demands of training and deploying TGCN models for predictive analytics on large-scale power grids can be substantial, necessitating optimization techniques and hardware accelerators to ensure timely responses. Furthermore, the energy efficiency of the computational infrastructure used for TGCN implementation is a critical factor, especially in environmentally sustainable energy systems. In addition, the adaptability of the TGCN framework to dynamic network conditions is essential for a robust performance in real-world environments. Power grids are subject to continuous changes due to factors such as demand fluctuations, equipment failures, and renewable energy integration, requiring adaptive learning algorithms capable of capturing and responding to evolving network dynamics. Incorporating mechanisms for online learning and model updating can enhance the framework's adaptability and enable it to cope with unforeseen changes and disturbances in the power grid. Addressing these challenges and considerations for real-world implementation is crucial to ensure the practical relevance and effectiveness of our TGCN framework in enhancing the cybersecurity and resilience of power grids. By leveraging advanced algorithms, computational resources, and domain knowledge, the TGCN framework holds promise as a valuable tool for industry practitioners and policymakers tasked with safeguarding critical energy infrastructure against cyber threats.

V. DISCUSSION

When it has been detected that a power grid has been attacked, protective measures must be taken to maintain the grid integrity and functions. A variety of actions can be taken, such as topology, status, redispatch, curtailment, and set-storage actions, as shown in Fig. 1. The question is which actions to take with respect to the attack that has just occurred. The issue is that a decision needs to be made in short time before any large-scale blackout occurs. For a regular power grid, both the number of possible attacks and the number of possible actions to take are exponentially large (with respect to the size of the network), making the required decision-making extremely challenging. While empirical methods based on, e.g., previous experiences might help, it is desired to develop a more rigorous framework to defend power grids against cyberattacks. Machine learning, especially reinforcement learning, provides a viable solution.

The main achievement of this work is the articulation of a reinforcement-learning centered predictive machine-learning framework capable of real-time decisions of choosing the best actions to protect a power grid from attack. Underlying the predictive framework are two ideas: heterogeneous RL agents and TGCN. In particular, the first idea is motivated by the following considerations: while employing reinforcement learning to protect power grids from cyberattacks has been extensively investigated in recent years, the challenge of dealing with an exponentially large action

space remains outstanding. For a regular power grid, using a single RL agent to explore the entire action space is computationally infeasible. Our idea is then to use an "army" of heterogeneous RL agents: each exploring a small part of the action space and altogether they cover the whole action space. The second idea of TGCN is motivated by two factors: (a) the graph or network nature of the power grid (spatial) and (b) the need to predict the dynamical evolution of the system, i.e., the time series of current flows in all the transmission lines (temporal). When an attack has been detected, for the defender of the power grid, the available information is the time series of the current flows in the network in a time interval that contains some time period before the attack and a short time period after, which constitute a spatiotemporal time series dataset. The defender's task is to choose the best course of actions based on the available spatiotemporal time series. It is precisely this spatiotemporal nature of the available information that leads us to the idea of TGCNs—a graph machine-learning architecture specifically suited to deal with spatiotemporal data. For each RL agent, we generate a unique TGCN that specializes in the specific type of RL actions. That is, we create an army of TGCNs whose number is equal to the number of heterogeneous RL agents. Each TGCN is trained according to the specific RL action type. Specifically, given an RL type, we take advantage of the Grid2Op platform to generate ample time series of the current flows in the power grid under a large variety of attack scenarios, which are used to train the TGCN. A well-trained TGCN accomplishes the prediction task by taking the available time series as the input and output an equal number of time series in a future time interval.

With a set of well-trained, specialized TGCNs, the defense strategy can be summarized. When an attack has occurred, the available spatiotemporal time series are fed into each TGCN to generate prediction of all the current flows in the network in the near future. The predicted spatiotemporal time series can be used to calculate a predefined grid health indicator. The RL actions associated with the TGCN that yields the highest value of the indicator are selected and delivered to protect the grid.

Our combined heterogeneous RL and TGCN framework provides a potential solution to defending power grids against cyberattacks. We have demonstrated the capability of the framework in accurately predicting the spatiotemporal time series and thereby selecting the corresponding RL actions using the benchmark RTE 14-Bus system. A future work may include adding preferences into the decision making. For example, realistically, topology changes are more expensive than the actions, such as reconnecting the transmission lines or changing the set points of a generator in a power grid. This introduces economic constraints into the problem so that the expensive solution is not selected unless absolutely required. While our framework demonstrates promise in simulation environments, challenges will arise in real-world implementation. Factors such as scalability, computational resource requirements, and adaptability to dynamic network conditions warrant careful consideration. Future research directions could focus on overcoming these limitations by exploring new architectures or algorithms to improve scalability and efficiency. Investigating methods to enhance adaptability to changing network conditions and integrating additional data sources for improved performance could also contribute to the framework's robustness and applicability in practical settings. Furthermore, evaluation metrics, such as convergence rates, computational efficiency, and robustness against various attack scenarios, should be investigated to provide a comprehensive assessment of the framework's efficacy. Addressing these unresolved issues and suggesting possible extensions or refinements can foster ongoing dialog and innovation in the field of RL-based defense mechanisms for cyber physical systems.

ACKNOWLEDGMENTS

This work was mainly supported by the U.S.-Israel Energy Center managed by the Israel-U.S. Binational Industrial Research and Development (BIRD) Foundation. This work was also supported by AFOSR under Grant No. FA9550-21-1-0438.

AUTHOR DECLARATIONS

Conflict of Interest

The authors have no conflicts to disclose.

Author Contributions

Mohammadamin Moradi: Conceptualization (lead); Data curation (lead); Formal analysis (lead); Investigation (lead); Methodology (lead); Writing – original draft (equal); Writing – review & editing (equal). **Shirin Panahi**: Investigation (supporting). **Zheng-Meng Zhai**: Investigation (supporting). **Yang Weng**: Conceptualization (supporting). **John Dirkman**: Conceptualization (supporting). **Ying-Cheng Lai**: Conceptualization (supporting); Formal analysis (supporting); Funding acquisition (lead); Project administration (lead); Supervision (lead); Writing – original draft (equal); Writing – review & editing (equal).

DATA AVAILABILITY

The data and code are available at https://github.com/AminMoradiXL/TGCN.

REFERENCES

[1] P. Pourbeik, P. S. Kundur, and C. W. Taylor, "The anatomy of a power grid blackout—Root causes and dynamics of recent major blackouts," IEEE Power Energy Mag. **4**, 22–29 (2006).

[2] G. Liang, S. R. Weller, J. Zhao, F. Luo, and Z. Y. Dong, "The 2015 Ukraine blackout: Implications for false data injection attacks," IEEE Trans. Power Syst. **32**, 3317–3318 (2017).

[3] A. Hobbs, "The colonial pipeline hack: Exposing vulnerabilities in us cybersecurity," in *SAGE Business Cases* (SAGE Publications SAGE Business Cases Originals, 2021).

[4] J. Cervini, A. Rubin, and L. Watkins, "Don't drink the cyber: Extrapolating the possibilities of Oldsmar's water treatment cyberattack," in *International Conference on Cyber Warfare and Security* (Academic Conferences International Limited, 2022), Vol. 17, pp. 19–25.

[5] S. Narvekar, B. Peng, M. Leonetti, J. Sinapov, M. E. Taylor, and P. Stone, "Curriculum learning for reinforcement learning domains: A framework and survey," J. Mach. Learn. Res. **21**, 7382–7431 (2020).

[6] P. Ammanabrolu and M. O. Riedl, "Playing text-adventure games with graph-based deep reinforcement learning," arXiv:1812.01628 (2018).

[7] J. He, J. Chen, X. He, J. Gao, L. Li, L. Deng, and M. Ostendorf, "Deep reinforcement learning with a natural language action space," arXiv:1511.04636 (2015).

[8]L. Zhao, Y. Song, C. Zhang, Y. Liu, P. Wang, T. Lin, M. Deng, and H. Li, "T-GCN: A temporal graph convolutional network for traffic prediction," IEEE Trans. Intell. Transp. Syst. **21**, 3848–3858 (2019).
[9]K. R. Davis, C. M. Davis, S. A. Zonouz, R. B. Bobba, R. Berthier, L. Garcia, and P. W. Sauer, "A cyber-physical modeling and assessment framework for power grid infrastructures," IEEE Trans. Smart Grid **6**, 2464–2475 (2015).
[10]B. Ning and L. Xiao, "Defense against advanced persistent threats in smart grids: A reinforcement learning approach," in *2021 40th Chinese Control Conference* (IEEE, 2021), pp. 8598–8603.
[11]N. I. Haque, M. H. Shahriar, M. G. Dastgir, A. Debnath, I. Parvez, A. Sarwat, and M. A. Rahman, "Machine learning in generation, detection, and mitigation of cyberattacks in smart grid: A survey," arXiv:2010.00661 (2020).
[12]M. Moradi, Y. Weng, and Y.-C. Lai, "Defending smart electrical power grids against cyberattacks with deep q-learning," PRX Energy **1**, 033005 (2022).
[13]M. Moradi, Y. Weng, J. Dirkman, and Y.-C. Lai, "Preferential cyber defense for power grids," PRX Energy **2**, 043007 (2023).
[14]J. Yan, H. He, X. Zhong, and Y. Tang, "Q-learning-based vulnerability analysis of smart grid against sequential topology attacks," IEEE Trans. Inf. Forensics Secur. **12**, 200–210 (2017).
[15]V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level control through deep reinforcement learning," Nature **518**, 529–533 (2015).
[16]V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. A. Riedmiller, "Playing atari with deep reinforcement learning," arXiv:1312.5602 (2013).
[17]Y. Li and J. Wu, "Low latency cyberattack detection in smart grids with deep reinforcement learning," Int. J. Electr. Power Energy Syst. **142**, 108265 (2022).
[18]Z. Wang, H. He, Z. Wan, and Y. Sun, "Coordinated topology attacks in smart grid using deep reinforcement learning," IEEE Trans. Ind. Inf. **17**, 1407–1415 (2020).
[19]C. Roberts, S.-T. Ngo, A. Milesi, S. Peisert, D. Arnold, S. Saha, A. Scaglione, N. Johnson, A. Kocheturov, and D. Fradkin, "Deep reinforcement learning for der cyber-attack mitigation," in *IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (IEEE SmartGridComm 2020)* (IEEE, 2020), pp. 1–7.
[20]B. Kiumarsi, K. G. Vamvoudakis, H. Modares, and F. L. Lewis, "Optimal and autonomous control using reinforcement learning: A survey," IEEE Trans. Neural Networks Learn. Syst. **29**, 2042–2062 (2018).
[21]S. E. Razavi, M. Moradi, S. Shamaghdari, and M. B. Menhaj, "Adaptive optimal control of unknown discrete-time linear systems with guaranteed prescribed degree of stability using reinforcement learning," Int. J. Dyn. Control **10**, 870–878 (2022).
[22]A. Marot, B. Donnot, K. Chaouache, A. Kelly, Q. Huang, R.-. Hossain, and J. L. Cremer, "Learning to run a power network with trust," Electr. Power Syst. Res. **212**, 108487 (2022).
[23]J. Duan, D. Shi, R. Diao, H. Li, Z. Wang, B. Zhang, D. Bian, and Z. Yi, "Deep-reinforcement-learning-based autonomous voltage control for power grid operations," IEEE Trans. Power Syst. **35**, 814–817 (2019).
[24]Dataset: B. Donnot (2020). "Grid2Op—A testbed platform to model sequential decision making in power systems," Github. https://GitHub.com/rte-france/grid2op
[25]M. Subramanian, J. Viebahn, S. H. Tindemans, B. Donnot, and A. Marot, "Exploring grid topology reconfiguration using a simple deep reinforcement learning approach," in *2021 IEEE Madrid PowerTech* (IEEE, 2021), pp. 1–6.
[26]Y. Yang, W. Pei, W. Deng, H. Xiao, and H. Sun, "Control method of power grid topology structure based on reinforcement learning," IOP Conf. Ser.: Earth Environ. Sci. **675**, 012073 (2021).
[27]Y. Zheng, Z. Yan, K. Chen, J. Sun, Y. Xu, and Y. Liu, "Vulnerability assessment of deep reinforcement learning models for power system topology optimization," IEEE Trans. Smart Grid **12**, 3613–3623 (2021).
[28]S. Taha, J. Poland, K. Knezovic, and D. Shchetinin, "Learning to run a power network under varying grid topology," in *2022 IEEE 7th International Energy Conference (ENERGYCON)* (IEEE, 2022), pp. 1–6.
[29]A. R. R. Matavalam, K. P. Guddanti, Y. Weng, and V. Ajjarapu, "Curriculum based reinforcement learning of grid topology controllers to prevent thermal cascading," IEEE Trans. Power Syst. **38**, 4206 (2022).
[30]M. Ghiasi, T. Niknam, Z. Wang, M. Mehrandezh, M. Dehghani, and N. Ghadimi, "A comprehensive review of cyber-attacks and defense mechanisms for improving security in smart grid energy systems: Past, present and future," Electr. Power Syst. Res. **215**, 108975 (2023).
[31]M. Ghiasi, S. Esmaeilnamazi, R. Ghiasi, and M. Fathi, "Role of renewable energy sources in evaluating technical and economic efficiency of power quality," Technol. Econ. Smart Grids Sustainable Energy **5**, 1–13 (2020).
[32]M. Dehghani, T. Niknam, M. Ghiasi, N. Bayati, and M. Savaghebi, "Cyber-attack detection in dc microgrids based on deep machine learning and wavelet singular values approach," Electronics **10**, 1914 (2021).
[33]H. Shirazi, M. Ghiasi, M. Dehghani, T. Niknam, M. G. Garpachi, and A. Ramezani, "Cost-emission control based physical-resilience oriented strategy for optimal allocation of distributed generation in smart microgrid," in *2021 7th International Conference on Control, Instrumentation and Automation (ICCIA)* (IEEE, 2021), pp. 1–6.
[34]J. Liu and W. Guan, "A summary of traffic flow forecasting methods," J. Highw. Transp. Res. Dev. **21**, 82–85 (2004).
[35]M. S. Ahmed and A. R. Cook, *Analysis of Freeway Traffic Time-Series Data by Using Box-Jenkins Techniques* (Transportation Research Record, 1979), p. 722.
[36]A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," Stat. Comput. **14**, 199–222 (2004).
[37]T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," arXiv:1609.02907 (2016).
[38]K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," arXiv:1409.1259 (2014).
[39]M. J. Hossain and M. Rahnamay-Naeini, "State estimation in smart grids using temporal graph convolution networks," in *2021 North American Power Symposium (NAPS)* (IEEE, 2021), pp. 01–05.
[40]Z. Wu, Q. Wang, and X. Liu, "State estimation for power system based on graph neural network," in *5th International Electrical and Energy Conference (CIEEC)* (IEEE, 2022), pp. 1431–1436.
[41]H. Wu, Z. Xu, and M. Wang, "Unrolled spatiotemporal graph convolutional network for distribution system state estimation and forecasting," IEEE Trans. Sustainable Energy **14**, 297 (2022).
[42]R. Liu and L. Chen, "Attention based spatial-temporal graph convolutional networks for short-term load forecasting," J. Phys.: Conf. Ser. **2078**, 012051 (2021).
[43]J. Liu, C. Yao, and L. Chen, "Time adaptive transient stability assessment based on gating spatial temporal graph neural network and gated neural network," Front. Energy Res. **398**, 885673 (2022).
[44]S. Guo and J. Cao, "Reactive power optimization for voltage stability in energy internet based on graph convolutional networks and deep q-learning," in *4th IEEE International Conference on Industrial Cyber-Physical Systems (ICPS)* (IEEE, 2021), pp. 511–516.
[45]E. Dai and J. Chen, "Graph-augmented normalizing flows for anomaly detection of multiple time series," arXiv:2202.07857 (2022).
[46]M. J. Kearns and U. Vazirani, *An Introduction to Computational Learning Theory* (MIT Press, 1994).
[47]L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement learning: A survey," J. Artif. Intell. Res. **4**, 237–285 (1996).
[48]R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction* (MIT Press, 2018).
[49]C. J. Dong, C. F. Shao, C. X. Zhuge, and M. Meng, "Spatial and temporal characteristics for congested traffic on urban expressway," J. Beijing Univ. Technol. **38**, 1242–1246+ (2012).
[50]Z.-M. Zhai, M. Moradi, L.-W. Kong, B. Glaz, M. Haile, and Y.-C. Lai, "Model-free tracking control of complex dynamical trajectories with machine learning," Nat. Commun. **14**, 5698 (2023).

[51] Z.-M. Zhai, M. Moradi, L.-W. Kong, and Y.-C. Lai, Phys. Rev. Appl. **19**, 034030 (2023).
[52] M. Moradi, Z.-M. Zhai, A. Nielsen, and Y.-C. Lai, "Random forests for detecting weak signals and extracting physical information: a case study of magnetic navigation," APL Mach. Learn. **2**, 016118 (2024).
[53] Z.-M. Zhai, M. Moradi, B. Glaz, M. Haile, and Y.-C. Lai, "Machine-learning parameter tracking with partial state observation," Phys. Rev. Res. **6**, 013196 (2024).
[54] J. L. Gross, J. Yellen, and M. Anderson, *Graph Theory and Its Applications* (Chapman and Hall;CRC, 2018).