applied sciences

MDPI

Review

A Survey on Machine-Learning Based Security Design for Cyber-Physical Systems

Sangjun Kim and Kyung-Joon Park *

Department of Information and Communication Engineering, Daegu Gyeongbuk Institute of Science and Technology (DGIST), Daegu 42988, Korea; sjkim@dgist.ac.kr
* Correspondence: kjp@dgist.ac.kr; Tel.: +82-53-785-6314

Abstract: A cyber-physical system (CPS) is the integration of a physical system into the real world and control applications in a computing system, interacting through a communications network. Network technology connecting physical systems and computing systems enables the simultaneous control of many physical systems and provides intelligent applications for them. However, enhancing connectivity leads to extended attack vectors in which attackers can trespass on the network and launch cyber-physical attacks, remotely disrupting the CPS. Therefore, extensive studies into cyber-physical security are being conducted in various domains, such as physical, network, and computing systems. Moreover, large-scale and complex CPSs make it difficult to analyze and detect cyber-physical attacks, and thus, machine learning (ML) techniques have recently been adopted for cyber-physical security. In this survey, we provide an extensive review of the threats and ML-based security designs for CPSs. First, we present a CPS structure that classifies the functions of the CPS into three layers: the physical system, the network, and software applications. Then, we discuss the taxonomy of cyber-physical attacks on each layer, and in particular, we analyze attacks based on the dynamics of the physical system. We review existing studies on detecting cyber-physical attacks with various ML techniques from the perspectives of the physical system, the network, and the computing system. Furthermore, we discuss future research directions for ML-based cyber-physical security research in the context of real-time constraints, resiliency, and dataset generation to learn about the possible attacks.

Keywords: cyber-physical systems; hierarchical CPS structure; CPS security; cyber-physical attacks; machine learning-based detection; learning-enabled CPS

Citation: Kim, S.; Park, K.-J. A Survey on Machine-Learning Based Security Design for Cyber-Physical Systems. *Appl. Sci.* **2021**, *11*, 5458. https://doi.org/10.3390/app11125458

Academic Editor: Jose Machado

Received: 25 May 2021
Accepted: 11 June 2021
Published: 12 June 2021

1. Introduction

Cyber-physical systems (CPSs) involve the integration of physical systems into the real world and control software in the cyber-world, where these two worlds are connected by networks that are responsible for the interchange of information between them [1,2]. Extensive developments in communications technology can support real-time communications with low latency, which makes it possible to control multiple physical systems remotely and provides various intelligent services to CPS users [3–5]. Moreover, adopting wired and wireless networks in a CPS enables the states of massive amounts of industrial equipment to be monitored, and therefore, it is possible to organize and flexibly manage a complex industrial system [5–8]. Thus, the CPS is one of the key technologies for various industrial domains, including intelligent transport systems [9–11], medical systems [12,13], and smart grids [14,15]. For example, in a communication-based train control (CBTC) system [6,9], which is a representative CPS, the communication technologies between trains and ground stations enable real-time feedback control by exchanging the states of the trains and the train control signals through a real-time wireless network. Therefore, the CBTC system reduces the dispatch interval between trains and guarantees better safety than conventional train control systems in guarding against accidents [16].

As the connectivity of the CPS increases and becomes more complex, the paths through which an attacker can infiltrate the CPS are increasing [17–19]. The networks that connect the physical systems and the control software are especially vulnerable to external attackers that aim to invade the CPS and cause malfunctions in the physical systems [20,21]. When an attacker accesses the network, the execution of control-critical software can be disturbed in the cyber-world, the control authority of physical system operations on the network can be seized, and the attacker can power-off the physical systems [16,18] or precisely manipulate the physical state with a deceitful attack detection system [14,15]. These cyber-physical attacks induce damage to industrial equipment and processes, causing economic losses and human casualties. In 2015, the BlackEnergy malware caused the malfunction of a power plant in Ukraine, resulting in a massive power outage [20]. In 2014, control of plant equipment was seized by a cyber attack on a German steel mill, and some blast furnaces were damaged [20]. To ensure the reliability of a CPS against adversaries, the need for cyber-physical security research is emerging [17,21].

Cyber-physical security is an extension of conventional cyber-security, where the operation of the physical system is additionally considered. For example, password cracking, which is password recovery process for a system, is one of the important security issues in the conventional cyber-security field due to the risk of personal information leaks. In cyber-physical security, a simple information leakage by password cracking cannot damage the CPS; however, the manipulation of the physical process by unauthorized access with a password can impact the dynamics of the physical systems. Therefore, a variety of cyber-physical security research is conducted by modeling physical dynamics with control theory. However, since CPSs are affected by various factors, such as rapid environmental changes and unexpected events, physical model-based cyber-physical security methods suffer from false alarms that degrade detection performance against cyber-physical attacks. Moreover, because the CPS becomes large and the relationships of each CPS component become complex, the level of accuracy shown by a conventional CPS model and a real CPS decreases, which generates additional attack vectors [22–24]. From a control–theoretical viewpoint, large and complex systems can be represented as high-order differential equations [24], where a mathematical model with a high-order term is vulnerable to noise on the state variables [23]. Therefore, it is difficult to obtain an exact mathematical model of a complex physical system, and unconsidered mathematical terms of the inaccurate dynamic system model become vulnerabilities of the model-based attack detector, resulting in inaccurate detection.

To overcome the limitations of legacy model-based cyber-physical security, data-driven anomaly detection methods (where abnormal data are acquired from numerous simulations and controlled experiments) are adopted in cyber-physical security [25,26]. In particular, machine learning (ML), which depicts correlations between an input and output using massive amounts of data without modeling based on physical laws, is adopted in cyber-physical security in order to satisfy high-level safety and reliability concerns [22]. Furthermore, ML techniques enable a model to be generated for the massive and complex relationships of each component of the CPS, including various physical systems in the real world, heterogeneous network protocols, and the complicated application software in the cyber-world, where the generated model can enhance the security level of the CPS.

In this paper, we provide a comprehensive survey of cyber-physical attacks and machine learning-based cyber-physical attack detection technologies. In particular, we focus on attacks that can damage physical systems and manipulate the physical processes. In addition, we mainly consider cyber-physical attack detection methods and attack handling methods with machine-learning techniques. There are a large number of surveys of physical model-based anomaly detection methods [27,28] and network intrusion detection systems (IDSs) [29–31]. Thus, conventional deviation-based attack detection methods in control theory and IDS-based anomaly detection techniques are not covered.

The rest of the paper is organized as follows. Section 2 introduces the hierarchical CPS structure and the roles in each layer. Section 3 provides the taxonomy of cyber-physical

attacks for each layer. Section 4 presents cyber-physical attack detection methods with machine learning techniques. Section 5 discusses the potential research directions for ML-based cyber-physical security in the context of real-time characteristics in the CPS, resiliency, and data generation methods for learning malicious behavior. Finally, we conclude this survey in Section 6.

2. Hierarchical Structure of Cyber-Physical Systems

A CPS can be constructed at a large size with massive components, and it can have complex relationships between each physical system in the real world and among control software in the cyber-world. Therefore, it is difficult to analyze the entire CPS [18]. We consider a hierarchical CPS structure as illustrated in Figure 1, which classifies and abstracts the complex CPS components as functions, and therefore the hierarchical CPS structure provides a CPS that is simpler and more intuitive [32]. The hierarchical CPS structure is applied in various fields, including in CBTC systems [9], smart production systems [33], and smart grids [34].

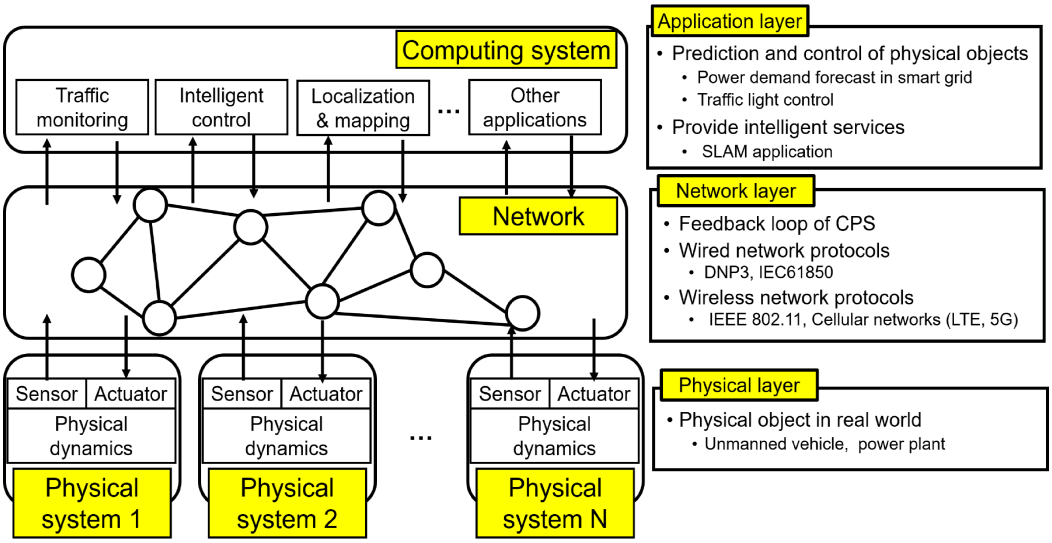

Figure 1. A hierarchical cyber-physical system structure.

The proposed CPS structure is similar to the Purdue enterprise reference architecture (PERA) [35], which is widely used in industrial control systems. The physical system layer of the proposed structure is matched to the lower layer of the PERA, including the specification layer, detailed design layer, manifestation layer, and operations layer. The network layer of the proposed structure corresponds to networks in the definition layer of the PERA. The application layer of the proposed structure is mapped to upper layer of the PERA, including concept layer and definition layer.

2.1. Physical System Layer

The physical system layer represents multiple physical systems, which are objects that operate in the real world. In the physical system layer, multiple physical systems have sensors and actuators in which sensors report states of the physical objects to the computing system in the cyber-world; the actuators operate the physical objects according to commands from the computing system [32]. From sensing and actuating in the physical system, the physical system fulfills the physical processes.

The operation of the physical system is represented as system dynamics in a continuous-time domain. To simplify the expression of physical dynamics, we consider a single-input single-output (SISO) linear time-invariant (LTI) system as follows:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t), \end{aligned} \tag{1}$$

where $x \in \mathbb{R}^n$ is the state of the physical system with n state variables, $A \in \mathbb{R}^{n \times n}$ is the system matrix, $B \in \mathbb{R}^n$ is the input matrix, $C \in \mathbb{R}^{1 \times n}$, $u \in \mathbb{R}$ is the control input signal, and $y \in \mathbb{R}$ is the sensor measurement. The system matrix term $Ax(t)$ represents the state transition by the characteristics of the physical system, the input matrix term $Bu(t)$ represents the state transition by the control input signal $u(t)$, and the output matrix term $Cx(t)$ represents the measurement of the state by the sensor attached on the physical system, where all physical state variables in the state vector $x(t)$ cannot be measured by the sensor. The physical systems periodically send a sensor measurement $y(t)$ to the computing system in order to report the state information, and then, the computing system returns the control input signal $u(t)$ to the physical systems to operate them as intended. We assume that matrix pairs (A, B) and (A, C) in (1) are controllable and observable, respectively.

The actuating process, which determines the dynamics of physical systems, is executed by the control input signal $u(t)$ on the computing system. The control input signal $u(t)$ is calculated as follows:

$$\begin{aligned} \dot{\hat{x}}(t) &= A\hat{x}(t) + Bu(t) + L(y(t) - C\hat{x}(t)) \\ u(t) &= -K\hat{x}(t), \end{aligned} \tag{2}$$

where $\hat{x}(t)$ is the state estimation of the physical system, L is the observer gain, and K is the controller gain. In general, the sensors cannot indicate full states of the physical systems; the computing system implements state estimation based on the system model (1) and sensor measurement $y(t)$ with an observer. From the state estimation in (2), the control input signal is calculated by the state feedback controller with controller gain K. We assume that the controller gain K and observer gain L are well-designed to stabilize the state of the physical systems.

Unmanned vehicles [7] and autonomous trains [9] are typical examples of physical systems within intelligent transport systems. Unmanned mobile objects are controlled by centralized control stations on the ground, where the dynamics of the unmanned mobile objects are determined according to the control command from the ground station [16]. Heating, ventilation, and air conditioning (HVAC) systems [36] and production machines [33] are examples of physical systems in industrial areas, and these industrial facilities belong to physical system-layer components. Likewise, in a smart grid, the power plants and the electric equipment that physically transmit electricity are elements of the physical system layer [14,15,37].

Most physical systems, especially unmanned aerial vehicles (UAVs), have power constraints because they have an external battery [38]. Therefore, it is difficult for the physical system to conduct tasks requiring complex computations. To overcome the power limitation, the physical system layer interacts with the computing system through the network layer.

2.2. Network Layer

The network layer is responsible for the communication between multiple physical systems in the real world and the computing systems in the cyber-world [2,5]. Due to the introduction of a network in the CPS, it is possible for the computing system to remotely control multiple physical systems, and therefore, in terms of the system configuration cost and flexible system management, the CPS becomes more advantageous than the conventional point-to-point control systems.

Sensor measurement from the physical systems and the control input signal from the application layer are exchanged over the network. When a network error occurs, such as control-related data that are missing [39] and/or a long transmission delay [40,41], physical systems can malfunction [42]. Therefore, a network that constructs a feedback control loop between the physical systems and the computing system must guarantee high-level reliability for data transmission.

In autonomous vehicle systems, wireless communication technologies, such as the IEEE 802.11-based standards, including 802.11p [43] and 802.11bd [44], and cellular vehicle-to-everything (C-V2X) communication standards, based on Long Term Evolution (LTE) [45] and 5G new radio (NR) [46], support real-time communication between multiple autonomous vehicles and road-side units (RSUs) fixed on the ground, in which communication enables real-time feedback control between the vehicles and the RSUs for autonomous driving. In the industrial area, wired networks such as EtherCAT [47] or Modbus [48], and wireless networks such as Zigbee [49] and WirelessHART [50], enable remote state reporting and the actuating of industrial facilities in real time. Furthermore, in a power grid, the distributed network protocol (e.g., DNP3) [51] connects the various power facilities and the centralized supervision system.

2.3. Application Layer

The application layer represents the computing system with which intelligent tasks are conducted by software in the cyber-world. The applications manage the physical systems [52], predict the state of the physical systems in the next time step [53,54], and provide intelligent functions to CPS users, where various CPS applications are executed based on sensor measurements from physical systems. From the results of the application execution, a computing system determines the system dynamics (1) in the next time step.

Due to the power limitation, the physical system depends on the computing system for the execution of intelligent functions requiring complex computations. In most CPSs, the application layer is supposed to have enough computing power and no electrical power constraints. In addition, due to real-time interactions in network layer, the power-limited physical systems can operate more intelligently than conventional embedded systems.

In intelligent transport systems, the RSUs and other traffic control equipment on the ground negotiate among autonomous vehicles, control the traffic lights at intersections, and distribute road traffic in congestion situations [55]. Furthermore, simultaneous localization and mapping (SLAM) [56], which positions a UAV and configures a map at the same time, and path planning, which determines the motion of vehicles in real time, are the most representative applications of a vehicular CPS. In a smart factory, digital twin technology [53,54], which realizes sophisticated and comprehensive factories in the cyber-world by using big data, is implemented at the application layer, where it is possible to expect the throughput of each production line per unit of time and to improve production lines continuously. Likewise, in a smart grid, a supervision system predicts power consumption for entire regions in real time based on sensor measurements from massive numbers of smart watt-hour meters in the physical system layer and from previous time-series data related to power consumption. Therefore, from power demand predictions, it is possible to feed back the control of electricity generation at the power plants [34].

3. Taxonomy of Cyber-Physical Attacks

A cyber-physical attack is defined as an exploitation of CPS components that causes a malfunction in a physical system and process, such as the divergence of state $x(t)$ in system dynamics (1). Since the CPS is the integration of the computing system, the networks, and the physical systems, if at least one CPS component is under attack, the states of all physical systems become unstable [32]. In contrast to the conventional embedded system, the CPS is especially vulnerable due to the connectivity of the networks, where an attacker can damage the computing systems, the network, and the physical systems, as illustrated in Figure 2. In addition, the CPSs are vulnerable due to a lack of proper protections for

the CPS such as design, configuration, and operation. In this section, we introduce the cyber-physical attacks that can occur against a CPS, where we classify cyber-physical attacks in the context of the hierarchical CPS structure discussed in Section 2.

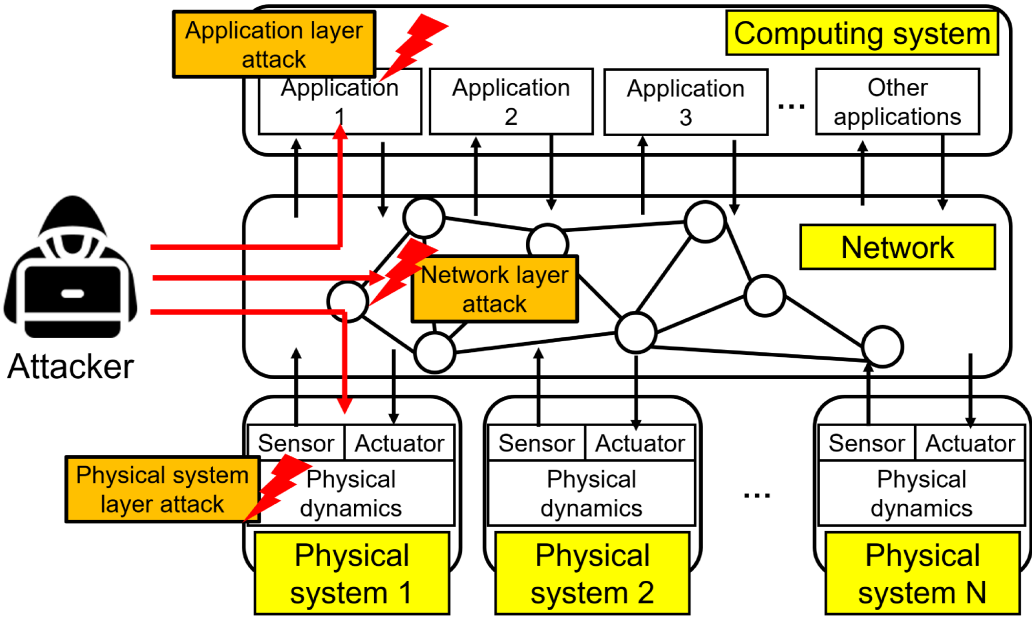

Figure 2. Cyber-physical attacks in hierarchical CPS architecture.

3.1. Physical System Layer

Physical systems exchange a control input signal $u(t)$ and sensor measurement $y(t)$ through the network. When an attacker intrudes into the network, the attacker can modify these two types of control-related data on the network, resulting in divergence from the physical state, $x(t)$.

Figure 3 shows attack locations for cyber-physical attacks in the physical system layer. There are three ways to manipulate control-related data on the network: first, the attacker only manipulates the sensor measurement packets $y(t)$ transmitted to the computing system; second, the attacker only manipulates control input signal packets $u(t)$ transmitted to actuators in the physical system; third, the attacker simultaneously manipulates both control input signal $u(t)$ and sensor measurement $y(t)$.

3.1.1. Sensor Attack

A sensor attack is defined as a manipulation of sensor measurement $y(k)$ on the network, which is represented as the addition of a sensor-attack signal to the legitimate sensor measurement, as follows:

$$\tilde{y}(t) = Cx(t) + y^a(t), \tag{3}$$

where $\tilde{y}(t)$ is the modified sensor measurement, and $y^a(t)$ is the sensor-attack signal from the attacker. The purpose of a sensor attack is to deceive the computing system that is conducting state estimation, and therefore an estimation error causes calculation faults in the control input signal $u(t)$ for the actuating process (2). Then, in the physical system (1), a fault control input signal in a sensor attack can cause malfunctions in the physical systems, such as divergence in the state trajectory of the physical system.

As cyber-physical security research advances, more sophisticated sensor attacks are being developed. Among the many cyber-physical security studies of the physical system layer, most focus on sensor attacks. For example, the pole-dynamics attack (PDA) [57,58] is one of the latest and most sophisticated sensor attacks, where a malicious user generates the attack signal by utilizing matrices A and C of a physical system (1). Although the

PDA rapidly alters state $x(t)$ in the physical system, conventional model-based anomaly detection methods such as the residual-based detector and the χ^2 detector cannot detect a PDA.

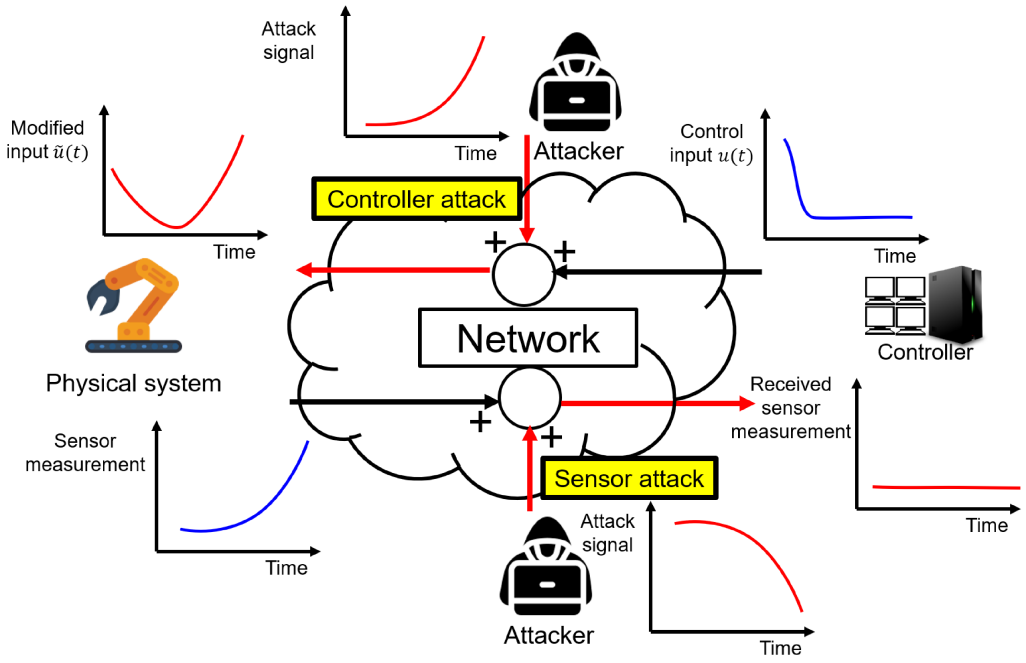

Figure 3. Cyber-physical attacks in physical system layer.

3.1.2. Controller Attack

A controller attack is defined as a modification of the control input signal $u(t)$ on a network, which is represented as the addition of the controller attack signal to the legitimate control input signal, as follows:

$$\dot{\tilde{x}}(t) = A\tilde{x}(t) + B(u(t) + u^a(t)), \tag{4}$$

where $\tilde{x}(t)$ is the attacked state of the physical system, and $u^a(t)$ is the control attack signal from the malicious user. The purpose of a controller attack is to destabilize the physical state $x(t)$ by injecting an unexpected control input signal $u^a(t)$. When a controller attack is launched, the physical system does not operate as intended by the computing system, because both the control input signal $u(t)$ from the computing system and controller attack signal $u^a(t)$ influence the physical dynamics, $x(t)$.

There are some advanced controller attacks used to avoid detection by conventional model-based detectors. In particular, sophisticated controller attacks exploit zero-dynamics [59], which is a specific behavior of the sensor measurement $y(t) = 0$ related to system dynamics. Attacks on the control input signal $u(t)$ and initial physical state $x(0)$ are reported in various cyber-physical security studies as the so-called zero-dynamics attack (ZDA) [60]. The ZDA targets the unstable zero-dynamics inherent in the physical system [60] or unstable zero-dynamics generated by discretization [61,62]. When a ZDA is launched, internal physical state $x(t)$ diverges to infinity; however, this divergence is not revealed in sensor measurement $y(t)$ due to a characteristic of zero-dynamics.

3.1.3. Combined Attack

A combined attack is defined as a simultaneous modification of a sensor measurement and the control input signal [63,64], which is represented as follows:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + B(u(t) + u^a(t)) \\ \tilde{y}(t) &= Cx(t) + y^a(t). \end{aligned} \tag{5}$$

Unlike a sensor-only attack and a controller-only attack, the combined attack requires eavesdropping on both channels over which the sensor measurement $y(t)$ and control input signal $u(t)$ are transmitted; therefore, it is difficult to successfully execute a combined attack. In other words, most combined attacks are more sophisticated than sensor-only and controller-only attacks, so detection strategies for the combined attack require more computational resources in order to detect and handle combined attacks.

The covert attack is one of the most typical combined attacks, where the attacker has perfect knowledge of the system dynamics and uses this to generate a sensor attack signal $y^a(t)$ and control input signal $u^a(t)$ [64]. On the sensor measurement transmission channel, the attacker seizes the legitimate sensor measurement $y(t)$ from the physical system and deceives the computing system by transmitting the generated sensor measurement $\tilde{y}$ with perfect system information [65]. Over the control input signal transmission channel, the physical system is controlled as the attacker intends by a malicious control input signal, $\tilde{u} = (u(t) + u^a(t)$, where that malicious control input signal is generated with perfect system information and with the legitimate sensor measurement signal $y(t)$ from the sensor measurement transmission channel.

3.2. Network Layer

The network layer is responsible for exchanging a variety of information between the physical system layer and the application layer with high-level reliability and real-time characteristics. In the network layer, the attacker can disturb data transmission, which violates the integrity of the data and real-time constraints on the CPS, resulting in the destabilization of the physical system.

Figure 4 illustrates three representative cyber-physical attacks in the network layer. First, the denial of service (DoS) attack forces dropped packets by exploiting physical characteristics or vulnerabilities in network protocols. Second, the flooding attack induces packet transmission delays in order to cause abnormal behavior in the physical systems. Third, the packet manipulation attack violates the data integrity on the network via packet modification. Although network layer attacks do not seem to have a relationship with physical dynamics (1) directly, these attacks can significantly affect the stability of the physical system.

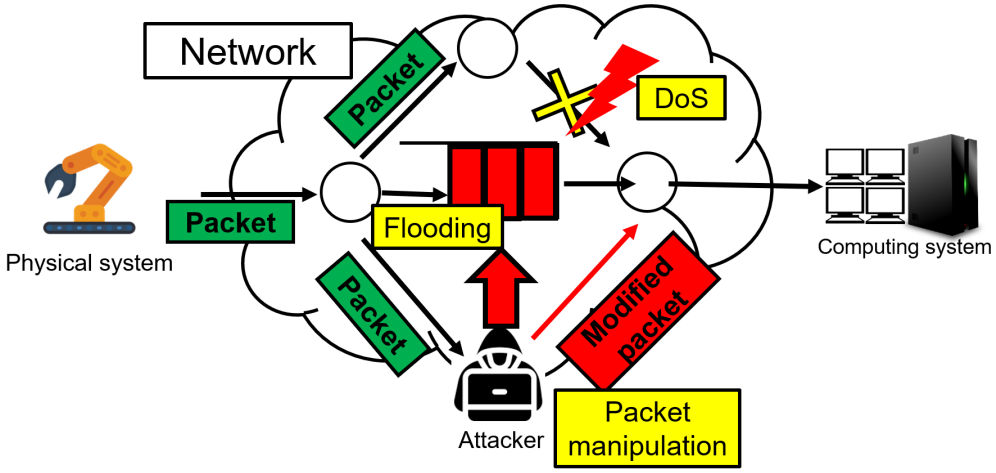

Figure 4. Cyber-physical attacks in the network layer.

3.2.1. Denial of Service Attack

In many control–theoretical studies, the DoS attack is defined as the prevention of delivery of control-related data, including control input signals and sensor measurements with certain network attacks [66–69]. Therefore, from the perspective of control theory, a DoS attack is modeled as a signal drop in the discrete-time domain, which has the same mean as the network disruption attack in the cyber security field. To cover the impact of

the physical dynamics (1) by an intentional packet drop from the DoS attack, in this paper, we define the DoS attack as an interruption of end-to-end packet delivery along the packet transmission path by forcing dropped packets, including communication jamming.

The DoS attack is realized in various ways; the implementation method is dependent on communication types and network protocols [70]. In the wired network environment, a physical link cut-off is the easiest way to interrupt transmissions of control-related data. In the wireless communications environment, the jamming attack is achieved by radiating a jamming signal with a high-gain antenna into the air, and this jamming signal reduces the signal-to-noise ratio (SNR), which interferes with the receiver [71]. From man-in-the-middle (MITM) attacks [42], which steal communication authority between two nodes by exploiting vulnerabilities in network protocols, the DoS attack can be implemented by intercepting packets without forwarding.

To express the DoS attack as viewed in the physical system layer, we rewrite physical dynamics (1) with discretization as follows:

$$\begin{aligned} x(k+1) &= A_d x(k) + B_d u(k) \\ y(k) &= C_d x(k), \end{aligned} \tag{6}$$

where $A_d \in \mathbb{R}^{n\times n}$, $B_d \in \mathbb{R}^n$, and $C_d \in \mathbb{R}^{1\times n}$ are the discretized system, input, and output matrices, respectively, via a zero-order hold (ZOH). When the DoS attack is launched against the network, the control input signal cannot be updated due to the dropped packets as follows:

$$x(k+1) = A_d x(k) + B_d u(k-1). \tag{7}$$

Intermittent packet drops from a DoS attack impede the control performance of the physical system. Moreover, if the DoS attack is launched persistently, it destabilizes the physical system with divergence in state $x(k)$ [66,69].

When the CPS is attacked at the network layer, physical system malfunctions trigger a fail-safe mode to avoid a divergence in the physical state via packet losses. The controller area network (CAN) [72], which is a wired in-vehicle network (IVN), connects massive electronic control units (ECUs) in vehicles. When a jamming signal is injected at some point on the CAN bus, most control signals on the CAN bus cannot be transmitted to ECUs, resulting in control errors in the vehicles [73]. In CBTC systems, where multiple trains and a ground station communicate through a leaky waveguide, an attacker can launch a long-range jamming attack to cause communication failures [74,75]. Because of the repeater on the track-side, which is installed to compensate for signal attenuation with distance, jamming signals by an attacker are also amplified, and therefore the passenger capacity per unit hour is reduced because of conservative operation due to communication failures.

3.2.2. Flooding Attack

The flooding attack is defined as an intentional exhaustion of network resources, such as network bandwidth or the memory of network devices, by generating massive amounts of network traffic [76,77]. When an attacker launches a flooding attack, the massive number of packets generated deprives legitimate communication nodes of transmission opportunities, or they fill up the memory in network devices. As the result of a flooding attack, transmission delays for legitimate packets increase and violate real-time constraints on the CPS. To succeed with a flooding attack, the attacker must simply rapidly generate a massive amount of data over the network, and therefore sophisticated knowledge about the network is not required.

In control theory, it is known that time-varying delays in a network negatively affect stability and the control performance of the physical system [41,78]. The effect of the network delay on physical dynamics (1) is represented as follows:

$$x(k+1) = e^{AT}x(k) + \sum_{j=0}^{\bar{d}-\underline{d}} \int_{T-t_{j+1}^k}^{T-t_j^k} e^{As} ds B u(k+j-\bar{d}), \tag{8}$$

where T is the sensor measurement sampling period of the physical system, t_j^k is the j-th time instance at time step k with $0 < t_j^k < T$, while $\bar{d}$ and $\underline{d}$ are $\lceil \tau_{max}/T \rceil$ and $\lfloor \tau_{min}/T \rfloor$, respectively, in which τ_{max} and τ_{min} are the maximum and minimum time-varying network delay bounds, respectively. The stability of the physical system with a large time-varying delay environment is explained in [41], where the allowable delay bound depends on the system dynamics and controller design. Network delay violates the stability condition; the physical state $x(t)$ diverges to infinity, and therefore the flooding attack makes the physical system unstable.

Like the DoS attack, when a flooding attack is launched against the network, physical systems are destabilized or switch into a fail-safe mode that stops or reduces the operation of the physical system to guarantee safety. For UAV control systems in IEEE 802.11-based wireless network environments, although a ground control station (GCS) sends control messages to the UAV, the UAV under an internet control message protocol (ICMP) flooding attack simply hovers in place, because the control-related packets are not received by the pre-configured transmission deadline due to the delay induced by ICMP flooding [79].

3.2.3. Packet Manipulation

Defined as a modification of the header or payload of packets transmitted over the network, packet manipulation consists of a packet-stealing phase and a packet-modification phase. In the packet-stealing phase, the attacker accesses the network and deceives communication nodes, including physical systems and the computing system, using vulnerabilities in the network protocol, CPS implementation, and communication nodes, etc. Then, the packets on the network are forwarded to the attacker. After the packet-stealing process, the attacker modifies the header or payload of the packets received from the legitimate source node and forwards the modified packets to the original destination.

When the attacker manipulates the error-checking field in the header during the modification phase, such as the checksum under the user datagram protocol (UDP), the destination node discards the manipulated packet because it determines that the received packet has an error due to the modified checksum. Therefore, the packet manipulation attack can also have a DoS attack effect. In other words, if the attacker modifies the payload while strictly adhering to the network protocols, then the destination nodes are deceived by the packet manipulation. When the sensor measurements or the control input signal are modified, this packet manipulation attack has an effect equal to physical layer attacks.

The packet manipulation attack can occur in various CPS fields. For a CBTC system, a packet manipulation attack targeting train control command packets is introduced with address resolution protocol (ARP) spoofing [16,32]. Due to the security vulnerability in the ARP process, the attacker can steal communication authority between the train and ground station; train collisions can occur due to control command manipulation. In [55], a sensor data-spoofing attack and various packet injection attacks targeting a vehicular ad-hoc network (VANET) are presented. Specifically, a Sybil attack, which deceives a number of nodes on the road in a VANET, makes it appear that traffic congestion occurs, although it does not, by sending incorrect messages, which results in inconvenience for traffic.

3.3. Application Layer

The application layer provides intelligent functions to CPS users and computing performance-limited physical systems. From the input/output (I/O) interfaces in the computing system, such as the serial communication port or network interface card (NIC), the attacker can intrude into the computing system and access important computing components, including file systems, cache memory, and process schedulers, where the attacker can launch attacks to disrupt the computing system. The more sophisticated the application layer, the larger the computing system becomes, and the higher the complexity; however, due to the complexity of the computing system, it is difficult to prevent an attack that targets the system.

Figure 5 shows two types of attack that can occur in the application layer. First, an application software attack disturbs the execution of CPS application software, resulting in the return of faulty control commands to the physical system or the generation of a system error. Second, the computing hardware attack executes malicious system commands that can damage computing hardware, such as the power supply, the CPU, and memory, and therefore the computing hardware attack disrupts the computing system itself.

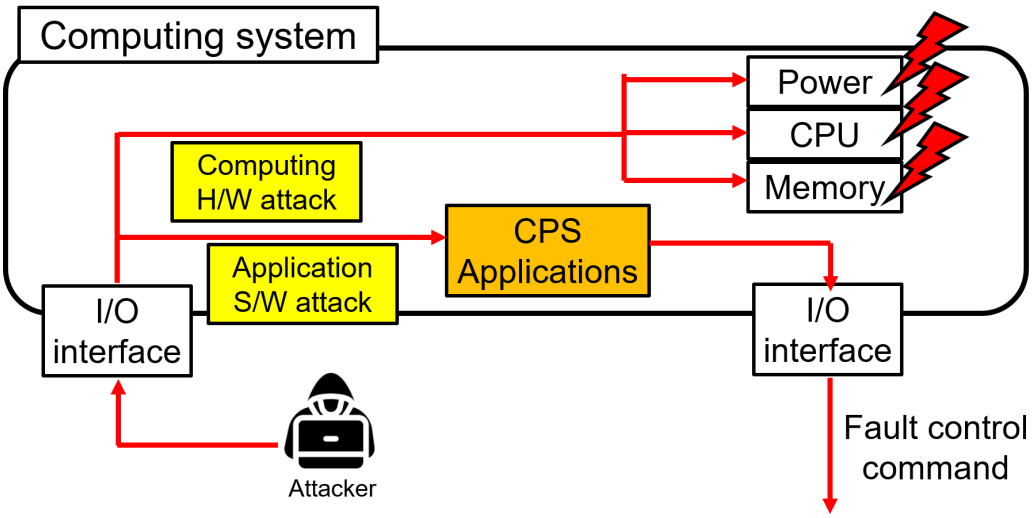

Figure 5. Cyber-physical attacks in the network layer.

3.3.1. Application Software Attack

An application software attack generates faulty results for service requests from the physical system layer, which then provides faulty services to CPS users and faulty commands to computation resource-limited physical systems. When faulty operational commands are injected into the physical systems, this causes the defective control of the physical system, which the CPS user does not intend.

Application software attacks are implemented by various methods including false code execution [80] and backdoor attacks [81]. False code execution, where the attacker installs malicious software in the computing system, returns false control commands and services to the physical system layer and the CPS user. The Stuxnet attack [82], which damaged an Iranian nuclear facility, BlackEnergy malware [20], which caused a massive power outage by damaging a Ukrainian power plant, and Triton [83], which triggered the fail-safe mode of a Saudi Arabian petrochemical plant, are typical application software attacks achieved with false code injection. Furthermore, the German railway infrastructure system was encrypted by the Wannacry ransomware, resulting in some failures of railway system components [84]. Backdoors, which access the computing system without a legitimate certification process, can also induce the malfunction of CPS applications. For example, object identification application for autonomous vehicles can be damaged by backdoors [85]; for example, modifying the training dataset related to traffic signs. Furthermore, a backdoor-based attack manipulates the setting of the applications, such as neural network parameters, resulting in the degradation of the safety-critical functions of the CPS, such as an emergency stop for an autonomous vehicle [81].

From the perspective of physical dynamics, an application software attack is represented as a fault in an actuation process (2). The modification of memory related to the state estimation process and the control input calculation process under the application software attack involve the manipulation of the observer gain L and controller gain K, respectively. When the observer gain L and controller gain K in the actuating process (2) are replaced to stabilize physical dynamics (1), even though the network layer and physical system layer are legitimate, the physical state $x(t)$ diverges to infinity.

In [86], the authors consider two application software attacks: one disables the feedback control software, and the other replaces legitimate control software with malicious software in order to destabilize a helicopter system. A controller gain change attack is considered in [87], which maliciously modifies the control gain in the control software, crashing a drone system.

Although eavesdropping cannot directly damage the computing system, the side-channel attack [88,89], which is advanced eavesdropping, can resemble an application software attack. For well-encrypted computing systems, the side-channel attack decrypts data related to the security of the computing system and attempts to find code execution information in the hardware, such as memory and the CPU, where the discovered hardware execution information can be used to configure malicious application software with reverse-engineering techniques. As an example of a typical side-channel attack, the cache-side attack periodically flushes a specific location in cache memory and eavesdrops on traces of the target processes [90]. From the eavesdropped traces, the attacker can decrypt the encryption policies of the computing system and can reconfigure malicious CPS applications to make the physical system unstable.

3.3.2. Computing Hardware Attack

The computing hardware attack is defined as disrupting a component of the computing system, such as the power supply, dynamic random access memory (DRAM), the CPU, and storage systems, directly or indirectly. We classify computing hardware attacks into two types of fault: one is a computer system-down attack, and the other is a data manipulation attack in memory via unauthorized accesses. Criteria for the computing hardware attack classification are determined by the impacts of the attacks on the physical system. A computer system-down attack refers to the impossibility of executing computing processes by intentional computer hardware faults, such as shutting down the power supply. Meanwhile, a data manipulation attack refers to modifying computing processes by the exploitation of a hardware vulnerability, but this does not break the computing hardware.

From the perspective of physical dynamics, a system crash from a computing system-down attack is equal to a DoS attack. When the computing system is disrupted by a malicious power-off, the control application software also stops, and therefore the physical system is no longer controlled by the computing system. Furthermore, the data manipulation type of attack is similar to an application software attack, where a physical bit-flip in DRAM is considered to represent the manipulation of controller gain K in (2).

Computer system-down attacks are implemented by injecting malware and requesting intensive tasks to be performed by the computing system maliciously. The power virus [91] leads to faults in the power supply of the computer by generating power surges that physically disrupt computer components. A CPU thermal attack [92] forces thermally intensive workloads to the CPU and leads to the physical disruption of the CPU or brings it into fail-safe mode, stopping the operation of the computing system. Meanwhile, data manipulation attacks are implemented by exploiting hardware vulnerabilities, but this does not break the computing hardware; an example of this is the row hammer attack [93]. The row hammer attack manipulates specific memory locations by intentional high-frequency access to rows of DRAM, exploiting an electromagnetic vulnerability [93,94]. The row hammer attack flips some bits in DRAM, which affects software operations in real time. In [95], a row hammer attack targeting software operation is proposed, which modifies a neural network model for image classification. This attack deteriorates image classification performance, which can do severe damage to a real CPS, such as for obstacle detection software in an autonomous driving system. Moreover, the authors in [96] analyze the performance deterioration from atomic-level bit-flips induced by computing hardware attacks.

4. ML-Based Cyber-Physical Attack Detection

Due to the advances in communication and computing technologies, the management of a large-scale CPS has been enabled, where each CPS component (including massive

physical systems), the communication networks, and the various CPS applications generate enormous amounts of data in each CPS layer of the hierarchical CPS structure proposed in Section 2. Because of the complexity of a large-scale CPS and the substantial amounts of data generated, ML techniques are adopted to detect cyber-physical attacks in order to overcome the detection limits of conventional, static, rule-based anomaly detection and misuse detection methods. In this section, we discuss ML-based cyber-physical attack detection strategies in each hierarchical CPS layer.

An ML-based anomaly detector consists of two phases [97], the training phase and the anomaly detection phase, as shown in Figure 6. In the training phase, the ML-based detector first collects the various CPS data related to cyber-physical security on each CPS layer in normal and attacked environments. On the physical system layer, safety-critical data including massive amounts of sensor measurements from multiple physical systems, control input signals, and control period information can be collected to train the ML-model in the detector. At the network layer, a variety of packet and network environment information related to threats, as discussed in Section 3, can be collected, such as packet headers, network channel state information, SNR, packet drops, and mean round trip time between the computing system and the physical system. At the application layer, information on the computing system, which can damage system software and hardware, can be collected, including the utilization of the CPU and RAM, the files in storage, and the frequency of specific command execution. After data collection, the ML-based detector labels data regarding whether they are generated under legitimate or abnormal situations. Finally, the ML model is built with features and labels, where various ML classification models can be adopted, such as a neural network [98], Q-learning [99], random forest [100], and the support vector machine (SVM) [101].

In the anomaly detection phase, the ML-based detector classifies abnormal behavior in the CPS with well-trained ML models in the training phase for unknown CPS data. When abnormal data, such as packets maliciously modified by an attacker, are injected into the CPS, the ML model analyzes abnormal behavior from that data. If the ML model classifies the data as abnormal, then the ML-based detector alerts CPS users to the attack and handles it to guarantee the stability of the CPS.

The performance of the ML-based detector is assessed with four metrics [102]: accuracy, precision, recall, and F1-score.

- Accuracy is defined as the number of correctly classified cases for the entire test dataset, whih is calculated as follows:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}, \tag{9}$$

  where TP is the number of correctly classified anomalies, TN is the number of samples correctly classified as normal, FP is the number of normal samples classified as anomalies, and FN is the number of anomalies wrongly classified as normal.
- Precision is defined as true-positive detection from samples the detector has determined to be abnormal, and is calculated as follows:

$$Precision = \frac{TP}{TP + FP}. \tag{10}$$

  Precision is related to false-positive detection, which degrades the control performance of physical systems.
- Recall is defined as detection performance with real anomalies, and is calculated as follows:

$$Recall = \frac{TP}{TP + FN}. \tag{11}$$

  Recall is related to misdetection probability, where a missed detection makes the physical system unstable.

- The F1-score is calculated as the harmonic mean between precision and recall, and is obtained as follows:

$$F1\text{-}score = 2 \times \frac{Precision \times Recall}{Precision + Recall}. \quad (12)$$

The F1-score shows the balance between precision and recall in an uneven sample distribution (a large number of normal samples).

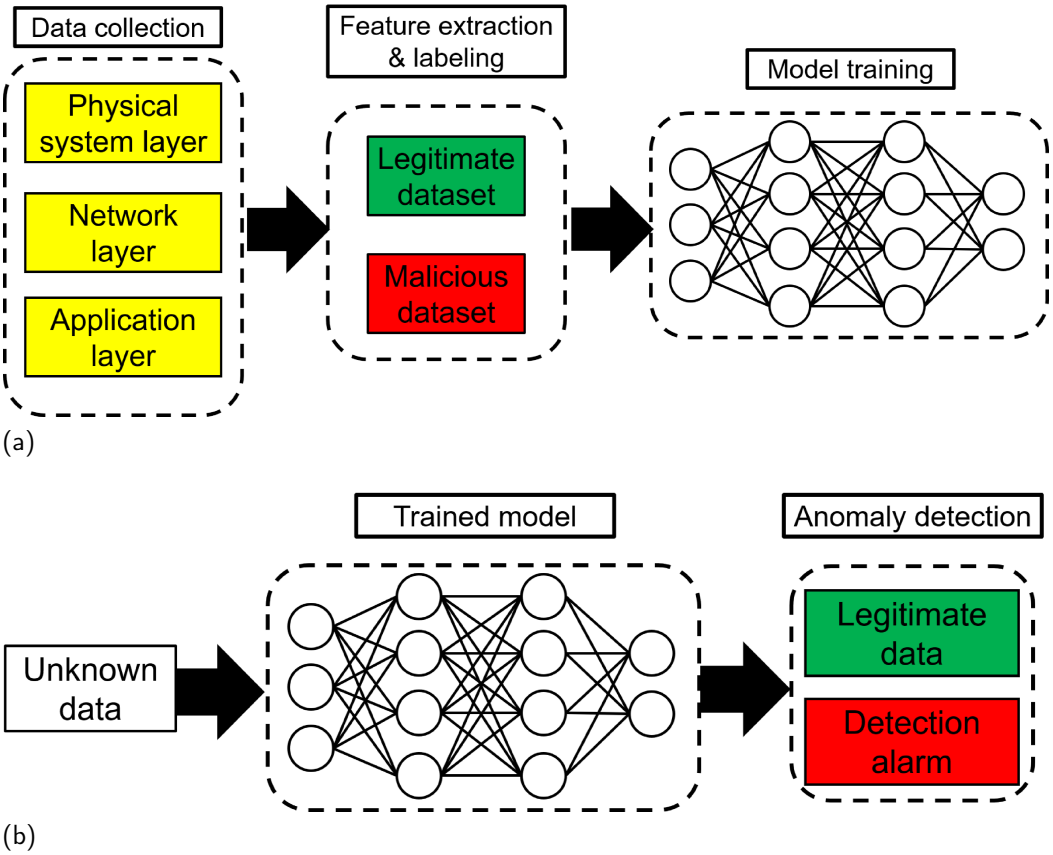

Figure 6. Machine learning in cyber-physical attack detection: (**a**) training the ML model for anomaly detection, and (**b**) anomaly detection with unknown data.

4.1. Physical System Layer

A recursive neural network (RNN)-based sensor attack detection strategy is proposed in [22], where an attacker targets a vehicle with four-wheel speed sensors. The authors in [22] consider multiple wheel speed sensor attack scenarios in which the proposed detection strategy classifies the location and number of attacked sensors. Experiments for the validation of the proposed RNN-based detector are carried out in a real road environment with actual sensors, where the accuracy of sensor attack detection is greater than 99%. An out-of-distribution detector for autonomous driving systems is proposed in [103], where the authors consider a malicious image injection attack against a visual-based autonomous driving system. For the attacks, the proposed detector adopts an auto-encoder and a deep support vector data description to learn convolutional neural network (CNN) models while reducing the computation time and guaranteeing real-time detection. An ML-based sensor attack-detection method targeting autonomous vehicles is proposed in [104] with a long short-term memory (LSTM)-based CNN model. The authors in [104] focus on various sensor attack scenarios with a small-magnitude attack signal, which poses problems for a conventional model-based detector. The LSTM-based CNN model is evaluated in terms of its accuracy, sensitivity, precision, and F1-score, where it enhances detection performance better than a Kalman filter (KF)-based detector and a KF-based CNN detector. A one-class SVM-based anomaly detector for connected vehicles is proposed in [105], where the

one-class SVM is substituted for a conventional χ^2 detector. The proposed detector on the vehicle allows it to utilize the states of surrounding vehicles connected through the network in order to classify anomalies. The evaluation results show that the proposed one-class SVM model enhances detection accuracy better than the χ^2 detector; however, the state transmission delay from the vehicles connected by the network deteriorates accuracy.

A Bayesian network-based attack detection strategy is proposed in [106] for water treatment systems, with data from multiple sensors and actuators under normal operation and attack situations. The authors in [106] evaluate the proposed detection strategy with precision, recall, and F1-score, in comparison with an SVM and a deep neural network (DNN) model. The proposed strategy has the advantage of a learning speed that is faster than other learning models. A robust supervised learning method to detect cyber-physical attacks on chemical processes is proposed in [107]. The proposed learning method adopts a non-linear SVM for the classification of normal behavior, disturbances, and anomalous behavior. An evaluation of the proposed method is conducted in a hardware-in-the-loop system (HILS) environment for a chemical process under sensor and controller attacks, where the proposed method provides real-time attack detection. A fusion of a physical model-based detector and a learning-based detector, defending against a sophisticated sensor attack targeting an HVAC system, is proposed in [108], where a one-class SVM model is used on the learning-based detector, which enables the detection of an attack that is not detected by a model-based detector. In [36], an SVM-based anomaly detection method is proposed for HVAC system anomalies, where a Gaussian process regression method [109] and an SVM method are combined to classify various faults in HVAC systems. The Gaussian process algorithm estimates the state of the HVAC system, and the SVM detection method is trained with the estimation from the Gaussian process and from sensor measurements. The demonstration result shows a low-level false detection rate and an execution time in milliseconds.

An anomaly detection module to detect cyber-physical attacks targeting a wide-area damping control system is proposed in [110]. The proposed anomaly detection strategy adopts various supervised learning algorithms, such as the SVM, the decision tree, k-nearest neighbors (KNN) and a neural network, with complete consideration of sensor attacks, controller attacks, and combined attacks. The performance of the anomaly detection module is evaluated in a hardware-in-the-loop testbed environment for two-area, four-machine power systems, resulting in a real-time attack detection with more than 96% accuracy. In [111], an anomaly detection method based on density ratio estimation (DRE) is proposed to detect a stealthy sensor attack targeting an AC microgrid. The proposed ML model with DRE does not require a dataset for abnormal behaviors, unlike other ML-based anomaly detection methods that require massive numbers of attack signals for detection. In simulations, the proposed DRE-based detection method shows a performance that is superior to conventional model-based anomaly detection methods and existing SVM-based detection methods. A covert attack detection method on a smart grid is proposed in [112], which utilizes an SVM model to learn the decision boundary between benign data and a covert attack. To improve the covert attack detection performance and to reduce computational complexity, the authors in [112] propose a genetic algorithm-based feature selection strategy. Compared with a conventional ML technique, such as multi-layer perceptron (MLP), naive Bayesian, and KNN methods, the proposed detection method with a feature selection strategy shows superior performance in terms of its accuracy and F1-score. A classification method for actual faults due to external disturbances and cyber-physical attacks in a large-scale smart grid is proposed in [113], which utilizes an unsupervised dynamic Bayesian network (DBN) model with time-series energy information. A symbolic dynamic filtering technique is adopted to extract features from collected information, which reduces computing resources and discovers interaction relationships between subsystems in large and complex power systems. For a sensor attack, the proposed methods are evaluated in simulations, where the performance of the proposed method achieves an accuracy of 98% and a false-positive detection rate of less than 2%.

We briefly review the ML-based attack detection methods against physical system layer attacks. Table 1 summarizes the existing ML-based detection methods.

Table 1. Summary of ML-based anomaly detection methods in the physical layer.

Reference	CPS Area	Defense Against	ML Model	Validation
[22]	Vehicle	Sensor	RNN	Experiment
[103]	Vehicle	Sensor	CNN	Simulation
[104]	Vehicle	Sensor	LSTM-CNN	Dataset
[105]	Vehicle	Sensor	SVM	Simulation
[106]	Water	Sensor	Various	Simulation
[107]	Chemical	Various	SVM	HILS
[108]	HVAC	Sensor	SVM	Simulation
[36]	HVAC	Various	SVM	Simulation
[110]	Power	Various	Various	HILS
[111]	Power	Sensor	DRE	Simulation
[112]	Power	Sensor	SVM	Simulation
[113]	power	Sensor	DBN	Simulation

4.2. Network Layer

For a vehicle platooning scenario in the IEEE 802.11p based VANET environment, the hybrid jamming attack detection method, combined with a protocol knowledge-based method and a learning-based method, is proposed in [114]. The jamming attack causes collisions of safety-critical cooperative awareness messages (CAMs); however, the CAMs are lost due to the inherent collision nature of the IEEE 802.11 communications protocol, and therefore inherent collisions and malicious collisions may not be classified. The proposed hybrid jamming attack detection method is evaluated by changing the number of platooned vehicles to as many as 25, with which the detection performance of the proposed method shows 95% accuracy. A CNN-based source node identification method for the CAN network is proposed in [115]. The proposed method can be utilized to detect abnormal jamming signals that exploit an inherent vulnerability in the CAN bus network, where the CNN model learns the channel characteristics and patterns of the CAN frame. A traffic sequence learning method for various types of network attack detection on a CAN is proposed in [73]. The authors in [73] consider three attack scenarios (DoS, random packet injection, and malicious packet injection with vehicle knowledge), and the proposed learning method, adopting a DNN model, learns these attacks with CAN traffic patterns for attack-free versus abnormal situations. A performance evaluation of the proposed method is conducted with real data from a CAN bus on a real vehicle, and the detection performance of the DNN model is superior to the decision tree algorithm and the KNN method. A supervised ML-based malicious node attack detector is proposed in [116] in consideration of a VANET with the IEEE 802.11p media access control (MAC) protocol and an ad-hoc on-demand distance vector (AODV) routing protocol, where a malicious node exploits the AODV protocol. The proposed detector adopts KNN models to learn various network features, including IP addresses, delay, jitter, dropped packets, and throughput. Simulation results in an NS-3 environment show 99% accuracy, with values less than 1% for the false-positive and false-negative detection ratios.

An SVM-based network attack detector is proposed in [117] for industrial control systems (ICSs). The proposed detector utilizes an SVM model and a K-means clustering method to alert administration to attacks, providing a classification of three types of network attack: the network scan, ARP spoofing, and the flooding attack. A two-stage, packet-level anomaly detector was proposed in [118], where the anomaly detector sequentially inspects the packet signatures and time-series characteristics of the packets. In the first stage, a Bloom-filter method detects packet-level anomalies in exchanged data, and a KNN learning model classifies an attack by inspecting time-series data. The authors validated the performance of the detector with real data from a gas pipeline system. A semi-supervised

learning technique to detect multiple cyber-physical attacks is proposed in [119], where the authors consider general industrial control systems adopting various industrial communications protocols. The proposed learning technique simultaneously utilizes a supervised learning model and an unsupervised learning model for automatic feature extraction and network anomaly detection. The proposed technique provides adaptive attack detection in a changing rapid attack-pattern environment, and therefore it has the strength to handle zero-day attacks. In [120], a fusion of learning methods with an LSTM model and a forward neural network (FNN) model is proposed to detect correlated network attacks. The FNN-only attack detection technique shows prominent detection performance only for single attacks, but low detection rates for correlated attacks. Meanwhile, the LSTM-only detection technique shows a remarkable detection capability for correlated attacks, but the accuracy in the detection of a single attack is not competitive. To overcome the defects of these two ML models, the proposed fusion method is adopted in the IDS, which enhances detection accuracy against both single and correlated network attacks.

In [121], the authors present malware detection algorithms with various machine learning models for networks under the DNP3 protocol. Stuxnet, which attacks industrial control systems, is selected as the target malware for the validation of the detection algorithms. A bidirectional RNN-based network attack detector for a power system with the IEEE 1815.1 protocol is proposed in [122]. The proposed RNN model separately learns the headers and payloads of the power system packets in order to classify various types of attack: five types of malware, three types of false-data injections, and a disabling reassembly attack are considered. The proposed detector not only identifies anomalies from the attacks under consideration, but it also detects trials of unauthorized command injections. A CNN-based network attack detector is proposed in [123] for supervisory control and data acquisition (SCADA) networks. The proposed CNN-based detector watches for network attacks in multiple network layers, from the data link layer to the application layer under the DNP3 protocol, where the authors consider 16 types of attacks and their combinations. The evaluation of the proposed detector is conducted on a real dataset from a power delivery system, with a detection performance above 94% precision for all attacks. A one-class SVM model is proposed in [124] to defend against DoS attacks targeting smart grid SCADA systems. In that work, the authors consider software-defined networking (SDN), where SDN is responsible for periodically capturing network device information on a centralized SDN controller. From the captured information, a one-class SVM model is trained, and it classifies DoS attacks against the network. The detection performance of the proposed one-class SVM shows an accuracy better than 99%.

We briefly review the ML-based attack detection methods against network layer attacks. Table 2 summarizes the existing ML-based detection methods.

Table 2. Summary of ML-based anomaly detection methods in the network layer.

Reference	CPS Area	Defense Against	ML Model	Validation
[114]	Vehicle (VANET)	DoS	DNN	Simulation
[115]	Vehicle (CAN)	DoS	CNN	Real-data
[73]	Vehicle (CAN)	Various	DNN	Real-data
[116]	Vehicle (VANET)	Manipulation	Various	Simulation
[117]	ICS	Various	Various	Experiment
[118]	Gas pipeline	Various	KNN	Real-data
[119]	ICS	Manipulation	Various	Experiment
[120]	ICS	Various	LSTM-FNN	Experiment
[121]	Power	Manipulation	Various	Simulation
[122]	Power	Various	RNN	Simulation
[123]	Power	Various	CNN	Experiment
[124]	power	DoS	SVM	Simulation

4.3. Application Layer

A light-weight ML-based software anomaly detector targeting a camera application in an embedded system is proposed in [125]. The proposed ML detector adopts a *k*-means algorithm, grouping data into *k* clusters, where the ML model utilizes the distribution of system call frequencies. A CNN model that detects malware is proposed in [126] to classify benign and malicious application software with various features related to permissions, code patterns, and application program interface (API) calls. The authors in [126] adopt a deep auto-encoder as a pre-training method of a CNN model to enhance the training speed of the proposed CNN model. The proposed detector reaches 99.8% accuracy in malware classification, which is higher than existing SVM models. In [127], an ML-based detector of malicious behavior in a computing system is proposed, using a thermal side channel of a CPU with thermal sensors. With a CNN algorithm, the proposed ML model utilizes temporal changes in a heat map of computing components under attacks such as code injection on a computational loop. The performance evaluation of the proposed detector is conducted on a multi-core processor, which shows robust real-time anomaly detection for the computing system with real-time thermal monitoring using a finite number of thermal sensors on chip.

The authors in [128] analyze the performance of various ML-based side channel-attack detection strategies to defend against micro-architectural side channel attacks. The analysis show that excessively frequent feature sampling for the side-channel attack not only increases the computing overhead from attack detection but also decreases accuracy. Therefore, appropriate feature sampling rates and computational overheads should be selected simultaneously. The same paper shows a trade-off between attack detection performance and detection latency [128] and proposes proper ML model selection for attack detection. An ML-based cache side-channel attack detection strategy is proposed in [129] under realistic computational load conditions, where the cache side-channel attack under zero-load, medium-load, and heavy-load conditions is considered. Hardware events on caches and system-wide information, including total CPU cycles and branch-miss prediction, are selected to train the ML model for attack detection. The evaluation of the ML-based detection strategy is conducted with 12 ML models, where most of them show performance degradation under heavy-load conditions, and with some ML models (such as KNN), a detection overhead is generated. A real-time cache side-channel attack detection method is proposed in [130] with a softmax classification algorithm. The proposed ML-based attack detection method is implemented with an Intel Performance Counter Monitor to measure and learn the state of the CPU for the ML model. Performance evaluation is conducted on various CPUs, where the proposed method can detect an attack within about one second, and the CPU usage is less than 1% for each CPU environment.

Against cyber-physical attacks, and exploiting computing hardware vulnerabilities, an ML-based attack classification method is proposed in [131], where the authors consider two cyber-physical attacks (row hammer and Spectre), which are side-channel attacks exploiting a structural vulnerability of the computer architecture. The authors choose three different ML models—logistic regression, the SVM, and MLP—to build attack classifiers. The proposed methods with three different ML models show detection performances of better than 99.7% accuracy and 98% accuracy against row hammer and Spectre, respectively. A CNN-based row hammer detection model is proposed in [132], which provides the online detection of row hammer without Linux kernel modification. The proposed CNN model monitors suspicious DRAM access patterns and learns complex patterns in DRAM accesses. An experiment is conducted in a desktop PC environment and shows that the proposed CNN model detects raw hammer within about 1.5 s. The detection time of the proposed detector [132] is sufficiently short because the average bit flip time by the raw hammer is 20 s in the experiment environment.

We briefly review the ML-based attack detection methods against physical application layer attacks. Table 3 summarizes the existing ML-based detection methods. In contrast to the physical system and network, computing systems use several common structures,

such as an Intel CPU, across the CPS area. Therefore, Table 3 does not contain a CPS area column, unlike Tables 1 and 2.

Table 3. Summary of ML-based anomaly detection methods in the application layer.

Reference	Defense Against	ML Model	Validation
[125,128,129]	Application software	Various	Experiment
[126,127]	Application software	CNN	Experiment
[130]	Application software	Softmax	Experiment
[131]	Computing hardware	Various	Experiment
[132]	Computing hardware	CNN	Experiment

5. Potential Research Directions

The ML technique is a powerful tool to detect various cyber-physical attacks targeting each CPS layer. However, the ML-based attack detection methods do not always guarantee the stability of the CPS due to the characteristics of the ML technique, such as the requirement of massive attack data and high computation load. Thus, there are some limitations to the adoption of the ML technique for real CPSs for all CPS layers, and it is necessary to overcome the characteristics of the ML in CPS design. In this section, we provide three potential research directions; real-time attack detection, resilient cyber-physical system design, and dataset generation for learning malicious behavior.

5.1. Real-Time Attack Detection in ML

General ML applications, such as obstacle detection, require high accuracy and precision. Therefore, general ML applications are evaluated with four performance metrics [102]: accuracy (9), precision (10), recall (11), and the F1-score (12). However, conventional evaluation metrics do not consider time-related metrics. Due to the real-time characteristics of physical dynamics in a CPS, security-critical ML applications must be evaluated with not only the four conventional metrics but also the time to detection metric.

Theoretically, physical dynamics can diverge to infinity, but the state of physical systems in the real world has a finite boundary. When the state of the physical system violates the boundary, the physical system becomes irreparable, which means disruption. Therefore, a cyber-physical attack must be detected before the state of the physical system exceeds the repairable boundary.

Detection deadlines are determined by system dynamics, the state of the physical system when the attack starts, and by the type of attack. Figure 7 shows one example of a real-time detection constraint under a DoS attack with a ball-beam control system, which is a well-known physical control system [133]. The ball-beam system tilts a beam to regulate the position of a ball rolling off the beam due to gravity; therefore, the ball-beam system is intrinsically unstable. When a DoS attack (7) is launched against a ball-beam control system, the ball rolls off the beam due to the inherently unstable characteristics of the system. We consider a control scenario in which the range of the beam is 0 m to 1 m and the reference ball position is 0.5 m. If the position of the ball exceeds the range of the beam, we then consider the physical system to be irreparable. In this scenario, we launch a DoS attack at 15 s; then, the position of the ball drastically decreases and exceeds the range of the beam at 25.75 s. Therefore, the ball-beam control system becomes irreparable. To avoid destroying the physical system, the cyber-physical attack detection strategy should succeed before the physical system reaches the irreparable state boundary. The attack detection deadline is determined to be after 10.75 s, which in this scenario is from the start of the DoS attack to the point where the physical system becomes irreparable.

Many studies into cyber-physical attack detection have been conducted; however, the research on cyber-physical security that takes real-time constraints into consideration is insufficient [27]. In the physical system layer, a system knowledge-based attack detection strategy with KF is proposed in [14], which supports real-time attack detection from sensor

attacks and DoS attacks on a smart grid. A real-time image sensor attack detection method with non-linear physical-dynamics knowledge is proposed in [134], where the proposed method is validated for a vehicle and detects the sensor attack within 0.1 s. In the network layer, a real-time packet manipulation-attack detection method is proposed targeting CBTC systems, where distributed network devices continuously monitor the exploitation of the ARP protocol [16]. In [58], a CPS framework provides detection against sophisticated sensor attacks through the network in real-time, where the real-time constraint is determined by the irreparable state condition of the physical system. In the application layer, a real-time anomaly detection method is proposed in [103] for autonomous driving systems, where the ML model classifies a normal image inputs versus malicious inputs causing unsafe conditions in a short time. In [87], a real-time software attack detection and mitigation strategy for a UAV system is proposed, where the attack is detected within a few hundred milliseconds.

ML-based cyber-physical attack detection methods must consider real-time constraints on physical systems. However, due to the enormous computation loads under a complex ML model, attack detection time is delayed, which causes a disruption of the physical system. To satisfy real-time constraints on cyber-physical security, reducing the complexity of an ML model is required while achieving high levels from conventional ML evaluation metrics (9)–(12).

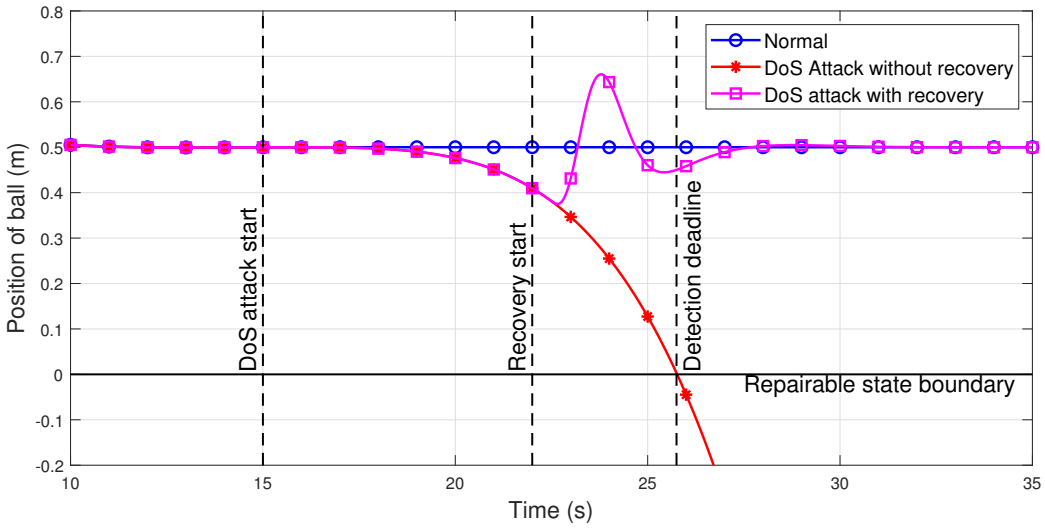

Figure 7. Real-time attack detection and recovery.

5.2. Resilient Cyber-Physical System Design

The purpose of a cyber-physical attack is to disrupt the physical system. In particular, the CPSs for societal infrastructure must have guaranteed safety and must provide normal services in spite of an attack. Therefore, a resilient CPS design that provides seamless performance recovery from an attack is required, with appropriate ML-based attack handling strategies after real-time attack detection.

Figure 7 shows an example of the recovery of a ball-beam control system under a DoS attack. When the DoS attack launches at 15 s, the position of the ball drastically leaves the range of the beam, resulting in the irreparable state of the physical system at 25.75 s (the red line). However, if a real-time detection and performance recovery strategy is adopted and operates starting at 22 s, then the position of the ball is regulated in a timely fashion to the reference position of 0.5 m with a temporary, small fluctuation (the magenta line). Through the resilience strategy, the stability of the ball-beam control system is guaranteed against the DoS attack in this control scenario.

In conventional physical system security, a hardware redundancy strategy is adopted to handle cyber-physical attacks and unexpected system faults. When one computing system of a CPS shows abnormal behavior, the previously configured auxiliary system

then substitutes for the abnormal computing system. A redundant controller architecture is adopted in [107] to mitigate the physical impact of a cyber-physical attack targeting chemical processes, switching controllers when the attack is detected by the SVM-model detector. The authors in [110] also consider a redundant system configuration to guarantee the stability of a power plant against cyber-physical attacks. In [135], a system-level simplex architecture that consists of a high-performance complex control module and a high-assurance safety controller with limited performance is proposed to guarantee the safety of the control system against unexpected faults in the complex control module. If the high-performance control module suffers an unexpected error, such as rebooting, the high-assurance control module takes over the control functions of the physical system, which provides robustness against system faults. The simplex architecture is also adopted for safety-critical CBTC systems in [32] to provide CPS resiliency against software faults and sensor attacks.

At the network layer, software-defined networking (SDN) can be adopted as one of the network recovery methods against the network layer attacks. SDN is a network paradigm that separates network functions for the control plane and data plane as software [136,137]. In the control plane, the centralized SDN controller periodically monitors the distributed network devices in general and installs network policies. On the data plane, distributed network devices are responsible for delivering the data and reporting the statistics of the network, such as link information, packet transmission successes, and changes in network topology. Interaction between the control plane and data plane enables the handling of network-layer cyber-physical attacks. When abnormal behavior, including attacks, is detected on the data plane, such as a link failure, the network devices send an alarm to the SDN controller, which then handles the abnormal behavior; e.g., with communications link reconfiguration for an alternative path routing method [138,139]. An SDN-based network recovery strategy is proposed in [140] for the performance recovery of a micro-grid against a link cut-off attack, where the proposed strategy detects malicious network topology changes from the attack and provides a seamless recovery of voltage with a small recovery overhead. Furthermore, the SDN provides a mitigation strategy against network delay generated by the flooding attack on the data plane [141,142]. In [58], the authors propose an SDN-based real-time cyber-physical attack detection method that simultaneously considers physical dynamics and network characteristics; the proposed SDN-based method also provides attacker isolation and communications path reconfiguration to recover the physical system from a PDA.

At the application layer, studies have been conducted to handle computing system attacks and guarantee the resiliency of a CPS. In [86], malicious code execution defense strategies are proposed, where the methods of restarting the computing system and utilizing a trust execution environment for the computing hardware platform guarantee safety against malicious code injection attacks. A simplex computing architecture for a secure UAV control is proposed in [87], which switches from the normal control mode to a safe control mode to stabilize the UAV when the attacker damages the virtual machine of the computing system, such as through the manipulation of the control parameter. Moreover, the ML-based applications are vulnerable to cyber-physical attacks against the computing layer, which drastically deteriorates the performance of ML applications [96]. If safety-critical ML applications are attacked, performance degradation in the ML-based detectors means that cyber-physical attacks will be missed. Thus, ML-based attack detection methods should be secure and should guarantee their own resilience. To mitigate application software attacks targeting a neural network model, a pruning-based and fine-tuning-based defense strategy is proposed in [81], which reduces backdoor attack success rates to 0% with only 0.4% accuracy degradation in the original ML application. A watermarking method for an ML model is proposed in [143], enhancing the security level of the ML model against a backdoor attack.

5.3. Dataset Generation for Learning Malicious Behavior

Unlike intelligent CPS applications, such as object detection using ML techniques with massive amounts of image data, adapting an ML technique for cyber-physical security is difficult in practice due to the lack of anomalous data for the CPS. Enormous amounts of fault and attack data are required to train the ML-based anomaly detector; however, there are limits to the ability to generate anomalous data such as car accident data and CPS faults in medicine, which involve human casualties [102,144]. Moreover, for unknown attack vectors and attack techniques, it is impossible to learn anomalous data that cause physical damage. Due to the lack of anomalous data collections for training, ML-based detectors provide poor accuracy and can generate false alarms.

The generative adversarial network (GAN) is an ML framework and consists of two neural network models: the generator and the discriminator. Figure 8 illustrates the learning processes of the two different learning models in the GAN. The generator makes malicious signals with a generation function G and random noise z to deceive the discriminator. The discriminator, with the classification function D, identifies legitimate signals from a database and malicious signals from the generator. The identification result from the discriminator, whether successfully classified or not, is back-propagated to both neural network models for training, and the neural network model is updated. From the iterations of malicious signal generation, the discrimination, and the back propagation, the malicious signal from the generator becomes more sophisticated. The learning process as illustrated in Figure 8 is finished when the discriminator no longer classifies the legitimate signals and the maliciously generated signals with the generating function G. A well-designed generator can be utilized to create malicious cyber-physical attack signals to train ML-based anomaly detectors.

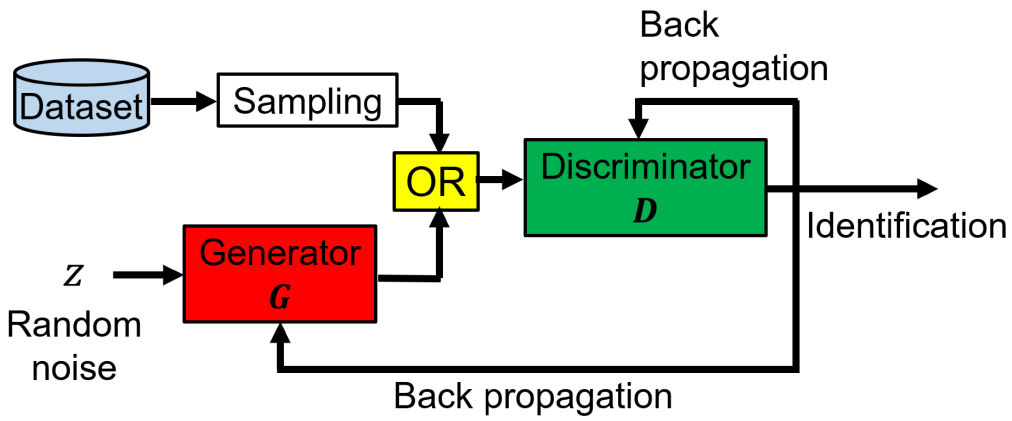

Figure 8. Structure of generative adversarial network (GAN) learning.

Multi-variable anomaly detection with a GAN model is proposed in [145], where an LSTM-RNN model is adopted as both a generator and discriminator in order to consider the time-varying characteristics of the physical system. The proposed GAN model is evaluated with a novel metric related to discrimination and anomaly reconstruction from two complex physical system datasets. Besides the generated anomaly signal, a GAN-based anomaly detection strategy that considers real-time requirements is proposed in [146]. In terms of GAN-based anomaly detection, the authors in [146] adopt fog computing to reduce the latency in anomaly detection, which is five times faster than the latency in [145]. In [147], the GAN modeling method is proposed for the analysis of cyber-physical security requirements, including integrity and availability, for production systems. The training of the proposed GAN model utilizes signal and energy information exchanged between the cyber-domain and the physical domain. The authors in [147] evaluate the performance of the security analysis with the proposed GAN modeling method in a 3D printer testbed environment.

A GAN-assisted network IDS is proposed in [148] to achieve a high-accuracy anomaly detection rate on a network with insufficient abnormal data. An evaluation with the KDD'99 dataset [149] shows that the proposed GAN-based IDS outperforms a standalone IDS in terms of its precision, recall, and F1-score. In [150], a hierarchical GAN framework is proposed for network intrusion detection in large-scale distributed networks with an auto-encoder model. The GAN training process on the local network gateway transmits a parametric model to the centralized network controller, which benefits from saving communication overheads rather than duplicating raw traffic. From the received models, the centralized network controller learns a global anomaly detection model.

6. Conclusions

Since networks combine the physical system and the computing system, a CPS becomes vulnerable to cyber-physical attacks, which may disrupt and cause malfunctions in a physical system in the real world. Moreover, due to the enhanced connectivity of the networks, the CPS will become large and complex; thus, modeling a complex CPS becomes difficult and inaccurate compared to a real CPS, reducing the security level of conventional model-based cyber-physical security strategies. To guarantee the safety and reliability of large and complex CPSs, it is necessary to adopt not only machine learning techniques but also conventional detection approaches in order to enhance the security level of the CPS.

This paper presented a comprehensive survey of the threats that damage the CPS and attack-detection strategies based on ML techniques. First, we presented a hierarchical CPS model that abstracts the complex CPS structure into three layers of CPS functions: the physical system layer, the network layer, and the application layer. Then, we presented cyber-physical attacks for each CPS layer in terms of attack implementations and examples. In particular, we analyzed various cyber-physical attacks from the perspective of physical system dynamics by introducing linear dynamics. In addition, we presented various ML-based cyber-physical attack-detection strategies and security evaluation metrics of the ML-based strategies, where the hierarchical CPS model was considered in order to detect and handle cyber-physical attacks targeting each layer. Finally, we discussed future research directions from the perspectives of real-time attack detection, resilient CPS design, and dataset generation to defend against cyber-physical attacks. These research directions enhance the security level of the CPS, despite the shortcomings of ML techniques, due to their tremendous computation power and data-driven nature.

Author Contributions: Conceptualization, S.K. and K.-J.P.; investigation, S.K.; writing—original draft preparation, S.K.; writing—review and editing, K.-J.P.; visualization, S.K.; supervision, K.-J.P.; project administration, K.-J.P.; funding acquisition, K.-J.P. All authors have read and agreed to the published version of the manuscript.

Funding: This work has been supported by the Unmanned Swarm CPS Research Laboratory program of the Defense Acquisition Program Administration and Agency for Defense Development (UD190029ED).

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

AODV	Ad-hoc on-demand distance vector
ARP	Address resolution protocol
CBTC	Communication-based train control
CAM	Cooperative awareness message
CAN	Control area network
CNN	Convolutional neural network
CPS	Cyber-physical system
CPU	Central processing unit
DNN	Deep neural network
DoS	Denial of service
DRAM	Dynamic random access memory
DRE	Density ratio estimation
DBN	Dynamic Bayesian network
ECU	Electronic control unit
FNN	Forward neural network
GAN	Generative adversarial network
HVAC	Heating, ventilation, and air conditioning
IDS	Intrusion detection system
KF	Kalman filter
KNN	K-nearest neighbor
LSTM	Long short-term memory
LTE	Long term evolution
MAC	Media access control
MITM	Man-in-the-middle
ML	Machine learning
MLP	Multi-layer perceptron
NIC	Network interface card
PDA	Pole-dynamics attack
RAM	Random access memory
RSU	Road side unit
RNN	Recursive neural network
SCADA	Supervisory control and data acquisition
SDN	Software-defined networking
SLAM	Simultaneous localization and mapping
SNR	Signal-to-noise ratio
SVM	Support vector machine
UAV	Unmanned aerial vehicle
VANET	Vehicular ad-hoc network
ZDA	Zero-dynamics attack

References

1. Park, K.J.; Zheng, R.; Liu, X. Cyber-physical systems: Milestones and research challenges. *Comput. Commun.* **2012**, *36*, 1–7. [CrossRef]
2. Kim, D.; Won, Y.; Kim, S.; Eun, Y.; Park, K.J.; Johansson, K.H. Sampling rate optimization for IEEE 802.11 wireless control systems. In Proceedings of the ACM/IEEE International Conference on Cyber-Physical Systems (ICCPS), Montreal, QC, Canada, 16–18 April 2019; pp. 87–96.
3. Rajkumar, R.; Lee, I.; Sha, L.; Stankovic, J. Cyber-physical systems: The next computing revolution. In Proceedings of the Design Automation Conference, Anaheim, CA, USA, 13–18 June 2010; pp. 731–736.
4. Kim, K.D.; Kumar, P.R. Cyber–physical systems: A perspective at the centennial. *Proc. IEEE* **2012**, *100*, 1287–1308.
5. Ahlén, A.; Akerberg, J.; Eriksson, M.; Isaksson, A.J.; Iwaki, T.; Johansson, K.H.; Knorn, S.; Lindh, T.; Sandberg, H. Toward wireless control in industrial process automation: A case study at a paper mill. *IEEE Control Syst. Mag.* **2019**, *39*, 36–57.
6. Wang, X.; Liu, L.; Tang, T.; Sun, W. Enhancing communication-based train control systems through train-to-train communications. *IEEE Trans. Intell. Transp. Syst.* **2019**, *20*, 1544–1561. [CrossRef]

7. Mozaffari, M.; Saad, W.; Bennis, M.; Nam, Y.; Debbah, M. A tutorial on UAVs for wireless networks: Applications, challenges, and open problems. *IEEE Commun. Surv. Tutor.* **2019**, *21*, 2334–2360. [CrossRef]
8. Shumeye Lakew, D.; Sa'ad, U.; Dao, N.; Na, W.; Cho, S. Routing in flying ad hoc networks: A comprehensive survey. *IEEE Commun. Surv. Tutor.* **2020**, *22*, 1071–1120. [CrossRef]
9. Farooq, J.; Soler, J. Radio communication for communications-based train control (CBTC): A tutorial and survey. *IEEE Commun. Surv. Tutor.* **2017**, *19*, 1377–1402. [CrossRef]
10. Cho, B.M.; Jang, M.S.; Park, K.J. Channel-aware congestion control in vehicular cyber-physical systems. *IEEE Access* **2020**, *8*, 73193–73203. [CrossRef]
11. Paranjothi, A.; Khan, M.S.; Zeadally, S. A survey on congestion detection and control in connected vehicles. *Ad Hoc Netw.* **2020**, *108*, 102277. [CrossRef]
12. Meng, W.; Li, W.; Wang, Y.; Au, M.H. Detecting insider attacks in medical cyber–physical networks based on behavioral profiling. *Future Gener. Comput. Syst.* **2020**, *108*, 1258–1266. [CrossRef]
13. Cho, B.M.; Park, K.J.; Park, E.C. Fairness-aware radio resource management for medical interoperability between WBAN and WLAN. *Ann. Telecommun.* **2016**, *71*, 441–451. [CrossRef]
14. Manandhar, K.; Cao, X.; Hu, F.; Liu, Y. Detection of faults and attacks including false data injection attack in smart grid using Kalman filter. *IEEE Trans. Control Netw. Syst.* **2014**, *1*, 370–379. [CrossRef]
15. Rawat, D.B.; Bajracharya, C. Detection of false data injection attacks in smart grid communication systems. *IEEE Signal Process. Lett.* **2015**, *22*, 1652–1656. [CrossRef]
16. Kim, S.; Won, Y.; Park, I.H.; Eun, Y.; Park, K.J. Cyber-physical vulnerability analysis of communication-based train control. *IEEE Internet Things J.* **2019**, *6*, 6353–6362. [CrossRef]
17. Koutsoukos, X. Systems science of secure and resilient cyberphysical systems. *Computer* **2020**, *53*, 57–61. [CrossRef]
18. Teixeira, A.; Pérez, D.; Sandberg, H.; Johansson, K.H. Attack models and scenarios for networked control systems. In Proceedings of the International Conference on High Confidence Networked Systems, Beijing, China, 17–18 April 2012; pp. 55–64.
19. Khalid, F.; Rehman, S.; Shafique, M. Overview of security for smart cyber-physical systems. In *Security of Cyber-Physical Systems*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 5–24.
20. Alladi, T.; Chamola, V.; Zeadally, S. Industrial control systems: Cyberattack trends and countermeasures. *Comput. Commun.* **2020**, *155*, 1–8. [CrossRef]
21. Dibaji, S.M.; Pirani, M.; Flamholz, D.B.; Annaswamy, A.M.; Johansson, K.H.; Chakrabortty, A. A systems and control perspective of CPS security. *Annu. Rev. Control* **2019**, *47*, 394–411. [CrossRef]
22. Shin, J.; Baek, Y.; Lee, J.; Lee, S. Cyber-physical attack detection and recovery based on RNN in automotive brake systems. *Appl. Sci.* **2019**, *9*, 82. [CrossRef]
23. Brunton, S.L.; Kutz, J.N. *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control*; Cambridge University Press: Cambridge, CA, USA, 2019; Volume 1.
24. Isidori, A.; Sontag, E.; Thoma, M. *Nonlinear Control Systems*; Springer: Berlin/Heidelberg, Germany, 1995; Volume 3.
25. Olowononi, F.O.; Rawat, D.B.; Liu, C. Resilient machine learning for networked cyber physical systems: A survey for machine learning security to securing machine learning for CPS. *IEEE Commun. Surv. Tutor.* **2021**, *23*, 524–552. [CrossRef]
26. Hassan, M.U.; Rehmani, M.H.; Chen, J. Differential privacy techniques for cyber physical systems: A survey. *IEEE Commun. Surv. Tutor.* **2020**, *22*, 746–789. [CrossRef]
27. Giraldo, J.; Urbina, D.; Cardenas, A.; Valente, J.; Faisal, M.; Ruths, J.; Tippenhauer, N.O.; Sandberg, H.; Candell, R. A survey of physics-based attack detection in cyber-physical systems. *ACM Comput. Surv. (CSUR)* **2018**, *51*, 1–36. [CrossRef]
28. Tan, S.; Guerrero, J.M.; Xie, P.; Han, R.; Vasquez, J.C. Brief survey on attack detection methods for cyber-physical systems. *IEEE Syst. J.* **2020**, *14*, 5329–5339. [CrossRef]
29. Alsubhi, K.; Bouabdallah, N.; Boutaba, R. Performance analysis in intrusion detection and prevention systems. In Proceedings of the IFIP/IEEE International Symposium on Integrated Network Management (IM 2011) and Workshops, Dublin, Ireland, 23–27 May 2011; pp. 369–376.
30. Mitchell, R.; Chen, R. Effect of intrusion detection and response on reliability of cyber physical systems. *IEEE Trans. Reliab.* **2013**, *62*, 199–210. [CrossRef]
31. Mitchell, R.; Chen, I.R. A survey of intrusion detection techniques for cyber-physical systems. *ACM Comput. Surv. (CSUR)* **2014**, *46*, 1–29. [CrossRef]
32. Won, Y.; Yu, B.; Park, J.; Park, I.H.; Jeong, H.; Baik, J.; Kang, K.; Lee, I.; Son, S.H.; Park, K.-J.; et al. An attack-resilient CPS architecture for hierarchical control: A case study on train control systems. *Computer* **2018**, *51*, 46–55. [CrossRef]
33. Aceto, G.; Persico, V.; Pescapé, A. A survey on information and communication technologies for industry 4.0: State-of-the-art, taxonomies, perspectives, and challenges. *IEEE Commun. Surv. Tutor.* **2019**, *21*, 3467–3501. [CrossRef]
34. Fang, X.; Misra, S.; Xue, G.; Yang, D. Smart grid—The new and improved power grid: A survey. *IEEE Commun. Surv. Tutor.* **2012**, *14*, 944–980. [CrossRef]
35. Williams, T.J. The Purdue enterprise reference architecture. *Comput. Ind.* **1994**, *24*, 141–158. [CrossRef]
36. Van Every, P.M.; Rodriguez, M.; Jones, C.B.; Mammoli, A.A.; Martínez-Ramón, M. Advanced detection of HVAC faults using unsupervised SVM novelty detection and Gaussian process models. *Energy Build.* **2017**, *149*, 216–224. [CrossRef]

37. Salinas, S.A.; Li, P. Privacy-preserving energy theft detection in microgrids: A state estimation approach. *IEEE Trans. Power Syst.* **2016**, *31*, 883–894. [CrossRef]
38. Wang, H.; Zhao, H.; Zhang, J.; Ma, D.; Li, J.; Wei, J. Survey on unmanned aerial vehicle networks: A cyber physical system perspective. *IEEE Commun. Surv. Tutor.* **2020**, *22*, 1027–1070. [CrossRef]
39. Peng, T.; Leckie, C.; Ramamohanarao, K. Survey of network-based defense mechanisms countering the DoS and DDoS problems. *ACM Comput. Surv. (CSUR)* **2007**, *39*, 3. [CrossRef]
40. Lou, X.; Tran, C.; Tan, R.; Yau, D.K.; Kalbarczyk, Z.T. Assessing and mitigating impact of time delay attack: A case study for power grid frequency control. In Proceedings of the ACM/IEEE International Conference on Cyber-Physical Systems (ICCPS), Montreal, QC, Canada, 16–18 April 2019; pp. 207–216.
41. Cloosterman, M.B.; Van de Wouw, N.; Heemels, W.; Nijmeijer, H. Stability of networked control systems with uncertain time-varying delays. *IEEE Trans. Autom. Control* **2009**, *54*, 1575–1580. [CrossRef]
42. Conti, M.; Dragoni, N.; Lesyk, V. A survey of man in the middle attacks. *IEEE Commun. Surv. Tutor.* **2016**, *18*, 2027–2051. [CrossRef]
43. Jiang, D.; Delgrossi, L. IEEE 802.11p: Towards an international standard for wireless access in vehicular environments. In Proceedings of the VTC Spring 2008-IEEE Vehicular Technology Conference, Marina Bay, Singapore, 11–14 May 2008; pp. 2036–2040.
44. Naik, G.; Choudhury, B.; Park, J.M. IEEE 802.11bd 5G NR V2X: Evolution of radio access technologies for V2X communications. *IEEE Access* **2019**, *7*, 70169–70184. [CrossRef]
45. Chen, S.; Hu, J.; Shi, Y.; Peng, Y.; Fang, J.; Zhao, R.; Zhao, L. Vehicle-to-everything (V2X) services supported by LTE-based systems and 5G. *IEEE Commun. Stand. Mag.* **2017**, *1*, 70–76. [CrossRef]
46. Garcia, M.H.C.; Molina-Galan, A.; Boban, M.; Gozalvez, J.; Coll-Perales, B.; Şahin, T.; Kousaridas, A. A tutorial on 5G NR V2X communications. *IEEE Commun. Surv. Tutor.* **2021**. [CrossRef]
47. Rostan, M.; Stubbs, J.E.; Dzilno, D. EtherCAT enabled advanced control architecture. In Proceedings of the IEEE/SEMI Advanced Semiconductor Manufacturing Conference (ASMC), San Francisco, CA, USA, 11–13 July 2010; pp. 39–44.
48. Dutertre, B. Formal modeling and analysis of the Modbus protocol. In *Proceedings of the International Conference on Critical Infrastructure Protection*; Springer: Boston, MA, USA, 2007; pp. 189–204.
49. Gislason, D. *Zigbee Wireless Networking*; Newnes: Oxford, UK, 2008.
50. Song, J.; Han, S.; Mok, A.; Chen, D.; Lucas, M.; Nixon, M.; Pratt, W. WirelessHART: Applying wireless technology in real-time industrial process control. In Proceedings of the IEEE Real-Time and Embedded Technology and Applications Symposium, St. Louis, MO, USA, 22–24 April 2008; pp. 377–386.
51. IEEE Standards Association. *IEEE Standard for Electric Power Systems Communications-Distributed Network Protocol (DNP3)*; IEEE: Piscataway, NJ, USA, 2014.
52. Figueiredo, J.; da Costa, J.S. A SCADA system for energy management in intelligent buildings. *Energy Build.* **2012**, *49*, 85–98. [CrossRef]
53. Haag, S.; Anderl, R. Digital twin—Proof of concept. *Manuf. Lett.* **2018**, *15*, 64–66. [CrossRef]
54. Tao, F.; Zhang, H.; Liu, A.; Nee, A.Y.C. Digital twin in industry: State-of-the-art. *IEEE Trans. Ind. Inform.* **2019**, *15*, 2405–2415. [CrossRef]
55. Hasan, M.; Mohan, S.; Shimizu, T.; Lu, H. Securing vehicle-to-everything (V2X) communication platforms. *IEEE Trans. Intell. Veh.* **2020**, *5*, 693–713. [CrossRef]
56. Lim, H.; Hwang, S.; Myung, H. ERASOR: Egocentric ratio of pseudo occupancy-based dynamic object removal for static 3D point cloud map building. *IEEE Robot. Autom. Lett.* **2021**, *6*, 2272–2279. [CrossRef]
57. Jeon, H.; Eun, Y. A stealthy sensor attack for uncertain cyber-physical systems. *IEEE Internet Things J.* **2019**, *6*, 6345–6352. [CrossRef]
58. Kim, S.; Eun, Y.; Park, K.J. Stealthy sensor attack detection and real-time performance recovery for resilient CPS. *IEEE Trans. Ind. Inform.* **2021**. [CrossRef]
59. Hoagg, J.B.; Bernstein, D.S. Nonminimum-phase zeros—Much to do about nothing—Classical control—Revisited part II. *IEEE Control Syst. Mag.* **2007**, *27*, 45–57. [CrossRef]
60. Teixeira, A.; Shames, I.; Sandberg, H.; Johansson, K.H. Revealing stealthy attacks in control systems. In Proceedings of the Allerton Conference on Communication, Control, and Computing (Allerton), Monticello, IL, USA, 1–5 October 2012; pp. 1806–1813.
61. Yuz, J.I.; Goodwin, G.C. *Sampled-Data Models for Linear and Nonlinear Systems*; Springer: Berlin/Heidelberg, Germany, 2014.
62. Kim, J.; Park, G.; Shim, H.; Eun, Y. Zero-stealthy attack for sampled-data control systems: The case of faster actuation than sensing. In Proceedings of the IEEE Conference on Decision and Control (CDC), Las Vegas, NV, USA, 12–14 December 2016; pp. 5956–5961.
63. Mo, Y.; Sinopoli, B. Secure control against replay attacks. In Proceedings of the Allerton Conference on Communication, Control, and Computing (Allerton), Monticello, IL, USA, 30 September–2 October 2009; pp. 911–918.
64. Smith, R.S. Covert misappropriation of networked control systems: Presenting a feedback structure. *IEEE Control Syst. Mag.* **2015**, *35*, 82–92.
65. Schellenberger, C.; Zhang, P. Detection of covert attacks on cyber-physical systems by extending the system dynamics with an auxiliary system. In Proceedings of the IEEE Conference on Decision and Control (CDC), Melbourne, VIC, Australia, 12–15 December 2017; pp. 1374–1379.

66. Cetinkaya, A.; Ishii, H.; Hayakawa, T. An overview on denial-of-service attacks in control systems: Attack models and security analyses. *Entropy* **2019**, *21*, 210. [CrossRef]
67. De Persis, C.; Tesi, P. Input-to-state stabilizing control under denial-of-service. *IEEE Trans. Autom. Control* **2015**, *60*, 2930–2944. [CrossRef]
68. Liu, J.; Yin, T.; Shen, M.; Xie, X.; Cao, J. State estimation for cyber–physical systems with limited communication resources, sensor saturation and denial-of-service attacks. *ISA Trans.* **2020**, *104*, 101–114. [CrossRef]
69. Cetinkaya, A.; Ishii, H.; Hayakawa, T. A probabilistic characterization of random and malicious communication failures in multi-hop networked control. *SIAM J. Control Optim.* **2018**, *56*, 3320–3350. [CrossRef]
70. Kim, W.; Park, J.; Jo, J.; Lim, H. Covert jamming using fake ACK frame injection on IEEE 802.11 wireless LANs. *IEEE Wirel. Commun. Lett.* **2019**, *8*, 1502–1505. [CrossRef]
71. Rose, S.H.; Jayasree, T. Detection of jamming attack using timestamp for WSN. *Ad Hoc Netw.* **2019**, *91*, 101874. [CrossRef]
72. Wu, W.; Li, R.; Xie, G.; An, J.; Bai, Y.; Zhou, J.; Li, K. A survey of intrusion detection for in-vehicle networks. *IEEE Trans. Intell. Transp. Syst.* **2020**, *21*, 919–933. [CrossRef]
73. Lin, Y.; Chen, C.; Xiao, F.; Avatefipour, O.; Alsubhi, K.; Yunianta, A. An evolutionary deep learning anomaly detection framework for in-vehicle networks—CAN bus. *IEEE Trans. Ind. Appl.* **2020**. [CrossRef]
74. Lakshminarayana, S.; Karachiwala, J.S.; Chang, S.Y.; Revadigar, G.; Kumar, S.L.S.; Yau, D.K.; Hu, Y.C. Signal jamming attacks against communication-based train control: Attack impact and countermeasure. In Proceedings of the ACM Conference on Security & Privacy in Wireless and Mobile Networks, Stockholm, Sweden, 18–20 June 2018; pp. 160–171.
75. Chang, S.Y.; Tran, B.A.N.; Hu, Y.C.; Jones, D.L. Jamming with power boost: Leaky waveguide vulnerability in train systems. In Proceedings of the IEEE International Conference on Parallel and Distributed Systems (ICPADS), Melbourne, VIC, Australia, 14–17 December 2015; pp. 37–43.
76. Ali, S.; Al Balushi, T.; Nadir, Z.; Hussain, O.K. WSN security mechanisms for CPS. In *Cyber Security for Cyber Physical Systems*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 65–87.
77. Hsiao, H.C.; Studer, A.; Chen, C.; Perrig, A.; Bai, F.; Bellur, B.; Iyer, A. Flooding-resilient broadcast authentication for VANETs. In Proceedings of the 17th Annual International Conference on Mobile Computing and Networking, Las Vegas, NV, USA, 19–23 September 2011; pp. 193–204.
78. Donkers, M.; Daafouz, J.; Heemels, W. Output-based controller synthesis for networked control systems with periodic protocols and time-varying transmission intervals and delays. *IFAC Proc. Vol.* **2014**, *47*, 6478–6483. [CrossRef]
79. Kwon, Y.M.; Yu, J.; Cho, B.M.; Eun, Y.; Park, K.J. Empirical analysis of MAVLink protocol vulnerability for attacking unmanned aerial vehicles. *IEEE Access* **2018**, *6*, 43203–43212. [CrossRef]
80. Chen, Y.; Poskitt, C.M.; Sun, J. Learning from mutants: Using code mutation to learn and monitor invariants of a cyber-physical system. In Proceedings of the IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 20–24 May 2018; pp. 648–660.
81. Liu, K.; Dolan-Gavitt, B.; Garg, S. Fine-pruning: Defending against backdooring attacks on deep neural networks. In *Proceedings of the International Symposium on Research in Attacks, Intrusions, and Defenses*; Springer: Cham, Switzerland, 2018; pp. 273–294.
82. Farwell, J.P.; Rohozinski, R. Stuxnet and the future of cyber war. *Survival* **2011**, *53*, 23–40. [CrossRef]
83. Sani, A.S.; Yuan, D.; Yeoh, P.L.; Qiu, J.; Bao, W.; Vucetic, B.; Dong, Z.Y. CyRA: A real-time risk-based security assessment framework for cyber attacks prevention in industrial control systems. In Proceedings of the IEEE Power & Energy Society General Meeting (PESGM), Atlanta, GA, USA, 4–8 August 2019; pp. 1–5.
84. Fang, D.; Xu, S.; Sharif, H. Security analysis of wireless train control systems. In Proceedings of the IEEE Globecom Workshops, Waikoloa, HI, USA, 9–13 December 2019; pp. 1–6.
85. Zhong, H.; Liao, C.; Squicciarini, A.C.; Zhu, S.; Miller, D. Backdoor embedding in convolutional neural network models via invisible perturbation. In Proceedings of the ACM Conference on Data and Application Security and Privacy, New Orleans, LA, USA, 16–18 March 2020; pp. 97–108.
86. Abdi, F.; Chen, C.Y.; Hasan, M.; Liu, S.; Mohan, S.; Caccamo, M. Preserving physical safety under cyber attacks. *IEEE Internet Things J.* **2019**, *6*, 6285–6300. [CrossRef]
87. Yoon, M.K.; Liu, B.; Hovakimyan, N.; Sha, L. VirtualDrone: Virtual sensing, actuation, and communication for attack-resilient unmanned aerial systems. In Proceedings of the ACM/IEEE International Conference on Cyber-Physical Systems (ICCPS), Pittsburgh, PA, USA, 18–20 April 2017; pp. 143–154.
88. Zhou, Y.; Feng, D. Side-channel attacks: Ten years after its publication and the impacts on cryptographic module security testing. *IACR Cryptol. ePrint Arch.* **2005**, *2005*, 388.
89. Lawson, N. Side-channel attacks on cryptographic software. *IEEE Secur. Priv.* **2009**, *7*, 65–68. [CrossRef]
90. Jang, M.; Lee, S.; Kung, J.; Kim, D. Defending against flush+reload attack with DRAM cache by bypassing shared SRAM cache. *IEEE Access* **2020**, *8*, 179837–179844. [CrossRef]
91. Li, C.; Wang, Z.; Hou, X.; Chen, H.; Liang, X.; Guo, M. Power attack defense: Securing battery-backed data centers. *ACM SIGARCH Comput. Archit. News* **2016**, *44*, 493–505. [CrossRef]
92. Gao, X.; Xu, Z.; Wang, H.; Li, L.; Wang, X. Why "some" like it hot too: Thermal attack on data centers. In Proceedings of the ACM SIGMETRICS/International Conference on Measurement and Modeling of Computer Systems, Urbana, IL, USA, 5–9 June 2017; pp. 23–24.

93. Fournaris, A.P.; Pocero Fraile, L.; Koufopavlou, O. Exploiting hardware vulnerabilities to attack embedded system devices: A survey of potent microarchitectural attacks. *Electronics* **2017**, *6*, 52. [CrossRef]
94. Mutlu, O. The RowHammer problem and other issues we may face as memory becomes denser. In Proceedings of the Design, Automation & Test in Europe Conference & Exhibition (DATE), Lausanne, Switzerland, 27–31 March 2017; pp. 1116–1121.
95. Zhao, P.; Wang, S.; Gongye, C.; Wang, Y.; Fei, Y.; Lin, X. Fault sneaking attack: A stealthy framework for misleading deep neural networks. In Proceedings of the 56th ACM/IEEE Design Automation Conference (DAC), Las Vegas, NV, USA, 2–6 June 2019; pp. 1–6.
96. Hong, S.; Frigo, P.; Kaya, Y.; Giuffrida, C.; Dumitraș, T. Terminal brain damage: Exposing the graceless degradation in deep neural networks under hardware fault attacks. In Proceedings of the USENIX Security Symposium (USENIX Security 19), Santa Clara, CA, USA, 14–16 August 2019; pp. 497–514.
97. Handa, A.; Sharma, A.; Shukla, S.K. Machine learning in cybersecurity: A review. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **2019**, *9*, e1306. [CrossRef]
98. Liu, W.; Wang, Z.; Liu, X.; Zeng, N.; Liu, Y.; Alsaadi, F.E. A survey of deep neural network architectures and their applications. *Neurocomputing* **2017**, *234*, 11–26. [CrossRef]
99. Elsayed, M.; Erol-Kantarci, M. Deep Q-learning for low-latency tactile applications: Microgrid communications. In Proceedings of the 2018 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm), Aalborg, Denmark, 29–31 October 2018; pp. 1–6.
100. Afanador, N.L.; Smolinska, A.; Tran, T.N.; Blanchet, L. Unsupervised random forest: A tutorial with case studies. *J. Chemom.* **2016**, *30*, 232–241. [CrossRef]
101. Burges, C.J. A tutorial on support vector machines for pattern recognition. *Data Min. Knowl. Discov.* **1998**, *2*, 121–167. [CrossRef]
102. Gómez, Á.L.P.; Maimó, L.F.; Celdran, A.H.; Clemente, F.J.G.; Sarmiento, C.C.; Masa, C.J.D.C.; Nistal, R.M. On the generation of anomaly detection datasets in industrial control systems. *IEEE Access* **2019**, *7*, 177460–177473. [CrossRef]
103. Cai, F.; Koutsoukos, X. Real-time out-of-distribution detection in learning-enabled cyber-physical systems. In Proceedings of the ACM/IEEE International Conference on Cyber-Physical Systems (ICCPS), Sydney, NSW, Australia, 21–25 April 2020; pp. 174–183.
104. Javed, A.R.; Usman, M.; Rehman, S.U.; Khan, M.U.; Haghighi, M.S. Anomaly detection in automated vehicles using multistage attention-based convolutional neural network. *IEEE Trans. Intell. Transp. Syst.* **2020**, 1–10. [CrossRef]
105. Wang, Y.; Masoud, N.; Khojandi, A. Real-time sensor anomaly detection and recovery in connected automated vehicle sensors. *IEEE Trans. Intell. Transp. Syst.* **2021**, *22*, 1411–1421. [CrossRef]
106. Lin, Q.; Adepu, S.; Verwer, S.; Mathur, A. TABOR: A graphical model-based approach for anomaly detection in industrial control systems. In Proceedings of the Asia Conference on Computer and Communications Security, Incheon, Korea, 4 June 2018; pp. 525–536.
107. Keliris, A.; Salehghaffari, H.; Cairl, B.; Krishnamurthy, P.; Maniatakos, M.; Khorrami, F. Machine learning-based defense against process-aware attacks on industrial control systems. In Proceedings of the IEEE International Test Conference (ITC), Fort Worth, TX, USA, 15–17 November 2016; pp. 1–10.
108. Paridari, K.; O'Mahony, N.; Mady, A.E.D.; Chabukswar, R.; Boubekeur, M.; Sandberg, H. A framework for attack-resilient industrial control systems: Attack detection and controller reconfiguration. *Proc. IEEE* **2018**, *106*, 113–128. [CrossRef]
109. KI Williams, C. *Gaussian Processes for Machine Learning*; Taylor & Francis Group: Abingdon, UK, 2006.
110. Ravikumar, G.; Govindarasu, M. Anomaly detection and mitigation for wide-area damping control using machine learning. *IEEE Trans. Smart Grid* **2020**. [CrossRef]
111. Chakhchoukh, Y.; Liu, S.; Sugiyama, M.; Ishii, H. Statistical outlier detection for diagnosis of cyber attacks in power state estimation. In Proceedings of the IEEE Power and Energy Society General Meeting (PESGM), Boston, MA, USA, 17–21 July 2016; pp. 1–5.
112. Ahmed, S.; Lee, Y.; Hyun, S.H.; Koo, I. Feature selection–based detection of covert cyber deception assaults in smart grid communications networks using machine learning. *IEEE Access* **2018**, *6*, 27518–27529. [CrossRef]
113. Karimipour, H.; Dehghantanha, A.; Parizi, R.M.; Choo, K.K.R.; Leung, H. A deep and scalable unsupervised machine learning system for cyber-attack detection in large-scale smart grids. *IEEE Access* **2019**, *7*, 80778–80788. [CrossRef]
114. Lyamin, N.; Kleyko, D.; Delooz, Q.; Vinel, A. AI-based malicious network traffic detection in VANETs. *IEEE Netw.* **2018**, *32*, 15–21. [CrossRef]
115. Jeong, W.; Han, S.; Choi, E.; Lee, S.; Choi, J.W. CNN-based adaptive source node identifier for controller area network (CAN). *IEEE Trans. Veh. Technol.* **2020**, *69*, 13916–13920. [CrossRef]
116. Singh, P.K.; Gupta, R.R.; Nandi, S.K.; Nandi, S. Machine learning based approach to detect wormhole attack in VANETs. In *Proceedings of the Workshops of the International Conference on Advanced Information Networking and Applications*; Springer: Cham, Switzerland, 2019; pp. 651–661.
117. Maglaras, L.A.; Jiang, J.; Cruz, T. Integrated OCSVM mechanism for intrusion detection in SCADA systems. *Electron. Lett.* **2014**, *50*, 1935–1936. [CrossRef]
118. Khan, I.A.; Pi, D.; Khan, Z.U.; Hussain, Y.; Nawaz, A. HML-IDS: A hybrid-multilevel anomaly prediction approach for intrusion detection in SCADA systems. *IEEE Access* **2019**, *7*, 89507–89521. [CrossRef]
119. Hassan, M.M.; Huda, S.; Sharmeen, S.; Abawajy, J.; Fortino, G. An adaptive trust boundary protection for IIoT networks using deep-learning feature-extraction-based semisupervised model. *IEEE Trans. Ind. Inform.* **2021**, *17*, 2860–2870. [CrossRef]

120. Gao, J.; Gan, L.; Buschendorf, F.; Zhang, L.; Liu, H.; Li, P.; Dong, X.; Lu, T. Omni SCADA intrusion detection using deep learning algorithms. *IEEE Internet Things J.* **2021**, *8*, 951–961. [CrossRef]
121. Yin, X.C.; Liu, Z.G.; Nkenyereye, L.; Ndibanje, B. Toward an applied cyber security solution in IoT-based smart grids: An intrusion detection system approach. *Sensors* **2019**, *19*, 4952. [CrossRef]
122. Kwon, S.; Yoo, H.; Shon, T. IEEE 1815.1-based power system security With bidirectional RNN-based network anomalous attack detection for cyber-physical system. *IEEE Access* **2020**, *8*, 77572–77586. [CrossRef]
123. Yang, H.; Cheng, L.; Chuah, M.C. Deep-learning-based network intrusion detection for SCADA systems. In Proceedings of the IEEE Conference on Communications and Network Security (CNS), Washington, DC, USA, 10–12 June 2019; pp. 1–7.
124. da Silva, E.G.; Silva, A.S.d.; Wickboldt, J.A.; Smith, P.; Granville, L.Z.; Schaeffer-Filho, A. A one-class NIDS for SDN-based SCADA systems. In Proceedings of the IEEE 40th Annual Computer Software and Applications Conference (COMPSAC), Atlanta, GA, USA, 10–14 June 2016; pp. 303–312.
125. Yoon, M.K.; Mohan, S.; Choi, J.; Christodorescu, M.; Sha, L. Learning execution contexts from system call distribution for anomaly detection in smart embedded system. In Proceedings of the Second International Conference on Internet-of-Things Design and Implementation, Pittsburgh, PA, USA, 18–21 April 2017; pp. 191–196.
126. Wang, W.; Zhao, M.; Wang, J. Effective android malware detection with a hybrid model based on deep autoencoder and convolutional neural network. *J. Ambient Intell. Humaniz. Comput.* **2019**, *10*, 3035–3043. [CrossRef]
127. Patel, N.K.; Krishnamurthy, P.; Amrouch, H.; Henkel, J.; Shamouilian, M.; Karri, R.; Khorrami, F. Towards a new thermal monitoring based framework for embedded CPS device security. *IEEE Trans. Dependable Secur. Comput.* **2020**. [CrossRef]
128. Wang, H.; Sayadi, H.; Sasan, A.; Rafatirad, S.; Mohsenin, T.; Homayoun, H. Comprehensive evaluation of machine learning countermeasures for detecting microarchitectural side-channel attacks. In Proceedings of the 2020 on Great Lakes Symposium on VLSI, 7–9 September 2020; pp. 181–186.
129. Mushtaq, M.; Akram, A.; Bhatti, M.K.; Chaudhry, M.; Yousaf, M.; Farooq, U.; Lapotre, V.; Gogniat, G. Machine learning for security: The case of side-channel attack detection at run-time. In Proceedings of the IEEE International Conference on Electronics, Circuits and Systems (ICECS), Bordeaux, France, 9–12 December 2018; pp. 485–488.
130. Cho, J.; Kim, T.; Kim, S.; Im, M.; Kim, T.; Shin, Y. Real-time detection for cache side channel attack using performance counter monitor. *Appl. Sci.* **2020**, *10*, 984. [CrossRef]
131. Li, C.; Gaudiot, J.L. Detecting malicious attacks exploiting hardware vulnerabilities using performance counters. In Proceedings of the IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC), Milwaukee, WI, USA, 15–19 July 2019; pp. 588–597.
132. Chakraborty, A.; Alam, M.; Mukhopadhyay, D. Deep learning based diagnostics for Rowhammer protection of DRAM chips. In Proceedings of the IEEE 28th Asian Test Symposium (ATS), Kolkata, India, 10–13 December 2019; pp. 86–91.
133. Li, J.; Xia, Y.; Qi, X.; Gao, Z. On the necessity, scheme, and basis of the linear–nonlinear switching in active disturbance rejection control. *IEEE Trans. Ind. Electron.* **2017**, *64*, 1425–1435. [CrossRef]
134. Quinonez, R.; Giraldo, J.; Salazar, L.; Bauman, E.; Cardenas, A.; Lin, Z. SAVIOR: Securing autonomous vehicles with robust physical invariants. In Proceedings of the USENIX Security Symposium (USENIX Security 20), 12–14 August 2020; pp. 895–912.
135. Bak, S.; Chivukula, D.K.; Adekunle, O.; Sun, M.; Caccamo, M.; Sha, L. The system-level simplex architecture for improved real-time embedded system safety. In Proceedings of the IEEE Real-Time and Embedded Technology and Applications Symposium, San Francisco, CA, USA, 13–16 April 2009; pp. 99–107.
136. Hu, F.; Hao, Q.; Bao, K. A survey on software-defined network and OpenFlow: From concept to implementation. *IEEE Commun. Surv. Tutor.* **2014**, *16*, 2181–2206. [CrossRef]
137. Togou, M.A.; Chekired, D.A.; Khoukhi, L.; Muntean, G.M. A distributed control plane for path computation scalability in software-defined networks. In Proceedings of the IEEE Global Communications Conference (GLOBECOM), Abu Dhabi, United Arab Emirates, 9–13 December 2018; pp. 1–6.
138. Sood, K.; Yu, S.; Xiang, Y. Software-defined wireless networking opportunities and challenges for Internet-of-Things: A review. *IEEE Internet Things J.* **2016**, *3*, 453–463. [CrossRef]
139. Yang, H.; Zhan, K.; Kadoch, M.; Liang, Y.; Cheriet, M. BLCS: Brain-like distributed control security in cyber physical systems. *IEEE Netw.* **2020**, *34*, 8–15. [CrossRef]
140. Jin, D.; Li, Z.; Hannon, C.; Chen, C.; Wang, J.; Shahidehpour, M.; Lee, C.W. Toward a cyber resilient and secure microgrid using software-defined networking. *IEEE Trans. Smart Grid* **2017**, *8*, 2494–2504. [CrossRef]
141. Wang, H.; Xu, L.; Gu, G. Floodguard: A DoS attack prevention extension in software-defined networks. In Proceedings of the IEEE/IFIP International Conference on Dependable Systems and Networks, Rio de Janeiro, Brazil, 22–25 June 2015; pp. 239–250.
142. Shang, G.; Zhe, P.; Bin, X.; Aiqun, H.; Kui, R. FloodDefender: Protecting data and control plane resources under SDN-aimed DoS attacks. In Proceedings of the IEEE INFOCOM 2017-IEEE Conference on Computer Communications, Atlanta, GA, USA, 1–4 May 2017; pp. 1–9.
143. Adi, Y.; Baum, C.; Cisse, M.; Pinkas, B.; Keshet, J. Turning your weakness into a strength: Watermarking deep neural networks by backdooring. In Proceedings of the USENIX Security Symposium (USENIX Security 18), Baltimore, MD, USA, 15–17 August 2018; pp. 1615–1631.
144. Creswell, A.; White, T.; Dumoulin, V.; Arulkumaran, K.; Sengupta, B.; Bharath, A.A. Generative adversarial networks: An overview. *IEEE Signal Process. Mag.* **2018**, *35*, 53–65. [CrossRef]

145. Li, D.; Chen, D.; Jin, B.; Shi, L.; Goh, J.; Ng, S.K. MAD-GAN: Multivariate anomaly detection for time series data with generative adversarial networks. In *Proceedings of the International Conference on Artificial Neural Networks*; Springer: Cham, Switzerland, 2019; pp. 703–716.
146. Freitas de Araujo-Filho, P.; Kaddoum, G.; Campelo, D.R.; Gondim Santos, A.; Macêdo, D.; Zanchettin, C. Intrusion detection for cyber–physical systems using generative adversarial networks in fog environment. *IEEE Internet Things J.* **2021**, *8*, 6247–6256. [CrossRef]
147. Chhetri, S.R.; Lopez, A.B.; Wan, J.; Al Faruque, M.A. GAN-Sec: Generative adversarial network modeling for the security analysis of cyber-physical production systems. In Proceedings of the Design, Automation & Test in Europe Conference & Exhibition (DATE), Florence, Italy, 25–29 March 2019; pp. 770–775.
148. Shahriar, M.H.; Haque, N.I.; Rahman, M.A.; Alonso, M. G-IDS: Generative adversarial networks assisted intrusion detection system. In Proceedings of the IEEE 44th Annual Computers, Software, and Applications Conference (COMPSAC), Madrid, Spain, 13–17 July 2020; pp. 376–385.
149. Lippmann, R.; Haines, J.W.; Fried, D.J.; Korba, J.; Das, K. The 1999 DARPA off-line intrusion detection evaluation. *Comput. Netw.* **2000**, *34*, 579–595. [CrossRef]
150. Zixu, T.; Liyanage, K.S.K.; Gurusamy, M. Generative adversarial network and auto encoder based anomaly detection in distributed IoT networks. In Proceedings of the GLOBECOM 2020—2020 IEEE Global Communications Conference, Taipei, Taiwan, 7–11 December 2020; pp. 1–7.