Energy-Adaptive and Robust Monitoring for Smart Farms Based on Solar-Powered Wireless Sensors

Dian Chen, Qisheng Zhang, *Student Member, IEEE,* Ing-Ray Chen, *Member, IEEE,* Dong Sam Ha, *Fellow, IEEE,* and Jin-Hee Cho, *Senior Member, IEEE*

***Abstract*—While smart farm technologies significantly aid in reducing costs and boosting productivity for farmers, they often lack the necessary robustness against cyberattacks and adaptability to dynamic environmental changes. We propose a solar-powered sensor-based smart farm system to provide high monitoring quality while preserving sensor energy in the presence of adversarial attacks. In a smart farm system, solar-powered sensors are attached to animals (e.g., cows) to monitor their health under varying weather conditions to provide energy-adaptive and high-quality monitoring services. Further, a smart farm system should be robust against adversarial attacks aiming to disrupt monitoring quality. We use deep reinforcement learning (DRL) to identify the optimal policy for maximizing monitoring quality and prolonging the system's lifetime while maintaining sufficient energy. We introduce transfer learning (TL) into the DRL process to achieve fast learning without experiencing a cold start problem in DRL. In addition, we develop an uncertainty-aware anomaly data detection method to filter out deceptive data caused by adversarial attacks. Via extensive comparative performance analysis conducted based on real datasets, we demonstrate the superior performance of the proposed TL-based DRL strategies over existing competitive counterparts in the system lifetime, the monitoring quality, the learning convergence time, and the energy consumption.**

***Index Terms*—Smart farm, energy-aware, transfer learning, deep reinforcement learning, solar-powered sensors, cyberattacks.**

I. Introduction

A. Motivation & Goal

The Food and Agriculture Organization (FAO) forecasts that the global consumption of meat proteins will increase by 14% over the next decade based on statistics from 2018 to 2020, mainly led by the growth of population and the income [1]. To accommodate this growth, animal agriculture has evolved a characteristic that increases the efficiency and the quality of animal monitoring, while expanding the scale and quantity. Animals are managed as large groups to improve productivity, requiring intelligent monitoring methods to reduce the labor costs. The adoption of the sensor-based animal monitoring system, a major component of smart farm technologies, satisfies the demand efficiently and cost-effectively. Smart farm technologies leverage the cooperation of sensors, Internet-of-Things (IoT), edge, and cloud computing techniques. Although smart farm research has been conducted over decades to monitor animal conditions and control the environment, it still lacks security-aware and energy-adaptive smart farm technologies in energy-constrained environments.

In a smart farm, each animal has a small sensor attached to its ear to collect data, such as heartbeat and body temperature, for monitoring the animal's health and behavior remotely. To avoid laborious and extravagant efforts of replacing sensor batteries, we consider sensors powered by solar energy harvesting. Constrained by the size of a solar panel and the nature of energy harvesting, the energy level of the sensor is limited and fluctuates, which requires an intelligent policy to sustain long and uninterrupted operation of the sensors.

We consider the presence of adversarial attackers that aim to disrupt monitoring quality and thus disable the normal operation of a smart farm system. With the increasing vulnerability of smart farm technologies to a wide array of cyberattacks, there is a pressing need for a smart farm system designed to be resilient against such attacks. This system is crucial for maintaining the sustainability and reliability of monitoring animal conditions. Consequently, our research addresses not only the broad spectrum of attacks peculiar to smart farm settings but also devises appropriate defenses to guarantee the uninterrupted operation of the monitoring system. See Section IV-C for the types of attacks considered in this paper. This research aims to develop an attack-resilient and energy-adaptive monitoring system for smart farms. A rule-based system requires extensive manual tuning and updates as the environment changes, rendering it less adaptable to new, dynamic environments. Moreover, it cannot handle inherent uncertainty with deterministic rules. In environments constrained by energy resources, smart farm systems face the significant challenge of balancing monitoring quality with energy consumption to achieve optimal operations and ensure sustainability. Traditional heuristic methods aim to simplify complex decision-making by establishing rules, especially when finding an optimal solution is unfeasible. However, the intricate and dynamic nature of smart farm environments, influenced by factors such as animal movements and varying weather conditions, renders simple rule-based solutions ineffective. These complexities introduce a level of unpredictabil-

 This research was partly sponsored by the National Science Foundation (NSF) under Grant Numbers CNS-2106987 and III-2107450. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein (*Corresponding author: Dian Chen*). Dian Chen, Qisheng Zhang, Ing-Ray Chen and Jin-Hee Cho are with the Department of Computer Science, Virginia Tech, Falls Church, VA, USA. Dong Sam Ha is with the Bradley Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg, VA, USA. E-mail: {dianc, qishengz19, irchen, ha, jicho}@vt.edu.

ity that heuristic approaches struggle to manage, highlighting the necessity for our DRL-based methodology to provide precise and adaptable solutions. Additionally, in addressing the dynamic challenges of smart farm monitoring with solar sensors under adversarial threats, Deep Reinforcement Learning (DRL) proves superior to Model Predictive Control (MPC). DRL's adaptability enables continuous strategy updates in response to unpredictable changes, which is a vital asset in the complex, nonlinear realm of smart agriculture. Moreover, DRL's scalability and robustness against adversarial threats provide effective, resilient monitoring solutions [2, 3]. In contrast, MPC's static, model-dependent approach and computational demands make it less viable for the flexible, evolving needs of modern smart farms [4, 5]. Thus, DRL not only fulfills the system's adaptive control and scalability but also offers strategic defenses against security threats, affirming its selection as the best approach. Therefore, we propose a deep reinforcement learning (DRL)-based monitoring approach for dynamic and autonomous decision-making to achieve a high monitoring quality of animal conditions, while maintaining the energy level for sensors under adversarial attacks causing uncertainties in transmitted data.

B. Key Contributions

Our work has the following **key contributions**:

- We propose *an attack-resilient, energy-adaptive monitoring system with solar sensors* in a wireless sensor network for smart farms. This research represents the inaugural effort to develop an energy-adaptive monitoring system incorporating solar sensors and animal behavior analytics, designed to optimally leverage limited and variable energy resources for superior monitoring quality.
- We leverage *transfer learning-based deep reinforcement learning (TL-DRL)* to accelerate the training time under fluctuating energy levels and adversarial attacks. In addition, we validate the robustness and effectiveness of the proposed DRL agents with extensive experiments on real datasets [6], demonstrating the outperformance of our proposed TL-DRL-based approach over existing, competitive counterparts in the system lifetime, monitoring quality, learning convergence time, and energy consumption.
- We propose an uncertainty-aware monitoring opinion update method to quantify the uncertainty in the proposed monitoring system. Moreover, we develop a *Subjective Logic*-based deceptive data detection algorithm that can detect anomaly data from compromised sensor nodes based on a new design notion of the degree of conflict estimated from the distance between opinions obtained from different gateways.
- We consider a full set of adversarial attacks that can happen to smart farms, including a neural trojan attack, a fast gradient sign method (FGSM) attack, and a projected gradient descent adversarial attack (PGDA), as described in Section IV-C. To our knowledge, no prior work has considered those adversarial attacks to ensure monitoring quality for smart farms.

C. Structure of the Paper

The paper is structured as follows: Section II reviews the literature on smart sensor systems, energy-efficient monitoring, and transfer learning. Section III outlines the problem statement. Section IV describes the system model. Section V details our DRL-based TL approach for smart farm monitoring. Section VI discusses parameterization, experimental settings, metrics, and comparison schemes. Section VII presents comparative and sensitivity analyses of the results. Finally, conclusions and key findings are summarized in Section VIII.

II. Background & Related Work

A. Smart Sensor Systems

Smart sensor systems have been mainly studied for energy-adaptive designs. Kumar et al. [7] proposed an IoT-based monitoring system, called `gCrop`, to measure conditions of leaf growth and then predict the age of leaves using ML and computer vision techniques. The system deployed a low-powered training model in energy-aware or resource-constrained environments. Liu et al. [8] proposed a meta-heuristics solution to improve the performance of dynamic, wireless sensor networks. They deployed an agent-assisted Quality-of-Service (QoS)-based routing algorithm to identify an optimal route that maximizes QoS and minimizes complexity. Unlike [7, 8], our work aims to develop energy-adaptive, secure, and robust transfer learning (TL) mechanisms for building a resilient smart farm against adversarial attacks. In addition, our work considers how to expedite the learning convergence in resource-constrained and adversarial environments.

Cybersecurity for wireless sensor networks (WSNs) has been studied for decades. However, cybersecurity for smart sensor systems (e.g., smart farms) has emerged recently. Gupta et al. [9] identified cybersecurity concerns related to data and network attacks in smart farm fields. Saheed and Arowolo [10] developed a cyber attack detector to prevent the Internet-of-Medical-Things (IoMT)-smart environments from various cyberattacks. They deployed a bio-inspired optimization algorithm to effectively and efficiently train the proposed deep recurrent neural network (RNN) for attack detection by refining features in sensor data. Chae and Cho [11] introduced a P2P-based smart farm system to prevent attackers from managing the communication and data stored in the smart farm system. They developed an efficient authentication method to reduce the operation time relatively compared to the traditional encryption/decryption. Relative to the cited works above, which focused on authentication/detection, our work addresses a full set of security threats in smart farm systems. Further, we apply TL first to achieve energy-efficient and attack-resilient monitoring quality for smart farm systems.

Aliyu and Liu [12] and Vangala et al. [13] proposed a blockchain-based smart farm system to improve security and privacy. The system can detect and respond to security threats by integrating blockchain transactions on smart farming requests. Although blockchain technology has been used to enhance security in smart systems, its high computational cost

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2024.3409525

and its adverse impact on system performance cannot provide practical solutions in resource-constrained WSNs.

Alemayehu and Kim [14] proposed a heuristic traveling salesman problem algorithm to improve data acquisition latency in WSNs. Their approach effectively eliminates duplicate sensor nodes in the data transmission path to not only conserve energy but also maximize data freshness within a fairly large transmission range. Since they proposed the adaptive-energy-distance (AED)-based monitoring system, we consider [14] as the state-of-the-art counterpart solution for performance comparison in Section VII. Unlike [14] focusing on performance issues in resource-constrained wireless sensor networks, our work considers both performance (including both energy and latency) and security issues of monitoring services in response to network dynamics in resource-constrained solar sensor-based smart farm systems, which has not been considered in the existing smart farm research.

Akhter et al. [15] developed a real-time smart system for water quality monitoring using multifunctional sensors and LoRa technology to measure various water attributes, including temperature and nutrient levels, applying a KNN model for nutrient prediction. Nagarajan et al. [16] introduced an IoT-based food supply chain for smart cities, enhancing food quality, vehicle routing, and contamination source tracing with a routing optimization algorithm. Aggarwal and Sharma [17] proposed a deep learning-enhanced smart home voice recognition system, utilizing a DNN to reduce noise and echo for improved performance in challenging conditions. Fan et al. [18] developed a BA-TENG-based smart glove, employing sensors and machine learning to recognize user finger movements. Li et al. [19] introduced a wearable sensor system for real-time health monitoring, diagnosing diseases from collected data like human breath. Catalano et al. [20] and Ghazal et al. [21] focused on smart agriculture and IoT device security, respectively, the former on machine learning-based anomaly detection to enhance data accuracy and the latter on a DDoS detection mechanism using an ensemble technique for improved accuracy against diverse malware activities. Numerous strategies have been developed to enhance smart sensor systems' efficiency, yet many overlook the challenges presented by resource-limited settings and the broad spectrum of security threats inherent to sensor technologies.

B. Energy-efficient Monitoring Systems

Saba et al. [22] introduced an energy-efficient IoMT framework (SEF-IoMT) to cut communication costs and energy use in patient health monitoring, employing Kruskal's algorithm for optimal routing. Lilhore et al. [23] enhanced a genetic algorithm for energy-efficient routing, selecting only optimum energy nodes to lessen data transmission and improve energy efficiency. Zhuo et al. [24] tackled underwater acoustic sensor networks (UWSNs) energy limitations with an optimization model for efficient data collection, minimizing energy use by identifying the shortest paths for autonomous underwater vehicles (AUVs). Similarly, Bharany et al. [25] optimized communication via an energy-efficient clustering protocol using glowworm swarm optimization for optimal cluster head selection. Haseeb et al. [26] developed an IoT-based wireless sensor network (WSN) for smart agriculture, introducing a decision function for efficient cluster head selection to minimize energy consumption. Kocherla et al. [27] proposed an energy-efficient routing algorithm for surveillance systems, focusing on minimizing energy consumption amidst dynamic environmental challenges. Our work diverges by developing energy-adaptive strategies for environments with fluctuating energy resources, aiming for high-quality monitoring in complex and dynamic settings, incorporating solar sensors and animal behavior analysis—marking a pioneering approach to addressing smart farm uncertainties with an energy-adaptive system.

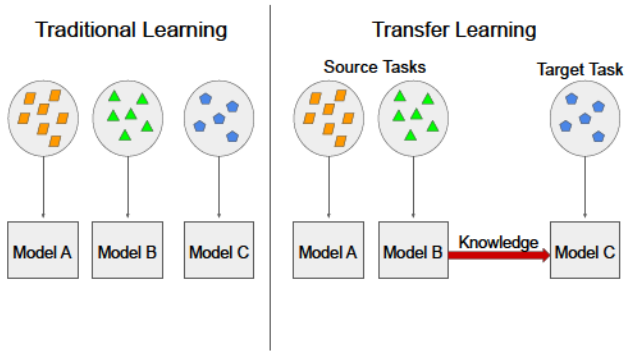


Fig. 1. The Process of Traditional Learning vs. Transfer Learning.

C. Transfer Learning

Transfer learning (TL) is a learning mechanism that uses learned knowledge from one problem solved previously to solve a different but related problem [28], as described in Fig. 1. Given a source domain $\mathcal{D}_S$, and a learning task $\mathcal{T}_S$, a target domain $\mathcal{D}_T$, and a learning task $\mathcal{T}_T$, TL aims to improve the learning of the target predictive function, $f_T(\cdot)$, in $\mathcal{D}_T$ using the knowledge in $\mathcal{D}_S$ and $\mathcal{T}_S$, where $\mathcal{D}_S \neq \mathcal{D}_T$ and $\mathcal{T}_S \neq \mathcal{T}_T$. TL research has mainly addressed *what-to-transfer* and *how-to-transfer* in the unexplored area of *when-to-transfer*, assuming that there exists a relationship between the source domain and the target domain. To answer these questions, TL has been mainly studied by the settings of the domains and tasks: *inductive TL*, *transductive TL*, and *unsupervised TL*. Each TL approach is detailed below.

1) **Inductive TL:** TL under inductive settings can be applied when $\mathcal{T}_S \neq \mathcal{T}_T$, for the case in which the source task and the target task are different. Inductive TL is studied based on four types. The first type is *instance-based TL (ITL)*, which reuses some parts of the data in the source domain for learning in the target domain. The main two techniques used in ITL are *instance reweighting* and *importance sampling* [28]. Dai et al. [29] proposed a boosting-based learning approach called `TrAdaBoost` to reduce the differences in the distributions between the outdated data and new data by setting weights to these training instances to reduce their impact on the model. The second type is feature-based TL (FTL), which transfers the learned feature

representation to a target domain [28]. Argyriou et al. [30] proposed a learning algorithm for feature representations with regularization and optimization techniques. They aimed to identify common features in both source and target tasks and determine the optimal representation for features. The third type is *parameter-based TL (PTL)* which considers the learning parameters of a model. For example, van Kasteren et al. [31] developed a parameter-based TL technique applied to real-world applications for activity recognition. Their approach relaxed the assumption that data are in identical-independent distribution (IID), and thus unlabelled data can be utilized to learn the parameters of a model. They used multiple source models to learn hyperparameters of the prior distribution. After then, they determined the parameters for the target model and all available data. The fourth type is *relation-based TL (RTL)* which learns the relations of knowledge across domains. To overleap the initial steps of learning new tasks in machine learning (ML), Mihalkova et al. [32] developed a transfer system based on *Markov Logic Network* (MLN) to transfer relations as knowledge across domains. After constructing the MLN from source domains, the system can automatically map to the target domain.

2) **Transductive TL:** This can be applied when the source task and the target task are the same while the source domain and the target domain are different [33], i.e., $\mathcal{T}_S = \mathcal{T}_T$ and $\mathcal{D}_S \neq \mathcal{D}_T$. Under this setting, two main TL approaches, instance-based and feature-based, are considered. Leveraging *instance-based TL*, Dai et al. [34] developed an algorithm applied on Expectation Maximization (EM)-based Naïve Bayes classifiers for text classifications. The main idea is to minimize the differences between distributions of train and test data using Kullback-Leibler divergence as a measurement. Zhang et al. [35] employed feature-based TL to develop a deep TL (DTL) framework using automation techniques in tuning parameters called `DeepRisk`. They constructed and transformed feature vectors and learned the optimal weights of nodes in networks through a set of neural networks. To our knowledge, no existing parameter-based or relation-based TL approach has been conducted under transductive settings.

3) **Unsupervised TL:** This approach can be applied for cases in which the source task is different from the target task while there is no labeled data that can be observed in the training stages. Song and Zhang [36] proposed a technique called *transferred dimensionality reduction*. It can be used for the case when existing labeled data are not in the same domain so they can be considered as unlabeled data. They proposed a *transferred discriminative analysis* method for determining and transferring valuable information from labeled classes to unlabeled classes in the target domain while improving the accuracy of clustering with unlabeled classes.

Coraci et al. [37] proposed an Online Transfer Training (OTL) strategy, enhancing learning efficiency and scalability for smart building systems. They trained DRL agents on various buildings as the pre-trained model, using OTL to fine-tune control policies to minimize electricity costs in a target building without prior knowledge. Gamrian and Goldberg [38] introduced Analogy-based Zero-Shot Transfer, a transfer learning method for overcoming generality issues in variants

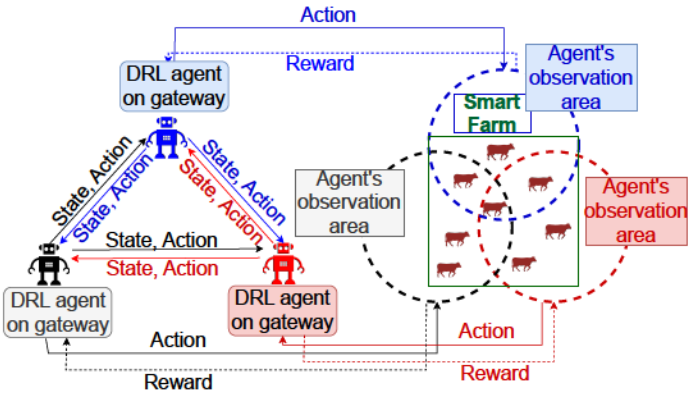


Fig. 2. The considered Multi-agent DRL environment.

of the Breakout game. This approach enables the agent to generate analogies between the target and source domains, mapping similar states to facilitate policy learning. Anwar and Raychowdhury [39] applied TL strategies to value-based DRL for autonomous navigation, significantly reducing energy consumption and training latency despite drone dynamics. Ke et al. [40] integrated TL into a Double Deep Q Network (DDQN)-based approach for variable speed limit (VSL) control, successfully transferring knowledge across multiple target domains in freeway merging scenarios. Additionally, Ciabatti et al. [41] evaluated the effectiveness of TL in a Deep Deterministic Policy Gradient (DDPG)-based DRL setting for landing tasks. Their findings indicated that landers (e.g., robotics) could perform tasks with high reward values on different celestial bodies, such as Mars and the Moon. Despite various advancements in combining TL and DRL, there remains a gap in addressing security concerns within systems and applying such integrations in smart farm systems.

TL has been used commonly to achieve faster learning convergence by transferring knowledge from one context to another context. However, there has been no prior work using TL to ensure high monitoring quality for an Internet-of-Things (IoT)-based smart environment under fluctuating energy and adversarial attacks. We fill this gap by proposing an efficient and secure monitoring system for a resilient smart farm by leveraging the merit of TL.

III. Problem Statement

The proposed smart farm system aims to maximize the monitoring quality and the remaining energy level of solar sensors in the presence of adversarial attacks and energy fluctuations. To attain this goal, we leverage multi-agent DRL, as described in Fig. 2, to identify the optimal policy $\pi : \mathcal{S} \times \mathcal{A} \rightarrow [0, 1]$, $\pi(s, a) = Pr(A_t = a \mid S_t = s)$ by:

$$\arg\max_{\mathbf{a}^*} \sum_{t=1}^{T} w_1 \mathcal{MQ}(a_t) + w_2 \mathcal{RE}(a_t), \tag{1}$$

where $\mathbf{a}^*$ denotes the optimal set of actions selected by an agent to maximize the combined metrics of $\mathcal{MQ}(a_t)+\mathcal{RE}(a_t)$ at any given time t, here, $\mathcal{MQ}(a_t)$ represents the monitoring quality at time t, and $\mathcal{RE}(a_t)$ signifies the average remaining energy level of sensor nodes at time t. The coefficients w_1 and

w_2 are weights allocated to each respective term. In our model, $\mathcal{MQ}(a_t)$ and $\mathcal{RE}(a_t)$ are assigned equal importance, highlighting our prioritization of maintaining both high monitoring quality and energy efficiency concurrently. Furthermore, we normalize these terms to fall within the $[0, 1]$ range, ensuring the reward function remains non-negative. This normalization is crucial as it does not impede the DRL agent's learning progression. The essence of DRL lies in acquiring the optimal policy by maximizing expected returns (rewards) consequent to actions taken [42]. Therefore, the differentiation in reward values across various actions will guide the DRL agent toward a policy that optimizes the reward function. $\mathcal{MQ}(a_t)$ refers to the monitoring quality at time t, estimated by:

$$\mathcal{MQ} = \frac{\sum_{t=1}^{T}\sum_{i=1}^{X}\sum_{j=1}^{d} \mathrm{mq}(i,j)}{|X| \times d}, \tag{2}$$

where T is the system's total operation period, X is the number of sensed data, d is the number of attributes for each animal, described in Table I, $GT_{i,j}$ is the ith ground truth data for jth attribute, and $x_{i,j}$ is our observed data. The $\mathrm{mq}(i, j)$ indicates the degree of monitoring quality in a jth attribute compared to the ith ground truth data and returns 1 when $x_{i,j} == GT_{i,j}$; 0 otherwise. In addition, $\mathcal{RE}(a_t)$ refers to sensor nodes' average remaining energy level at time t, given by:

$$\begin{aligned} \mathcal{RE} &= 1 - \left(\mathcal{E}_{\mathrm{SG}} + \mathcal{E}_{\mathrm{SS}} + \mathcal{E}_{\mathrm{active}} + \mathcal{E}_{\mathrm{sleep}}\right) \\ &= 1 - \left(\frac{(e_{\mathrm{SG}} + e_{\mathrm{SS}})}{E_{\mathrm{S}}} + \frac{T_u}{E_{\mathrm{S}}}(d_{\mathrm{active}} + d_{\mathrm{sleep}})\right), \end{aligned} \tag{3}$$

where e_{SG} and e_{SS} are the energy consumed per data transmission from a sensor to a gateway and a sensor to a sensor, respectively. The d_{active} and d_{sleep} are energy levels consumed per second in active and sleep modes. The E_{S} indicates the energy level when a sensor is fully charged.

A smart farm often leverages Long Range (LoRa) communications [43] to transmit the sensed data on animal conditions to LoRa gateways and then to the cloud server through the Internet. The proposed work develops an energy-aware DRL algorithm to identify the optimal policy for a given smart farm system, where the optimal policy indicates a set of low-energy sensor nodes to transmit data to LoRa gateways by requesting nearby high-energy sensor nodes to relay their data. It involves development of an effective and efficient energy policy and the TL-based DRL, uncertain data aggregation and update under adversarial attacks, and detection of deceptive data, as detailed in Section V.

To enhance readability, Table I is provided to summarize all the notations and corresponding meanings used in this work.

TABLE I
NOTATIONS & THEIR DEFINITIONS

Notation	Definition
HES	High energy sensor node
LES	Low energy sensor node
$\mathcal{MQ}(a_t)$	Monitoring quality by taking action a at time t
$\mathcal{RE}(a_t)$	Average remaining energy level of sensor nodes by taking action a at time t
$e_{\mathrm{SG}}/e_{\mathrm{SS}}$	Energy consumed per data transmission from a sensor to a gateway/a sensor to a sensor
$d_{\mathrm{active}}/d_{\mathrm{sleep}}$	Energy levels consumed per second in active/sleep modes
E_{S}	Energy level when a sensor is fully charged
$\mathcal{E}_{SG}$	Energy consumption for transmitting data from a sensor node to a gateway
$\mathcal{E}_{\mathrm{SS}}$	Energy consumption for transmitting data through BLE (from a sensor node to a nearby sensor node)
$\mathcal{E}_{\mathrm{active}}$	Energy drained in active mode in a time interval
$\mathcal{E}_{\mathrm{sleep}}$	Energy drained in sleep mode in a time interval
$\mathcal{S}_t$	State space
$\mathcal{A}_t$	Action space
r_t	Immediate reward
$\mathcal{R}_t$	Accumulated reward
$V_i(s)$	Total number of times the student agent i has visited state s
π	Policy learned by a DRL agent
ω_X^A	Aagent A's opinion about a given proposition X
a_X	Base rate (i.e., prior belief) distribution of variable X
b_X	Belief masses distribution
u_X	Uncertainty mass
DC	*degree of conflict*
PD	Projected distance
C_T	Convergence time
DQN	Deep Q-Network
PPO	Proximal Policy Optimization
TL-DQN	Transfer learning-based DQN
TL-PPO	Transfer learning-based PPO
TFT-PPO	Transfer learning with fine-tuning-based PPO
AED	Adaptive-Energy-Distance
FE	Fixed-energy

IV. SYSTEM MODEL

A. Network Model

The network comprises solar-powered sensors, LoRa gateways, and a cloud server, as described in Fig. 3. Sensors transmit data on animals' conditions to gateways directly or other sensors for relay of the data. A gateway aggregates sensed data as all animals' average conditions and periodically transfers them to the cloud server. Hence, a gateway connects sensors and the cloud server, enabling connectivity for IoT devices to be less expensive and have a longer range.

A TL model will be deployed on the gateways to maintain the monitoring system's normal operation. We assume that the communication between sensors and LoRa gateways may be vulnerable to cyberattacks [44] as the system does not use data encryption because encryption is not a viable solution under severe resource constraints in IoT environments. Therefore, multiple adversarial attacks (as stated in Section IV-C) may exist in data transmission to influence the quality of sensed data transferred from gateways to the cloud server. Our work investigates the robustness of our proposed approach to ensure monitoring quality under such threats.

B. Node Model

In a smart farm environment, sensors transmit data to LoRa gateways or other sensors based on the energy levels of the sensors. The sensors' energy levels constantly change from sunrise to sunset and are influenced by environmental conditions, such as the positions of animals and ambient temperature of the sensors. Thus, it is required for the system to be able to keep its monitoring quality, while maintaining sufficient sensors' energy levels. We define sensor node i at

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2024.3409525

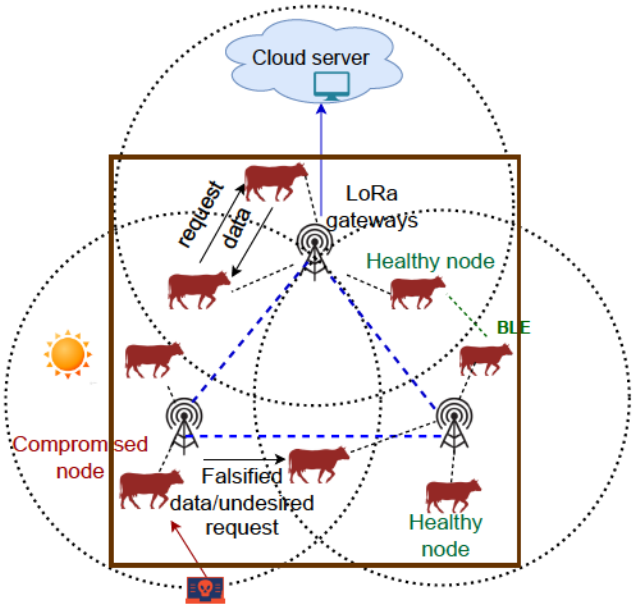


Fig. 3. The considered wireless solar sensor-based smart farm network.

time t, denoted by sn_t^i, with four attributes:

$$\text{sn}_t^i = [\text{temp}_t^i, \text{hb}_t^i, \text{ma}_t^i, \text{bl}_t^i], \tag{4}$$

where temp_t^i is the temperature of sensor node i at time t, hb_t^i is the heartbeat of node i at time t, ma_t^i refers to the moving activity of node i at time t, and bl_t^i means sensor node i's battery life. The ma_t^i and bl_t^i are scaled in $[0, 1]$, respectively, as percentage. Moreover, some sensor nodes are regarded as monitor nodes with a network-based intrusion detection system (NIDS) deployed. We describe how the NIDS is considered in Section IV-D. We describe the energy model used for LoRa, the procedures of Bluetooth low-energy data transmission, and the simulation settings in Section VI-A.

C. Attack Model

We consider the following adversarial attack behaviors that may disturb the normal monitoring services of the smart farm system concerned in this work.

1) **False data injection attacks:** This attack is performed by a compromised sensor node sending falsified or modified data to gateways [45]. Furthermore, when man-in-the-middle attacks occur, attackers can intercept our data and transmit their information to high-energy sensor (HES) nodes. We model this attack based on the probabilities of data to be affected by both inside and outside attackers.

2) **Non-compliance to the protocol:** A compromised sensor node may not comply with a given data delivery protocol [46]. We apply this attack that can request the sensor node to send its sensed data to a LoRa gateway. The compromised sensor node may be selfish and not transmit its sensed data to save energy. Further, it may discard the request from another low-energy sensor (LES) node, aiming to make the compromised sensor node transmit its sensed data.

3) **Denial-of-Service (DoS):** A compromised HES node can perform a DoS attack [47] by requesting its neighbor nodes to transmit its sensed but falsified data to a gateway even if it has a sufficient level of energy. This can expedite the energy depletion of other sensor nodes in the system. The DoS attacks can only be performed on healthy HES sensor nodes, as other compromised nodes will discard the request easily. We also consider distributed DoS (DDoS) attacks by allowing multiple compromised sensor nodes to perform DoS attacks on legitimate, healthy nodes. Since each node is limited in processing capacity, it may introduce errors in transmitting legitimate data to the LoRa gateways.

4) **Sensor data obstruction:** This is one impact of the WiFi Deauthentication attacks as a major availability attack in the smart farm environment [47]. Such an attack prevents sensor nodes from connecting to the network, making it lose real-time communication with other sensor nodes and LoRa gateways. The compromised HES nodes are disconnected from the network so that their sensed data cannot reach the gateways and be utilized in the decision-making and monitoring process.

5) **Neural Trojan Attack:** This attack inverses the neural networks (NNs) to produce a general Trojan trigger, a small input data [48]. After that, the attacker uses reverse engineering to inject malicious behaviors into the NN model. The model shows the malicious behaviors only when the Trojan trigger is activated on the input data. This attack is performed by a compromised gateway.

6) **Fast Gradient Sign Method (FGSM):** This attack is initially designed to disrupt an image classification model [49]. An inside attacker computes a loss function from input data (e.g., image) and generates an adversarial image to maximize the loss. This attack aims to mislead an agent to consider the worst action the most preferred. Thus, the actions can be considered class labels using an RL-based model while we can still determine the gradients by our initial loss function for the TL model. A compromised gateway can perform this attack to mislead the agents to act unwillingly.

7) **Projected Gradient Descent Adversarial Attack (PGDA):** This gradient-based adversarial attack is based on the Projected Gradient Descent (PGD) optimization technique to cause misclassification in classification tasks [50]. The attacker performs PGDA attacks by iteratively computing the sign of gradient with a small step size. In DRL settings, PGDA can mislead agents into taking poor or suboptimal actions. Compromised RoLa gateways may perform the PGDAs to force the agents to move to a certain action, impacting the system's monitoring performance.

In our model, we posit that a predetermined proportion of sensors, specifically 30%, are designated as compromised at the onset of the network's initial bootstrapping phase, a parameter we denote as P_{AE}. This parameter, P_{AE}, represents the critical threshold—the maximum quota of compromised sensors that the system is engineered to withstand without hindering its operational integrity. A Byzantine failure scenario ensues if the count of compromised nodes surpasses this threshold. This scenario pertains to a critical challenge within distributed systems, wherein nodes must achieve consensus or collective decision-making [51]. Byzantine failures signify a

system's inability to maintain consensus due to malfunctioning or malicious nodes exceeding the tolerable limit, P_{AE}. We model a compromised sensor performing an attack with the probability of P_A%. We summarize all the considered attack behaviors in Table II.

D. Defense Model

This section describes intrusion detection and intrusion response mechanisms considered in this work.

1) **Intrusion Detection:** Our proposed system will detect deceptive data received on each LoRa gateway. We assume that each gateway has an intrusion detection engine to monitor data coming from sensor nodes and detect the anomaly traffic and, accordingly, compromised sensor nodes. Hence, the gateways will reject any deceptive data if a sensor node is detected as sending δ (e.g., 5) times of rejected data, considering the node being compromised. We provide the details of the detection algorithm for deceptive data in Section V-F. To mitigate the impact of sensor data obstruction attacks, we introduced data redundancy in monitoring information, enabling a sensor node to broadcast its data to all gateways within its transmission range. This redundancy significantly enhances the system's resilience to such attacks. Furthermore, our IDS has been refined to detect compromised nodes by identifying selfish behaviors and DoS attacks, quantifying the detection accuracy with false positive (FP) and false negative (FN) rates set at 0.05 each. This setting ensures that the IDS system accurately identifies healthy sensor nodes as healthy with a 95% probability and correctly flags compromised nodes with the same level of precision.

2) **Intrusion Response:** Upon detecting a compromised sensor node, the system will take action to recover or remove the compromised sensor node. We consider the following actions to make the system secure and free from the compromised nodes [52]: The recovery action is selected based on the remaining energy of the compromised sensor node. If the compromised node has the remaining energy above 40%, it should be repaired; it will be replaced otherwise. To be specific, we will have the following defenses:

- *Repair*: A compromised node will be assessed if it can be repaired. Since repairing too many sensor nodes simultaneously may break down the monitoring system, we consider a probability threshold (i.e., $\zeta_1 \in [0, 1]$) to determine whether to repair the compromised node. If the system decides to repair the compromised node, it will need to allow the node's downtime, T_{repair}, impacting the monitoring quality of the animal with the compromised sensor node.
- *Replace*: If the compromised node is determined as 'not repaired' due to its low remaining energy, it must be replaced with a new sensor. Since there will be a delay in replacing the sensor, it will need to allow the node's downtime, T_{replace} (e.g., 1-6 hours), which can also affect the monitoring quality of the animal with the compromised sensor node.

3) **Adversarial training:** In addition, we have implemented adversarial training techniques for DRL agents to counteract adversarial attacks on gateways, such as those executed via the Fast Gradient Sign Method (FGSM). This training approach equips DRL agents to learn from both standard and adversarially perturbed data inputs, enhancing their robustness and resilience against such attacks [53].

V. DRL-based TL for Monitoring Smart Farms

A. Energy Policy

We denote the minimum battery level of a sensor node that can send its sensed data to gateways by L_{bl}. If a node's battery level is higher than L_{bl}, then it is an HES, transmitting sensed data regularly per interval (e.g., 30 seconds). Otherwise, it is an LES and will send its data to a nearby HES. To extend a sensor's battery lifetime, which will significantly impact the lifetime of the proposed monitoring system, we develop an intelligent energy policy that determines *what sensed data a sensor node should update to gateways and when to update them*. The key idea is to keep a list of nearby sensors ordered by their remaining energy in ascending order. Based on the sensors' battery levels and the system's current monitoring quality, the policy determines a set of LES nodes based on threshold ρ. Only the top ρ percentage of the LESs can transmit its data to the nearby HES nodes. We use a DRL algorithm to optimize the energy policy further to reduce the frequency of data transmissions and save energy when the monitoring system can properly operate without degrading the system's monitoring quality. We normalize the level of energy consumption to $[0, 1]$. A fully charged sensor node will be set to 1, which is set as the initial state.

We have four energy consumption levels for both data transmission and energy drainage under active mode and sleep mode over time. A sensor node operates normally in an active mode with a battery level greater than L_{bl}. When a sensor node does not have the ability to transmit data, $bl < L_{bl}$, it goes to sleep mode. As discussed above, when a sensor node's battery level, bl, is higher than L_{bl}, it will transmit data to gateways per time interval, T_u (e.g., 30 sec.). We denote a fully charged sensor's energy by E_S and energy consumed to send a data packet from a sensor to a gateway and another sensor through the BLE (Bluetooth Low Energy) protocol by e_{SG} and e_{SS}, respectively. We define the amount of energy drained per second under active mode and sleep mode by d_{active} and d_{sleep}, respectively. For simplicity, we normalize the energy consumption by (1) $\mathcal{E}_{SG}$: Energy consumption for transmitting data from a sensor node to a gateway, calculated by e_{SG}/E_S; (2) $\mathcal{E}_{\text{SS}}$: Energy consumption for transmitting data through BLE (from a sensor node to a nearby sensor node), calculated by e_{SS}/E_S; (3) $\mathcal{E}_{\text{active}}$: Energy drained in active mode in a time interval, calculated by $(d_{\text{active}}T_u)/E_S$; and (4) $\mathcal{E}_{\text{sleep}}$: Energy drained in sleep mode in a time interval, estimated by $(d_{\text{sleep}}T_u)/E_S$.

B. DRL-based Monitoring System

In our smart farm framework, environmental dynamics play a pivotal role in influencing both monitoring quality and energy consumption. For example, the efficiency of sensor node charging decreases in overcast conditions or when animals seek shade, directly impacting the sensors' energy reserves and, subsequently, the monitoring quality. Traditional static or

TABLE II
DESCRIPTIONS OF ATTACK BEHAVIORS (HES: HIGH ENERGY SENSOR; LES: LOW ENERGY SENSOR; LG: LORA GATEWAY)

Node type	Attack type	Meaning
Healthy HES	Outside false data injection attacks	A healthy HES's data is intercepted by the outside attackers (e.g., man-in-the-middle-attacks) and sent to gateways; send correct data otherwise.
Compromised HES	Non-compliance	A compromised HES does not send its own and other LESs' data to gateways
	Inside false data injection attacks	A compromised HES's data is intercepted by the inside attackers and sent the data to gateways; send correct data otherwise.
	DoS attacks	A compromised HES sends fake requests to nearby LESs
	Sensor data obstruction	A compromised HES is prevented from connecting to the network, resulting in its data not reaching the gateways.
Healthy LES	Outside false data injection attacks	A healthy LES's data is intercepted by outside attackers (e.g., man-in-the-middle-attacks) and sent to healthy HES, which can deliver the data to gateways; otherwise, send correct data.
Compromised LES	Non-compliance	A compromised LES does not send its sensed data to HESs
	Inside false data injection attacks	A compromised LES's data is intercepted by the inside attackers and sent to a healthy HES, which can deliver the data to the gateway; otherwise, send correct data.
Compromised LG	Neural Trojan attacks	A compromised LG can inject malicious behaviors into the NN model deployed on DRL agents; otherwise, it does not affect the NN model.
	FGSM attacks	A compromised LG can mislead DRL agents to take unwilling actions; DRL agents perform correctly.
	PGD attacks	A compromised LG forces DRL agents to take undesired actions; DRL agents take correct action.

rule-based energy management strategies are inadequate for maintaining peak system performance over time due to these fluctuations. To overcome this, we have adopted a dynamic energy policy tailored for sensor data transmission, which is further refined by our DRL agent. This refinement process ensures optimal monitoring quality and energy efficiency, allowing for real-time adjustments to environmental variations and ensuring the system's responsiveness to such changes. Additionally, our DRL agents are equipped with adversarial training that incorporates both conventional and adversarial updates, including those from FGSM attacks, as elaborated in Section IV.D (Defense Model). This method significantly bolsters the system's defense mechanisms, augmenting its resilience to cyber threats.

- **State space** ($\mathcal{S}_t$): Each LoRa gateway maintains a local database to record all sensed data received from sensor nodes. Each DRL agent on the gateway will compute and update a set of opinions from sensed data. Each DRL agent's ability to observe the monitoring environment is limited, and each DRL agent can only access its dataset. However, since our uncertainty-aware system uses the *degree of conflict* (see Section V-D1) to identify if received data is compromised, the DRL agent is required to periodically request another DRL agent which receives sensed data from the same node to share its opinion with it to determine if it supposes to accept the new data or not. In a system state, the DRL agent can also observe each sensor's remaining energy level within its transmission range. We define the state space at time step t by $\mathcal{S}_t = \{s_1^t, s_2^t, \ldots, s_n^t\}$, where n is the total number of DRL agents running on LoRa gateways in the given smart farm network. Each state in state space $\mathcal{S}_t$ is denoted by $s_i^t = \{\{\mathrm{re}_1^t, \mathrm{o}_1^t\}, \ldots, \{\mathrm{re}_j^t, \mathrm{o}_j^t\}\}$, where o_j^t is the opinion computed for animal j at time t, and re_j^t is a list of remaining energy on each sensor node close to sensor node j at time t.
- **Action space** ($\mathcal{A}_t$): Our proposed smart farm monitoring system aims to maximize the monitoring quality of animals' status as well as the lifetime of the monitoring system with healthy sensor nodes which are not compromised by attackers and not energy-depleted. To ensure sensor nodes are not being energy-depleted, we use a threshold ρ representing a percentage threshold of LESs to determine which sensor nodes have sufficient energy to transmit data. At each step, we first rank LESs by their remaining energy, and only the top ρ percentage of the LESs can send their request to nearby HESs. After the initial value of ρ is given, it can be adjusted over time by the DRL agents, which can identify an optimal ρ. Identifying an optimal ρ is critical because HES can transmit LES's data when the LES requests the data transmission to a nearby HES. We define the action space with three discrete actions at time t, $\mathcal{A}_t = \{\text{increase}, \text{decrease}, \text{stay}\}$ with the increment or decrement with τ (e.g., 0.05 where the ρ is scaled in $[0, 1]$ as a real number), in which the DRL agent can select an action in each step per update interval. A low ρ value means fewer sensor nodes can send out their data to gateways, resulting in lower monitoring quality but higher remaining energy. On the other hand, a high ρ value will result in high monitoring quality at the expense of a lower remaining energy level, which may shorten the system's lifetime.
- **Immediate reward** (r_t): The DRL agent will receive the immediate reward after taking action at time t, $r_t = \mathcal{MQ}(a_t) + \mathcal{RE}(a_t)$. $\mathcal{MQ}(a_t)$ is the monitoring quality at time t and $\mathcal{RE}(a_t)$ is sensor nodes' average remaining energy levels at time t (see Eqs. (2) and (4)).
- **Accumulated reward** ($\mathcal{R}_t$): The DRL agent will select an action to maximize the accumulated expected return, formulated by $\mathcal{R}_t = \sum_{t=0}^{T} \gamma^t r_t$, where r_t is a reward at time t, γ is a discount factor, and T is the period of an episode.

C. Transfer Learning for Robust Monitoring

To identify the source and target domains in our monitoring system, we run TL agents on the three LoRa gateways. Two LoRa gateways will each run a target TL agent, while one LoRa gateway will run a source TL agent. In applying the TL algorithm, we use the following protocol to determine *when-to-transfer* and *what-to-transfer*.

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2024.3409525

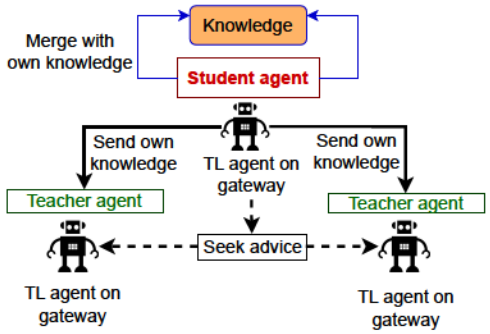


Fig. 4. Process of Transfer Learning.

1) **When-to-Transfer:** This operation determines how often a source agent transfers its knowledge to a target agent. For example, source agents can periodically transfer their knowledge about a sequence of states to target agents. This approach allows target agents to obtain additional knowledge from source domains. However, the proactive nature of periodic knowledge transfer may increase the computing time and overload target agents to determine which part of knowledge should be leveraged. For efficient knowledge transfer, it is critical to identify when the source agent has meaningful knowledge to share with a target agent.

We call an agent sharing knowledge *a teacher agent* and an agent receiving the knowledge *a student agent*. When a system is initialized or has not received any data due to poor environmental conditions, such as rainy or snowy days, no teacher agent may be available for knowledge transfer. To handle such situations, we will employ a pre-trained model with a set of policies learned by each agent. Before training our agents, each agent will select an optimal policy using a *semi-Markov decision process value learning* method [54]. In this method, an agent will follow each policy by taking a few steps further respectively, and choose a policy with the highest accumulated rewards. On the other hand, when the system operates normally by receiving data collected by solar sensors, we will leverage a *visited-based advising* method [55]. This approach will allow each agent to periodically ask other agents to provide advice based on their current states. At each time step, each student agent i will calculate the probability it will use a teacher model in another DRL agent j, denoted by P_s. Agent i will ask agent j's advice based on the frequency agent j has visited the current state s. This means teacher agent j has sufficiently visited state s. We define a probability that agent i becomes a student agent by:

$$P_i(s, \Upsilon_i) = (1 + \nu_i)^{-\Upsilon_i(s)}, \tag{5}$$

where ν_i is a scale factor and $\Upsilon_i(s)$ is a confidence function of current state s by the student model in agent i. We formulate $\Upsilon_i(s)$ by:

$$\Upsilon_i(s) = \sqrt{V_i(s)}, \tag{6}$$

where $V_i(s)$ is the total number of times the student agent i has visited state s. The teacher agent j will calculate the probability of being a teacher agent to the student agent i for current state s. Agent j will use a confidence function to determine if it can provide useful knowledge to other agents. The confidence function $\Psi_j(\cdot)$ representing that agent j becomes a teacher agent is calculated by:

$$\Psi_j(s) = \log_2 V_j(s), \tag{7}$$

Then, the probability that agent j provides advice to a student agent is formulated by:

$$P_j(s, \Psi) = 1 - (1 + \nu_j)^{-\Psi_j(s)}, \tag{8}$$

where ν_j is the scale factor. A larger ν_j means a higher probability that an agent provides its knowledge to other agents while it has high confidence in the current state s. Note that each agent can be a teacher agent and a student agent simultaneously in TL where the student agent can learn from the teacher agent to refine its model. Although an agent cannot manage its current state, it may learn something valuable for the current state other agents visit, as described in Fig. 4.

2) **What-to-Transfer:** After a source agent decides to transfer knowledge to target agents, we need to determine what knowledge to transfer that can be meaningful and valuable for the target agents' learning process. After the target agents receive knowledge, they will determine if they should accept or discard it. If the target agents have no prior knowledge, they simply accept the knowledge. Otherwise, the target agents will aggregate the received knowledge with their current knowledge. The key decision criterion to accept or discard received knowledge from the source agent is how much the state will be changed close to the goal state (i.e., a converged, optimal state). To this end, we employ the *trust region optimization method* [56] based on Kullback-Leibler (KL)-Divergence, which is a measure of minimizing the distance between source agents' policy distribution and target agents' current policy distribution. We use the KL-divergence of the two distributions as our loss function to minimize by:

$$\text{Minimize } \mathrm{KL}(\pi_s \parallel \pi_t) = \sum_{x \in X} \pi(x) \log \left(\frac{\pi_s(x)}{\pi_t(x)} \right), \tag{9}$$

where π_t is the policy of source agent, and π_s is the policy of target agent. We keep updating the target agent's policy, π_t, until the distance between these two policy distributions is sufficiently small. After the target agents learn the new policy, they should check if it leads to the converged state. It will take several steps to check if the state value approaches the convergence value. This guarantees that the TL agents will not cause negative transfers resulting in performance degradation due to transfer learning.

D. SL-based Opinion Formulation

At each gateway, the DRL agent runs and aggregates observations of animal conditions sent by solar sensors in the smart farm system. This section discusses how to aggregate the collected observations by considering the uncertainty caused by a lack of evidence and dissonance.

1) **Aggregation of Uncertain Observations:** Each record of animal conditions will be regarded as evidence to update information about a given animal's condition. The DRL agents will form an opinion about the animal's condition using a belief model, called *Subjective Logic* (SL), to explicitly deal

with multiple types of uncertainty. In SL, an agent A can form its opinion about a given proposition X, denoted by $\omega_X^A = \{b_X, u_X, a_X\}$, where b_X is belief masses distribution, u_X is the uncertainty mass, and a_X is the base rate (i.e., prior belief) distribution of variable X. The components satisfy the additivity requirement with $u_X + \sum b_X(x) = 1$. Each evidence of the animal's condition (e.g., temperature) will be recorded as one of the K classes of the range, e.g., for the temperature, $K = 3$, meaning that there are 3 classes of ranges: 37 or below as lower than normal, 38-41 as normal, 42 or above as higher than normal. In this case, without prior knowledge, we initialize the base rate equally for each belief mass, i.e., $a_X(x_i) = 1/K$ for any x_i.

2) **Opinions of Animal Conditions' Attributes:** We summarize each attribute of animal conditions in Table III, formulated using SL-based opinions in this work. When sensors transmit sensed data to gateways within their wireless radio range, the DRL agent in each gateway will formulate an opinion based on the received data from each sensor. For LESs, it is possible to request sending their data to multiple nearby HESs, resulting in a gateway with two opinions about the same animal. In this case, we need to conduct an aggregation of opinions using SL. When multiple sensors send their sensed data of the same animal, we will use a consensus operator, called the *cumulative fusion operator* [57], which combines two opinions, ω_X^A and ω_X^B, held by two different sources (i.e., different sensor nodes). The cumulative fusion opinion of ω_X^A and ω_X^B is denoted as $\omega_X^A \oplus \omega_X^B = \{b_X^A(x) \oplus b_X^B(x), u_X^A \oplus u_X^B, a_X^A(x) \oplus a_X^B(x)\}$ where

$$b_X^A(x) \oplus b_X^B(x) = \frac{b_X^A(x)u_X^B + b_X^B(x)u_X^A}{u_X^A + u_X^B - u_X^A u_X^B},$$
$$u_X^A \oplus u_X^B = \frac{u_X^A u_X^B}{u_X^A + u_X^B - u_X^A u_X^B}, \tag{10}$$
$$a_X^A(x) \oplus a_X^B(x) = \begin{cases} \frac{a_X^A(x)u_X^B + a_X^B(x)u_X^A - (a_X^A(x)+a_X^B(x))u_X^A u_X^B}{u_X^A + u_X^B - 2u_X^A u_X^B}, \\ \quad \text{if } u_X^A \neq 1 \vee u_X^B \neq 1, \\ \frac{a_X^A(x)+a_X^B(x)}{2} \qquad \text{if } u_X^A = u_X^B = 1, \end{cases}$$

in the case for $u_X^A \neq 0 \vee u_X^b \neq 0$.

3) **Dissonance Uncertainty in SL-based Opinions:** *Dissonance* indicates uncertainty due to conflicting evidence and is estimated based on the distance between different belief masses in a given multinomial opinion, ω_X and domain $\mathbb{X}$ by:

$$\dot{b}_X^{\text{Diss}} = \sum_{x_i \in \mathbb{X}} \left(\frac{b_X(x_i) \sum\limits_{x_j \in \mathbb{X} \setminus x_i} b_X(x_j) \text{Bal}(x_j, x_i)}{\sum\limits_{x_j \in \mathbb{X} \setminus x_i} b_X(x_j)} \right), \tag{11}$$

where the relative mass balance between a pair of belief masses $b_X(x_j)$ and $b_X(x_i)$ is expressed by:

$$\text{Bal}(x_j, x_i) = 1 - \frac{|b_X(x_j) - b_X(x_i)|}{b_X(x_j) + b_X(x_i)}. \tag{12}$$

The relative mass balance has its maximum at 1 when $b_X(x_j) = b_X(x_i)$. The relative mass balance has the minimum at 0 when one of the belief masses equals 0.

TABLE III
EVD DATASET DESCRIPTION

Metric	Description
Serial	A unique animal identifier
HR	Heart Rate of the animal
Average temperature	Average body temperature in Celsius
Min-temperature	Minimum temperature in Celsius
Max-temperature	Maximum temperature in Celsius
Average-activity	Average activity recorded by the number of steps taken
Battery-level	Residual battery life
Timestamp	Date and time of transmission

E. Uncertainty-Aware Monitoring Opinion Update

Whenever a gateway receives new evidence on an animal's condition, it needs to update the uncertain opinion ω_X^A about the corresponding animal. We will estimate two types of uncertainty: vacuity and dissonance. *Vacuity* refers to uncertainty caused by a lack of evidence, which is uncertainty mass, u_X, in an opinion. *Dissonance* is caused by conflicting evidence, estimated by Eq. (11).

After receiving a sufficient amount of evidence from sensors, the opinion update may be terminated because uncertainty is minimized. Since vacuity is zero or close to zero, no further significant update can be made. However, even with a sufficient amount of evidence, one may not be able to make a decision when the received evidence supports the two opposite beliefs (almost) equally. To make the opinion keep being updated by new evidence received and resolve the inconclusive opinion due to high dissonance, we use an uncertainty (vacuity) maximization technique [57] for the opinion to be updated by applying new evidence. Given opinion $\omega_X = (b_X, u_X, a_X)$ where $P_X = b_X + a_X \cdot u_X$ for a domain $\mathbb{X}$, the corresponding vacuity-maximized opinion is denoted by $\ddot{\omega}_X = (\ddot{b}_X, \ddot{u}_X, a_X)$ where $\ddot{u}_X$ and $\ddot{b}_X$ are given by:

$$\ddot{u}_X = \min_i \left[\frac{P_X(x_i)}{a_X(x_i)} \right], \tag{13}$$
$$\ddot{b}_X(x_i) = P_X(x_i) - a_X(x_i) \cdot \ddot{u}, \quad \text{for } x_i \in \mathbb{X}.$$

The vacuity maximization is performed with a threshold ϕ, which the above Eq. (13) will only be performed when $u_X < \phi$ where ϕ is sufficiently low (e.g., 0.05).

F. Detection of Deceptive Data

Adversarial attacks can poison data transmitted to gateways and thus can introduce conflicting evidence, resulting in degrading monitoring quality. To tackle this problem, we introduce a measure of the *degree of conflict* (DC) in SL [57] to identify suspicious data gateways received from compromised sensors (i.e., compromised HESs or LESs) performing false data injection. Specifically, we use projected distance (PD) [57] to measure the difference between a given opinion, ω_X^A, and other opinions formulated by:

$$\widehat{PD}(\omega_X^A) = \frac{\sum_{j \in B} PD(\omega_X^A, \omega_X^j)}{|B|}, \tag{14}$$
$$\text{where } PD(\omega_X^A, \omega_X^j) = \frac{\sum_{x \in \mathbb{X}} | \omega_X^A(x) - \omega_X^j(x) |}{2},$$

where ω_X^A and ω_X^j (where $j \in B$) are opinions of agent A and j, respectively, for proposition X. Each gateway will update its opinion every time receiving sensed data from sensor nodes. After updating, we compute the DC based on $\widehat{PD}(\omega_X^A, \omega_X^B)$. If the estimated PD exceeds threshold $\phi \in [0, 1]$, we consider the sensed data from A suspicious and possibly modified because A is compromised or the data is modified or forged by external attackers. We keep track of the number of times the sensed data by each sensor node is discarded as the ratio to the total number of data transmission times. Then, we use this record to detect whether the node is compromised. If the percentage of discarding for a node exceeds a threshold ξ, we consider this node compromised.

VI. Experimental Setup

A. Parameterization

The farm is a square area of 40 acres (i.e., ~160K square meters) with each side of length 400 meters. The farm has 20 cows and is fully covered by three gateways. We obtained real-world datasets from a smart farm operated by Virginia Tech's College of Agriculture and Life Sciences to conduct our simulations [6]. These datasets were collected from various devices, including EmbediVet Implantable Temperature Devices (EVD), Halter Sensors, Heart Rate Sensors, and Implantable Temperature Sensors, whose attributes are summarized in Table III. Each DRL agent running on a gateway will hold an opinion for each cow's condition attributes, including heart rate, temperature, and activity. We define three beliefs for each attribute: lower than normal, normal, or higher than normal. For a healthy cow, the normal ranges of its temperature, heart rate, and moving activity are given respectively as $[37.8, 39.2]$ Celsius, $[48, 84]$ beats per minute, and $[1, 2]$ meters per sec. Since there may be some sick cows on the farm, their attributes may not be in the normal range of the healthy cows. For a sick cow with Bovine Viral Diarrhoea, the most common disease in cows, its temperature range is $[40.0, +\infty]$ Celsius as the sign of disease. Other attributes have no big difference from the normal ranges. We also use P_{mv}^i for cow i's moving probability. We assume cows move in a random pattern with a normal distribution of their speeds with an average of 1.5 m/s and a standard deviation of 0.1 m/s. The whole simulation is considered a 24-hour monitoring period. Each gateway selects an action to identify the optimal number of LES to send data with the interval $T_a = 60\ sec$. We assume there are 5 HES with initial energy level 1 and 15 LES with random initial battery levels in $[0, E_{init}^{LES}]$ in the monitoring area. Table IV summarizes the key design parameters, meanings, and default values used for our experiments.

B. Energy Consumption in LoRa Gateway and BLE

In the considered wireless solar sensor-based smart farm environment, there are two conditions: a sensor node transmits sensed data either regularly to LoRa gateways or a nearby node via Bluetooth Low Energy (BLE). The LoRa protocol is deployed for long-distance communication, usually with a distance of 5 to 15 km and a data transfer speed of 27 $kbps$ [43]. By comparison, the BLE protocol is for short-distance communication with a distance of 100 meters and a transfer speed of 2 $Mbps$. Regarding energy consumption, the LoRa radio of SAM R34/35 consumes 170 mW for transmitting data while BLE radio dissipates $11mW$. In this case, transmission for one-bit data through LoRa radio consumes 1,100 times more energy than through BLE radio [58]. A fully charged sensor node has 5 kW as the initial energy level, and the efficiency of solar power under outdoor light and indoor light is around $10mW/cm^2$ and $0.1mW/cm^2$, respectively.

TABLE IV
Key Design Parameters, Their Meanings, and Default Values

Notation	Meaning	Value
n	Total number of sensors(cows)	20
P_{mv}^1	Probability of cow i to move	[0.3,0.7]
ρ	Percentage of the LESs can send data	0.8
τ	Adjust step size when an agent takes action	0.1
ϕ	Acceptable degree of conflict	0.3
ξ	Threshold determining if a node is compromised	0.6
ζ_1	Probability determining if a node should be repaired	0.3
δ	Threshold counting the number of rejected data to detect a compromised node	5
P_A	Probability for an attacker or a compromised node to perform a certain attack	0.3
P_{AE}	Probability for a compromised gateway to perform a certain attack	0.3
T_u	Time interval for a sensor to send sensed data	30 s
T_a	Time interval for an agent to select an action	60 s
L_{bl}	Threshold determining if a sensor is LES	0.3
E_{init}^{LES}	Initial energy level of low energy sensors	[0.1,0.2]

Specifically, we simulate the charging behavior of our solar-powered sensors as follows: we define $P(x, y, t)$ as the probability that a sensor located at (x, y) charges at time t. This probability is calculated using a quadratic form $P(x, y, t) = \max[0, -\frac{1}{6}(t - t_{xy})^2 + 1]$, where t_{xy} is the time corresponding to the location (x, y), calculated as $t_{xy} = \frac{t_0}{d}(x - \frac{d}{2}) + 12$. Here, t_0 is a hyperparameter, d represents the length of the farm's operation area, and both t and t_{xy} are scaled within the 24-hour day range. To incorporate weather conditions into our simulation, we modify the charging probability by applying the sun exposure rate α, which ranges from 0 (overcast) to 1 (sunny). The final charging probability is then given by $\alpha P(x, y, t)$, influencing the policy determined by the DRL agents based on varying animal locations.

C. Metrics

We use the following **metrics** for our experiments.

- **Accumulated Reward** ($\mathcal{R}$): This represents the sum of accumulated reward in all simulation runs based on our discussion in Section V-B.
- **Remaining Energy** ($\mathcal{RE}$): This measures the degree of remaining energy in LESs per time interval, T_u. At each time interval, there are four possible events on sensors consuming energy: data transmission through LoRa ($\mathcal{E}_{\text{SG}}$) and BLE ($\mathcal{E}_{\text{SS}}$), and, energy drained in active mode ($\mathcal{E}_{\text{active}}$) and sleep mode ($\mathcal{E}_{\text{sleep}}$). $\mathcal{RE}$ is calculated by Eq. (4).
- **Model Convergence Time** (C_T): This measures the time spent from the beginning of the training process to the time

the loss function reaches its minimum. This is estimated by: $C_T = T_c - T_b$, where T_c is the time the model reaches a convergence, and T_b is the starting time of the training process. We define T_c as the time when the immediate reward settles to within a range $[-1, +1]$ around the final value for at least 10 episodes.

- **Monitoring Quality** ($\mathcal{MQ}$): This estimates the monitoring quality during the system's operating times, where this metric indicates monitoring accuracy based on the amount of true sensed data received. It is given as Eq. (2) in Section III.

D. Comparing Schemes

We consider the following schemes for evaluation:

- **Deep Q-Network (DQN)** [3]: DRL agents select the best action from the learned Q-table. In the multi-agent environment, each agent learns its own local Q-function.
- **Proximal Policy Optimization (PPO)** [59]: DRL agents select the optimal actions based on learned policy. The PPO uses an actor-critic style algorithm deploying multiple echos of stochastic gradient ascent to update the policy.
- **TL-DQN**: DQN DRL agents obtain knowledge from their learning and other agents' experience using our proposed TL to learn their local Q-function.
- **TL-PPO**: PPO DRL agents learn knowledge from each other using our proposed TL to update their policy.
- **TFT (TL with Fine-Tuning)-PPO** [60]: PPO DRL agents learn hyper-parameters from other agents via fine-tuning.
- **Adaptive-Energy-Distance (AED)** [14]: Agents select an action to achieve both a high remaining energy level ($\mathcal{RE}$) and low total transmission distance ($\mathcal{D}$) for reducing the data acquisition latency efficiently. The ($\mathcal{D}$) is defined by the total distance between each LES and its nearby HES, $\mathcal{D} = \sum_i d(p_i, q)$, where p_i is the position of a LES i, and q is the position of LES i's nearby HES. The agents will select an action with the maximum value of $\mathcal{RE} - \mathcal{D}$.
- **Random**: Agents will randomly select an action from the action space at each step.
- **Fixed-energy (FE)**: Agents will choose top 30% of sensor nodes with least remaining energy.

As discussed in Section II, Alemayehu and Kim [14] proposed an energy-efficient monitoring mechanism called AED for a wireless sensor network whose aim aligns well with our work in terms of energy preservation and latency reduction. Thus, we chose AED [14] as a state-of-the-art counterpart scheme for performance comparison. Altogether, our experiments conduct a comparative performance analysis of our proposed DRL-based schemes (i.e., DQN, PPO, TL-DQM, TL-PPO, and TFT-PPO) against the state-of-the-art heuristic-based (i.e., AED) and baseline (i.e., Random, FE) schemes.

VII. Numerical Results & Analysis

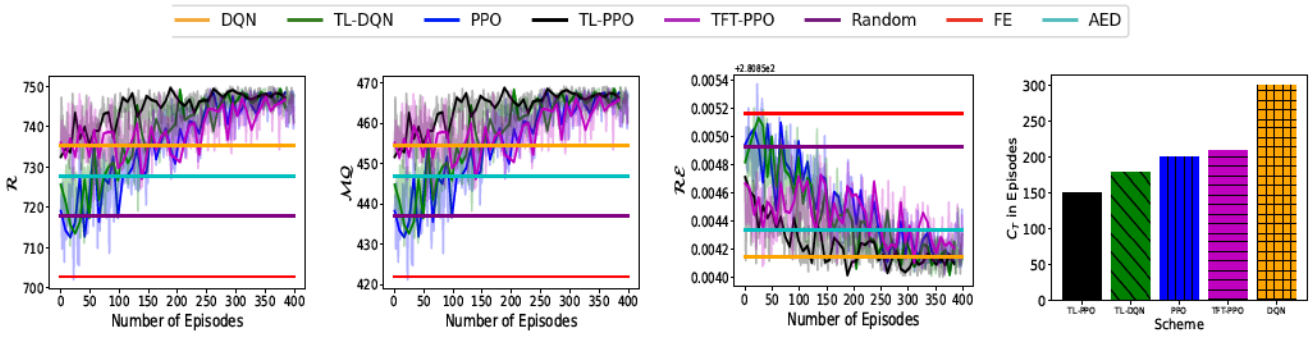
We run 100 simulations to evaluate the performance of the proposed TL-based schemes and the 4 baseline schemes based on the parameter settings described in Section VI-A. Each data point represents the average results obtained from the 100 simulation runs. For PPO, TL-PPO, and TFT-PPO, we set the batch size and learning rate to 500 and 0.008, respectively. For DQN and TL-DQN, we set the batch size and learning rate to 500 and 0.02, respectively. These hyperparameter settings are chosen based on each scheme's optimal performance.

A. Comparative Performance Analysis During Training Time

Fig. 5 demonstrates the DRL training process of the eight schemes in Section VI-D with $P_A = 0.1$, as default. The training curve through episodes for the Random, FE, and AED schemes is a horizontal line, with the converging time being zero since they do not have a learning process. One thing that is noteworthy is that FE performs worse than Random since FE does not take any action at each step, while Random has a probability of approximately 0.3 to take the best action. Assuming the system's inherent vulnerability with a default presence of attackers (encompassing 30% of sensors compromised and an attack severity level of 0.1), the superior performance of our DRL-based approaches underlines the system's robustness and adaptability. This adaptability is crucial for adjusting to and determining the energy policy that ensures optimal monitoring quality and energy conservation amidst adversarial conditions. The scheme's ability to converge despite adversarial attacks further emphasizes its resilience. We also observe that PPO-based schemes outperform DQN-based schemes, respectively, for metrics accumulated reward ($\mathcal{R}$) in Fig. 5(a), monitoring quality ($\mathcal{MQ}$) in Fig. 5(b), and model convergence time (C_T) in Fig. 5(d), while it is just the opposite in the remaining energy ($\mathcal{RE}$) in Fig. 5(c). This is because our objective function has two conflicting objectives (i.e., more energy-consuming operations can lead to higher monitoring quality and vice-versa), leading to different policies. In the given environment, the increase of monitoring quality by receiving data from a sensor node is much larger than the cost of that sensor to send data. For example, TL-PPO has the lowest remaining energy while outperforming other schemes in accumulated reward. In addition, TL-PPO performs better than TFT-PPO and PPO, while TL-DQN performs better than DQN because leveraging TL effectively accelerates the training process of DRL agents. This proves TL can significantly contribute to increased performance in $\mathcal{R}$, $\mathcal{MQ}$, and $\mathcal{RE}$. The overall performance order of the proposed schemes is TL-PPO $\geq$ TFT-PPO $\geq$ PPO $\geq$ AED $\approx$ TL-DQN $\geq$ DQN $\geq$ Random $\geq$ FE. Fig. 5(d) shows that the convergence time order of the proposed scheme is Random = FE = AED $\leq$ TL-PPO $\leq$ TL-DQN $\leq$ PPO $\leq$ DQN. Overall, PPO performs better than DQN because PPO can directly learn from the environment, making the smart farm system highly adaptive to ensure its performance and security under non-stationary settings. This is also the reason why the rule-based approach results in a similar approach to the DQN-based approach, but worse performance than the PPO-based. It has a lack of ability to deal with uncertainty and to update rules to accommodate new scenarios.

B. Sensitivity Analyses

1) **Effect of Varying Attack Severity (P_A) on Sensors:** Fig. 6 shows the effect of varying attack severity (P_A) on the performance metrics in Section VI-C. Higher P_A leads



(a) Accumulated Reward ($\mathcal{R}$) (b) Monitoring Quality ($\mathcal{MQ}$) (c) Remaining Energy ($\mathcal{RE}$) (d) Convergence Time (C_T)

Fig. 5. Comparative performance analysis during training time with $P_A = 0.1$.

to decreasing monitoring quality ($\mathcal{MQ}$) because Higher P_A introduces more compromised data. In addition, higher P_A introduces higher remaining energy ($\mathcal{RE}$) for all schemes because sending more data may introduce the increased compromised data. Therefore, less energy is consumed considering the utility of data transmission. DRL approaches using TL (i.e., TL-DQN and TL-PPO) outperform their counterparts (i.e., DQN and PPO). Under varying P_A, the overall performance is observed in the following order: TL-PPO ≥ TFT-PPO ≥ PPO ≥ AED ≈ TL-DQN ≥ DQN ≥ Random ≥ FE.

2) **Effect of Varying Attack Severity (P_{AE}) on Gateways:** Fig. 7 shows how the varying attack level (P_{AE}) performed on gateways affects the system performance in terms of the performance metrics. For this performance analysis, we only consider DRL-based approaches because these attacks (see the attacks under compromised LG in Table II) are performed in disrupting DRL operations. Higher P_{AE} results in lowering accumulated reward ($\mathcal{R}$) and monitoring quality ($\mathcal{MQ}$) while increasing remaining energy ($\mathcal{RE}$). This is because the poisoned data by this adversarial attack can mislead DRL agents to use a different policy which may not be optimal to its true state. We notice the critical impact of P_{AE} on the convergence time (C_T) for DQN and TL-DQN. DQN's C_T is the shortest among all under attacks while the converged reward is not optimal, showing the lowest $\mathcal{R}$. The reason is that the attacks on gateways greatly influence the DRL agent's exploration ability of DQN. Once the DQN agent is misled to an undesired state, it cannot find an optimal state again. Thus, when attacks are performed in the DQN training process, even though the training curve converges fast, it may not converge to an optimal state. With respect to varying P_{AE}, the overall performance is observed in the following performance order: TL-PPO ≥ PPO ≥ TL-DQN ≥ DQN.

3) **Effect of Varying Cyber Attack (P_A) and Adversarial Example (P_{AE}) Severity:** Fig. 8 demonstrates the effects on the performance of DRL-based schemes while varying both P_A and P_{AE}. We observe more severe attacks lead to lower accumulated rewards ($\mathcal{R}$) in Fig. 8(a) and monitoring quality ($\mathcal{MQ}$) in Fig. 8(b), and higher remaining energy level ($\mathcal{RE}$) in Fig. 8(c) and the convergence time (C_T) in Fig. 8(d) for all DRL schemes. Regarding accumulated rewards, the TFT-PPO and TL-PPO have almost the same performance since they are both PPO-based DRL schemes with different TL strategies having the potential to learn similar policies. Concerning varying both P_A and P_{AE}, the overall performance of the proposed schemes is ordered as TL-PPO ≥ TFT-PPO ≥ PPO ≈ TL-DQN ≥ DQN.

TABLE V
ALGORITHMIC ASYMPTOTIC COMPLEXITY ANALYSIS

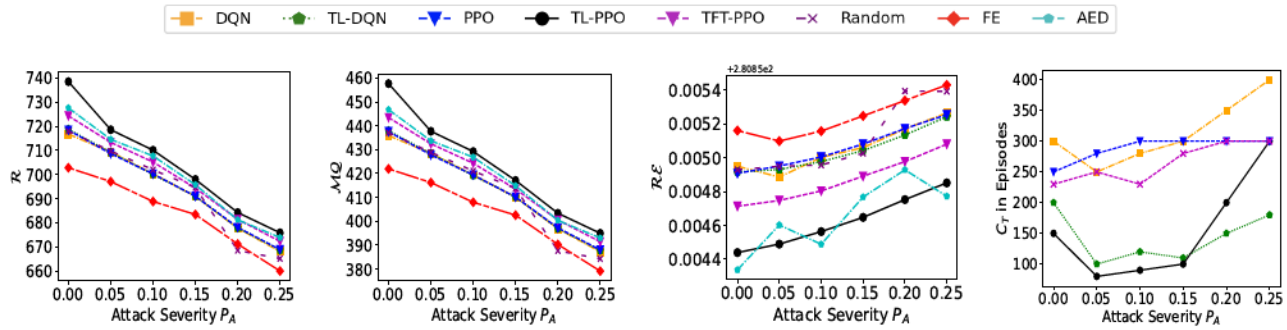
Scheme	Complexity
TL-PPO	$O(TL[n_e] \times T_{ppo} \times n_s)$
TL-DQN	$O(TL[n_e] \times T_{dqn} \times n_s)$
TFT-PPO	$O(TFT[n_e] \times T_{ppo} \times n_s)$
PPO	$O(n_e \times T_{ppo} \times n_s)$
DQN	$O(n_e \times T_{dqn} \times n_s)$
FE	$O(n_s)$
Random	$O(n_s)$
AED	$O(n_s)$

4) **Effect of the Varying Initial Energy Level of LESs (E_{init}^{LES}):** Fig. 9 shows how sensors' different initial energy levels (E_{init}^{LES}) impact the system's performance in the four metrics. As shown in Figs. 9(a)-(b), higher E_{init}^{LES} leads to a higher $\mathcal{R}$ and $\mathcal{MQ}$ because more energy is available for sensor nodes to send data. The remaining energy ($\mathcal{RE}$) is the least sensitive to the changes in E_{init}^{LES}, as shown in Fig. 9(c). Although the upper and lower bounds of the initial energy level are different, the range size is the same in the analysis. With the same range size, different DRL schemes will determine a very similar policy, showing almost the same performance. The convergence time with an initial energy level interval $(0.05, 0.15]$ is much higher than others because, with a low energy level, a policy preferred not to send data can be determined. DRL agents obtain fewer data per step, which requires more training episodes to converge.

Our analysis, which adjusts for variations in attack severity and the system's energy levels, consistently shows our proposed schemes outperforming both baseline and heuristic models. This underscores our system's proficiency in adapting the optimal policy to uphold normal operations even under severe and challenging conditions.

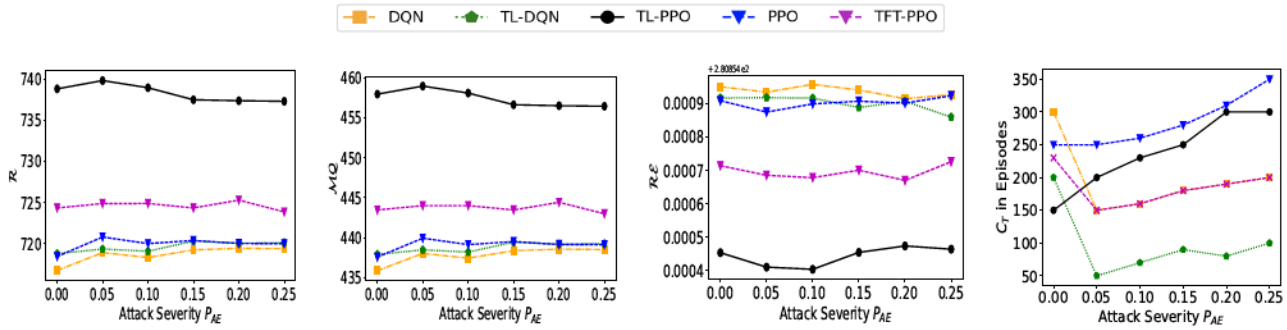
C. Algorithmic Asymptotic Complexity Analysis

Table V summarizes the algorithmic asymptotic complexity where n_e is the number of episodes in the training process, $TL[n_e]$ and $TFT[n_e]$ means the number of episodes using TL and TFT. Since TL improves the efficiency and performance of both PPO and DQN, it requires fewer episodes to converge.



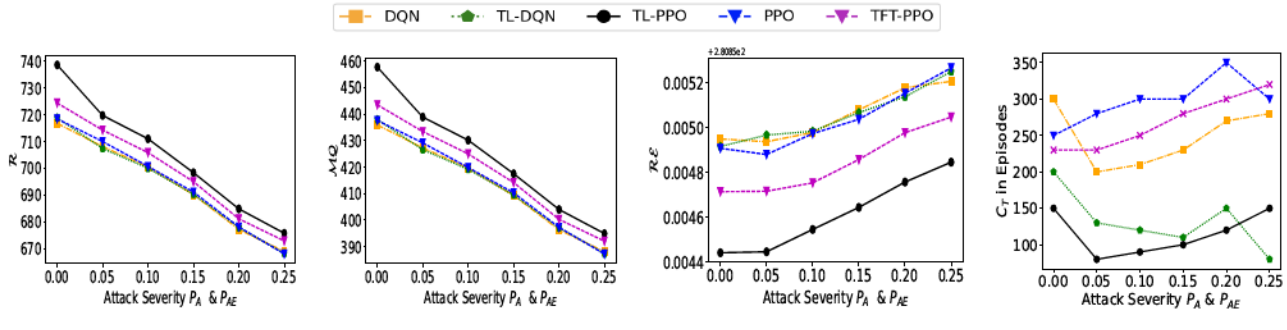
(a) Accumulated Reward ($\mathcal{R}$) (b) Monitoring Quality ($\mathcal{MQ}$) (c) Remaining Energy ($\mathcal{RE}$) (d) Convergence Time (C_T)

Fig. 6. Effect of Varying Attack Severity (P_A) on Solar Sensors.



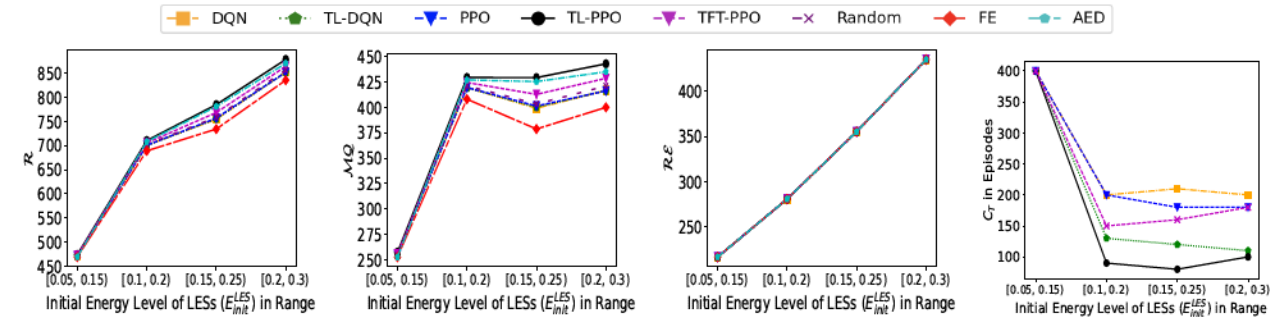
(a) Accumulated Reward ($\mathcal{R}$) (b) Monitoring Quality ($\mathcal{MQ}$) (c) Remaining Energy ($\mathcal{RE}$) (d) Convergence Time (C_T)

Fig. 7. Effect of Varying Adversarial Example Severity (P_{AE}) on Gateways.



(a) Accumulated Reward ($\mathcal{R}$) (b) Monitoring Quality ($\mathcal{MQ}$) (c) Remaining Energy ($\mathcal{RE}$) (d) Convergence Time (C_T)

Fig. 8. Effect of Varying Cyber Attack (P_A) and Adversarial Example (P_{AE}) Severity.



(a) Accumulated Reward ($\mathcal{R}$) (b) Monitoring Quality ($\mathcal{MQ}$) (c) Remaining Energy ($\mathcal{RE}$) (d) Convergence Time (C_T)

Fig. 9. Effect of the Varying Initial Energy Level of LESs (E_{init}^{LES}).

Thus, when TL is used, we use $TL[n_e]$ and $TFT[n_e]$ to distinguish it from n_e with $TL[n_e] \ll n_e \ll TFT[n_e]$, thus

manifesting the effectiveness of TL implementation. The T_{ppo} and T_{dqn} are the training times per episode using the PPO algorithm and DQN algorithm, respectively. The n_s is the number of simulation times. It is proven that the PPO is more resilient in a complex environment, in which $T_{ppo} \ll T_{dqn}$. FE, Random, and AED algorithms only depend on n_s, showing the highest efficiency among all, while their performance is the worst among all, as demonstrated in Figs. 5, 6, and 9.

VIII. Conclusion & Future Work

The advent of IoT technologies has spurred extensive research within smart environments, with a significant focus on enhancing monitoring systems through reduced communication costs. Contrasting with these studies, our research introduces an energy-adaptive and attack-resilient methodology tailored for smart farming. This approach ensures the system's energy sufficiency and high monitoring performance, even amidst cyberattacks and energy variability. Our empirical findings underscore the efficacy of our strategy, demonstrating the system's capability to sustain normal operations against both cyber threats and environmental challenges.

We obtained the following **key findings** from this work: (1) Our TL-based DRL agents tend to trade energy consumption for achieving high monitoring quality as the optimal policy in data transmission, especially when the environment is highly dynamic and hostile. This is because the benefit of high monitoring quality with fresh data often outweighs the energy consumption cost for data transmission. (2) Our proposed TL-PPO scheme (with PPO as the scheme for DRL) has the best performance in finding the optimal policy that maximizes the monitoring quality while preserving energy. (3) Using TL in DRL reduced the training time significantly for both PPO and DQN, as it takes much fewer training episodes for learning convergence. (4) Our proposed TL-PPO scheme is robust and outperforms all state-of-the-art counterpart schemes in the presence of adversarial attacks.

For **future work**, we plan to explore the following research directions: (1) We will further improve the proposed TL-based DRL scheme to identify and prevent negative transfer in the learning process to see if agents can learn a better policy. (2) We will investigate the scalability issue of deploying our proposed TL-based DRL scheme with more DRL agents in a large, dynamic, and complex smart farm system.

Acknowledgement

This work is partly funded by NSF Grant 2106987 and 2107450, the Commonwealth Cyber Initiative (CCI), and Virginia Tech's ICTAS EFO Opportunity Seed Investment Grant.

References

[1] OECD, Food, and A. O. of the United Nations, *Meat*, 2022. [Online]. Available: https://www.oecd-ilibrary.org/content/component/ab129327-en

[2] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[3] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[4] W.-H. Chen and F. You, "Sustainable building climate control with renewable energy sources using nonlinear model predictive control," *Renewable and Sustainable Energy Reviews*, vol. 168, p. 112830, 2022.

[5] D. Mayne, "Model predictive control theory and design," *Nob Hill Pub, Llc*, 1999.

[6] Center for Advanced Innovation in Agriculture (CAIA). (2023) Virginia Tech SmartFarm Innovation Network (TM). [Online]. Available: https://caia.cals.vt.edu/caia-s-research-platforms/vtsmartfarm.html

[7] S. Kumar, G. Chowdhary, V. Udutalapally, D. Das, and S. P. Mohanty, "gCrop: Internet-of-Leaf-Things (IoLT) for monitoring of the growth of crops in smart agriculture," in *2019 IEEE International Symposium on Smart Electronic Systems (iSES) (Formerly iNiS)*, 2019, pp. 53–56.

[8] M. Liu, S. Xu, and S. Sun, "An agent-assisted qos-based routing algorithm for wireless sensor networks," *Journal of Network and Computer Applications*, vol. 35, no. 1, pp. 29–36, 2012, collaborative Computing and Applications. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1084804511000853

[9] M. Gupta, M. Abdelsalam, S. Khorsandroo, and S. Mittal, "Security and privacy in smart farming: Challenges and opportunities," *IEEE Access*, vol. 8, pp. 34 564–34 584, 2020.

[10] Y. K. Saheed and M. O. Arowolo, "Efficient cyber attack detection on the internet of medical things-smart environment based on deep recurrent neural network and machine learning algorithms," *IEEE Access*, vol. 9, pp. 161 546–161 554, 2021.

[11] C.-J. Chae and H.-J. Cho, "Enhanced secure device authentication algorithm in P2P-based smart farm system," *Peer-to-peer networking and applications*, vol. 11, pp. 1230–1239, 2018.

[12] A. A. Aliyu and J. Liu, "Blockchain-based smart farm security framework for the Internet of Things," *Sensors*, vol. 23, no. 18, p. 7992, 2023.

[13] A. Vangala, A. K. Sutrala, A. K. Das, and M. Jo, "Smart contract-based blockchain-envisioned authentication scheme for smart farming," *IEEE Internet of Things Journal*, vol. 8, no. 13, pp. 10 792–10 806, 2021.

[14] T. S. Alemayehu and J.-H. Kim, "Efficient nearest neighbor heuristic TSP algorithms for reducing data acquisition latency of UAV relay WSN," *Wireless Personal Communications*, vol. 95, pp. 3271–3285, 2017.

[15] F. Akhter, H. R. Siddiquei, M. E. E. Alahi, K. P. Jayasundera, and S. C. Mukhopadhyay, "An iot-enabled portable water quality monitoring system with mwcnt/pdms multifunctional sensor for agricultural applications," *IEEE Internet of Things Journal*, vol. 9, no. 16, pp. 14 307–14 316, 2021.

[16] S. M. Nagarajan, G. G. Deverajan, P. Chatterjee, W. Alnumay, and V. Muthukumaran, "Integration of iot based routing process for food supply chain management in sustainable smart cities," *Sustainable Cities and Society*, vol. 76, p. 103448, 2022.

[17] S. Aggarwal and S. Sharma, "Voice based deep learning enabled user interface design for smart home application system," in *2021 2nd International Conference on Communication, Computing and Industry 4.0 (C2I4)*. IEEE, 2021, pp. 1–6.

[18] Y. Fan, L. Zhang, D. Li, and Z. Wang, "Progress in self-powered, multi-parameter, micro sensor technologies for power metaverse and smart grids," *Nano Energy*, p. 108959, 2023.

[19] Y. Li, C. Liu, H. Zou, L. Che, P. Sun, J. Yan, W. Liu, Z. Xu, W. Yang, L. Dong *et al.*, "Integrated wearable smart sensor system for real-time multi-parameter respiration health monitoring," *Cell Reports Physical Science*, vol. 4, no. 1, p. 101191, 2023.

[20] C. Catalano, L. Paiano, F. Calabrese, M. Cataldo, L. Mancarella, and F. Tommasi, "Anomaly detection in smart agriculture systems," *Computers in Industry*, vol. 143, p. 103750, 2022.

[21] T. M. Ghazal, N. A. Al-Dmour, R. A. Said, A. Omidvar, U. Y. Khan, T. R. Soomro, H. M. Alzoubi, M. Alshurideh, T. M. Abdellatif, A. Moubayed *et al.*, "Ddos intrusion detection with ensemble stream mining for iot smart sensing devices," in *The Effect of Information Technology on Business and Marketing Intelligence Systems*. Springer, 2023, pp. 1987–2012.

[22] T. Saba, K. Haseeb, I. Ahmed, and A. Rehman, "Secure and energy-efficient framework using internet of medical things for e-healthcare," *Journal of Infection and Public Health*, vol. 13, no. 10, pp. 1567–1575, 2020.

[23] U. K. Lilhore, O. I. Khalaf, S. Simaiya, C. A. Tavera Romero, G. M. Abdulsahib, and D. Kumar, "A depth-controlled and energy-efficient routing protocol for underwater wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 18, no. 9, p. 15501329221117118, 2022.

[24] X. Zhuo, M. Liu, Y. Wei, G. Yu, F. Qu, and R. Sun, "Auv-aided energy-efficient data collection in underwater acoustic sensor networks," *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 10 010–10 022, 2020.

[25] S. Bharany, S. Sharma, N. Alsharabi, E. Tag Eldin, and N. A. Ghamry, "Energy-efficient clustering protocol for underwater wireless sensor networks using optimized glowworm swarm optimization," *Frontiers in Marine Science*, vol. 10, p. 1117787, 2023.

[26] K. Haseeb, I. Ud Din, A. Almogren, and N. Islam, "An energy efficient and secure iot-based wsn framework: An application to smart agriculture," *Sensors*, vol. 20, no. 7, p. 2081, 2020.

[27] R. Kocherla, M. Chandra Sekhar, and R. Vatambeti, "Enhancing the energy efficiency for prolonging the network life time in multi-conditional multi-sensor based wireless sensor network," *Journal of Control and Decision*, vol. 10, no. 1, pp. 72–81, 2023.

[28] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[29] W. Dai, Q. Yang, G.-R. Xue, and Y. Yu, "Boosting for transfer learning," in *Proceedings of the 24th International Conference on Machine Learning*, ser. ICML '07. New York, NY, USA: Association for Computing Machinery, 2007, pp. 193–200. [Online]. Available: https://doi.org/10.1145/1273496.1273521

[30] A. Argyriou, T. Evgeniou, and M. Pontil, "Multi-task feature learning," in *Advances in Neural Information Processing Systems*, B. Schölkopf, J. Platt, and T. Hoffman, Eds., vol. 19. MIT Press, 2006.

[31] T. L. M. van Kasteren, G. Englebienne, and B. J. A. Kröse, "Transferring knowledge of activity recognition across sensor networks," in *Pervasive Computing*, P. Floréen, A. Krüger, and M. Spasojevic, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 283–300.

[32] L. Mihalkova, T. Huynh, and R. J. Mooney, "Mapping and revising markov logic networks for transfer learning," in *Proceedings of the 22nd National Conference on Artificial Intelligence - Volume 1*, ser. AAAI'07. AAAI Press, 2007, p. 608–614.

[33] N. Agarwal, A. Sondhi, K. Chopra, and G. Singh, "Transfer learning: Survey and classification," *Proceedings of ICSICCS 2020 Smart Innovations in Communication and Computational Sciences*, pp. 145–155, 2021.

[34] W. Dai, G.-R. Xue, Q. Yang, and Y. Yu, "Transferring naive bayes classifiers for text classification," in *Proceedings of the 22nd National Conference on Artificial Intelligence - Volume 1*, ser. AAAI'07. AAAI Press, 2007, pp. 540–545.

[35] Y. Zhang, H. Wang, D. Zhang, and D. Wang, "Deeprisk: A deep transfer learning approach to migratable traffic risk estimation in intelligent transportation using social sensing," in *2019 15th International Conference on Distributed Computing in Sensor Systems (DCOSS)*, 2019, pp. 123–130.

[36] Y. Song and C. Zhang, "Transferred dimensionality reduction," 09 2008, pp. 550–565.

[37] D. Coraci, S. Brandi, T. Hong, and A. Capozzoli, "Online transfer learning strategy for enhancing the scalability and deployment of deep reinforcement learning control in smart buildings," *Applied Energy*, vol. 333, p. 120598, 2023.

[38] S. Gamrian and Y. Goldberg, "Transfer learning for related reinforcement learning tasks via image-to-image translation," in *International conference on machine learning*. PMLR, 2019, pp. 2063–2072.

[39] A. Anwar and A. Raychowdhury, "Autonomous navigation via deep reinforcement learning for resource constraint edge nodes using transfer learning," *IEEE Access*, vol. 8, pp. 26 549–26 560, 2020.

[40] Z. Ke, Z. Li, Z. Cao, and P. Liu, "Enhancing transferability of deep reinforcement learning-based variable speed limit control using transfer learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 7, pp. 4684–4695, 2020.

[41] G. Ciabatti, S. Daftry, and R. Capobianco, "Autonomous planetary landing via deep reinforcement learning and transfer learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 2031–2038.

[42] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, "Deep reinforcement learning: A brief survey," *IEEE Signal Processing Magazine*, vol. 34, no. 6, pp. 26–38, 2017.

[43] *SAM R34/R35 Low Power LoRa® Sub-GHz SiP Datasheet*, Microchip, 2018. [Online]. Available: http://ww1.microchip.com/downloads/en/DeviceDoc/SAMR34-R35-Low-Power-/LoRa-Sub-GHz-SiP-Data-Sheet-DS70005356B.pdf

[44] F. Hessel, L. Almon, and M. Hollick, "Lorawan security: An evolvable survey on vulnerabilities, attacks and their systematic mitigation," *ACM Trans. Sen. Netw.*, vol. 18, no. 4, mar 2023. [Online]. Available: https://doi.org/10.1145/3561973

[45] M. A. Rahman and H. Mohsenian-Rad, "False data injection attacks with incomplete information against smart power grids," in *2012 IEEE Global Communications Conference (GLOBECOM)*, 2012, pp. 3153–3158.

[46] J. Ophoff and K. Renaud, "Revealing the cyber security non-compliance" attribution gulf"," 2021.

[47] S. Sontowski, M. Gupta, S. S. Laya Chukkapalli, M. Abdelsalam, S. Mittal, A. Joshi, and R. Sandhu, "Cyber attacks on smart farming infrastructure," in *2020 IEEE 6th International Conference on Collaboration and Internet Computing (CIC)*, 2020, pp. 135–143.

[48] Y. Liu, S. Ma, Y. Aafer, W.-C. Lee, J. Zhai, W. Wang, and X. Zhang, "Trojaning attack on neural networks," in *Proceedings 2018 Network and Distributed System Security Symposium*. San Diego, CA: Internet Society, 2018.

[49] I. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *International Conference on Learning Representations*, 2015. [Online]. Available: http://arxiv.org/abs/1412.6572

[50] M. S. Ayas, S. Ayas, and S. M. Djouadi, "Projected gradient descent adversarial attack and its defense on a fault diagnosis system," in *2022 45th International Conference on Telecommunications and Signal Processing (TSP)*, 2022, pp. 36–39.

[51] M. Nasreen, A. Ganesh, and C. Sunitha, "A study on byzantine fault tolerance methods in distributed networks," *Procedia Computer Science*, vol. 87, pp. 50–54, 2016.

[52] S. Yoon, J.-H. Cho, G. Dixit, and I.-R. Chen, "Resource-aware intrusion response based on deep reinforcement learning for software-defined Internet-of-Battle-Things," *Game Theory and Machine Learning for Cyber Security*, pp. 389–409, 2021.

[53] A. Pattanaik, Z. Tang, S. Liu, G. Bommannan, and G. Chowdhary, "Robust deep reinforcement learning with adversarial attacks," in *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, ser. AAMAS '18. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems, 2018, p. 2040–2042.

[54] R. S. Sutton, D. Precup, and S. Singh, "Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning," *Artificial Intelligence*, vol. 112, no. 1, pp. 181–211, 1999. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0004370299000521

[55] F. L. da Silva, R. Glatt, and A. H. R. Costa, "Simultaneously learning and advising in multiagent reinforcement learning," in *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*, ser. AAMAS '17. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems, 2017, p. 1100–1108.

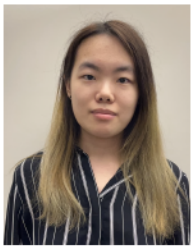
[56] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust region policy optimization," in *Proceedings of the 32nd International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, F. Bach and D. Blei, Eds., vol. 37. Lille, France: PMLR, 07–09 Jul 2015, pp. 1889–1897. [Online]. Available: https://proceedings.mlr.press/v37/schulman15.html

[57] A. Jøsang, *Subjective Logic: A Formalism for Reasoning Under Uncertainty*, 1st ed. Springer Publishing Company, Incorporated, 2016.

[58] *CC2640R2F SimpleLink™ Bluetooth® 5.1 Low Energy Wireless MCU*, Texas Instruments, 2016, rev. C. [Online]. Available: https://www.ti.com/product/CC2640R2F

[59] J. Schulman, et al., "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[60] G. Vrbančič and V. Podgorelec, "Transfer learning with adaptive fine-tuning," *IEEE Access*, vol. 8, pp. 196 197–196 211, 2020.



Dian Chen is currently a Ph.D. student in the Department of Computer Science at Virginia Tech since 2022. She received a B.S. degree in Computer Science from Worcester Polytechnic Institute in 2020 and an M.S. degree in Computer Science from Northwestern University in 2022. Her research interests include network security in cyber-physical systems.

Qisheng Zhang is currently a Ph.D. student in the Department of Computer Science at Virginia Tech since 2019. He received a B.S. degree in mathematics from Shandong University in 2017 and an M.S. degree in mathematics from the University of Warwick in 2018. His research interests include network security and network science.

Ing-Ray Chen received a BS degree from the National Taiwan University and MS and Ph.D. degrees in computer science from the University of Houston, University Park. He is a professor at the Department of Computer Science at Virginia Tech. His research interests are primarily in trust, network, and service management as well as reliability, security, and performance analysis of mobile systems and wireless networks, including the Internet of Things, wireless sensor networks, service-oriented peer-to-peer networks, ad hoc networks, mobile social networks, mobile web services, mobile cloud services, and cyber-physical systems. Dr. Chen has published over 120 journal papers, with more than one-third of them appearing in IEEE/ACM Transactions journals. He is a recipient of the IEEE Communications Society William R. Bennett Prize in the field of Communications Networking, The 2023 IEEE ComSoc Network Operations & Management (CNOM) Test of Time Paper Award, and the U.S. Army Research Laboratory (ARL) Publication Award. Dr. Chen is a member of the IEEE and ACM.

Dong Sam Ha (Life Fellow) received the B.S. degree in electrical engineering from Seoul National University, Seoul, South Korea, in 1974, and the M.S. and Ph.D. degrees in electrical and computer engineering from The University of Iowa, Iowa City, IA, USA, in 1984 and 1986, respectively. Since Fall 1986, he has been a Faculty Member with The Bradley Department of Electrical and Computer Engineering, Virginia Polytechnic Institute and State University (Virginia Tech), Blacksburg, VA, USA. He is currently a Professor and the Founding Director of the Multifunctional Integrated Circuits and Systems (MICS) Group, composed of five faculty members and about 30 graduate students. His research interests include power management circuits for energy harvesting, intelligent analog and RF circuits and systems, wireless sensor nodes for smart farms, and high-temperature RF circuits and systems for downhole communications, jet engine monitoring, and extreme environment sensing.

Jin-Hee Cho (M'09; SM'14) is currently an Associate Professor in the Department of Computer Science at Virginia Tech since Aug. 2018 and a director of the Trustworthy Cyberspace Lab. Before joining Virginia Tech, she worked as a computer scientist at the U.S. Army Research Laboratory (US-ARL), Adelphi, Maryland, since 2009. Dr. Cho has published peer-reviewed technical papers in leading journals and conferences in cybersecurity, decision-making under uncertainty, and network science. She received the best paper awards in IEEE TrustCom'2009, BRIMS'2013, IEEE GLOBECOM'2017, 2017 ARL's publication award, and IEEE CogSima 2018. She won the 2015 IEEE Communications Society William R. Bennett Prize in the Field of Communications Networking and The 2023 IEEE ComSoc Network Operations & Management (CNOM) Test of Time Paper Award. Dr. Cho was selected for the 2013 Presidential Early Career Award for Scientists and Engineers (PECASE), the highest honor bestowed by the U.S. government on outstanding scientists and engineers in the early stages of their independent research careers. She also received the 2022 Faculty Fellow Award from the College of Engineering at Virginia Tech. Dr. Cho earned a Ph.D. degree in computer science from Virginia Tech in 2008. She is currently serving on the editorial board as an associate editor in IEEE Transactions on Network and Service Management and IEEE Transactions on Services Computing and served on the editorial board for the Computer Journal (Oxford). She is a senior member of the IEEE and a member of ACM.