**Journal of Information Systems Engineering and Management**
2025, 10(30s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

**Research Article**

# Self-Repairing Security Systems for Manufacturing Networks: Proactive Threat Defense and Automated Recovery

[1]Aakarsh Mavi (Corresponding Author), [2]Sanat Talwar
*mavi.aakarsh4@gmail.com*
*sanattalwar1994@gmail.com*

**ARTICLE INFO**

Received: 30 Dec 2024

Revised: 05 Feb 2025

Accepted: 25 Feb 2025

**ABSTRACT**

As manufacturing networks become more connected, having strong and flexible security measures is more important than ever. This paper dives into self-repairing security systems for manu- facturing environments, emphasizing how these systems can automatically spot, tackle, and recover from security issues in real time. The goal is to create advanced solutions that let manufacturing networks reconfigure defenses—like firewalls and intrusion detection systems—on their own when- ever a breach or vulnerability is detected. The paper also looks at building automated systems that can restore compromised data or functionality quickly, which helps cut down on downtime and keep the network running smoothly. By using energetic monitoring, machine learning, and automatic response protocols, this research suggests a framework for self-healing systems that not only address security threats but also evolve to prevent future problems. The findings aim to boost the resilience and security of manufacturing networks, ensuring business continuity and reducing reliance on manual interventions.

**Keywords:** Self-repairing security systems, automated detection, response automation, vulnera- bility scanning, network defense, patch management, continuous improvement, incident recovery, busi- ness continuity, legacy systems integration, network segmentation, ICS, SCADA systems, operational technology (OT), disaster recovery, threat intelligence, system resilience, security breach recovery, manufacturing environment, Industrial Internet of Things (IIoT), cloud computing, Ansible, Python.

## 1 INTRODUCTION

Digital technologies are changing the game for manufacturing networks, turning old-school industrial systems into interconnected, data-driven powerhouses. With the rise of the Industrial Internet of Things (IIoT), cloud computing, and smart manufacturing solutions, these networks are now more complex and expansive than ever. While these advancements offer a ton of operational perks, they also bring new vulnerabilities and expand the areas at risk of cyberattacks. As a result, cybersecurity is a major concern because breaches in manufacturing networks can cause not just production downtime but also theft of intellectual property and even safety hazards.

Old-school security measures like firewalls, intrusion detection systems, and antivirus software are having a hard time keeping up with the new threats facing manufacturing. Many of these systems depend on human oversight to spot, handle, and recover from security issues, and this can lead to frustrating delays, prolonged downtimes, and even more vulnerabilities. Plus, as cyber threats get smarter and more persistent, we need a faster, automated approach to cybersecurity now more than ever.

To tackle these challenges, self-repairing security systems show a lot of promise for beefing up the resilience of manufacturing networks. These systems can automatically detect, fight off, and recover from security incidents in real-time without needing manual help. They keep a constant eye on network traffic, devices, and applications to spot any weird behavior or breaches. This means they can instantly adjust defenses and fix any compromised data or functions. By using machine learning, adaptive security protocols, and automation, self-repairing security systems can react much quicker and more effectively than traditional methods, making sure

that critical manufacturing operations experience minimal disruption. As cloud-native infrastructures and connected devices proliferate in industrial environments, automated detection of vulnerabilities, like those found in cloud configurations, becomes crucial for strengthening network security (Talwar, 2022) [Tal22].

This paper dives into the design and implementation of these self-repairing security systems for manufacturing networks, emphasizing how they can autonomously find vulnerabilities, adjust defenses, and return operations to normal. We suggest a framework that ties together real-time monitoring, auto- mated responses, and ongoing recovery to secure and maintain the integrity of industrial environments. Through this research, we aim to help create more resilient manufacturing networks that can stand up to cyber threats, keep downtime to a minimum, reduce needing human intervention, and ensure the ongoing safety and reliability of industrial systems.

## 2 LITERATURE REVIEW

In recent years, needing strong cybersecurity in manufacturing networks has really come into focus. With industrial systems getting more complex and interconnected, it's essential to keep everything safe. As manufacturing welcomes digital tools like IoT devices, cloud solutions, and industrial control systems (ICS), these environments are becoming more exposed to cyber threats. This part takes a look at existing research on cybersecurity in manufacturing, self-healing systems, and automation techniques that support self-repairing security measures, laying the groundwork for this study.

### 2.1 Cybersecurity Challenges in Manufacturing Networks

The rise of smart manufacturing and the Industrial Internet of Things (IIoT) has brought about a whole new set of challenges for securing manufacturing networks. A study by Kassler et al. (2018)[Kas18] points out that the digital transformation in manufacturing grows the attack surface, as older systems with limited security often connect with newer, advanced technologies. A lot of industrial networks weren't built with security in mind, and they lack the strong defenses needed to handle today's cyber threats. The vulnerabilities in ICS and SCADA (Supervisory Control and Data Acquisition) sys- tems are especially worrisome since they play a critical role in industrial operations. Hristov et al. (2019)[Hri19] stress that the complexity of securing these systems comes from mixing legacy protocols, outdated software, and minimal security monitoring.

Also, research by Li et al. (2020)[Li20] simplifies the rising threat of ransomware and cyber-physical attacks on manufacturing systems. These attacks can lead to production delays, loss of intellectual property, and even jeopardize worker safety, making it clear that we need advanced, adaptive cyberse- curity measures that can proactively tackle vulnerabilities.

### 2.2 Traditional Approaches to Security in Manufacturing Networks

Traditionally, security measures like firewalls, intrusion detection systems (IDS), and antivirus software have been widely used to protect manufacturing networks. But these conventional systems often rely on manual efforts to spot and fix security issues. Kiss et al. (2018)[Kis18] argue that while IDS and firewalls provide essential security features, they might not be enough to handle the nature of cyber threats, especially in tight timing where human intervention could slow things down. What's more, Fernandez et al. (2021)[Fer21] point out the downsides of traditional cybersecurity frameworks, which tend to be reactive instead of proactive, making organizations more vulnerable in the early moments of an attack.

### 2.3 Self-Repairing Security Systems

Self-healing security systems are all the buzz lately. These smart systems can spot problems, react, and bounce back from security issues all on their own. According to Rasmussen et al. (2017)[Ras17], these automated setups can restore compromised network functions without needing people to step in. They keep an eye on everything with real-time monitoring and detection algorithms, so they can quickly adjust their security settings if something seems off. With features like automatically reconfiguring firewalls or retrieving files from secure backups, they help keep downtime to a minimum and limit the chances for attackers to cause harm.

A big part of these self-healing systems is using machine learning (ML) and AI for smart, adaptive reactions.

Tufail et al. (2020)[Tuf20] point out that when AI gets mixed into cybersecurity, it helps systems learn from past events and stay ahead of future threats, which makes defending against new

forms of attacks much easier. For example, Liu et al. (2021)[Liu21] display how ML algorithms can detect odd behavior in manufacturing networks and automatically kick off corrective actions based on set security rules.

### 2.4 Automated Incident Response and Recovery

Automated incident response is hugely important for these self-healing systems. Alcaraz et al. (2018)[Alc18] discuss how automation can speed up incident response times because it cuts out needing humans to

get involved right away during a security breach. These systems can determine threats, isolate affected systems, and start predefined recovery processes, like patching vulnerabilities or restoring services from backups. It's especially critical in manufacturing settings where keeping everything running smoothly is essential.

Research by Sharma et al. (2020)[Sha20] dives into using blockchain technology in self-healing systems to ensure data integrity and autonomous recovery. Blockchain's decentralized setup can create tamper-proof records of network events, making sure that data recovery is both secure and auditable. Also, Sharma et al. (2021)[Sha21] emphasize how automated backup systems can help industrial networks restore data without needing any human assistance.

## 3 FRAMEWORK DESIGN

In manufacturing networks, a self-repairing security system works like having a security team that never sleeps. It combines several layers of security measures, automated responses, and recovery options. The goal is to spot, fix, and bounce back from security problems in real-time, keeping downtime to a minimum and ensuring everything runs smoothly. This system includes key features like real-time monitoring, machine learning for spotting anomalies, automated reactions to incidents, and mechanisms for recovering data. It's a comprehensive take on security automation.

### 3.1 System Overview

Here's a breakdown of the framework, which has four main layers:

· **Detection Layer:** This layer keeps an eye on the manufacturing network, looking out for security threats and weaknesses.

· **Response Layer:** When a security issue pops up, this layer automatically takes action based on a set plan to address it.

· **Recovery Layer:** If something goes wrong, this layer can autonomously restore data, system functions, or settings.

· **Continuous Improvement Layer:** This layer learns from feedback provided by the others to boost security measures and responses over time. Each layer works in real-time, using automation to quickly identify threats and recover from incidents without needing a human to step in.

### 3.2 Detection Layer

The Detection Layer is all about keeping tabs on the network for any potential security threats. It gathers information from various components like industrial control systems (ICS), IoT devices, fire- walls, and intrusion detection systems (IDS). In addition to machine learning-based anomaly detection, a robust vulnerability scoring framework, such as the one described by Talwar and Mavi (2023), can assist in evaluating network security and prioritizing remediation actions [TM23]. Here's how it works:

· **Anomaly Detection:** Using machine learning (ML), the system analyzes network traffic, device behavior, and logs to catch odd patterns that could signal security breaches. These ML models learn from past data and get regular updates from new threat intelligence.

· **Vulnerability Scanning:** It runs real-time checks on network components with automated tools that spot known vulnerabilities like outdated software, misconfigurations, or weak authentication practices. This alerts the system to areas that could be at risk.

**Key Technologies:**

· Machine learning algorithms for spotting anomalies (like supervised and unsupervised learning).

· Integrated threat intelligence feeds for the latest threat updates. Vulnerability scanners (like OpenVAS and Nessus) for ongoing vulnerability assessments.

### 3.3 Response Layer

When a security issue pops up, the Response Layer jumps into action, taking steps we've already set up to tackle the threat and stop any more damage from happening. What it does next really depends on how bad the threat is, as flagged by the detection layer. Here are some of the moves it might make:

· **Network Defense Reconfiguration:** This means automatically tweaking firewalls, intrusion prevention systems (IPS), or access control lists (ACLs) to block or limit access to compromised parts of the network. It might involve changing up the network segments on the fly or isolating affected systems.

· **Service Shutdown or Quarantine:** For serious threats like ransomware or malware infections, the system can automatically quarantine or shut down the affected systems or services to stop the attack from spreading.

· **Patch Management:** The system can kick off patching processes to fix vulnerabilities or soft- ware bugs that hackers could exploit. Updates are rolled out based on the system's setup and what's been identified as a risk.

**Key Technologies:**

· Automation tools (like Ansible or Puppet) for reconfiguring defenses and patch management.

· Endpoint security tools for quick isolation and quarantine (like Symantec or CrowdStrike).

· Intrusion prevention systems (IPS) to block threats in real time.

### 3.4 Recovery Layer

The Recovery Layer is essential for keeping downtime to a minimum and getting things back on track after an attack. It helps the system recover compromised data, services, or functions automati- cally.Ensuring business continuity after a breach is vital for manufacturing networks, and integrating a comprehensive disaster recovery plan, as outlined by Rana et al. (2023), is crucial for minimizing downtime and restoring operations swiftly.[MRH23] Key recovery processes include:

· **Automated Data Restoration:** Using secure backup systems, this layer can automatically bring back lost or corrupted data from encrypted backups, ensuring essential manufacturing data, settings, and historical logs are restored.

· **Service and System Restoration:** The system automatically brings back essential appli- cations or processes that might have been disrupted during an attack. This covers restarting servers, regaining access to control systems, and making sure production systems are up and running without needing manual input.

· **Configuration Rollback:** If system settings were messed with during an attack—like firewalls being misconfigured or access credentials changed—the system can roll back to the last known good configuration automatically.

**Key Technologies:**

· Backup and recovery systems (like Veeam or Acronis) for getting data back.

- Version control tools for managing configurations (like GitOps or Terraform).
- Disaster recovery orchestration tools to automate getting services back up.

### 3.5 Continuous Improvement Layer

The Continuous Improvement Layer is all about learning from what's happened before to keep our security strong. This layer takes in data from the Detection, Response, and Recovery elements, and uses it to sharpen security settings, response options, and detection methods.

Here are the main components:

- **Incident Feedback Loop:** We keep track of every security incident, including how it happened, how we spotted it, what actions we took, and how we got things back to normal. Analyzing this info helps us find patterns and figure out any weaknesses in our defenses.
- **Adaptive Learning:** The machine learning models in our Detection Layer get updated with new insights from past incidents. These models continuously learn using data from our network and outside threat intelligence to stay effective against new threats.
- **Policy Refinement:** We fine-tune our automated security policies based on the lessons learned from earlier incidents, making sure our response strategies stay sharp and aligned with current risks and best practices.

**Key Technologies:**

- AI/ML-based feedback systems for continuous learning.
- Data analytics tools for digging into and reporting on security incidents.
- Policy management systems to tweak security settings based on experience.

### 3.6 Integration with Existing Infrastructure

To really work well, our framework needs to mesh smoothly with existing manufacturing setups, which might include older systems, IoT devices, and industrial control systems. It's important that we integrate with current cybersecurity tools like firewalls and IDS/IPS to ensure we have complete monitoring and control. The framework is built to support mixed environments, so both new and older systems can work together and benefit from our self-repairing capabilities.

**Key Integration Points:**

- API integrations with current security tools and industrial control systems.
- Real-time data syncing across manufacturing assets and monitoring platforms.
- Support for various industrial protocols like Modbus, OPC, and DNP3 for keeping an eye on and managing IoT devices.

## 4 IMPLEMENTATION

Setting up a self-repairing security framework for manufacturing networks means rolling out several components that work together to offer automated detection, quick responses, recovery, and ongoing enhancements. Here, we'll dive into the design, tools, and implementation code needed for each part of the framework. We're using a mix of Python, Ansible, machine learning models, and various cybersecurity tools to create a strong, self-healing solution.

### 4.1 Detection Layer Implementation

The Detection Layer keeps an eye on the manufacturing network for any possible security threats. It uses machine learning (ML) for spotting anomalies and conducts vulnerability scans to identify threats in real time.

**Tools Used:**

- **Machine Learning Libraries:** Scikit-learn, TensorFlow (to train and predict models).
- **Vulnerability Scanners:** OpenVAS (for assessing vulnerabilities in real time).
- **Syslog/Log Collectors:** ELK Stack (Elasticsearch, Logstash, Kibana) for analyzing logs.

### 4.1.1 Anomaly Detection Model

The anomaly detection system employs a machine learning model trained on network traffic data to spot unusual behavior as it happens. Here's how you can implement a basic anomaly detection model using Scikit-learn.

```
import pandas as pd
from sklearn.ensemble import IsolationForest
from sklearn.model selection import train test_split from sklearn.metrics import classification report
# Load the network traffic data (e.g., from a CSV or database)
data = pd.read csv('network traffic.csv')
# Split the data into features and labels X = data.drop('label', axis =1)
# Features (network traffic data) y = data['label']
# Labels (normal/attack)
# Split the data into training and testing sets
X train, X test, y train, y test =
train test split(X, y, test size =0.3, random state =42)
# Train the anomaly detection model
model = IsolationForest(n estimators =100, contamination =0.1, random state =42)
model.fit(X train)
# Predict anomalies
y pred = model.predict(X test)
# Evaluate the model
print(classification report(y test, y pred))
```

### 4.1.2 Vulnerability Scanning with OpenVAS

To perform real-time vulnerability scanning, OpenVAS can be used to spot known weaknesses through- out the network.

- Start by installing OpenVAS and set it up to scan your network.
- Use the OpenVAS API for triggering scans automatically. import openvas lib

```
# Initialize OpenVAS client and connect to the OpenVAS server
client = openvas lib.OpenVASClient
(username='admin', password='password')
# Start a new scan job
task id = client.create task
```

```
('Network Vulnerability Scan', '192.168.0.0/24')
# Run the scan task
scan id = client.start task(task id)
# Retrieve the results once the scan is complete scan results = client.get task results(scan id)
# Process and analyze the scan results
for result in scan results['vulnerabilities']:print(result['title'], result['severity'])
```

### 4.2 Response Layer Implementation

The Response Layer takes charge of automating responses to security incidents that are detected. This includes reconfiguring network defenses, isolating compromised systems, and applying patches.

**Tools Used:**

- Ansible for automating network reconfigurations and patch management.
- **Firewall Configuration:** iptables, UFW (Uncomplicated Firewall) for Linux-based defenses.

#### 4.2.1 Automated Firewall Reconfiguration with Ansible

You can use Ansible playbooks to adjust firewall rules or isolate compromised systems based on threat detection. For instance, if a system is flagged as compromised, Ansible can cut it off from the network.

```
---
— name: Isolate compromised system hosts: compromised system
tasks:
— name: Block all incoming trafficufw:
state: enabled rule: deny
direction: incoming
from ip: 192.168.0.100
```

#### 4.2.2 Automated Patch Management with Ansible

Ansible is also handy for applying patches to vulnerable systems that have been flagged during the scanning process.

```
---
— name: Apply security patches to vulnerable systems hosts: vulnerable systems
tasks:
— name: Update all packages apt:
upgrade: dist
update cache: yes
```

### 4.3 Recovery Layer Implementation

The Recovery Layer is all about restoring compromised data, services, or system configurations. This step is key to minimizing downtime and getting systems back to a secure state as quickly as possible.

**Tools Used:**

- **Backup and Restore Tools:** Veeam, Acronis, or rsync for restoring data.
- **Configuration Management:** GitOps, Terraform for rolling back configurations.

#### 4.3.1 Automated Data Restoration

If data corruption or loss occurs, the system can automatically restore from secure backups. For example, using rsync to recover critical data.

```
rsync –avz /backup/critical data/ /var/data/
```

#### 4.3.2 Service Restoration with Ansible

If any services or systems go down, Ansible can be used to automatically bring them back online.

```
---
— name: Restart affected serviceshosts: compromised systems
tasks:
— name: Restart production application service:
name: production servicestate: restarted
```

#### 4.3.3 Configuration Rollback

If a security incident alters system configurations, you can roll back to a previously known good configuration using tools like GitOps.

```
# Git commands to rollback configuration changes git checkout HEAD~1 /etc/confi
g/
```

### 4.4 Continuous Improvement Layer Implementation

The Continuous Improvement Layer makes the system smarter by learning from past incidents. It takes feedback from the Detection, Response, and Recovery layers to refine detection models, response policies, and recovery protocols.

**Tools Used:**

- **Feedback Loop Analysis:** Employ machine learning models for ongoing learning.
- **Data Analytics:** Use Python and Pandas to dig into incident reports.

#### 4.4.1 Feedback Loop for Adaptive Learning

Every time there's a security incident, we log all the details to analyze later and enhance our detection algorithms. We can even automate this by plugging incident logs into the anomaly detection model.

```
import pandas as pd
from sklearn.ensemble import IsolationForest
# Load incident logs (e.g., from Elasticsearch)
incident logs = pd.read csv('incident logs.csv')
# Re–train the anomaly detection model with new incident data
X = incident logs.drop('label', axis=1)y = incident logs['label']
# Re–train model
model = IsolationForest(n estimators=100, contamination=0.1)
model.fit(X)
# Save the updated model import joblib
```

```
joblib.dump(model, 'updated anomaly detection model.pkl')
```

### 4.5 Integration with Existing Infrastructure

The self-repairing security system needs to blend smoothly with what's already out there in manufac- turing networks, such as ICS, SCADA systems, and IoT devices.

**Integration Points:**

· **API-based Integration:** Connect the self-healing system with existing ICS/SCADA systems through APIs for effective monitoring and defense.

· **Device Compatibility:** Make sure the system works with industrial protocols like Modbus, OPC, and DNP3.

**Example API Integration for ICS**:

```
import requests

# Make an API call to the ICS system to monitor its health

response = requests.get('http://ics-system.local/api/status')

# Check if the system is compromised and respond accordingly if response.status
code == 200 and

response.json()['status'] == 'compromised': # Trigger response actions

(e.g., isolate system, reconfigure firewall)

trigger isolation(response.json()['ip address'])
```

## 5 FUTURE WORK

The self-repairing security system framework introduced in this research is a solid way to protect manufacturing networks, but there are definitely some areas we can dive into for future improvements. Here are a few ideas for what we could work on next:

· **Scalability and Performance Optimization:** As manufacturing networks become larger and more complex, we need to make sure our self-repairing security system can keep up. Future efforts should focus on tweaking the framework for larger setups, so automated detection, response, and recovery can handle the extra load from more devices and possible vulnerabilities without slowing down. This might involve refining the system's design and using smarter ways to handle and

process data.

· **Integration with Legacy Systems:** A lot of manufacturing networks still depend on older systems that might not play nice with modern cybersecurity tools. A key area for future research is figuring out how to connect the self-repairing system with these legacy setups. This could mean developing compatibility solutions, API layers, or protocols that allow for smooth integration while keeping the integrity of older industrial control systems (ICS) and SCADA systems intact.

· **Regulatory Compliance and Auditing:** Staying compliant with industry regulations like NIST, IEC 62443, or ISO/IEC 27001 is essential to make sure our self-repairing security system meets legal and operational standards. Future projects should aim to automate compliance checks within the system, ensuring every security measure and response not only works but also follows the necessary regulations. This would include creating detailed compliance reports and

logs that are easy to audit for adherence to security policies.A key part of self-repairing security systems is ensuring compliance with relevant cybersecurity regulations, such as ISO/IEC 27001, which can be supported by frameworks for cloud-based data governance (Naik, 2023).[Nai23]"

· **Network Segmentation and Isolation Strategies:** Boosting the system's ability to segment and isolate compromised network sections is key to preventing threats from spreading. Future efforts should

look at adding automated, flexible network segmentation techniques that can react to various security events and threats. This way, the system can isolate affected parts without causing disruptions in the overall operation of the manufacturing environment.

· **Patch Management and Update Automation:** Right now, we're using Ansible to patch vulnerable systems, but future work could simplify the patch management process with better testing and validation methods. This might include automatically testing patches in a staging area to confirm they don't introduce new issues or disrupt operations before going live. Also, we

could enhance automation to keep track of patching status and automatically revert updates if problems crop up.

· **Incident Recovery and Business Continuity:** We should refine the recovery process to guarantee that business operations can keep going during and after a security incident. Fu- ture work could dive into improving backup and restore methods, bolstering redundancy, and integrating disaster recovery plans to minimize downtime and quickly restore critical services.

Also, we should customize our recovery plans to meet the unique needs of the manufacturing environment, considering both IT and operational technology (OT) systems.

By focusing on these areas, the self-repairing security system can become a more well-rounded, flexible, and resilient solution for keeping manufacturing networks safe against growing cyber threats.

## 6 CONCLUSION

This paper dives into creating and implementing self-repairing security systems for manufacturing networks. The goal? To build a strong, automated security framework that can spot issues, respond to them, and bounce back with little to no human input. By combining things like anomaly detection models, vulnerability scans, automated responses, and recovery processes, the system offers a flexible way to boost security and keep things running smoothly in manufacturing settings.

We're using machine learning for real-time anomaly detection, using Ansible to automate responses and manage patches, and applying various recovery techniques like data restoration and service roll- backs. Plus, there's a continuous improvement aspect, which takes feedback from incidents to help the system adapt to new threats and get more effective over time.

This self-healing method could really cut down on downtime and lessen the blow from security breaches, especially since manufacturing networks can be pretty sensitive to disruptions. Automating both preventive and corrective measures means organizations can keep their security tight without being heavily reliant on manual input, making risk management much more efficient.

While this framework shows a lot of promise for securing industrial networks, there's definitely room for future exploration—like integrating advanced threat intelligence platforms, scaling the system for different network architectures, and optimizing machine learning models for better threat detection precision. Plus, making sure it works with older systems and meets industry regulations is an important area for improvement too.

In the end, this research emphasizes the huge potential of self-repairing security systems in man- ufacturing networks. It offers a proactive and flexible approach to cybersecurity that can really help manage risks and boost overall resilience against cyber threats.

## REFERENCES

[1] [Alc18] et al. Alcaraz, C. Automated incident response in manufacturing networks: A survey. *Computers Security*, 75:1–20, 2018.

[2] [Fer21] et al. Fernandez, E. Reactive vs. proactive cybersecurity frameworks in industrial environments. *Computers Industrial Engineering*, 156:107243, 2021.

[3] [Hri19] et al. Hristov, I. Challenges in securing industrial control systems and scada networks. *Computers Security*, 83:1–15, 2019.

[4] [Kas18] et al. Kassler, J. The rise of smart manufacturing and the industrial internet of things (iiot) and its impact on cybersecurity. *Journal of Manufacturing Science and Engineering*,

140(10):101001, 2018.
[5] [Kis18] et al. Kiss, I. Limitations of traditional cybersecurity measures in manufacturing networks. *Procedia CIRP*, 72:101–106, 2018.
[6] [Li20] et al. Li, X. Ransomware and cyber-physical attacks on manufacturing systems: A survey. *Journal of Manufacturing Processes*, 56:1–14, 2020.
[7] [Liu21] et al. Liu, Y. Machine learning for anomaly detection in manufacturing networks. *Journal of Manufacturing Science and Engineering*, 143(4):041012, 2021.
[8] [MRH23] CHAFIK Khalid Milankumar Rana and EL HASSANI Hajar. Disaster recovery plan for business continuity. *Indian Journal of Economics and Business*, 22(1), 2023.
[9] [Nai23] S. Naik. Cloud-based data governance: Ensuring security, compliance, and privacy. *The Eastasouth Journal of Information System and Computer Science*, 1(01):69–87, 2023.
[10] [Ras17] et al. Rasmussen, M. Self-healing security systems: Automated restoration of network functions. *Journal of Network and Computer Applications*, 85:1–12, 2017.
[11] [Sha20] et al. Sharma, S. Blockchain technology in self-healing systems: Ensuring data integrity and autonomous recovery. *Future Generation Computer Systems*, 108:100–115, 2020.
[12] [Sha21] et al. Sharma, S. Automated backup systems for industrial networks: Enhancing data recov- ery without human intervention. *Journal of Industrial Information Integration*, 22:100191, 2021.
[13] [Tal22] S. Talwar. Securing cloud-native dns configurations: Automated detection of vulnerable s3-linked subdomains. *International Journal of Applied Engineering and Technology*, 4(2):270– 278, 2022.
[14] [TM23] S. Talwar and A. Mavi. An overview of dns domains/subdomains vulnerabilities scoring framework. *International Journal of Applied Engineering and Technology*, 5(S4):274–280, 2023.
[15] [Tuf20] et al. Tufail, M. Integrating ai into cybersecurity: Enhancing defense against emerging threats. *Artificial Intelligence Review*, 53:1–27, 2020.