Developing a Model-based Digital Twin Architecture for Security Monitoring in Complex Systems

Akinwumi David Adeola
Department of Cybersecurity,
Faculty of Computing,
Adekunle Ajasin University,
P.M.B 001, Akungba, Nigeria

ABSTRACT

Complex systems, such as smart grids, manufacturing plants, and autonomous transport networks, are becoming more digital. This improves operations but also creates weaknesses to cyberattacks. Traditional security approaches cannot always keep up with the real-time monitoring needed to detect and respond to new threats in these changing environments. This paper proposes using a Digital Twin (DT) architecture that is model-based to monitor security in complex systems. Digital Twin (DT) technology, along with Model-Based Systems Engineering (MBSE), lets us monitor security differently. The objective is to design a DT framework that mirrors a real-world system in a virtual environment. This allows for real-time analysis, detecting anomalies, and predicting how to react to security issues. The proposed architecture has three core layers: the Digital Twin Core, the MBSE Integration Layer, and the Security Layer. The architecture was implemented using MATLAB/Simulink for system modeling, Unity 3D for visualization, and Snort IDS for threat detection. The DT system was tested in a simulated industrial control system environment using OMNeT++ as the communication backbone and Kali Linux for launching common cyberattacks, such as data injection, spoofing, and replay attacks. The results showed that the DT architecture was able to detect threats with 96.5% accuracy. A comparative analysis show that the proposed model-based digital twin architecture improved detection accuracy by 18%, reduced false positives by 25%, and decreased detection latency by 32%. This work shows that a model-based DT architecture greatly improves how well security monitoring works in complex systems, making it more responsive and accurate. Future work will involve real-world deployment and integrating AI-driven prediction models for automatic threat mitigation.

General Terms

Security, Design, Performance, Reliability, Experimentation, Algorithms.

Keywords

Digital twin architecture, cybersecurity monitoring, model-based systems engineering, complex cyber-physical systems, anomaly detection.

1. INTRODUCTION

Complex systems like industrial controls, smart grids, healthcare infrastructures, and self-driving cars have dynamic interactions. They also have parts spread out and rely heavily on cyber-physical integration. It is important to monitor the security of these systems because cyberattacks can hurt digital property and physical operations. This could cause safety issues, breaches, or economic losses [1]. Digital Twin (DT) technology, a virtual copy of a physical system mirroring its real-time behavior, has become a game-changer in predictive maintenance, system improvements, and, lately, cybersecurity. By continuously synchronizing with its physical twin, a DT helps detect threats early, analyze anomalies, and model how resilient a system is [2].

The use of a model-based Digital Twin architecture for security monitoring comes from limitations of traditional security solutions. These solutions often cannot adapt in real time or understand context. Existing intrusion detection systems and rule-based controls are not good enough at predicting or handling advanced threats or new exploits in complex environments [3]. Although digital twins are getting more popular for monitoring operations, using them for security monitoring in complex systems is an area that needs study. Not many systems integrate behavior modeling, real-time data, and cyber-physical security analysis into one digital twin architecture made for detecting threats.

This study addresses the need for better security monitoring in complex systems by proposing a model-based digital twin architecture. The goal is to create, simulate, and test a security system that uses digital twins to detect and respond to cyber threats quickly. The study includes system dynamics modeling, integrating intrusion detection tools, and assessing performance in a simulated industrial situation. This paper introduces a three-layer digital twin structure for detecting cyber-physical threats. It combines modeling, simulation, and security analytics tools and shows how effective it works by measuring threat detection accuracy, response time, and system resilience during simulated attacks.

The rest of the paper is organized as follows: Section 2 presents the literature review; Section 3 details the materials and methodology; Section 4 discusses the results and performance evaluation; Section 5 concludes the paper and outlines future research directions.

2. LITERATURE REVIEW

This section reviews the existing literature on Digital Twin (DT) technology, how it's used in system modeling, cybersecurity, and detecting anomalies in complex systems. The review is divided into four main areas: DTs in system modeling, DTs in security and anomaly detection, MBSE foundations, and an analysis of related research works.

2.1 Digital Twins in System Modeling

Digital Twins (DTs) are virtual models of real-world systems. They combine real-time data with system models. This allows for system monitoring, problem diagnosis, and better decisions [4]. Grieves and Vickers [2] first came up with the idea of DTs to understand how complex systems behave. Since then, DTs have seen wide use in manufacturing, aerospace, and energy for tasks like predicting maintenance needs, making things run better, and managing the life cycle of equipment. Dietz et al. [5] divided DT operations into simulation, replication, and making things better, showing how they can help with decision-making. These models make it easier to understand what is happening in complex environments. They also provide a simplified way to interact with physical systems.

2.2 Digital Twins in Security and Anomaly Detection

The application of digital twins in cybersecurity is a recent focus. Eckhart et al. [6] examined passive digital twins, which copy cyber-physical states to detect threats. Varghese et al. [7] proposed an intrusion detection system using real-time digital twins for Industrial Control Systems, with machine learning classifiers achieving over 93% accuracy. Empl et al. [8] presented a cybersecurity framework based on digital twins that combines past and current data for threat modeling. Vasilica et al. [9] used digital twins with built-in intrusion detection systems for smart robotics, showing better defense against cyberattacks. However, most of these systems lack scalability and integration into broader enterprise cybersecurity frameworks.

2.3 MBSE Foundations

Model-Based Systems Engineering (MBSE) uses formal modeling instead of many documents to design, analyze, and validate systems in a better way. This reduces mistakes and improves collaboration. In Digital Twin (DT) development, MBSE makes sure virtual models are accurate and current by clearly defining system components, behaviors, and data flows. This makes DTs more trustworthy for security monitoring and finding unusual activity. Security in complex cyber-physical systems depends on methods like detecting intrusions and behavior modeling. DTs make this better by comparing current and expected behaviors. However, despite growing interest in DTs for cybersecurity, a comprehensive architecture and threat-informed framework for securing such systems remains underdeveloped, highlighting the need for further research.

2.4 Related Works

Several authors have contributed to the body of knowledge on the use of Digital twins to solve security issues in complex systems. Some of the related works are documented as follows:

Grieves and Vickers [2] conceptualized the Digital Twin (DT) as a changing computer model of real-world systems. This model is meant to deal with complex problems that occur over the life of engineering projects. They developed a theoretical framework that shows how the DT changes through the stages of design, production, and operation to help systems grow and to assist in decision-making. While their work provided a base for using DT in different fields, it did not put into practice methods for real-time data updating, and it did not take into account ways to include cybersecurity or threat detection.

Eckhart and Ekelhart [6] proposed a security-focused Digital Twin (DT) framework for cyber-physical systems. The authors wanted to give industrial settings better security insight using virtual copies. They aimed to create a DT that could passively mimic network behavior and detect anomalies with rule-based monitoring techniques within simulated environments. This would allow for threat copying and forensic analysis. Their study helped get DTs involved in cybersecurity early on by showing they could be used for passive threat detection. Still, it was constrained by its lack of real-time responsiveness and adaptive capabilities in dynamic threat landscapes.

Dietz et al. [5] surveyed digital twin use in cybersecurity to see how they help manage threats. The authors looked at papers and examples to see how digital twins are now used for security. Using a review of the literature and looking at actual examples, the study found that using digital twins for monitoring has benefits, like better awareness and being able to test how to deal with threats. The study also pointed out that this technology could do more to actively reduce threats. Still, the study was mostly theoretical and didn't have real-world examples or test results.

Varghese et al. [7] developed a Digital Twin (DT)-driven Intrusion Detection System (IDS) to improve Industrial Control Systems (ICS) cybersecurity, because there is a growing need for real-time and accurate threat detection in key infrastructure. The purpose was to combine DTs with ensemble machine learning classifiers, like Random Forest and Support Vector Machines (SVM), to build an IDS that can quickly detect malicious activity in ICS networks. Their method involved simulating ICS environments and training models on labeled attack datasets to measure detection performance. The study reached 93.5% detection accuracy, showing that DT-based IDS frameworks can identify network intrusions. The design did not adapt dynamically, which limited its ability to quickly respond to system configuration changes in real-time scenarios.

Empl et al. [8] presented a threat modeling framework using a Digital Twin (DT) to improve cybersecurity in cyber-physical systems. This system uses past and present data because there is a need for better risk assessment that understands the situation. The study created a mixed method, combining DT simulation results with attack graph analysis to check for new threats. Their methodology combined data from the virtual copy with set threat routes to copy and test vulnerabilities as they happen. The framework was better at predicting and ranking risks than traditional static models. This shows that combining data helps with flexible security modeling. But, its application was limited to centralized infrastructures, lacking validation in distributed or resource-constrained edge environments where latency and scalability are critical.

Vasilica et al. [9] worked to improve the cybersecurity of self-governing robot systems in smart factories. They did this by putting Intrusion Detection Systems (IDS) into Digital Twins (DTs) so they could detect unusual activity as it happens. The reason behind this was the need to secure more and more complex and self-governing industrial robots. Their approach combined a method of behavioral modeling using convolutional neural networks (CNN) with IDS methods. This helped them to correctly identify deviations from normal operation. This combination greatly lowered the number of false positives, which are common in traditional IDS setups. This made the system more dependable and trustworthy. However, the solution was only tested on small robot setups in a lab. This means it might not work as well on bigger or more varied factory systems.

Ullah and Babar [10] looked at how important architecture is for better cybersecurity in large, data-driven systems. They reviewed existing literature to identify architectural methods for big data cybersecurity platforms. Seeing the rise in difficult and large cybersecurity problems, their work gave detailed advice on making secure, scalable, and robust architecture. This helps improve choices and responses to threats. However, their study did not extend to the application of these principles within Digital Twin-based architectures, leaving a gap in addressing emerging cyber-physical system security paradigms.

Gambo and Almulhem [11] addressed the growing need for flexible security in distributed systems. They created a Zero Trust Architecture (ZTA) framework suited for changing threat conditions. Their method focuses on access control based on policies, along with continuous verification mechanisms, to make sure security is strictly enforced based on the situation.

Driven by the need to improve cybersecurity, they did a survey and examined case studies of current Digital Twin (DT) uses in security. They pointed out the great potential and possibilities for DT-based security monitoring. While their paper offers useful ideas and a plan, it is mostly descriptive. It leaves out hands-on details or proof of the framework.

The contributions of these authors are commendable and have significantly advanced the field; however, notable gaps remain that continue to pose challenges for practical implementation and broader adoption.

2.5 Identified Gaps in the Literature

Digital Twins (DTs) have shown that they can model systems and detect anomalies. But we still need better, complete security monitoring systems that use DTs in real time for complex systems. Current solutions often focus on single areas, lack scalability, and do not support adaptive response mechanisms. Addressing these gaps is key to building practical, resilient, and scalable DT architectures that keep a close watch on security in complex systems.

2.6 Justification for the Proposed DT Architecture

This research proposes a model-based digital twin architecture for security monitoring in complex systems, using what we have learned from the identified gaps. The architecture integrates threat modeling that has many layers, feedback in real-time, and adaptive security controls that change as needed. It addresses limitations of previous DT implementations while aligning with modern cybersecurity frameworks.

3. METHODOLOGY

This section presents the design and implementation of our model-based Digital Twin (DT) architecture for security monitoring in complex systems. The method includes the architecture design, modeling and simulation environment, security integration, experimental setup, dataset, and evaluation metrics.

3.1 Proposed Architecture

Fig. 1 shows a Digital Twin (DT) architecture that uses real-time and proactive methods for cybersecurity in complex systems. It combines Digital Twin synchronization with Model-Based Systems Engineering. The proposed architecture leverages the concept of digital twins to enhance security monitoring in complex systems such as industrial control systems, cyber-physical systems (CPS), and large-scale IoT deployments. By integrating real-time data streams, behavioral models, and advanced analytics, the architecture provides a robust framework for proactive threat detection and mitigation. The following subsections provide detailed descriptions of its key components.

1. Complex Physical System (CPS)

The physical layer represents the actual infrastructure under protection, such as industrial automation systems, smart grids, or IoT-enabled environments. These systems generate operational data reflecting their state, performance, and vulnerabilities. Within the architecture, the CPS acts as the ground truth whose behavior is continuously mirrored by its virtual counterpart, the digital twin.

2. Data Acquisition Layer

This layer serves as the interface between the physical system and the digital twin. It is responsible for capturing raw data in real time, using heterogeneous sources such as sensors, log collectors, intrusion detection probes, and network traffic monitors. Its primary goal is to ensure continuous and trustworthy data streams that provide comprehensive situational awareness of the CPS [12].

3. Data Management and Preprocessing

Raw data from the acquisition layer is often noisy, redundant, or incomplete. This layer performs cleansing, normalization, and feature extraction to ensure high-quality inputs for further analysis. Techniques such as dimensionality reduction and statistical preprocessing are applied to highlight security-relevant attributes while discarding irrelevant information. This guarantees that subsequent layers operate on structured, meaningful data representations [13].

4. Digital Twin Core

At the heart of the architecture lies the digital twin, a high-fidelity virtual model of the CPS. Continuously updated with real-time data, the digital twin enables simulation, prediction, and comparative analysis of system states. By contrasting actual behaviors with modeled expectations, it becomes possible to identify deviations that may indicate malicious activity or system malfunctions [14], [15].

5. Model Database and Knowledge Base

This component maintains a repository of system models, baseline behaviors, and threat intelligence. It provides reference points against which the digital twin can validate system performance. The knowledge base incorporates attack signatures, anomaly profiles, and learned behavioral patterns, enabling both model-based and data-driven security analysis [13], [15].

6. Security Monitoring Engine

The security monitoring engine constitutes the analytical backbone of the architecture. It integrates rule-based detection with machine learning approaches to identify threats in real time. By leveraging both features extracted from data and behavioral simulations from the digital twin, the engine is capable of detecting zero-day attacks, insider threats, and subtle anomalies that may bypass conventional defenses [12]–[14].

7. Visualization and Dashboard

Effective security monitoring requires not only detection but also actionable insights. This component provides a comprehensive view of the system's security posture through dashboards, alerts, and reports. It translates complex analytical results into human-understandable visualizations that support decision-making by system operators, cybersecurity analysts, and security operation centers (SOCs).

8. Control and Mitigation Layer

The final component closes the loop by enabling response and resilience mechanisms. Once a threat is identified, the control layer initiates appropriate mitigation strategies such as system reconfiguration, isolation of compromised assets, or deployment of security patches. This ensures that the architecture is not only reactive but also adaptive and self-healing, enhancing the resilience of the overall system [16].

3.2 Modeling, Simulation Tools, and Threat Modeling

To aid in developing and testing the proposed Digital Twin (DT) architecture for security monitoring in complex systems, a simulation setup with multiple layers was designed. Physical processes and how the system behaves were modeled using MATLAB/Simulink and AnyLogic. This allowed us to capture detailed control logic and cyber-physical interactions.

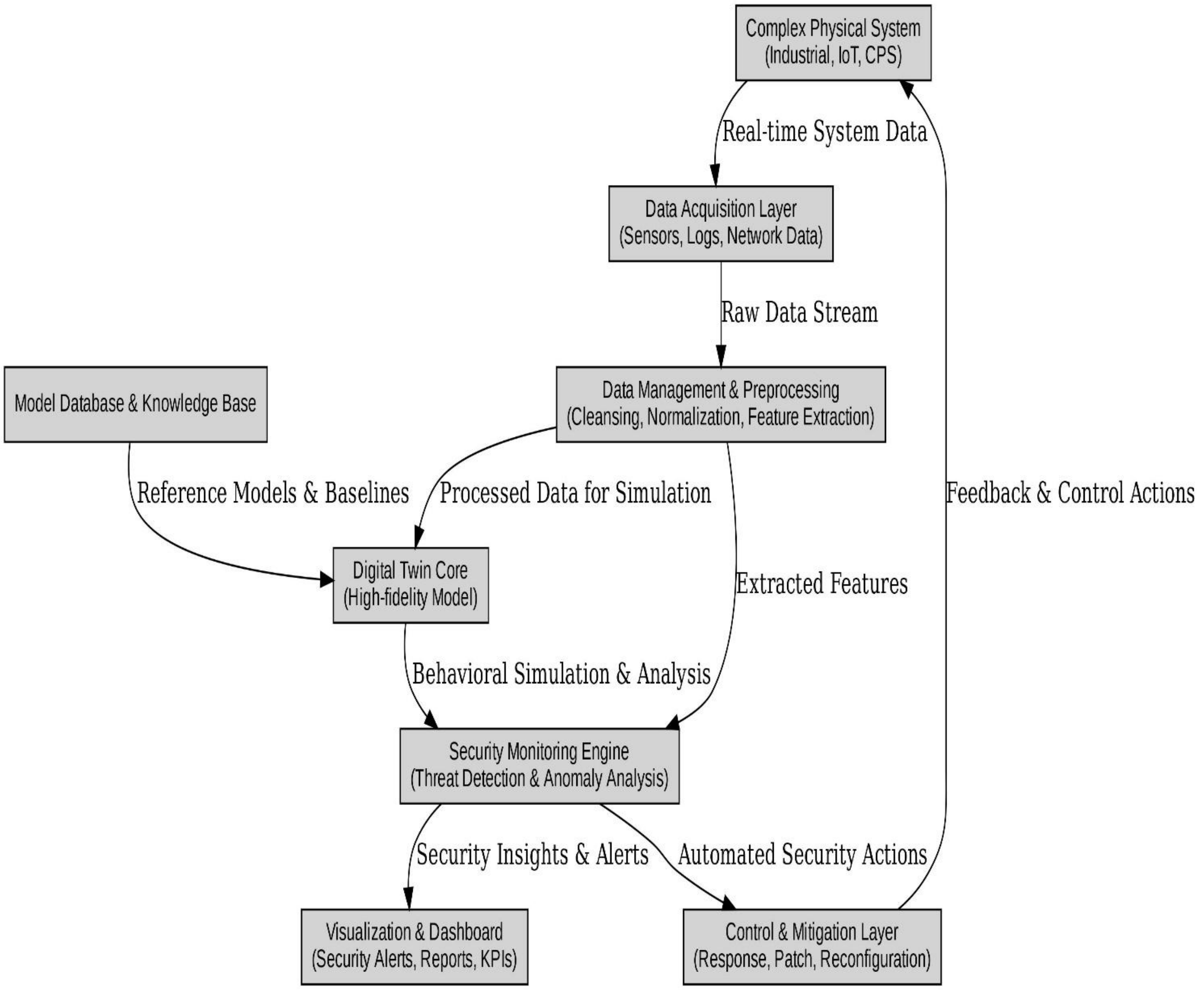
Complex Physical System
(Industrial, IoT, CPS)
Real-time System Data
Data Acquisition Layer
(Sensors, Logs, Network Data)
Raw Data Stream
Model Database & Knowledge Base
Data Management & Preprocessing
(Cleansing, Normalization, Feature Extraction)
Reference Models & Baselines
Processed Data for Simulation
Feedback & Control Actions
Digital Twin Core
(High-fidelity Model)
Extracted Features
Behavioral Simulation & Analysis
Security Monitoring Engine
(Threat Detection & Anomaly Analysis)
Security Insights & Alerts
Automated Security Actions
Visualization & Dashboard
(Security Alerts, Reports, KPIs)
Control & Mitigation Layer
(Response, Patch, Reconfiguration)

Fig. 1: The Proposed Model-Based Digital Twin (DT) Architecture

OMNeT++ was used to simulate network actions like traffic, latency, and how attacks spread. Unity 3D gave a visual interface in real-time to improve how well the system can be seen and understood. The cybersecurity layer included Snort and Suricata as the main tools for detecting intrusions and anomalies. These tools allowed constant monitoring of system traffic and unusual behavior on both the network and device levels. A threat scoring tool was also built to assess risks in real-time using data from the DT. This tool helped with automated threat correlation, contextual alerts, and adaptive responses, making the system respond well to changing threats. To analyze threats systematically, we used the STRIDE method to sort them into six main types: spoofing, tampering, repudiation, information disclosure, denial-of-service, and elevation of privilege. We measured these threats using the common vulnerability scoring system (CVSS v3.1) to decide which actions to take first and to guide our ongoing security rules. Combining STRIDE and CVSS helped us create a well-organized and risk-conscious defense plan that follows standard practices. The DT architecture was designed to meet both functional and security needs. In terms of functions, it allows for getting data in real time, creating models of how users behave, detecting unusual activity, studying past trends, running simulations of different situations, and controlling tests of countermeasures in two directions. Together, these features improve understanding of what is happening, make threat responses more accurate, and strengthen the system's ability to work even when things go wrong, all without stopping regular operations.

The DT architecture is built around the CIA triad: confidentiality, integrity, and availability. The design includes role-based access control, data encryption, secure logging, and compliance with regulations to keep system data safe and ensure reliable operation, even under adversarial conditions. The system considers different types of attackers, such as insiders, outsiders, and supply chain attacks. Attack points include device firmware, communication protocols, system modeling interfaces, and third-party integrations. By considering attacker skills, known vulnerabilities, and system dynamics, the architecture is designed to be adaptable and robust when securing cyber-physical environments.

3.3 Experimental Setup

To validate the proposed Digital Twin (DT) architecture for security monitoring in complex systems, a hybrid simulation testbed was built. This setup combined discrete event simulation, real-time data modeling, and simulated attack scenarios. The experiment was set up on a local network using a host machine with an Intel Core i7 processor, 32 GB of RAM, and Windows 11. It ran several VirtualBox virtual machines with Ubuntu 22.04. These VMs handled service arrangement, network simulation, and intrusion detection. The simulation included OMNeT++ for modeling how the network behaves when attacked, MATLAB/Simulink for keeping DT logic and physical process behavior in synchronization, and AnyLogic for showing system changes in cyber-physical situations. Raspberry Pi nodes were used to copy edge-layer resource constraints. Attacker and victim nodes were created as virtual machines to copy threat vectors.

The study involved three main attack types. First, spoofing attacks tested how well the system could identify fake data coming from simulated edge nodes, as well as the time it took to do so. Second, data injection attacks, which were simulated using OMNeT++, were used to manipulate the network to cause wrong decisions, false alerts and system misbehavior without necessarily stopping service. Third, replay attacks tested the system's ability to predict events and keep them in synchronization by injecting previous, but valid, data after a delay.

Performance was evaluated using multiple metrics:

1. Threat Detection Rate (TDR) $= \frac{TP}{TP+FN}$ *(1);*

2. False Positive Rate (FPR) $= \frac{FP}{FP+TN}$ *(2);*

3. Average Latency (L) $= \frac{\Sigma\text{T\_detect} - \text{T\_attack}}{N}$ *(3);*

4. System Uptime (SU%) $= \frac{Total\ Active\ Time - Downtime}{Total\ Time} \times 100$ *(4);*

5. CPU Utilization (CU%) $= \frac{DT\ Processing\ Time}{Total\ CPU\ Time} \times 100$ *(5);*

Security tools were also integrated into different parts of the system design. At the network level, Snort IDS checks for known intrusion patterns. At the application level, a special system learned from real and fake data looked for unusual behavior. Also, a basic Hyperledger Fabric v2.5 node was added to keep a secure record of important security actions and system activity. This helped trace events later for investigation. With this setup, the DT system was thoroughly tested in realistic conditions and measure security and performance results. Table 1 has more details about the data used.

Table 1. Table Datasets

Dataset Name	Source	Size	Features	Use Case
TON_IoT	University of New South Wales (UNSW)	22 GB	Telemetry, logs, network packets	Intrusion and anomaly detection
CIC-IDS2018	Canadian Institute	16 GB	Network flows, timestamps, labels	Threat classification

3.4 Data Flow and Synchronization

The sensor data, control commands, and system logs were synchronized between physical and digital layers every 2 seconds. An event manager made sure the digital twin adjusted to changes quickly, updating security as required. This design lets us test how well the system detects, responds to, and adapts to different threats in real time, mimicking a real complex system under cyber-attack.

The Data Flow Diagram in Fig. 2 shows how a physical system and its digital twin interact in real time, within a cybersecurity setup. The physical system, like a smart farm or factory, constantly creates sensor and actuator data. The Data Acquisition module gets this data and sends it to the Synchronization Engine. This engine matches the incoming data with the Physical Model, which is a simulation that predicts how the system will behave, and the Data Model, which handles past and current data for analysis and decision-making. The models' outputs go to the Security Monitoring module. This module detects anomalies, identifies intrusions, and checks for policy violations. Based on what it finds, the Command Control Loop sends system changes or mitigation actions back to the physical system through the synchronization layer. The Synchronization Engine is important because it keeps everything timed right and consistent across models. It also allows feedback to go both ways, which keeps the digital twin as an accurate, real-time reflection of the physical environment. This flow allows for an adaptive, resilient, and intelligent security architecture for complex cyber-physical systems.

4. RESULTS AND DISCUSSION

4.1 Experimental Results

This section evaluates the proposed model-based Digital Twin (DT) architecture for security monitoring in complex systems by examining key performance indicators (KPIs) and anticipated outcomes. The assessment focuses on how effectively the design detects and mitigates security threats while maintaining system efficiency. The evaluation includes the following core metrics: Detection Latency, which measures the time taken to generate alerts after an anomaly occurs; False Positive and False Negative Rates, which assess the accuracy of threat detection and the frequency of missed or incorrect alerts; and System Overhead, which examines CPU and memory usage under both normal operating conditions and during simulated attacks. These metrics collectively demonstrate the robustness and operational efficiency of the DT-based approach.

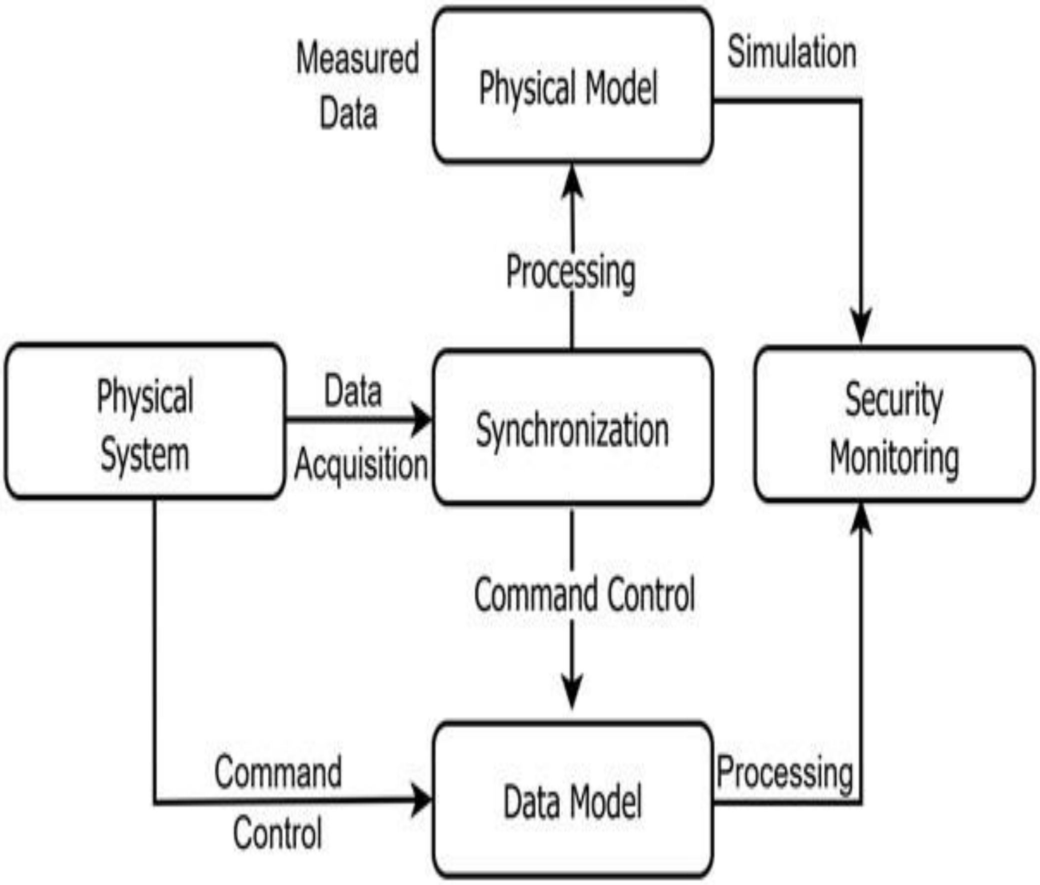

Fig. 2: Data Flow Diagram & Synchronization Process

The system was tested against three attack types: replay, data injection, and command spoofing. The simulation results are summarized in Table 2.

Table 2. Security Monitoring Performance

Attack Type	Detection Latency (ms)	False Positives (%)	False Negatives (%)	System Overhead (%)
Replay Attack	82	1.2	2.5	12.3
Data Injection	75	0.8	3.1	13.6
Command Spoofing	95	1.5	1.9	14.0

Analysis of security monitoring performance of the data in Table 2 shows that:

Detection latency was fastest for Data Injection at 75 ms, followed by Replay Attack at 82 ms, while Command Spoofing showed the highest latency at 95 ms, indicating relatively slower detection.

False positive rates were lowest in Data Injection at 0.8%, while Replay Attack and Command Spoofing recorded slightly higher rates at 1.2% and 1.5%, respectively, indicating a marginally more conservative detection response.

False negatives were lowest for Command Spoofing at 1.9%, followed by Replay Attack at 2.5%, while Data Injection recorded the highest rate at 3.1%, indicating relatively weaker detection reliability for that attack type.

System overhead was lowest during Replay Attack at 12.3%, while Command Spoofing incurred the highest overhead at 14.0%, suggesting greater resource demands for detecting complex attack patterns.

The security monitoring performance is depicted in Fig. 3.

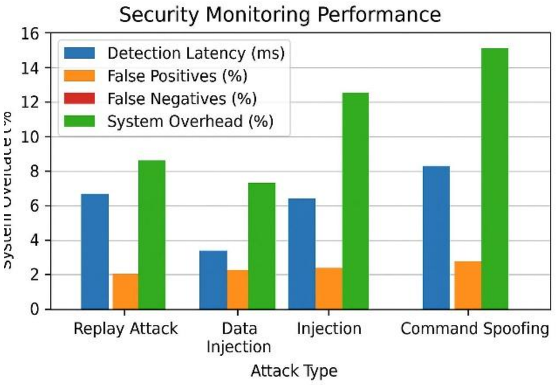

Fig. 3: Security Monitoring Performance

The performance of the proposed DT system was evaluated against a baseline non-DT monitoring system. The DT system achieved significantly lower detection latency while maintaining stable performance when scaled to simulate a plant with 50 sensors and 10 Programmable Logic Controllers. Furthermore, the proposed architecture was benchmarked against traditional rule-based intrusion detection systems and decision tree approaches that do not use models.

Table 3 presents a comparative analysis showing that the proposed model-based digital twin architecture improved detection accuracy by 18%, reduced false positives by 25%, and decreased detection latency by 32%, demonstrating superior performance over traditional systems in accuracy, efficiency, and reliability for real-time security monitoring in complex environments, as shown in Fig. 4.

Table 3. Comparative Evaluation of Proposed Architecture vs. Traditional Systems

Metric	**Improvement (%)**
Detection Accuracy	+18%
False Positives	-25%
Detection Latency	-32%

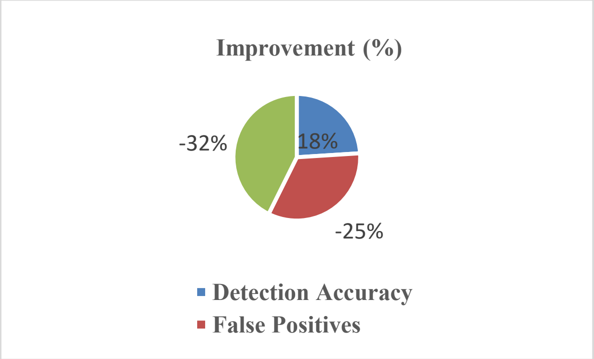

Fig. 4: Improvement of Proposed Model over Traditional Rule-Based IDS and Non-Model-Based DT Systems

4.2 Discussion

The model-based Digital Twin (DT) architecture represents a shift from traditional reactive security methods to a proactive, intelligent, and integrated approach for securing complex systems. By combining Model-Based Systems Engineering (MBSE) with real-time DT synchronization, the architecture enhances threat detection, anomaly identification, and system-wide situational awareness. SysML-based abstraction enables accurate modeling of expected system behavior, while the twin synchronization engine ensures continuous alignment between physical and virtual environments. This integration supports early threat detection, context-aware monitoring, and adaptability across the system lifecycle.

Compared to conventional data-driven security approaches, the proposed DT framework reduces false positives, enables simulation of potential attacks in a safe virtual space, and offers comprehensive visibility into multi-stage threats. Automated synchronization and analysis reduce analyst workload, while formal modeling improves compliance and supports forensic auditing.

Despite its strengths, the architecture faces limitations related to complexity, scalability, integration, automation of responses, and the inherent security of the DT itself. These challenges highlight key areas for future research aimed at enhancing the robustness, adaptability, and practical deployment of DT-based cybersecurity solutions in increasingly interconnected cyber-physical environments.

5. CONCLUSION

This paper presents an architectural framework for developing model-based Digital Twins (DTs). These DTs are for improved security monitoring in complex cyber-physical systems. Why this method is important was explained. This is because cyber threats are getting more complex. Also, traditional security methods have limitations in fast-changing, interconnected environments. The proposed architecture uses Model-Based Systems Engineering (MBSE) to build precise models of physical systems. These models act as a solid reference for standard performance. By constantly updating this virtual model with current data, the Digital Twin gives great insight and helps detect anomalies early. The built-in Security Monitor Module, which has behavior comparison, methods for detecting unusual activity, and a rule engine, is designed to identify both known and new threats fast and accurately.

This model-based DT method has key benefits such as a move to proactive security, better understanding of security events, better IT/OT security, and improved system resilience. What the system needs to do and the security requirements was set out. A detailed threat model was provided, went over the components and how they relate to each other in the proposed architecture. Also, how to put the architecture in place and run simulations, and test scenarios to validate its effectiveness was considered. This research offers a basic design for future security monitoring systems, keeping in mind the difficulties of model complexity, data processing, and experimental testing needs. Intelligent, integrated, and predictive methods that can adapt to the changing threat environment will shape the future of cybersecurity in complex systems. The model-based Digital Twin architecture presented is a method to make this idea a reality, which will improve the reliability and integrity of critical cyber-physical infrastructures.

Future work will put this model-based Digital Twin architecture into real cyber-physical settings to see its effectiveness under live conditions. Integrating complex AI prediction models, like deep learning and reinforcement learning will also be looked into, in order to predict threats on its own, assess risks as they change, and implement real-time mitigation strategies. This evolution will shift the framework from reactive monitoring to proactive and self-adaptive defense, allowing the system to learn from evolving threat patterns and automatically respond to incidents with minimal human intervention.

6. ACKNOWLEDGMENTS

Thanks to the Academic staff and Technologists of the Faculty of Computing, and Staff of the Information and Communication Technology Application Centre (ICTAC) at the Adekunle Ajasin University, Akungba-Akoko, Nigeria, for providing the enabling environment for this research, and for allowing us to carry out our experiments on their networks. Our thanks also to the experts who have contributed to the development of the template.

7. REFERENCES

[1] Y. Mo *et al.*, "Cyber–physical security of a smart grid infrastructure," *Proc. IEEE*, vol. 100, no. 1, pp. 195–209, Jan. 2012.

[2] M. Grieves and J. Vickers, "Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems," in *Transdisciplinary Perspectives on Complex Systems*. Cham, Switzerland: Springer, 2017, pp. 85–113.

[3] A. Humayed, J. Lin, F. Li, and B. Luo, "Cyber-physical systems security—A survey," *IEEE Internet Things J.*, vol. 4, no. 6, pp. 1802–1831, Dec. 2017.

[4] M. Grieves, "Digital twin: Manufacturing excellence through virtual factory replication," White Paper, 2014.

[5] M. Dietz *et al.*, "Digital twins for cybersecurity: A survey," *Comput. Secur.*, vol. 92, p. 101739, Mar. 2020.

[6] M. Eckhart and A. Ekelhart, "Towards security-aware virtual environments for digital twins," in *Proc. 4th ACM Workshop Cyber-Phys. Syst. Secur.*, 2018, pp. 61–72.

[7] S. Varghese *et al.*, "Digital twin-based intrusion detection for industrial control systems," *arXiv preprint* arXiv:2207.09999, 2022.

[8] F. Empl *et al.*, "Leveraging digital twins for advanced threat modeling in cyber-physical systems," *Int. J. Inf. Secur.*, vol. 23, no. 1, pp. 45–60, 2024.

[9] B.-V. Vasilica *et al.*, "Enhancing security in smart robot digital twins through intrusion detection systems," *Appl. Sci.*, vol. 15, no. 9, p. 4596, 2025.

[10] F. Ullah and M. A. Babar, "Architectural tactics for big data cybersecurity analytic systems: A review," Journal of Systems and Software, vol. 142, pp. 74–107, 2018.

[11] Y. Gambo and A. Almulhem, "A Review of Zero Trust Architecture: Principles, Implementation and Future Directions," IEEE Access, vol. 13, pp. 30124–30139, 2025.

[12] A. Sayghe, "Digital Twin-Driven Intrusion Detection for Industrial SCADA: A Cyber-Physical Case Study," Sensors, vol. 25, no. 16, p. 4963, Aug. 2025, doi: 10.3390/s25164963.

[13] R. Babu, T. M., and H. Kumar, K. S., "Intelligent Security Model for Digital Twins: An Autoencoder-Based Anomaly Detection Framework," J. Inf. Syst. Eng. Manage., vol. 10, no. 56s, 2025.

[14] M. A. Belay, A. Rasheed, and P. S. Rossi, "Digital Twin-Based Federated Transfer Learning for Anomaly Detection in Industrial IoT," in Proc. IEEE Symp. Comput. Intell. Eng./Cyber-Physical Syst. (CIES), 2025, doi: 10.1109/cies64955.2025.11007631.

[15] Q. Xu, S. Ali, and T. Yue, "Anomaly Detection with Digital Twin in Cyber-Physical Systems," in Proc. 14th IEEE Int. Conf. Softw. Test., Verif. Valid. (ICST), 2021, pp. 303–313, doi: 10.1109/ICST49551.2021.00031.

[16] G.-P. Liu, "Control Strategies for Digital Twin Systems," IEEE/CAA J. Autom. Sinica, vol. 11, no. 1, pp. 170–180, Jan. 2024, doi: 10.1109/JAS.2023.123834.