# Hybrid Digital Twin Architecture for Power System Cyber Security Analysis

V. Ayyalusamy [1], B. Sivaneasan [1], NK Kandasamy [2], J. F. Xiao [3] and Abidi K [3]
[1] Singapore Institute of Technology, Singapore
[2] Lite-On, Singapore
[3] Newcastle University, Singapore
sivaneasan@singaporetech.edu.sg

***Abstract*—This paper presents the system architecture and communication system design of a new hybrid digital twin (HDT) system for smart grid cyber security studies. The HDT system emulates the smart grid cyber-physical system using MATLAB-SIMULINK model (digital) and single board computers (physical). The methodology to establish the HDT including the network segmentation using configurable network switch and communication protocol using Node-RED is described in detail. The performance of the HDT communication system in terms of round trip time between the digital and physical models and within network components in the physical model is also studied. The development of the HDT will provide a new reconfigurable, scalable and low-cost platform to study the cyber-security vulnerabilities in smart grids and other cyber-physical systems.**

***Index Terms*—Hybrid digital twin, cyber security, industrial control system, cyber physical system, smart grid.**

## I. Introduction

Supervisory control and data acquisition (SCADA) system forms the backbone of smart grids. From the cyber security aspect of a smart grid, the SCADA system are the greatest concerns as it provides the cyber entry point to the smart grid through wired/wireless communication. For many years, the SCADA systems benefited from security through obscurity. As a result, these systems often did not even implement the most basic security measures. The industry is only now beginning to implement modest security measures.

The smart grids are dominated by power electronics converters used for interfacing distributed generations, energy storage and loads. In such systems, the physical electrical components are tightly interconnected by information and communication technologies, and their operations are tightly coupled to cyber system functionality. Vulnerabilities are brought to the system as a result of this tight interconnection between cyber and physical components.

However, it is impractical to test real-world security flaws, countermeasures, and performance impact on a live system without interfering with critical functions that could affect grid operations [1]. Researchers and system operators place a high premium on test-beds that accurately simulate the operation of critical infrastructures. According to Cintuglu et al. [2], evaluating defense mechanisms in a physical test bed enables a more seamless translation of developed technologies. However, the feasibility of reproducing, scaling, and modifying such test-beds is relatively low. It is prohibitively expensive to replicate such test-beds for researchers working on cyber security technologies with a TRL lower than TRL6 (technology readiness level 6). The total cost of ownership, which includes operation and maintenance, limits the use case to a smaller community of researchers.

As a result, there is a practical requirement for Digital Twins (DTs) that can replicate the physical system with sufficient fidelity and an emulated virtual network that is equivalent to a physical test-bed. Another critical advantage of such DTs is their reproducibility, scalability, and adaptability. However, even with DTs, the exercise(s) are not trivial, because creating a digital twin itself is a complex and expensive process given the real-time modelling and simulation requirements. Many real-time simulation platforms such as Simulink Real-time, Opal-RT, RTDS and Typhoon HIL, etc., provide options for creating physical models for different CPS on same hardware, however, no such technology is available for creating controllers and the corresponding network. Pure DT is not able to test the actual Industrial Control Systems like intelligent electronic devices (IEDs), programmalble logic controllers (PLCs), etc.

In this paper, a Hybrid Digital Twin (HDT) is proposed, designed and developed to bridge this gap and provide cost-effective solution for cyber-security studies in any industry sector. The major benefits of the HDT is that it can replicate any real industrial hardware and network component and quickly establish highly configurable, low cost and scalable prototype. The procedure used for creating the HDT in this paper can be extended to any type of system whether a test-bed or an actual plant.

## II. Literature Review

In order to maintain the efficient and secure operation of the smart grid, it is necessary to integrate the physical power system with it's cyber system [3]. The integration of the physical power system with the cyber system [4], [5] evolves into a strongly coupled cyber-physical power system (CPPS). A CPPS is a system that combines and coordinates the virtual and physical power system elements. These systems are distributed networks executing in unpredictable environments and built from control and embedded systems to monitor and regulate the physical power system in real time.

At a high level, a digital twin can be defined as a virtual replica of an object from the physical world [6]. The environment that maintains a digital twin requires a standard structure, end-to-end connectivity, a communication protocol with backward compatibility, and a standardized data format with which the virtual replica can communicate with the real world object or system.

Many studies in the literature have started developing and describing conceptual designs and applications of digital twins. A description of the expected main building blocks for a general cyber-physical system was introduced in [7]. In [8], the authors presented a digital twin architecture for the security of an industrial automation system, where they proposed a security-oriented digital twin for the Programmable Logic Control (PLC) software update process. In [9], the author provided a cloud-based digital twin for a cyber-physical system with an application to the social internet of vehicles for driving assistance application.

The main characteristic of digital twin are the strong interdependency between the cyber and physical systems. In [10], [11], [12] the authors have investigated the impacts of various cyber contingency on a physical system using model-based methods. The authors in [13], [14], [15], performed extensive research on the analysis of different types of cyber-attacks like denial-of-service attack, false data injection attack, and man-in-the-middle attack in CPPS and the results has shown that these attacks jeopardize of stability of the power system. To protect the complex power grid control networks of CPPS, it is necessary to perform the risk and vulnerability assessment under cyber-attacks [16], [17], [18].

There is only one work in the literature on hybrid DT for smart grids fault prediction where a low latency IIoT network and data driven machine learning computational resources is used to represent the cyber system while the physical system is represented by a digital model of the grid [19]. To the best of the authors' knowledge, the work presented in this paper will be the first highly configurable, low cost and scalable hybrid digital twin established using single board computers to mimic the smart grid cyber-physical components and the communication network with high fidelity.

## III. Architecture of Hybrid Digital Twin (HDT)

Figure 1 shows the proposed HDT architecture where the physical system will be represented by a MATLAB-SIMULINK digital model (referred as digital model of the HDT) while the cyber components are establish using multiple single board computers (referred as physical model). Simscape Electrical Toolbox provides necessary tools for modeling a smart grid with various components which will be executed on the development computer. The network components of the smart grid such as IEDs, PLCs, smart meters, human machine interface (HMI), and historian will be replaced with Raspberry Pi 3 Model B+. A 24 port TPLink TL-SG3428MP configurable switch together with OC200 Omada Hardware Controller are used to cluster the Raspberry Pis into different physical network segments to mimic actual cyber system configuration.

In actual power system, the current transformers (CTs) and voltage transformers (VTs) are used in the process level to convert the electrical signals into physical signals that are connected via wires to metering devices. Similarly, control signals to close/open circuit breakers are physical signals send to IEDs/PLCs in the bay level area via wires. As shown in Fig. 1 this information flow between the digital and physical models of the HDT is achieved through MQTT (Message Queuing Telemetry Transport) protocol. The communication between the 24 stack of Raspberry Pis is achieved through different power system protocols including Modbus, IEC61850 MMS, IEC61850 Goose and IEC104. Node-red programming tool is used to established the different functionalities of the network component and their communication protocols.

### A. HDT Network Segments

As an illustration, in Fig. 2, the developed HDT consists of five network segments, namely, Generation, Transmission, Microgrid, Smart home and Controller. The generation, transmission, microgrid and smart home segments has one IED, one PLC, two smart meters and one attacker gateway. The Controller segment has SCADA HMI, PLC, historian and attacker gateway. However, it must be noted that, the number of segments and the associated network components can be quickly reconfigured according to the CPS system being modelled.

The developed physical model of the HDT that replicates the five network segments is shown in Fig. 3. All 24 Raspberry Pis are connected to the switch using Ethernet LAN cables. Network segment and device mapping are defined by the switch using VLAN based on IP address and port configuration. In this case, all the network components in the generation segment will start with the address 191.168.1.2XX, and network components in transmission network will start with 192.168.2.2XX. Network components in micorgrid, smarthome and controller segments will have subnet addresses ending with 3.2XX, 4.2XX, 5.2XX, respectively.

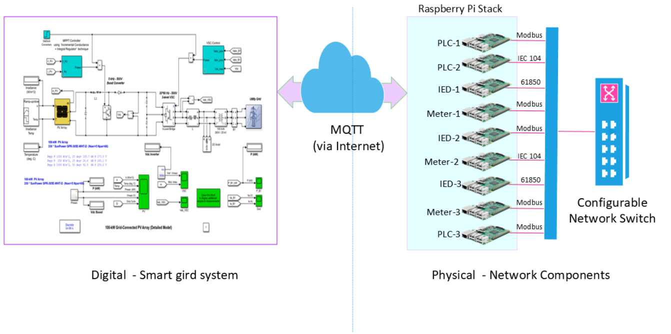


Fig. 1. Hybrid Digital Twin Architecture

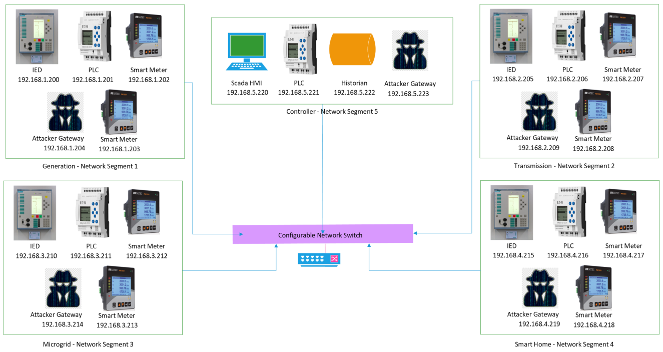


Fig. 2. Network Segment Diagram

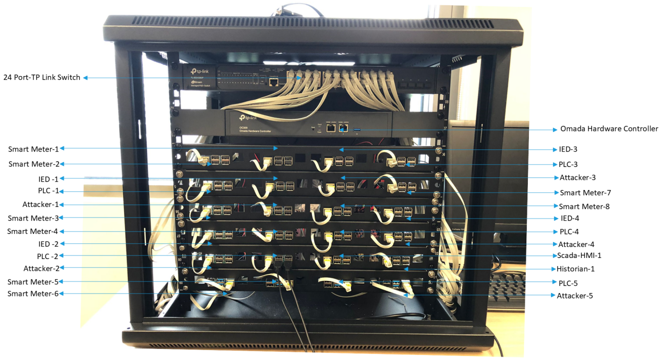


Fig. 3. Physical Hardware Components

Debian OS is installed in all 24 Raspberry PIs. For each Raspberry Pi, a script is provided for installing Node.js, npm, and Node-RED. Additionally, the script can be used to upgrade an existing installation when a new version becomes available. Node-RED is a programming language that enables novel and interesting connections between hardware devices, APIs, and online services. It includes a browser-based editor that simplifies the process of wiring together flows by utilizing a diverse palette of nodes that can be deployed to the runtime with a single click. After installing Node-red in all 24 Raspberry PIs, the server is started from terminal. Using localhost or with the respective raspberry Pi IP address, Node-RED visual interface can be launched via browser.

For the benefit of the research community, the authors have shared the script and command for Raspberry PIs setup as different network components at GitHub. **https://raw.githubusercontent.com/node-red/linux-installers/master/deb/update-nodejs-and-nodered**

### *B. HDT Communication*

Figure 4 illustrates the communication between an IEDs in the physical model sending and receiving information from the MATLAB digital model and other network components on the physical model. The CT and VT signals in the digital model in MATLAB is transferred to the IED in the physical model via MQTT protocol. MQTT is a network protocol based on the publisher-subscriber paradigm for interacting between devices connected to a centralized intermediary broker (a host running on a cloud). It is well suited for non-synchronous communication, which best represents the data bus in actual power system. Modbus, IEC61850, and IEC104 protocols can be used to communicate between the network components in the physical model using Node-RED in the Raspberry PIs.

In this paper, two protocols which have been established and tested in HDT is discussed, namely, (a) MQTT protocol which is used to transferring data between digital model and physical model, and (b) Modbus protocol which is used for communication between all 24 Raspberry PIs.

## IV. Communication Performance Evaluation

MQTT is a publish/subscribe communication protocol that uses TCP/IP sockets or WebSockets. A client device connects to the MQTT broker and can either publish to or subscribe to a channel's updates. In the HDT, mosquitto is used as the MQTT broker and paho-mqtt python library act as mqtt client. MATLAB model via the MQTT API and 4 Raspberry Pis via Node-RED are configured as clients. MATLAB MQTT API establishes connection with MQTT broker and then publishes the voltage/current value under the specific topic. These message will be received by everyone who are have subscribed this topic. The values published by the MATLAB MQTT API is received by Node-RED subscriber as shown in Fig. 5.

Figure 6 shows the Modbus communication flow design in Node-RED. It represents both server and client, one for read (Modbus Flex Getter) and another to write (Modbus Flex Writer). It therefore can act as both master as well as slave. Once the Modbus server is active, client device can read/write to coil and holding registers but only can read from discrete/input registers.

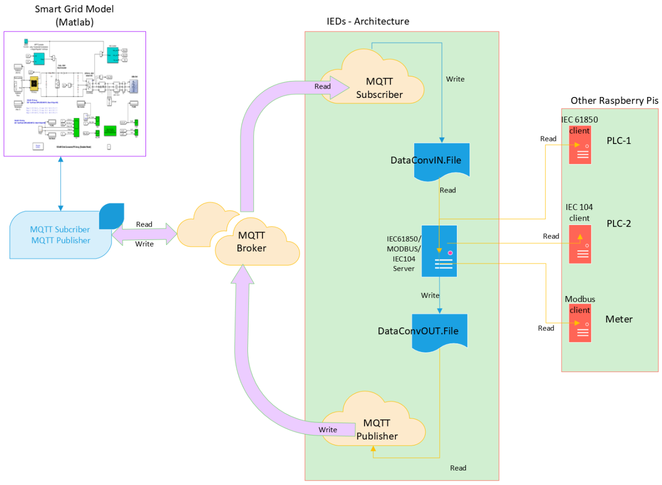
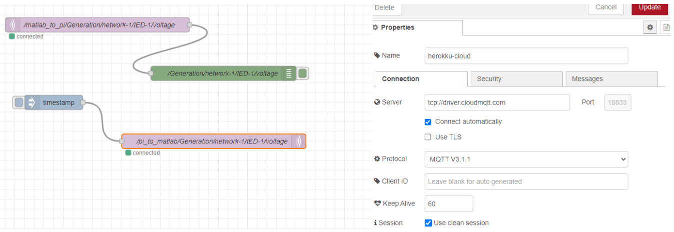


Fig. 4. Comms Protocol - Dataflow in HDT


/matlab_to_pi/Generation/network-1/IED-1/voltage
connected
/Generation/network-1/IED-1/voltage
timestamp
/pi_to_matlab/Generation/network-1/IED-1/voltage
connected
Delete
Cancel
Update
Properties
Connection
Security
Messages

Name: herokku-cloud

Server: tcp://driver.cloudmqtt.com Port: 18833

☑ Connect automatically

☐ Use TLS

Protocol: MQTT V3.1.1

Client ID: Leave blank for auto generated

Keep Alive: 60

Session: ☑ Use clean session

Fig. 5. MQTT Publish and Subscribe in Node-Red

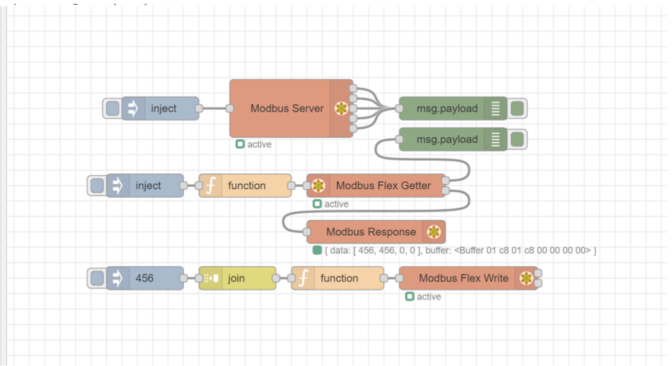


Fig. 6. Modbus Client/Server in Node-Red

A performance test is conducted to measure the data transmission delay between the digital model and physical model of the HDT (named TEST1) and between two Raspberry Pis (2) in the physical model (named TEST2). To execute TEST1, the MATLAB MQTT API will publish a message under a topic ”matlab to pi” on the mosquitto broker using the publish function. The Raspberry Pi which has subscribed to the topic ”matlab to pi” awaits for the message and when received, it will post a status message under the topic ”pi to matlab,” to which the MATLAB MQTT API has subscribed. A script runs on the MATLAB to determine the total round trip time (RTT) from the difference between the first publish time to the time the status message is received.

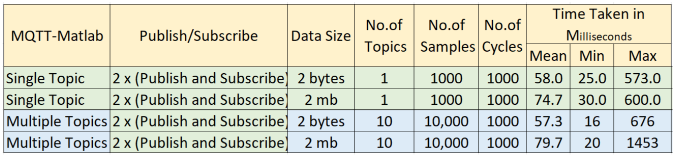
TEST1 considers three different factors that affect transmission time:

- Number of Topics The experiment considers messages sent both in sequential and in parallel under single and multiple topics (1 and 10 topics)
- Packet Size The experiment considers both small and large data packet size (2 bytes and 2MB)
- Number of Samples The experiment considers the number of samples to provide a better measure of the RTT. For single topic, 1000 samples are published and for multiple topics, 10,000 samples are published.

Each scenario goes through 1000 iteration cycles to obtain a trustworthy result. Fig. 7 provides the results obtained for different scenarios. It can be observed that the mean RTT for a single topic with a small packet size of 2 bytes is 58ms, with minimum and maximum values of 25ms and 573ms, respectively. However, the mean, minimum, and maximum for larger message sizes of 2MB under a single topic are 74ms, 30ms, and 600ms, respectively. The mean difference between 2 bytes and 2 MB for 1000 cycles is about 16ms.

For multiple topics (message sent via 10 parallel topics), the mean, minimum, and maximum times for smaller packet size of 2 bytes are 57ms, 16ms, and 676ms, respectively. However, for larger packet sizes of 2mb, the mean, minimum, and maximum response times are 79ms, 20ms, and 1453ms, respectively. There is a mean difference of 22ms between 2 bytes and 2 megabytes for 1000 cycles.

Interestingly, regardless of the number of topics, the mean RTT of the MQTT system is not significantly affected. Throughout all rounds of test, there was no packet loss. TEST1 shows that the MQTT performance is acceptable to replace the wired data-bus communication between the digital and physical models in HDT.



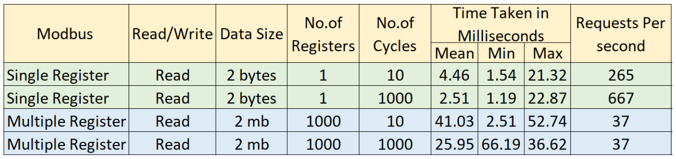
| MQTT-Matlab | Publish/Subscribe | Data Size | No.of Topics | No.of Samples | No.of Cycles | Time Taken in Milliseconds | | |
|---|---|---|---|---|---|---|---|---|
| | | | | | | Mean | Min | Max |
| Single Topic | 2 x (Publish and Subscribe) | 2 bytes | 1 | 1000 | 1000 | 58.0 | 25.0 | 573.0 |
| Single Topic | 2 x (Publish and Subscribe) | 2 mb | 1 | 1000 | 1000 | 74.7 | 30.0 | 600.0 |
| Multiple Topics | 2 x (Publish and Subscribe) | 2 bytes | 10 | 10,000 | 1000 | 57.3 | 16 | 676 |
| Multiple Topics | 2 x (Publish and Subscribe) | 2 mb | 10 | 10,000 | 1000 | 79.7 | 20 | 1453 |

Fig. 7. MQTT Data Tranmission Rate Between Raspberry Pis and Laptop(Matlab Model)

TEST2 is conducted between two Raspberry Pi in order to simulate the HDT operation for data exchange between network devices using Modbus protocol. To execute TEST2, the Modbus server is deployed using pymodbus on a Raspberry Pi. Client devices send requests to slave devices for reading Modbus registry values. The start time is recorded the moment when the client request is triggered. The request then reaches the server, which returns the values from the requested registries, before sending to the client device. The time at which the register values is received is recorded as the stop time. The RTT is the difference between the start and end times in the master Raspberry Pi device.

TEST2 also considers three different factors that affect transmission time:

- Number of Modbus Registry The experiment considers messages sent both in sequential and in parallel under single and multiple registries (1 and 1000 registers)
- Packet Size The experiment considers both small and large data packet size (2 bytes and 2MB). Since each Modbus registry stores a maximum of 2 bytes of data; for 2MB of data, the client will request data from 1000 registers to retrieve the data.
- Number of Samples The experiment considers the number of samples to provide a better measure of the RTT. For single register, 10,000 samples are read and for multiple registers, 100,000 samples are read.



| Modbus | Read/Write | Data Size | No.of Registers | No.of Cycles | Time Taken in Milliseconds | | | Requests Per second |
|---|---|---|---|---|---|---|---|---|
| | | | | | Mean | Min | Max | |
| Single Register | Read | 2 bytes | 1 | 10 | 4.46 | 1.54 | 21.32 | 265 |
| Single Register | Read | 2 bytes | 1 | 1000 | 2.51 | 1.19 | 22.87 | 667 |
| Multiple Register | Read | 2 mb | 1000 | 10 | 41.03 | 2.51 | 52.74 | 37 |
| Multiple Register | Read | 2 mb | 1000 | 1000 | 25.95 | 66.19 | 36.62 | 37 |

Fig. 8. Modbus Data Transmission Rate Between Raspberry Pis

Each scenario goes through 10 and 1000 iteration cycles to obtain a trustworthy result. Fig. 8 provides the results obtained for different scenarios. It can be seen that the mean, minimum, and maximum RTT for small packet size of 2 bytes fetched from single register for 10 cycles are 4.5ms, 1.5ms, and 21ms, respectively. However, the mean, minimum, and maximum RTT for 1000 cycles read from a single register are 2.5ms, 1.2ms, and 22ms, respectively.

For larger data sizes of 2MB, 1,000 register values are retrieved, and their respective mean, minimum, and maximum times for 10 iteration cycles are 41ms, 2.5ms, and 52ms. When run at 1,000 iteration cycles, the mean, minimum, and maximum times are 26ms, 66ms, and 36ms, respectively. There is a 15ms mean difference between 10 and 1000 cycles, and performance improves as the number of cycles increases. Comparing the mean of a single register to that of a multiple register reveals a roughly 10-fold increase. This is primarily due to the fact that 256 bytes is the maximum size of data that can be read per second from the holding registers (i.e., number of holding registers read per second is 123 registers). Therefore, the request per second for reading 1,000 registers is 37, regardless of the number of cycles. However, the RTT of the Modbus communication between the Raspberry Pis are well within the acceptable time for HDT application [20]

## V. Conclusion

This paper explored a new concept of hybrid digital twin (HDT) of a smart grid for cyber security analysis.The physical system was represented by a MATLAB-SIMULINK digital model while the cyber components are establish using multiple single board computers. The hardware architecture and communication system for the HDT was described. A performance evalaution of the developed HDT was conducted based on the round trip time (RTT) for the communication between the digital model and physical model as well as between network components in the physical model. Both the MQTT and Modbus communication showed acceptable results for HDT application with a mean RTT of 79ms and 26ms for worst case scenario, respectively. The development of the HDT will provide a reconfigurable, scalable and cost-effective solution for cyber-security studies.

## References

[1] M. Khan, O. Rehman, I. M. H. Rahman, and S. Ali, "Lightweight testbed for cybersecurity experiments in scada-based systems," in *2020 International Conference on Computing and Information Technology (ICCIT-1441)*, 2020, pp. 1–5.

[2] M. H. Cintuglu, O. A. Mohammed, K. Akkaya, and A. S. Uluagac, "A survey on smart grid cyber-physical system testbeds," *IEEE Communications Surveys & Tutorials*, vol. 19, pp. 446–464, 2017.

[3] L. Shi, Q. Dai, and Y. Ni, "Cyber–physical interactions in power systems: A review of models, methods, and applications," *Electric power systems research*, vol. 163, pp. 396–412, 2018.

[4] S. Suryanarayanan, T. M. Hansen, and R. Roche, *Cyber-physical-social systems and constructs in electric power engineering*. IET, 2016, vol. 2.

[5] Y. Cao, Y. Li, X. Liu, and C. Rehtanz, *Cyber-Physical Energy and Power Systems*. Springer, 2020.

[6] M. Eckhart and A. Ekelhart, "Digital twins for cyber-physical systems security: State of the art and outlook," *Security and quality in cyber-physical systems engineering*, pp. 383–412, 2019.

[7] S. Karnouskos, "Cyber-physical systems in the smartgrid," in *2011 9th IEEE International Conference on Industrial Informatics*, 2011, pp. 20–23.

[8] C. Gehrmann and M. Gunnarsson, "A digital twin based industrial automation and control system security architecture," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 1, pp. 669–680, 2020.

[9] K. M. Alam and A. El Saddik, "C2ps: A digital twin architecture reference model for the cloud-based cyber-physical systems," *IEEE Access*, vol. 5, pp. 2050–2062, 2017.

[10] S. Zonouz, C. Davis, K. D. n Smart Grid, and undefined 2013, "Socca: A security-oriented cyber-physical contingency analysis in power infrastructures," *ieeexplore.ieee.org*. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/6687271/

[11] C. Ten, A. Ginter, R. B. I. T. on Power, and undefined 2015, "Cyber-based contingency analysis," *ieeexplore.ieee.org*. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7284712/

[12] G. Wu, G. Wang, J. Sun, J. C. I. T. on, and undefined 2020, "Optimal partial feedback attacks in cyber-physical power systems," *ieeexplore.ieee.org*. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9042205/

[13] A. Patel, S. P. I. C.-P. S. T. . Appl., and undefined 2019, "Switching attacks on smart grid using non-linear sliding surface." *ieeexplore.ieee.org*, vol. 4, pp. 382–392, 12 2019. [Online]. Available: https://ieeexplore.ieee.org/iel7/7805360/8953040/08953046.pdf

[14] H. Khalid, S. Muyeen, J. P. I. S. Journal, and undefined 2019, "Cyber-attacks in a looped energy-water nexus: An inoculated sub-observer-based approach," *ieeexplore.ieee.org*. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8861290/

[15] Q. Wang, W. Tai, Y. Tang, M. N. I. C.-P. Systems, and undefined 2019, "Review of the false data injection attack against the cyber-physical power system," *ieeexplore.ieee.org*. [Online]. Available: https://ieeexplore.ieee.org/iel7/7805360/8744670/08744664.pdf

[16] L. Yu, X. Sun, T. S. I. T. on Systems, undefined Man, undefined, , and undefined 2019, "False-data injection attack in electricity generation system subject to actuator saturation: Analysis and design," *ieeexplore.ieee.org*. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8726150/

[17] S. Ahmadian, X. Tang, H. Malki, Z. H. I. Access, and undefined 2019, "Modelling cyber attacks on electricity market using mathematical programming with equilibrium constraints," *ieeexplore.ieee.org*. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8651454/

[18] X. Lyu, Y. Ding, S. Y. I. Access, and undefined 2020, "Bayesian network based c2p risk assessment for cyber-physical systems," *ieeexplore.ieee.org*. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9090827/

[19] N. Tzanis, N. Andriopoulos, A. Magklaras, E. Mylonas, M. Birbas, and A. Birbas, "A hybrid cyber physical digital twin approach for smart grid fault prediction," in *2020 IEEE Conference on Industrial Cyberphysical Systems (ICPS)*, vol. 1, 2020, pp. 393–397.

[20] P. Church, H. Mueller, C. Ryan, S. V. Gogouvitis, A. Goscinski, and Z. Tari, "Migration of a scada system to iaas clouds – a case study," *Journal of Cloud Computing*, vol. 6, p. 11, 12 2017.