Journal of Information Systems Engineering and Management
2025, 10(33s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Security Automation Framework for Digital Twins in Smart Manufacturing

Aakarsh Mavi
mavi.aakarsh4@gmail.com

ARTICLE INFO

Received: 26 Dec 2024

Revised: 14 Feb 2025

Accepted: 22 Feb 2025

ABSTRACT

Digital twin technology is shaking things up in the manufacturing world by creating realtime virtual copies of physical assets. But as this technology evolves, keeping these systems safe and secure is more important than ever. Digital twins work by constantly exchanging data between the real world and their virtual counterparts, which makes them a target for cyberattacks. If these attacks succeed, they can throw off the accuracy of the virtual models and disrupt manufacturing processes. This research looks into security automation solutions specifically for digital twin systems in manufacturing settings, prioritizing monitoring and defending against new cyber threats. We're suggesting automated security tools that use anomaly detection algorithms to keep an eye on the data flowing into digital twins, allowing us to spot and react to any unusual or harmful activities in real-time. Plus, we're introducing automated lockdown features that protect the digital twin environment by isolating any compromised components to stop further damage. The goal of this framework is to boost the security, accuracy, and reliability of digital twin systems, ensuring these innovative technologies can be used safely and effectively in critical manufacturing operations. This paper dives into the design, implementation, and challenges of securing digital twins in the manufacturing space and adds to the ongoing push to weave security automation into the Industry 4.0 framework.

Keywords: Digital Twins, Security Automation, Anomaly Detection, Manufacturing Systems, Real-Time Monitoring, Machine Learning, Threat Intelligence, Automated Lockdown, Cybersecurity Framework, Continuous Learning, Isolation Forest, Edge Computing, SIEM Integration, SOAR Platforms

1 Introduction

Digital twin technology is basically about creating virtual copies of real-world assets, and it's shaking things up in the manufacturing sector. With the ability to monitor operations in real-time, run simulations, and fine-tune processes, digital twins really boost efficiency, help with predictive maintenance, and aid in smarter decision-making. As more manufacturers jump on board with this tech, making sure these digital twin systems are secure and reliable becomes super important. These virtual models depend on constant data flowing from their physical counterparts, which opens up new ways for cybercriminals to potentially disrupt things, manipulate information, or cause big financial and reputational issues.

Regular security measures often aren't enough when it comes to protecting the digital twin ecosystem. These systems are constantly changing and work with a ton of different industrial technologies. So, to keep digital twins secure, we need fresh, automated solutions that can sift through huge amounts of real-time data, spot anything unusual, and act quickly if there's a threat. Without strong security automation, any weaknesses in digital twin systems could affect the accuracy of these virtual models, leading to flawed simulations and wrong predictions that could mess up production and maintenance schedules.

This paper digs into security automation techniques specifically designed to protect digital twin systems in manufacturing settings. We're particularly looking at how to create automated tools that can monitor the data streams feeding into digital twins, detect anomalies, and quickly implement lockdown procedures to isolate any compromised parts. By using advanced security automation, we're aiming to strengthen the resilience and credibility of digital twin technology, ensuring it continues to play a critical role in key manufacturing operations. The following sections will go over the design, implementation, and potential challenges of securing digital twins, giving insight into how automation can fit into the security frameworks of Industry 4.0 technologies.

2 Literature Review

Digital twins are really making waves in the manufacturing world. They're being used more and more for things like monitoring assets in real time, predicting maintenance needs, and optimizing processes. As these technologies grow, it's super important to think about how we can keep digital twin systems safe from cyber threats, especially since the risks are getting higher in industrial settings.

2.1 Digital Twins in Manufacturing

So, what are digital twins? Basically, they create a energetic virtual representation of a physical asset or system. This virtual model gets updated with data from sensors, IoT devices, and other sources all the time. As [TZL18] point out, having digital twins can really boost operational efficiency by letting manufacturers see real-time insights and make better decisions based on accurate simulations of real-world conditions. They're being used across the board for tasks like predictive maintenance, quality control, and fine-tuning production processes [GS12]. But with this increased use comes a higher risk of cyber threats, especially in environments where there's a constant flow of data between the physical and virtual worlds.

2.2 Cybersecurity Challenges for Digital Twin Systems

Digital twins thrive on a ton of continuous data to keep the virtual model in step with its physical counterpart. However, this data flow opens up several pathways for potential cyber-attacks, ranging from threats to data integrity to tampering with control systems. [FZLL20] emphasize that while digital twins offer massive advantages, they also expose manufacturers to important vulnerabilities, especially since they connect to many IoT devices, cloud platforms, and third-party networks that might not be secure. If these digital twins are compromised, it can lead to serious issues like production downtime, financial losses, and even safety risks. As manufacturers strive to secure digital twins and their data flows, balancing privacy concerns with threat visibility is crucial, similar to the challenges explored in DNS over HTTPS (DoH) implementations in gaming [Tal24], where privacy and security must be carefully managed.

Plus, as van der [MG20] point out, there's a clear need for strong cybersecurity protocols to keep the integrity of these virtual models intact and to ward off unauthorized access to critical data streams. Since digital twins rely on real-time data, any disruption or changes can mess up simulations and lead to poor decision-making, which can seriously affect how manufacturing systems function. This is why it's so important to safeguard digital twins from cyber threats if we want to keep using them in key industries.

2.3 Security Automation for Industrial Systems

With manufacturing environments becoming increasingly complex and automated, traditional cybersecurity methods just don't cut it anymore. That's where security automation comes into play. As [BA18] discuss, automated patch management and vulnerability scanning are critical in industrial systems, allowing them to stay secure without needing constant manual oversight. Automation can greatly improve detection and response times, which is critical in environments where real-time operations matter, like with digital twin technologies.

Anomaly detection is a key part of security automation and has been a hot topic in relation to industrial control systems (ICS) and operational technology (OT). Basically, according to [HL17], these algorithms keep an eye on how a system behaves and compare it to what's expected. If something strays from the norm, it could mean there's a possible security issue or cyber attack going on. For digital twin systems, anomaly detection helps monitor data inputs, communication channels, and overall system behavior to spot any oddities that might indicate data tampering, system compromise, or unauthorized access. The real trick is in building automated systems that can handle the complex, fast-moving data that digital twins churn out.

2.4 Implementing Security Automation in Digital Twins

Recently, there's been a big push to add automated lockdown mechanisms and real-time monitoring to cybersecurity strategies for digital twin systems. [MG20] point out that keeping a constant watch on the digital twin environment, along with automated containment steps (like isolating compromised parts or restricting system access), can stop attacks in their tracks and lessen the blow of any security breaches. Automating lockdown actions means that when anomalies get spotted, immediate steps can be taken to protect the virtual twin, keeping it accurate and secure. Recent advancements in security automation techniques, such as passive enumeration methodologies for DNS scanning [Tal25], could be adapted to improve the detection and response capabilities of digital twin systems in manufacturing environments.

Bringing threat intelligence into security automation tools is another essential area of research. [RS22] suggest that linking real-time threat intelligence feeds to security automation frameworks can boost the system's response to new threats. Considering digital twins, this could mean updating security measures whenever new vulnerabilities or cyber-attacks are discovered, ensuring defenses evolve with the threat environment.

2.5 Challenges and Opportunities in Securing Digital Twins

While adding security automation to digital twin systems shows a lot of promise, there are still some challenges to clear. First off, the complexity of digital twin architectures—which blend physical and virtual components—makes it tough to achieve comprehensive security. As [ZC18] emphasize, older systems and devices can be tricky to mesh with modern security automation tools, potentially creating blind spots in the security of digital twins used in manufacturing.

Another issue is needing constant tweaking and adjusting of security measures. Digital twin systems are energetic and keeping security solutions aligned with changes in data, configurations, and system behavior is a major concern. As digital twins become more autonomous, it's super important that automated security tools can adapt right along with them.

Despite these challenges, there's a lot of potential for boosting the security of digital twins through automation. Developing smart security frameworks that can learn from past issues, recognize attack patterns, and anticipate future vulnerabilities shows a really promising way to create resilient and secure digital twin systems.

This literature review pulls from various studies and research that look at the intersection of digital twin technology, cybersecurity, and automation in manufacturing. It lays out the groundwork for the proposed security automation framework for digital twins.

3 Framework Design

Designing a framework to secure digital twins in manufacturing is all about keeping those virtual models representing real assets safe and accurate. The main goal here is to set up an automated, real-time security system that can monitor things, spot any odd behavior, and deal with cyber threats effectively—all without needing someone to step in manually. This framework is divided into three key parts: Monitoring and Data Integrity, Anomaly Detection and Response, and Automated Lockdown Mechanisms.

3.1 Monitoring and Data Integrity Layer

The first part focuses on making sure the data flowing between physical assets and their digital twins stays intact. Here, monitoring tools are constantly checking data inputs, system behavior, and network traffic to ensure the virtual twin accurately reflects its real-world counterpart. Real-time monitoring of digital twin systems could benefit from Power BI dashboards, which provide key performance indicators (KPIs) for decision-making, similar to their use in inventory management [Yer25a].

3.1.1 Components:

- **Real-Time Data Monitoring:** Keeping a constant eye on data streams from sensors, IoT devices, and other sources feeding into the digital twin. This way, we can make sure only valid and authorized data updates the virtual models.

- **Data Validation:** Setting up checks to confirm that the data being sent over is genuine and accurate. This means making sure sensor readings are solid and fit within expected ranges or patterns.

- **Digital Twin Integrity Check:** Regular checks to compare how the virtual twin behaves against the physical asset, catching any differences that pop up.

3.1.2 Tools and Technologies:

- **SIEM Systems:** Tools like Splunk or IBM QRadar help us keep track of security events in real time across the manufacturing network.

- **Blockchain for Data Integrity:** Using a blockchain ledger allows us to securely store and validate data input into the digital twin, ensuring that it remains untouchable.

3.2 Anomaly Detection and Response Layer

After monitoring the data, the next step is to spot potential security incidents through anomaly detection algorithms. This part is essential for automatically catching any discrepancies that might suggest malicious activity or issues with how the digital twin operates. Python and machine learning models, such as those used for anomaly detection in order tracking systems [Yer23a], can be applied in the anomaly detection layer of the digital twin framework to identify potential cybersecurity threats in real-time.

3.2.1 Components:

- **Anomaly Detection Algorithms:** Machine learning models, like Isolation Forest or Autoencoders, are trained on what normal looks like, helping to flag any unusual patterns. These deviations could signal a cyber-attack, data issues, or tampering with the digital twin.

- **Behavioral Analytics:** These tools observe the usual behavior of the digital twin system to establish baselines. Any unusual activity, like sudden changes or unapproved access attempts, sets off an alert.

- **Event Correlation:** The framework ties together different events or anomalies from various system parts, creating a clearer view of potential threats. This capability helps spot complex attacks that cross both physical and digital boundaries.

3.2.2 Tools and Technologies:

- **Machine Learning Frameworks:** Tools like TensorFlow, PyTorch, and Scikit-learn help us develop and deploy machine learning models that catch anomalies in real-time.

- **Edge Computing Devices:** Edge devices, placed close to where the digital twin's data comes from, can run lightweight anomaly detection algorithms. This setup cuts down on latency and enables quicker responses to potential threats.

3.3 Automated Lockdown Mechanisms Layer

Next up is the third layer, which focuses on isolating any compromised parts of the digital twin system to stop attacks from spreading. With automated lockdown mechanisms, the system jumps into action as soon as it spots an anomaly, taking steps like isolating compromised components or limiting system access.

3.3.1 Components:

- **Automated Isolation of Affected Components:** If the system detects an anomaly, it automatically isolates the troubled digital twin component or virtual model to prevent further data tampering or spread of cyber attacks. For instance, if a sensor is sending faulty data, the system can disconnect it from the network until an operator can check it out.

- **Quarantine and Access Restrictions:** Once a threat is identified, the framework restricts access to the affected digital twin models or network segments, keeping attackers from spreading their attack to other connected systems.

- **Intrusion Prevention System (IPS):** An IPS like Suricata or Snort can be set up to detect and block potential intrusions, making sure that any unauthorized access is stopped in its tracks.
- **Automated Reporting and Alerts:** After a lockdown, the system creates reports that detail the anomaly, the actions taken, and which systems were affected. Alerts get sent out to operators and administrators for them to review and confirm the security incident.

3.3.2 Tools and Technologies:

- **Firewall and Network Segmentation:** To isolate any compromised systems, the framework uses network segmentation and firewalls to limit access between components. Tools like Cisco's Next-Generation Firewall (NGFW) or Palo Alto Networks can automatically block or segment compromised digital twin nodes.
- **Automated Security Playbooks:** Security orchestration, automation, and response (SOAR) platforms such as Demisto or Swimlane help automate the incident response and lockdown processes, ensuring a fast reaction to cyber incidents.

3.4 Integration of Threat Intelligence

To further boost security for digital twins, the framework integrates threat intelligence to provide realtime updates on new threats and vulnerabilities. This helps us update security protocols and tweak anomaly detection models based on the most current cyber threat data. Incorporating AI-driven data cleansing into the threat intelligence layer of the system could enhance the accuracy and integrity of the data used for cybersecurity analysis, as seen in supply chain systems where it aids in master data management [Yer24a]

3.4.1 Components:

- **Threat Intelligence Feeds:** The framework subscribes to threat intelligence platforms like VirusTotal, IBM X-Force, or open-source repositories like MISP (Malware Information Sharing Platform) to continually receive updates about new vulnerabilities and threats that could impact digital twins.
- **Threat Intelligence Correlation:** The system matches threat intelligence data with internal security logs and real-time system activity, allowing for quicker detection of threats specific to manufacturing environments or digital twin systems.le.

3.5 Continuous Learning and Adaptation

To wrap it up, this framework is all about continuous learning. It's designed to keep up with new threats as they pop up. The machine learning models we use for spotting anomalies get updated with new data whenever it's available. This way, the system continues to evolve along with the ever-changing world of cyber threats.

3.5.1 Components:

- **Model Retraining:** We periodically retrain our anomaly detection models with the freshest data from both our digital twin system and sources of threat intelligence. This keeps our models in tune with new attack trends.
- **Feedback Loop:** We've set up a feedback system where security teams can look over security events and fine-tune how the system reacts to certain threats. This boosts the system's effectiveness in spotting and handling future attacks.

4 Implementation

To secure digital twins in manufacturing, we need to implement a security automation framework that protects both virtual models and the real-time data coming from physical systems. Let's break down the key components, along with some sample code snippets that show how we can automate data monitoring, detect anomalies, and set up an automated lockdown mechanism.

4.1 Real-Time Data Monitoring for Digital Twins

For monitoring data in real-time, we can use tools like Splunk or the ELK Stack (which stands for Elasticsearch, Logstash, and Kibana). These help us capture and process data from sensors, IoT devices, and other data sources. Imagine we're in a manufacturing setup where a Python script is collecting sensor data, and we're keeping an eye out for any unusual patterns. The integration of automated ETL processes, similar to those used in logistics to improve operational efficiency [Yer23b], could be applied to digital twin systems for real-time monitoring and data integrity checks in manufacturing. **Sample Python Code for Data Monitoring:**

```
import time import random import logging

# Configure logger to log data logging . basicConfig ( filename='sensor _data . log ' , level=logging .INFO)

# Simulated sensor data collection def collect sensor _data ():

sensordata = random . randint (50 , 100)

# Simulated sensor reading between 50−100 return sensor _data

# Real−time data monitoring def monitor data (): while True :

sensor value = collect _sensor _data ()

timestamp = time . strftime(”%Y−%m−%d %H:%M:%S”)

logging . info ( f ”{timestamp} − Sensor Reading : {sensor _value }”)

# Implement threshold for flagging data i f sensor _value > 90: logging . warning

      ( f ”ALERT: High sensor value detected − {sensor _value }”) time . sleep (1) # Collect data every second

i f name == ” main ”: monitor data ()
```

Explanation:

- **Data Collection:** The collect sensor data() function mimics the process of reading sensor data from a manufacturing asset.
- **Data Monitoring:** The monitor data() function logs sensor data continuously, checking if the readings go over a set threshold (90 in this case). If that happens, it logs a warning.
- **Logging:** All the data is saved in a file (sensor data.log), and any major anomalies are marked with a warning.

4.2 Anomaly Detection with Machine Learning

The next step is to identify anomalies in the data using machine learning techniques. Here, we'll rely on the Isolation Forest, which is great for spotting anomalies in complex datasets. **Sample Python Code for Anomaly Detection:**

```
import numpy as np from sklearn . ensemble import IsolationForest import logging

# Sample data for sensor readings ( for illustration purposes ) sensor _data = np. array ([[65] , [70] , [80] ,

[100] ,    [95] ,    [50] ,    [90] ,    [60] ,    [85] ,    [110]])

# Initialize Isolation Forest model model = IsolationForest ( contamination =0.2)

# Contamination    represents    the proportion    of    outliers

# Fit the model

model . fit ( sensor _data )

# Predict anomalies predictions = model . predict ( sensor _data )

# Log the results logging . basicConfig ( filename='anomaly _detection . log ' , level=logging .INFO) for i , prediction in enumerate( predictions ):
     i f      prediction == −1:
     logging . warning( f ”Anomaly Detected at Index { i } with Sensor Value : {sensor _data [ i ][0]}”) else :
logging . info ( f ”Normal Data at Index { i } with Sensor Value : {sensor _data [ i ][0]}”)
```

Explanation:

- **Isolation Forest Model:** We use the IsolationForest from Scikit-learn to find anomalies in the data. The contamination parameter helps us set how many outliers we expect to see in our dataset.
- **Model Prediction:** The predict() function will yield -1 for anomalies and 1 for normal data.

We log these results, marking anomalies with warnings.

4.3 Automated Lockdown Mechanism

After catching an anomaly, we need to automatically respond by isolating or locking down the affected component. This could mean disabling the faulty sensor or disconnecting the impacted digital twin from our system.

Let’s simulate an automated lockdown by stopping access to a compromised sensor (we’ll show this by halting the data collection process).

Sample Python Code for Automated Lockdown:

```
import time import logging

# Simulate disabling a sensor def disable _sensor ( sensor _id ):

logging . warning( f ”Sensor { sensor _id } disabled due to anomaly detection .”)

# Simulate a component lockdown def lockdown( sensor _id ):

logging . info ( f ”Locking down sensor { sensor _id }. All communication stopped .”)

# Add code here to disconnect the sensor from the network (e . g . , disabling API, shutting down communication channels ) disable _sensor ( sensor _id )

# Simulate the detection of an anomaly def check for anomalies ( sensor _data ):

# Assuming sensor _data is a l i s t of sensor readings for index , value in enumerate( sensor _data ):

i f     value > 90:          # Threshold      for anomaly detection lockdown( index )

# Sample sensor data sensor _data = [65 , 70 , 80 , 100 , 95 , 50 , 90 , 60 , 85 , 110]

# Monitor and lockdown compromised sensors check for anomalies ( sensor _data )
```

Explanation:

- **Disable Sensor:** The disable sensor() function simulates the action of turning off a sensor when we detect an anomaly.
- **Lockdown Mechanism:** The lockdown() function records the lockdown and simulates the removal of the compromised sensor from the network.
- **Anomaly Detection and Lockdown:** The check for anomalies() function keeps an eye on sensor readings and triggers the lockdown process if an anomaly pops up.

4.4 Integrating Threat Intelligence

To boost our security, we weave in external threat intelligence feeds that let us adjust our security measures based on new threats. For simplicity, we’ll simulate pulling from a sample threat feed. **Sample Python Code for Threat Intelligence Integration:**

```
import requests import logging
```

```
# Simulated threat intelligence feed URL threat feed url = ”https :// samplethreatfeed .com/api/threats”
 # Fetch threat              intelligence       data
def fetch _threat _intelligence (): response = requests . get ( threat feed url )
          i f        response . status _code == 200:
         return response . json () # Return threat data as a JSON object else :
logging . error (” Failed to fetch threat intelligence data .”) return None
# Integrate threat intelligence with the existing system def integrate threat intelligence ():
threat data = fetch threat intelligence () i f threat data :
                   for    threat    in     threat data :
# Simulate updating system defenses based on threat intelligence logging . info ( f ”New Threat Identified : {threat [ ’
threat _name ’]} – {threat [ ’ sev
                                  # Here you would implement further           logic          to adjust anomaly detection     threshol
i f  __ name __  == ” __ main __ ”:
integrate threat intelligence ()
```

Explanation:

- **Threat Feed Integration:** The fetch threat intelligence() function pretends to gather threat intelligence from an external source.
- **Threat Response:** The integrate threat intelligence() function updates our system's defenses according to the threat data we receive, logging any new threats along the way.

4.5 Continuous Learning and Adaptation

Lastly, we focus on continuous learning by regularly updating our anomaly detection model. This helps it keep up with new data and changing attack patterns. **Sample Code for Continuous Learning:**

```
from sklearn . ensemble import IsolationForest import numpy as np
# Simulated sensor data sensor\ _data = np. array ([[65] , [70] , [80] ,
[100] ,    [95] ,    [50] ,    [90] ,    [60] ,    [85] ,    [110]])
 # Train the model i n i t i a l l y model = IsolationForest ( contamination =0.2) model . f i t ( sensor\ _data )
 # Simulate new sensor data over time new\ _data = np. array ([[75] , [85] , [95] , [105] , [ 5 0 ] ] )
# Retrain the model periodically with new data def retrain \ _model (model , new\ _data ): model . f i t (new\ _data )
              print (”Model retrained           with new data .”)
# Simulate         periodic            retraining retrain \ _model (model , new\ _data )
```

Explanation:

- **Model Retraining:** The retrain model() function refreshes the Isolation Forest model with new sensor data to adjust to changing attack patterns or system behaviors.
- **Ongoing Improvement:** This ongoing process makes sure that the system keeps learning and getting better at spotting anomalies and handling threats.

5 Future Work

This research lays a solid groundwork for protecting digital twins in manufacturing settings, but there are plenty of opportunities to take the security automation framework to the next level. Here are some areas where we can really boost security and scale up the system:

- **Integration with Advanced Machine Learning Models:** Looking ahead, we can dive into using more advanced machine learning techniques, like deep learning, to enhance how we spot anomalies. Using methods like Recurrent Neural Networks (RNNs) or Autoencoders could help catch those subtle changes in sensor data that might signal a security threat, especially in complex and busy environments.Future work could explore the use of AI-driven predictive analytics to forecast cybersecurity disruptions in digital twin systems, similarly to how predictive analytics are used in supply chain systems to optimize efficiency [Yer25b]

- **Collaborative Security Frameworks:** We could also expand this framework to include collaborations across several manufacturing sites or within a supply chain. By setting up secure communication channels between different systems, manufacturers can share real-time security updates, threat intel, and mitigation tactics, helping to build a stronger ecosystem. This may involve decentralized security methods that support peer-to-peer threat detection and response.

- **Edge Computing for Localized Security:** Since digital twins depend on real-time data from sensors and IoT devices, adding edge computing capabilities can be a revolutionary. Processing data right at the edge of the network could speed up security operations, cutting down on delays and ensuring we catch and handle potential threats quickly before they impact central systems.

- **Simulations and Red Teaming for Testing:** To make sure our security automation framework is solid, future efforts could include creating simulated environments for testing. By setting up virtual spaces where different cyber-attacks are staged, researchers can see how well the system reacts to various attack types. Engaging in red teaming and ongoing penetration testing can help fine-tune security measures, ensuring we're ready for whatever comes our way.

- **Integration with Industry Standards and Regulations:** Staying compliant with cybersecurity standards is essential in manufacturing. Future initiatives can focus on aligning the security framework with specific industry standards like ISO/IEC 27001, NIST SP 800-82, and IEC 62443. This way, we can guarantee the solution meets not just technical needs, but also keeps us on the right side of legal and regulatory requirements. Plus, this integration could simplify generating compliance reports for audits.

- **Scalability and Performance Optimization:** As we see an increase in the number of digital twins in manufacturing environments, making sure our security framework can handle this growth will be critical. Future research could look into optimization methods to manage large amounts of data from numerous sensors and digital twins without hampering performance. Exploring strategies like distributed computing, data compression, and load balancing could help keep the system efficient and scalable.

By focusing on these areas, we can enhance the security automation framework for digital twins in manufacturing, making it even more strong, scalable, and adaptable to new threats.

6 Conclusion

This research paper dives into creating a security automation framework specifically for digital twins in the manufacturing world. The main goal? To protect those virtual models of important physical assets from cyber-attacks. As digital twin technology becomes a bigger part of modern manufacturing, keeping these virtual replicas and their data secure is super important. The framework we've come up with includes real-time data monitoring, detecting odd behaviors, automated lockdown features, and using threat intelligence to keep our digital twins safe.Future work could involve integrating predictive analytics within digital twin systems to enhance decision-making, reduce delays, and optimize operational processes, much like how predictive analytics is used to enhance logistics operations [Yer24b]

By implementing things like automated patch management and continuous anomaly detection with machine learning, along with real-time lockdown capabilities, we can show how proactive steps can really cut down the risks of cyber-attacks in complex manufacturing settings. Plus, by incorporating external threat intelligence sources, the system can stay nimble and adjust to new threats as they pop up, making sure our defenses keep pace with the ever-changing environment of cyber threats.

The findings from this research indicate that using automation to secure digital twins isn't just a good idea—it's essential for keeping modern manufacturing systems reliable and secure. With less dependence on manual intervention, manufacturers can help preserve the integrity of their virtual models, reduce downtime, and safeguard critical physical assets against sophisticated cyber threats.

As digital twin technology keeps changing, our research underlines needing automated, scalable, and adaptive security frameworks. Future efforts can build on this groundwork by adding more advanced machine learning models for detecting anomalies, improving lockdown mechanisms, and tapping into broader threat intelligence sources to boost the overall security of digital twin systems in manufacturing.

References

[1] [BA18] Imran Barkat and Asim Arshad. Cybersecurity automation in industrial systems: A survey. Journal of Industrial Information Integration, 10:12–22, 2018.

[2] [FZLL20] Hui Feng, Yifan Zhang, Kai Li, and Hongyu Lu. Security and privacy for digital twins in the industrial internet of things: Challenges and solutions. IEEE Access, 8:31346–31358, 2020.

[3] [GS12] Edward H. Glaessgen and David Stargel. The digital twin paradigm for future nasa and u.s. air force vehicles. Proceedings of the 53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, pages 1–14, 2012.

[4] [HL17] Ching-Wen Hsu and Chia-Ter Lin. Anomaly detection in industrial control systems with machine learning algorithms. Journal of Computer Science and Technology, 32(6):1159– 1175, 2017.

[5] [MG20] Peter Mell and Timothy Grance. The nist definition of cloud computing (special publication 800-145). National Institute of Standards and Technology, 2020.

[6] [RS22] Patrick Robinson and Rajeev Singh. Leveraging threat intelligence in automated cybersecurity systems: Case studies and best practices. Journal of Information Security, 45(7):1342– 1361, 2022.

[7] [Tal24] Sanat Talwar. Dns over https (doh) in gaming: Balancing privacy and threat visibility. Computer Fraud and Security, 2024(12):349–356, 2024.

[8] [Tal25] Sanat Talwar. Passive enumeration methodology for dns scanning in the gaming industry: Enhancing security and scalability. ESP International Journal of Advancements in Computational Technology, 3(1):102–110, 2025.

[9] [TZL18] Fei Tao, Li Zhang, and Yang Liu. Digital twin and its applications in industry. International Journal of Advanced Manufacturing Technology, 94(9-12):3971–3987, 2018.

[10] [Yer23a] Srikanth Yerra. Leveraging python and machine learning for anomaly detection in order

[11] tracking systems. International Journal of Scientific Research in Computer Science, Engineering and Information Technology, 2023.

[12] [Yer23b] Srikanth Yerra. Reducing shipping delays through automated etl processing and real-time data insights. International Journal of Scientific Research in Computer Science, Engineering and Information Technology, 2023.

[13] [Yer24a] Srikanth Yerra. The impact of ai-driven data cleansing on supply chain data accuracy and master data management. Smart Computing Systems, 4(1):187–191, 2024.

[14] [Yer24b] Srikanth Yerra. Improving customer satisfaction with predictive analytics in logistics and delivery systems. Smart Computing Systems, 4(1):187–191, 2024.

[15] [Yer25a] Srikanth Yerra. Enhancing inventory management through real-time power bi dashboards and kpi tracking. International Journal of Scientific Research in Computer Science, Engineering and Information Technology, 2025.

[16] [Yer25b] Srikanth Yerra. Optimizing supply chain efficiency using ai-driven predictive analytics in logistics. International Journal of Scientific Research in Computer Science, Engineering and Information Technology, 2025.

[17] [ZC18] Hui Zhu and Jinyuan Chen. Security challenges of digital twin technologies in industrial applications. Proceedings of the 2018 IEEE International Conference on Industrial Technology, pages 2774–2779, 2018.