DEFINING THE DIGITAL TWIN FOR INDUSTRY 4.0

BY

ZACHARY A. DEWARDENER

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE

REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

IN

SYSTEMS ENGINEERING

UNIVERSITY OF RHODE ISLAND

2023

MASTER OF SCIENCE THESIS

OF

ZACHARY A. DEWARDENER

APPROVED:

Thesis Committee:

Major Professor: Manbir S. Sodhi

David G. Taggart

Resit Sendag

David Chelidze
DEAN OF THE GRADUATE SCHOOL

UNIVERSITY OF RHODE ISLAND

2023

ABSTRACT

This manuscript offers a comprehensive review of Digital Twin (DT) technology, first presenting the proposal of our novel DT amendment that looks to refine the conceptual understanding of DTs across various industrial use-cases. The study synthesizes findings from an extensive literature review, emphasizing the need for precise use-case-centric definitions and categorization to advance the field effectively. Chapter 1 introduces the current state of DT research, advocating for a more nuanced approach to the identification, classification, and validation of DT systems. It challenges the broad definitions prevalent in current literature and proposes a more application-specific research trajectory.

Subsequent chapters build on this foundation by curating an extensive review of Supervisory Control And Data Acquisition (SCADA) systems. This explanation illustrates that industry's current definition of DT follows suit with the capabilities of such technology when an automatic, bi-directional data transfer is required for simulation models to be classified as DTs. After this, we explore the application of DT technology in dataset creation for the training of Intrusion Detection Systems (IDS) at the industrial application layer. Lastly, we propose a novel validation method for DTs by utilizing machine learning techniques while identifying key challenges such as system complexity and the need for diverse similarity measures.

This manuscript brings to question the current literature's tendency to create self-validating information silos and highlights the discrepancy in validation approaches for different systems. It acknowledges the infancy of practical applications and calls for a standardized validation framework.

PREFACE

This Thesis is presented in a manuscript format in accordance with the University of Rhode Island Graduate School Guidelines. The Thesis is composed of four manuscripts that have been either published or prepared for submission. These manuscripts are listed in the following aspects:

- **Manuscript I: Redefining the Digital Twin**

  This manuscript has been prepared for publication but is not submitted yet.

- **Manuscript II: SCADA Systems**

  This manuscript has been prepared for publication but is not submitted yet.

- **Manuscript III: Data-Driven Defense for Smart Manufacturing: A Novel Anomaly Detection Dataset and Comparative Algorithmic Study**

  This manuscript has been prepared for publication but is not submitted yet.

- **Manuscript IV: Digital Twin Validation**

  This manuscript has been prepared for publication but is not submitted yet.

TABLE OF CONTENTS

Page

Page

MANUSCRIPT 1

Redefining the Digital Twin

Zachary A. deWardener[1*], and Manbir S. Sodhi[1]

[1]Department of Industrial Engineering, University of Rhode Island, Kingston, Rhode Island, 02881

[*]Corresponding author: zdewardener@uri.edu

Drafted for submission, XXX–XXX.

ACM ISBN XXX-X-XXXX-XXXX-X/XX/XX

https://doi.org/XX.XXXX/XXXXXXX.XXXXXXX

1.1 Abstract

The field of Digital Twin (DT) research is constantly evolving, necessitating the need for a more precise definition and categorization to guide its development and application across multiple domains. This paper offers a theoretical proposal for the redefinition of DTs, aiming to refine the conceptual understanding of DT technology. By analyzing recently published literature reviews, the paper synthesizes the current state of DT research, focusing on the identification, classification, and validation of DT systems. An underlying goal within this study is to emphasize that an automatic bidirectional data transfer between physical and virtual entities should not be a defining characteristic of a DT, making the point that no simulation model can be a 100% accurate mirror of their physical counterpart. This situation causes the digital twin model to converge with the physical system, in essence creating a SCADA system with predictive capabilities. This paper advocates for a more goal-oriented research approach, particularly in application-specific contexts like machining, where the physical and virtual dimensions of DTs must align with the operational realities and technological infrastructure. It also addresses the ethical concerns and limitations of DT technology, suggesting a cautious expansion into sensitive areas.

By proposing a renewed definition of DT technologies, while running the risk of further diluting the concept, this paper contributes to a more structured discourse on DTs. It identifies gaps in knowledge, particularly in the virtual-to-physical connections, and calls for research on validation techniques tailored to specific DT applications across various industries. The ultimate goal is to enhance the practical utility of DTs and ensure their secure implementation.

1.2 Introduction

Digital twin (DT) technology, emerging as a key topic in the 21st century, is gaining attention in both academic and industrial sectors. This concept aims to create high-fidelity simulation models of physical systems, facilitating informed decision-making through 'what if' scenarios. Additionally, when combined with Machine Learning (ML), digital twins can be used for predictive tasks such as maintenance and schedule optimization, though they require a significant amount of real-time data. Another notable aspect is their capability for real-time monitoring and the interactive manipulation of both the physical system and its virtual counterpart.

Immediately apparent from the outlined capabilities of Digital Twin (DT) systems is the emergence of three distinct definitions. [i] A high-fidelity simulation model whose purpose is performing scenario testing, [ii] A digital proxy of the real system for implementation of ML algorithms with the goal of performing predictive operations, and [iii] A real-time visualization of the physical system, allowing for the bi-directional manipulation of physical and virtual spaces. While this is not the first article made for the purpose of refining DT definitions into a comprehensive grouping, our approach is much different from the others. The objective of this paper is to shed light on the fallacies of DT research and propose a novel categorization methodology that will hopefully allow the concept of DTs to be fully explored within the industrial realm.

The article is structured as follows: Section 1.3 outlines the methodology for our DT technology review. Section 1.4 traces the evolution of DTs. Section 1.5 examines academic perspectives on DTs. Section 1.6 presents our DT definition, with Section 1.7 analyzing its implications. The paper concludes with future research directions in Section 1.8 and a summary in Section 3.8.

1.3 Methodology

The objective of our work was to consolidate and analyze recently published works pertaining to the concept of DT to realize the history and classification methods used throughout its inception for various applications. We selected our search keywords based on finding the most information on relevant terminology for the concept of interest such as "digital twin", "survey", "review", "classification", and developed the following search query: ("digital twin" AND "survey") OR ("digital twin" AND "review") OR ("digital twin" AND "classification"). This search provided a total of 278,000 articles published as of November 2023 and were further refined through the screening of their titles, matching keywords, abstracts, and languages to exclude those sources which were not relevant to our research. Additional articles outside of this search criteria were found as citations mentioned in other articles and were included based on their relevance.

1.3.1 Sources of literature

We conducted a systematic literature review of academic articles and book chapters published to various reputable journals, mainly using the comprehensive publication database Google Scholars (`scholars.google.com`).

1.4 Historical Development of Digital Twins

The following section outlines the development of the DT concept from its earliest mention and publication through its evolution, concurrent with technological innovations, leading to the current trends most commonly seen in literature today.

1.4.1 Origin of the Digital Twin

The digital twin, while still in its infancy, was first conceptualized and utilized by NASA during preparation for the Apollo 13 mission in the 1960s as a means

for engineers to simulate possible solutions for a rescue mission in space [1]. Subsequently, the earliest publication of the term 'digital twin' was found in an article by Hernández [2] in 1997. Although only mentioned once within their article, the authors used the term to describe a novel method for iterative modifications in the design of urban road networks using digital modelling. The next, and arguably most influential, reference of the phrase was from Grieves [3] in a presentation at the Society of Manufacturing Engineering (SME) conference in Troy, Michigan in October of 2002. His presentation highlighted the potential benefits of creating a digital twin as an application towards Product Lifecycle Management (PLM), as seen in Figure 1. However, due to technological limitation of the time and a blossoming interest in the concept of cyber-physical systems (CPS), the work of Grieves had not received widespread attention. That is until a conference paper published in 2012 by Glaessgen and Stargel [4] which provided the first specific definition of digital twin for the domain of aerospace and aeronautical engineering:

"A Digital Twin is an integrated multiphysics, multiscale, probabilistic simulation of an as-built vehicle or system that uses the best available physical models, sensor updates, fleet history, etc., to mirror the life of its corresponding flying twin." [4]

This re-invigoration of the digital twin concept pushed Grieves to publish his white paper in 2014, further defining three main aspects of a digital twin within the domain of PLM, those being [i] physical entities in real space, [ii] virtual models in virtual space, and [iii] the data which connects physical and virtual entities together [5].

1.4.2 Evolution of the Digital Twin

Following Grieves' publication in 2014, his subsequent papers published in 2017 with Vickers [6] and later alone clarifying his intent for the digital twin in [7]

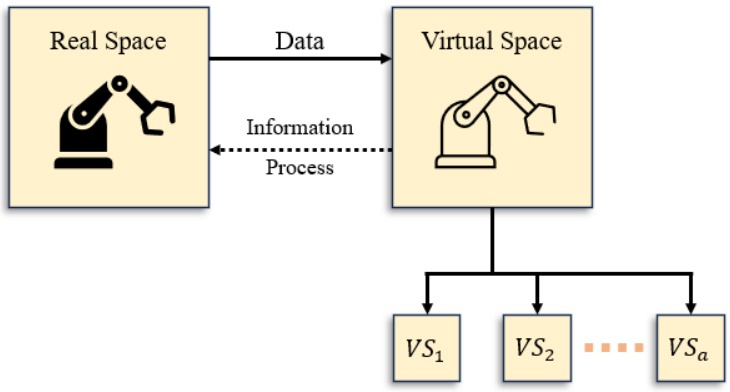


Figure 1. Grieves' proposed structure for a PLM-based DT

further elucidated the current fallacies proposed by literature and the differences between a Digital Twin Prototype (DTP), Digital Twin Instance (DTI), and Digital Twin Aggregate (DTA) as seen in Figure 2 with a closer look at the base-level DT structure in Figure 3.

- **Digital Twin Prototype (DTP)** – Consists of geometrical, physical characteristic, and manufacturing process data assembled within a virtual entity during the create phase.

- **Digital Twin Instance (DTI)** – Which is generated once an individual product of interest is manufactured is further linked to that physical entity and collects data throughout and exceeding its lifecycle. Containing information on any variations that occur within an actual production, pulled from sensor readings, and stored in a historical database to illustrate state changes, performance, and resulting outcomes.

- **Digital Twin Aggregate (DTA)** – Is comprised of multiple DTIs, amassing all the digital information and data from these instances, allowing the integration of machine learning (ML) and artificial intelligence (AI) algorithms for predictive operations to ensue.

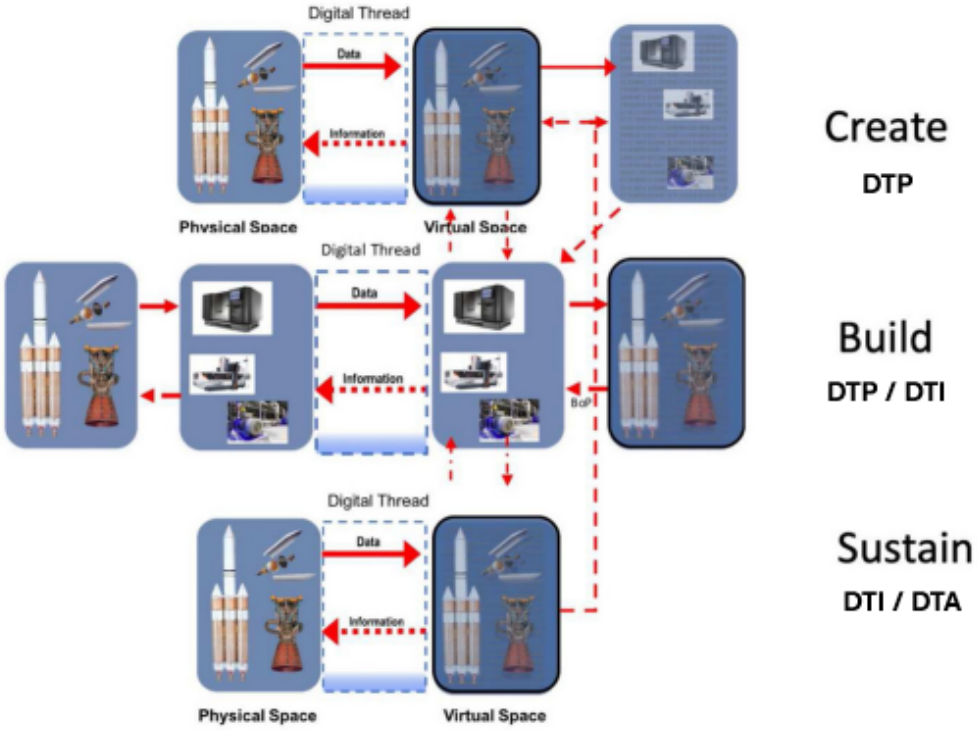


Figure 2. DTP, DTI, and DTA visualized, including interrelationships [8]

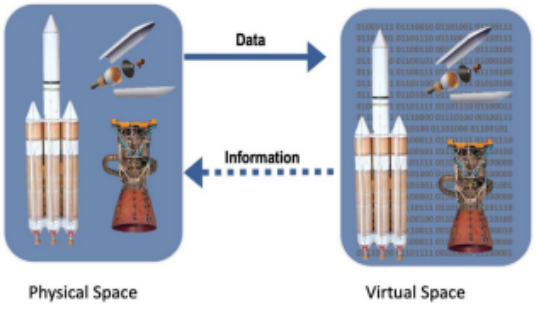


Figure 3. A closer look at the base-level DT structure [8]

Between 2014 and present day, publications covering digital twin technology across multiple industries have been shared and peer reviewed. However, much discourse remains among industry professionals and researchers alike when approaching the definition of a generalized digital twin for all domains. One major proponent of this discourse, in our eyes, is the work of Kritzinger in his approach to categorize digital twins via levels of data integration.

The work of Kritzinger et al. [9] is widely regarded as one of the main contributors to the field of digital twin research. The authors analyzed a total of 52 papers published between 2014-2017. They performed a systematic literature review of digital twins used in manufacturing and classified those works into three levels of data integration (Figure 4):

- **Digital Model (DM):** A virtual model of a physical product or system with no automatic data transfer between the physical and virtual entities.
- **Digital Shadow (DS):** A virtual model containing a one-way automatic transfer of data from the physical entity to its virtual counterpart.
- **Digital Twin (DT):** Defined as having an automatic bidirectional data transfer between physical and virtual entities, providing a synchronization of real and virtual spaces.

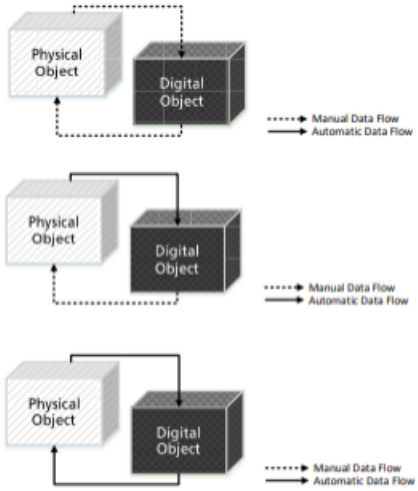


Figure 4. Krizinger's levels of data integration [9]

Their categorization methods have led to countless researcher citing his work (i.e., [10], [11], and supporting sources referenced in: [12], [13], [14], etc.) with the intention of stating that their implementation of digital twin technology is correct and supported by peer review. However, the authors of this paper note that there is a considerable number of studies which see data connectivity as a main problem with the current digital twin definition and further conclude their analysis by stating that the development of the digital twin is still at its infancy as literature mainly consists of concept papers without concrete case-studies. Our main issue with Kritzinger's categorization is that if a DT model is synchronized automatically with data from the physical twin and vice-versa, this creates a situation where the DT converges with the physical system [15]. In essence, this creates a high-fidelity Supervisory Control And Data Acquisition (SCADA) system with visualization and predictive capabilities via ML and AI algorithm integration. SCADA systems serve as the focal point for collecting, processing, and presenting data from automated or semi-automated operations, while also facilitating the centralized management of machinery and equipment, both on-site and remotely. Due to

advancements in Industrial Internet of Things (IIoT) technology, modern SCADA systems do not merely encompass monitoring and control capabilities; they also include predictive maintenance, real-time analytics, and optimizing operations using data-driven insights [16].

1.4.3 Current Trends

According to [17], and as anticipated by Research&Markets, the global digital twin market will reach $73.5 billion by 2027, with a 60.6% compound annual growth rate during 2022-2027. However, the only way to sustain this type of growth is to make sure that the research community is on the same page. As it stands, no stable consensus has been drawn with respect to the definition of a digital twin which, in part, is due to the vast application areas of DT. Researchers have applied their unique renditions of DT technology across various domains [18] from the construction of smart cities and utilities management [17], to manufacturing [19] and healthcare [13].

1.5 Theoretical Perspectives on Digital Twins

Due to the relatively recent indoctrination of DT technology and its vast applicability across multiple domains, there are many theoretical perspectives on the concept. This section aims to provide a comprehensive view for some of the most relevant proposals and adaptations supplied by literature.

1.5.1 Existing Definitions

Table 1 shows some of the existing DT definitions used throughout the years of its development in literature. A trend can be seen within these results and those of other survey papers, such as [20] and [21]. DT publications within the earlier years of conception focused more on high-fidelity simulation modelling while later implementations stress the apparent need for a bi-directional data transfer and

automatic synchronization with the physical entity.

Source	Year	Definition
[22]	2015	Very realistic models of the current state of the process and their behaviors in interaction with their environment in the real world – typically called the "Digital Twin".
[23]	2018	The digital twin is a digital representation of a physical asset that can be used to describe the asset's properties, condition, and behavior through modeling, analysis, and simulation.
[24]	2019	A Digital Twin is a virtual instance of a physical system (twin) that is continually updated with the latter's performance, maintenance, and health status data throughout the physical system's life cycle.
[20]	2021	A virtual representation of a physical system (and its associated environment and processes) that is updated through the exchange of information between the physical and virtual systems.
[12]	2022	The Digital Twin is a virtual construct that represents a physical counterpart, integrates several data inputs with the aim of data handling, data storing, and data processing, and provides an automatic, bi-directional data linkage between the virtual world and the physical one. Synchronization is crucial to the Digital Twin to display any changes in the state of the physical object. Additionally, a Digital Twin must comply with data governance rules and must provide interoperability with other systems.
[17]	2023	A DT means a virtual representation of a real-world entity, system, process, or other abstraction, which can be instanced by a computer program or encapsulated software model that interacts and synchronizes with its physical counterpart.
[25]	2023	A digital twin (DT) is a mathematical model with an updating mechanism that generates data which are indistinguishable from its physical counterpart. A digital twin system (DTS) is a structured system that processes data from an experiment (EX) and a digital twin (DT) via analysis methods (M) and decision-making (DM).

Table 1. Digital Twin definitions

1.5.2 Related Concepts

There are numerous concepts within industry that can be seen as synonymous and may even run parallel to that of digital twins. Most notably, the Cyber-Physical System (CPS) has drawn an increasing resemblance to the capabilities of DTs. CPSs are defined by [26] as "physical facilities with embedded sensors, processors, and actuators controlled or monitored by computers" in 2018, and later in 2021 [27] was defined as "an engineering system that seeks to enhance the performance of manufacturing systems, aggregates the information of heterogeneous manufacturing elements and applications, and uses it to predict future conditions in manufacturing plants wherein abnormal scenarios occur". The main differences being the necessity of virtual modelling and simulation in DTs as opposed to CPS being more focused on controlling the operation of physical equipment through raw data acquisition and analysis [28].

Another research field that is highly related and is often seen as overlapping with digital twins is that of Building Information Modelling. Building Information Models (BIMs) provide users and stakeholders a means to visualize and control operations within a given structure throughout its lifecycle via 3D modelling along with concurrent data connections from physical to virtual and vice-versa. While these models loosely follow the DT levels of integration set by Kritzinger in [9], (DM corresponding to BIM level 1, DS with BIM level 2, and DT with BIM level 3 as seen in Figure 5) their main difference lies in the modelled structure itself. While a digital twin of a building's information can be modelled, a BIM of a manufacturing process or specific machine itself is not applicable. In other words, BIMs are limited to the planning, design, construction, and management of built structures rather than the industrial processes within.

BIM level	Description
0	Unmanaged two-dimensional CAD shared via paper/electronic paper
1	Managed two/three-dimensional CAD adhering to BS1192:2007 and within a Common Data Environment that allows collaboration.
2	Level 1 within a three-dimensional virtual environment with attached data. Representations for Architectural, Structural, Facilities, Building Sources and Bridges.
3	Level 2 plus interoperable data.

Figure 5. BIM levels as described by the BIM Industry Working Group [29]

1.5.3 Theoretical Frameworks
5-dimensional DT model

In their comprehensive analysis, Thelen et al. [1] reviewed digital twin applications across 230 scholarly articles published from 2017 to 2022. This paper undertakes an in-depth exploration of contemporary research in the digital twin domain, incorporating theoretical frameworks from notable scholars like Tao [30], who identified five essential components of a digital twin (a real machine, a virtual machine, service components, digital twin data, and the data connections), and Kritzinger, who emphasized the elements of DM, DS, DT, noting that not all scenarios necessitate automatic bidirectional data transfer.

Building upon this extensive literature review, the authors formulated their unique digital twin framework, examining key technologies that facilitate system modeling, the transfer of data between physical and virtual realms, model updating methods, the integration of virtual-physical control (V2P), physical-virtual control (P2V), and decision-making processes. Their proposed formula, DT = F(PS, DS, P2V, V2P, OPT), expresses the interaction of five dimensions: PS (Physical System), DS (Digital System), P2V (Updating Engine), V2P (Prediction Engine), and OPT (Optimization Dimension), with F(.) representing the integration function of these dimensions. This conceptual framework is visually depicted in Figure 6.

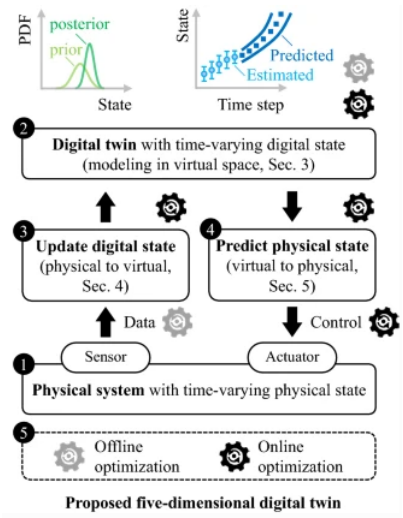


Figure 6. 5 Dimensional Digital Twin Model [1]

DTMaS Framework

In their scholarly review, Liu et al. [19] examined 157 academic publications from 2013 to 2022, concentrating on the application of digital twins in machining processes. Their research led to the development of a framework for constructing a Digital Twin-based Machining System (DTMaS). The primary objective of this survey was not to consolidate various definitions of digital twins but to examine the rationale behind their unique digital twin model. Their classification approach is grounded in three distinct perspectives: Point, Line, and Face, as illustrated in figure 7.

In the point perspective, the "Point" itself refers to the multiple digital twin models (virtual entities) within the virtual space. Since the authors of this study have adopted the definition of digital twins requiring a bidirectional data transfer, they look to analyze the difference between the composition modes of digital twin models in existing papers (i.e., geometric models, behavior models, etc.). The "Line" perspective represents the data connections, encompassing the bidirectional data flow between the physical and virtual realms. This aspect covers data collec-

tion, analysis, and cognition within their DTMaS. Finally, the "Face" perspective pertains to the cyclical operational services provided by the digital twin model. These services span various functions, including design, process monitoring, and intelligent operation, although they are not limited to these areas.

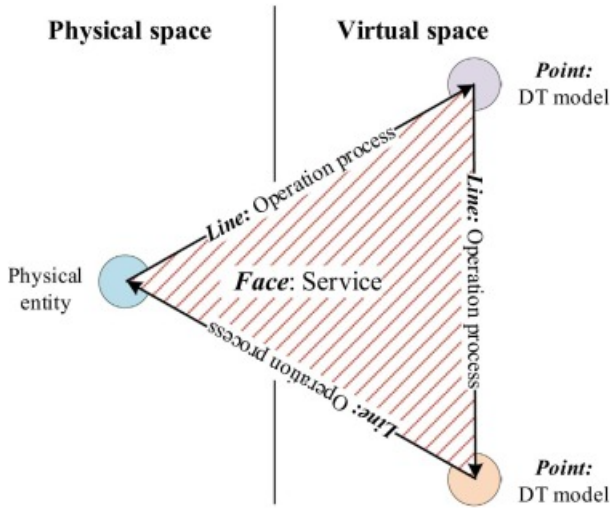


Figure 7. Sketch of DTMaS [19]

IoDT Framework

In their review, Wang et al. [17] analyzed 10 state-of-the-art digital twin survey papers published from 2019 to 2022, along with 131 supplementary articles, to present their perspective on the Internet of Digital Twins (IoDT), as depicted in Figure 8. This IoDT concept is integral to the vision of smart cities, encompassing a network of interconnected digital twin models that exchange vast quantities of IoT data, thereby facilitating data-driven decision-making in daily life. The authors also explored the overarching architecture, communication modes (inter-twin, referring to communication among digital twin models, and intra-twin, denoting data transfer between the physical and virtual realms), fundamental characteristics, enabling technologies, and contemporary prototypes within the IoDT framework.

Their classification of digital twin models is somewhat aligned with Grieves'

conceptualization, necessitating three core elements for a system to be considered an IoDT: [i] physical entities in the real space, [ii] digital twins and their virtual assets in cyberspace, and [iii] an IoDT engine that bridges the cyber and physical worlds through the integration of big data and feedback mechanisms.

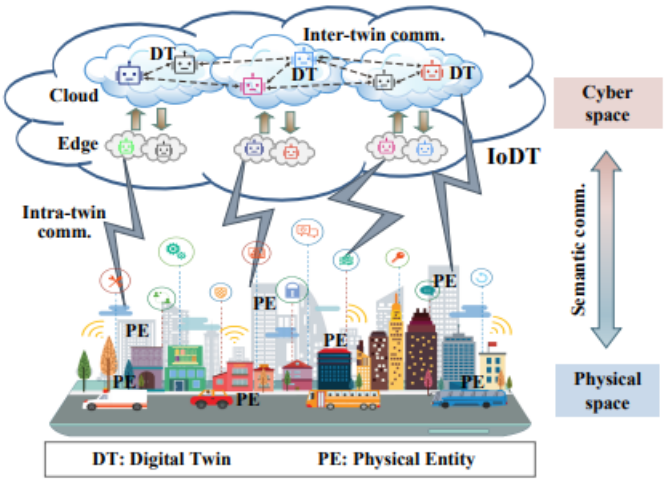


Figure 8. IoDT concept visualization [17]

1.6 Proposed Theoretical Definition

The following section proposes and elaborates on our theoretical definition of how a digital twin should be classified.

1.6.1 Necessity for Definition

The rapid advancement of technology, particularly in the integration of IoT devices and other Industry 4.0 innovations, have caused the focus of digital twin technology to shift from what it once was to what is essentially a SCADA system with visualization capabilities and an embedded predictive engine for automated task allocation. Although this represents a significant stride in the digitalization of systems, it is crucial to recognize that such an advanced configuration may not be a universal requirement for all digital twins across various industries or use-cases.

1.6.2 Proposed General Definition

The definition, in which we support as independent researchers, is as follows:

The digital twin (DT) is a super high-fidelity simulation model which replicates the behavior of a given physical entity, with an emphasis on achieving a high level of similarity in its data output. The inclusion of a bi-directional data transfer is inherently optional and should only be applied to situations which warrant its use (for example in the application of predictive operations, for which these DTs should be referred to as something along the lines of Predictive Twin (PT)[1]).

Rather than focusing on how to implement generalized DT frameworks that could apply to multiple use-cases across various industries, DTs should be defined on a problem specific basis with a standardized validation method. This standardized method would subsequently be used to realize a level of fidelity above a statistically significant threshold, allowing DT models to reliably solve a given question through 'what-if' scenarios.

The classification framework for DTs proposed in this study, illustrated in Figure 9, is based on the fidelity levels of simulation models. A model is considered a DT only when it exceeds a defined validation threshold. Yet, there is a maximum fidelity level; surpassing this level results in the model merging with its physical counterpart, transforming the DT into what is essentially a SCADA system with enhanced visualization and predictive capabilities.

[1]Digital Twins, often linked with inherent predictive capabilities enabled by bidirectional data transfer, have led to the term Predictive Twin (PT) being interchangeably used with Digital Twin (DT) in academic literature. This overlap in terminology accounts for the limited distinct definition of PT in this article.

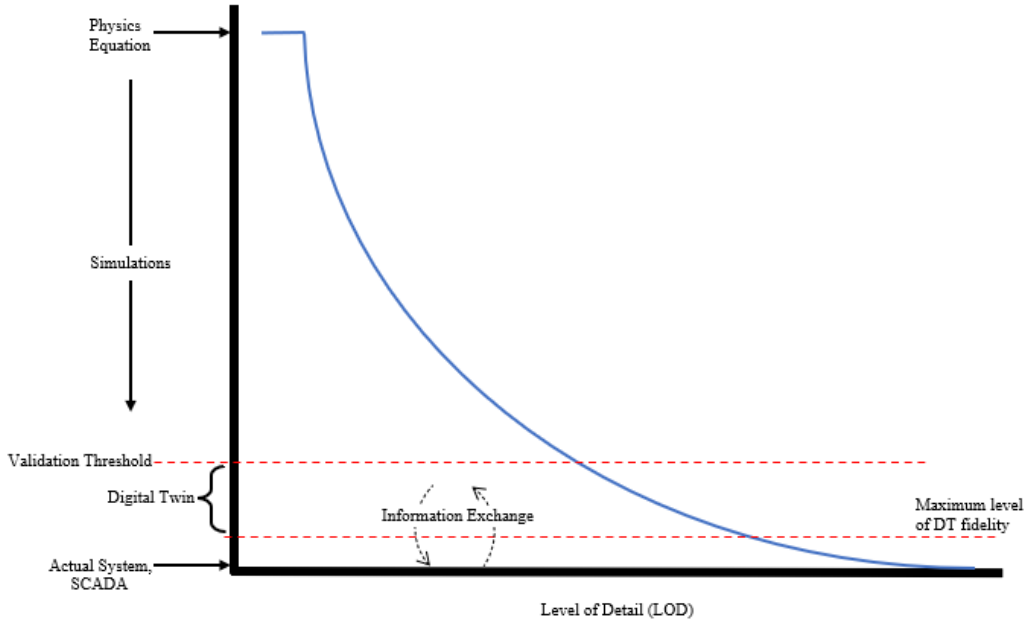


Figure 9. Comparison of asset modelling techniques with respect to level of detail.

1.6.3 Proposed DT for Industry 4.0

In terms of creating DT models specifically related to Industry 4.0 applications, we can generally follow the same principles outlined in the prior section. As illustrated in Figure 10, the proposed Industry 4.0 DT framework is as follows:

- **Step 1:** We start off with determining the physical object or system of interest. The data is then manually transferred (denoted by the dashed arrow) from the physical system to the next step in the process.

- **Step 2:** Entails identifying physics-based and mathematical equations that explain the mechanical behavior of components within the system. This is followed by fitting the real data with statistical distributions in the hopes of elucidating the remaining behavior that might be influenced by external or unmodeled forces.

- **Step 3:** Utilizing this gathered information, we begin the creation of a

base DT model, whose responsibility is the replication of normal system behavior. In this step it is crucial for Industry 4.0 scenarios to model the interconnection of devices, whether it be at the application layer - modelling ROS communication between machines, or the control layer - imitating the behavior of Programmable Logic Controllers (PLCs) and other Internet of Things (IoT) devices.

- **Step 4:** The validation of our base DT model is pivotal to the successful implementation of reliable changes to the physical system. To realize a robust DT model, the integrator must first determine a threshold value for which the model must surpass. If the model is not sufficiently valid in the mirroring of its physical counterpart, the modelling conducted in step 3 should be revised.

- **Step 5:** Once the base DT model sufficiently passes validation testing, the effects of varying operational conditions can be modelled. This could include geometrical changes, such as reorganizing machines in a shop layout, or measuring the affect to process throughput from the inclusion of autonomous machinery in a mainly human-based production line.

- **Step 6:** Lastly, model inference is the process of identifying the best outcome realized from aggregating DT test models. In this step, insights on system performance and recommendations for future action are passed to the integrator, who is inevitably responsible for carrying out these changes in the physical system.

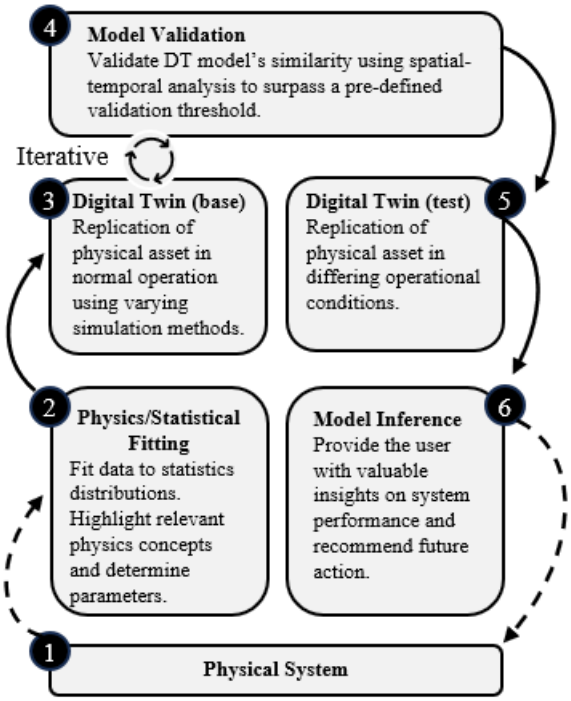


Figure 10. Proposed DT framework for industry 4.0

1.6.4 Components

The essential components of our digital twin definition is as follows: [i] The physical object or system being modelled, [ii] The simulated counterpart to that physical entity (i.e., the DT itself), and [iii] The desired level of fidelity which must be reached for making reliable, model-driven, decisions.

1.6.5 Characteristics

What makes our definition unique to others is the inherent focus on a standardized validation procedure. If researchers can agree upon a universal method of ensuring simulation models match real-world behavior to a certain level of fidelity, the generalizability of DT models themselves becomes irrelevant. Another discrepancy to mainstream DT definitions is the necessity of automatic, bi-directional, data transfer. If a simulation model is continuously fed real-time data from the physical object or system, the model converges with the real system, no longer

making it a simulation but rather an additional means of controlling and visualizing the current state of its physical counterpart.[15]

1.6.6 Scope

With a standardized validation method, any industry can use digital twinning technology for their specific needs as long as they surpass a predetermined, statistically significant, threshold of model fidelity. Our proposed definition provides a general starting point for which an integrator can apply technologies associated with the fourth and fifth industrial revolution. With the focus of researchers shifted away from manipulating DT models to be inter-operable with numerous applications, they can fully focus on how DT technology can be utilized within their specific industrial use cases.

1.7 Discussion

The following subsections detail the strengths and limitations associated with the adoption of our proposed definition along with highlighting the importance of standardization throughout the field of DT research.

1.7.1 Strengths

The merits coinciding with our proposed definition are twofold. A simulation-based digital twin provides users with a means to model complex systems at any layer within the automation pyramid (Figure 11), whether it be at the lowest layer - emulating physical actuator and sensor readings, or at the supervisory layer - modelling the communication infrastructure between devices. Additionally, research on DT technology can be shifted away from its current position of attempting to theorize generalized definitions of DT that can be used in any industry to the position of creating Predictive Twin models that could aide in the development of autonomous systems. Without the burden of attempting to generalize the DT

definition, researchers can continue with their integration of emergent technologies and concentrate more on creating application-specific twins.

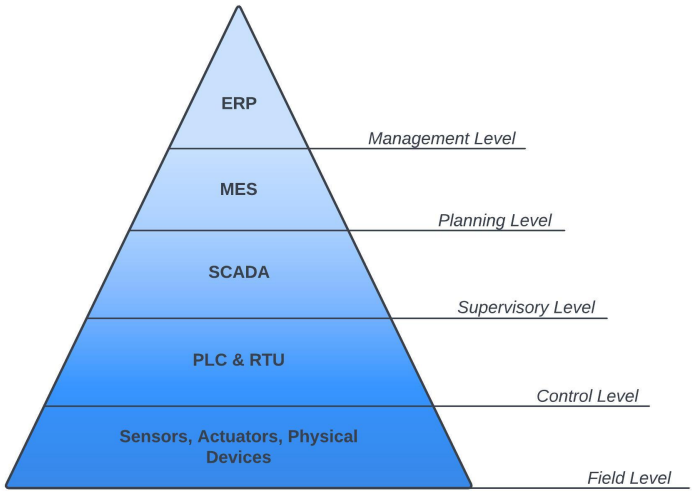


Figure 11. Automation Layer Pyramid [31]

1.7.2 Limitations

The DT concept is in an exciting stage of technological and theoretical development. However, there is still a need for specific steps in the creation of these baseline models for scientific replication within distinct industry applications. Additionally, while our proposed definition creates an opportunity for open interpretation as it applies to multiple industries and use-cases, there may still be confusion in what exactly constitutes as a digital twin when compared to other simulation models. We want to stress that the DT is a simulation model with the main discerning factor of achieving a high level of fidelity with its physical counterpart, of which can only be measured through a standardized validation method in hopes of surpassing a pre-determined threshold, whether it be through statistical formulation or ML/AI aided correspondence testing.

1.7.3 Standardization

The standardization of DT technology architectures is inherently problematic due to the wide range of industrial use-cases. In our definition, we focus less on creating a generalized DT model that could be applicable to various industries, and more on standardizing the validation of variable-fidelity simulation models. In terms of the simulation model itself, this can be comprised of any technique that would be suitable for matching the desired behavior of the physical entity. From physics-based formula integration to statistical distribution mapping and iterative heuristic approaches, the main goal is to develop a simulation model whose data sufficiently matches that of the real world for constructive scenario testing.

1.7.4 Comparison

Reflecting on the five-dimensional DT framework introduced by Thelen[1], we recognize parallels between their model and ours, particularly in the use of physics-based and statistical modeling structures capable of conducting offline optimization. Similarities are evident in the offline handling of positional and process data, transitioning from physical system sensors to the P2V updating engine, and state estimation within the DT model. These aspects resonate with our proposed model's data management and iterative validation abilities. However, our approaches diverge with the author's emphasis on the integration of predictive capabilities affecting control mechanisms in the physical system. Despite this divergence, Thelen's work provides a wealth of insightful information on cutting-edge methods for DT modeling, aiming to establish a versatile DT framework applicable across various industrial settings.

As for the three-dimensional Digital Twin Driven Machining System (DT-MaS) framework proposed by Liu et al.[19], the authors focus more specifically on the machining process within manufacturing environments. This approach aligns

with our view that Digital Twin (DT) definitions should be specific to use cases. Regarding alignment, the author's framework and ours both utilize multiple DT models to enhance physical processes. However, our proposed DT frameworks deviate with respect to data handling operations. The authors advocate for real-time bidirectional data transference between physical and virtual spaces to control machinery during processing and enable predictive operations through ML integration.

The works of Wang et al.[17] on the Internet of Digital Twins (IoDT) framework align with our perspective on defining DTs based on specific objectives. However, the scope of their work leads to significant differences from our approach, as they focus on smart cities and cyber-physical systems modeling. While real-time synchronization is not mandatory within their framework, it emphasizes the need for near real-time or delay-tolerant bidirectional data transfer. The authors also address the cyber security challenges of handling extensive data from physical IoT devices. In contrast, our model reduces this risk by relying on manually fitted data that aligns with statistical distributions and physics equations rather than direct data sourcing from physical entities. Nevertheless, their research provides valuable insights into smart cities' development and future potential.

1.8 Future Research Directions

In the following excerpts, we will bring to light potential research directions that should be considered with regards to the current gaps in literature and technological advancements of the 21st century.

1.8.1 Literature Gaps

While many publications extensively address potential architectures of DT models, very few of these works fully grapple with the development of a standardized validation technique for their models. In part, this is due to current DT definitions incorporating a bi-directional data transfer from physical-virtual space, evading the need for validation due to the data within their virtual model automatically matching that of the real system.

1.8.2 Technological Trends

With advancements in technology rapidly being introduced, for instance the introduction of various IoT devices, the necessity for DT concepts to evolve with these trends is paramount. The age of autonomous systems integration is approaching, and DT researchers are already working at incorporating data-driven decision making into various, highly complex, systems. However, working with such large amounts of potentially sensitive data could expose these systems to various security risks. For this reason, attention must be paid to securing these models, especially if they are linked in real-time to physical assets. If such a system is not thoroughly secured, malicious actors could potentially gain access to these systems and either leak or control sensitive data being passed from virtual to physical layers.

1.8.3 Research Recommendations

To summarize, we recommend that research be conducted into both the development of a standardized validation method for DT models along with the overall cyber security protocols which should be utilized for Predictive Twin models.

1.9 Conclusion

In our detailed exploration of Digital Twin (DT) technology, we have conducted an extensive analysis of a wide range of literature on DTs, leading to the proposal of a nuanced definition that aims to advance academic research beyond its current state. This revised definition challenges the notion that automatic bidirectional data transfer between physical and virtual entities is essential to DTs. We argue that a simulation model should not be an exact replica of its physical counterpart, as such a convergence results in a system resembling a SCADA system with predictive capabilities, diverging from the original concept of DTs as initially envisioned. Our study underscores the critical need of a standardized validation method for DT models, addressing a notable gap in existing literature. We emphasize the importance of DT models accurately mirroring real-world behavior to ensure their reliability and trustworthiness. Furthermore, we highlight the technological trends and advancements, such as the integration of various IoT devices, which necessitate the evolution of DT concepts. We also stress the importance of securing these models, particularly in scenarios involving large, sensitive data sets and real-time connections to physical assets, to mitigate potential security risks.

In summary, our work advocates for a more goal-oriented research approach, particularly in application-specific contexts. We conclude with a call to action for future research directions, emphasizing the need for a comprehensive understanding and application of DT technology in various industrial realms to necessitate the growth of this exciting field.

List of References

[1] A. Thelen, X. Zhang, O. Fink, Y. Lu, S. Ghosh, B. D. Youn, M. D. Todd, S. Mahadevan, C. Hu, and Z. Hu, "A comprehensive review of digital twin—part 1: modeling and twinning enabling technologies," *Structural and Multidisciplinary Optimization*, vol. 65, no. 12, p. 354, 2022.

[2] L. Hernández and S. Hernández, "Application of digital 3d models on urban planning and highway design." *WIT Transactions on The Built Environment*, vol. 33, 1997.

[3] M. Grieves, "Completing the cycle: Using plm information in the sales and service functions [slides]," in *SME Management Forum*, 2002.

[4] E. Glaessgen and D. Stargel, "The digital twin paradigm for future nasa and us air force vehicles," in *53rd AIAA/ASME/ASCE/AHS/ASC structures, structural dynamics and materials conference 20th AIAA/ASME/AHS adaptive structures conference 14th AIAA*, 2012, p. 1818.

[5] M. Grieves, "Digital twin: manufacturing excellence through virtual factory replication," *White paper*, vol. 1, no. 2014, pp. 1–7, 2014.

[6] M. Grieves and J. Vickers, "Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems," *Transdisciplinary perspectives on complex systems: New findings and approaches*, pp. 85–113, 2017.

[7] M. W. Grieves, "Digital twins: Past, present, and future," in *The Digital Twin*. Springer, 2023, pp. 97–121.

[8] M. Grieves, "Intelligent digital twins and the development an d management of complex systems [version 1; peer review," 2022.

[9] W. Kritzinger, M. Karner, G. Traar, J. Henjes, and W. Sihn, "Digital twin in manufacturing: A categorical literature review and classification," *IfacPapersOnline*, vol. 51, no. 11, pp. 1016–1022, 2018.

[10] F. Abdoune, L. Ragazzini, M. Nouiri, E. Negri, and O. Cardin, "Toward digital twin for sustainable manufacturing: A data-driven approach for energy consumption behavior model generation," *Computers in Industry*, vol. 150, p. 103949, 2023.

[11] A. Fuller, Z. Fan, C. Day, and C. Barlow, "Digital twin: Enabling technologies, challenges and open research," *IEEE access*, vol. 8, pp. 108 952–108 971, 2020.

[12] H. van der Valk, H. Haße, F. Möller, and B. Otto, "Archetypes of digital twins," *Business & Information Systems Engineering*, pp. 1–17, 2021.

[13] D. M. Botín-Sanabria, A.-S. Mihaita, R. E. Peimbert-García, M. A. Ramírez-Moreno, R. A. Ramírez-Mendoza, and J. d. J. Lozoya-Santos, "Digital twin technology challenges and applications: A comprehensive review," *Remote Sensing*, vol. 14, no. 6, p. 1335, 2022.

[14] C. Alcaraz and J. Lopez, "Digital twin: A comprehensive survey of security threats," *IEEE Communications Surveys & Tutorials*, 2022.

[15] M. Batty, "Digital twins," *Environment and Planning B: Urban Analytics and City Science*, vol. 45, no. 5, pp. 817–820, 2018. [Online]. Available: https://doi.org/10.1177/2399808318796416

[16] I. Automation, "The new scada," Jun 2014. [Online]. Available: https://inductiveautomation.com/resources/article/old-scada-vs-the-new-scada

[17] Y. Wang, Z. Su, S. Guo, M. Dai, T. H. Luan, and Y. Liu, "A survey on digital twins: architecture, enabling technologies, security and privacy, and future prospects," *IEEE Internet of Things Journal*, 2023.

[18] D. Helbing and J. A. Sánchez-Vaquerizo, "Digital twins: Potentials, ethical issues, and limitations," *arXiv preprint arXiv:2208.04289*, 2022.

[19] S. Liu, J. Bao, and P. Zheng, "A review of digital twin-driven machining: From digitization to intellectualization," *Journal of Manufacturing Systems*, vol. 67, pp. 361–378, 2023.

[20] E. VanDerHorn and S. Mahadevan, "Digital twin: Generalization, characterization and implementation," *Decision support systems*, vol. 145, p. 113524, 2021.

[21] M. Liu, S. Fang, H. Dong, and C. Xu, "Review of digital twin about concepts, technologies, and industrial applications," *Journal of Manufacturing Systems*, vol. 58, pp. 346–361, 2021.

[22] R. Rosen, G. Von Wichert, G. Lo, and K. D. Bettenhausen, "About the importance of autonomy and digital twins for the future of manufacturing," *Ifac-papersonline*, vol. 48, no. 3, pp. 567–572, 2015.

[23] M. Helu, A. Joseph, and T. Hedberg Jr, "A standards-based approach for linking as-planned to as-fabricated product data," *CIRP Annals*, vol. 67, no. 1, pp. 487–490, 2018.

[24] A. M. Madni, C. C. Madni, and S. D. Lucero, "Leveraging digital twin technology in model-based systems engineering," *Systems*, vol. 7, no. 1, p. 7, 2019.

[25] F. Emmert-Streib, "Defining a digital twin: A data science-based unification," *Machine Learning and Knowledge Extraction*, vol. 5, no. 3, pp. 1036–1054, 2023.

[26] L. D. Xu and L. Duan, "Big data for cyber physical systems in industry 4.0: a survey," *Enterprise Information Systems*, vol. 13, no. 2, pp. 148–169, 2019.

[27] Y. H. Son, K. T. Park, D. Lee, S. W. Jeon, and S. Do Noh, "Digital twin–based cyber-physical system for automotive body production lines," *The International Journal of Advanced Manufacturing Technology*, vol. 115, pp. 291–310, 2021.

[28] X. Liu, D. Jiang, B. Tao, F. Xiang, G. Jiang, Y. Sun, J. Kong, and G. Li, "A systematic review of digital twin about physical entities, virtual models, twin data, and applications," *Advanced Engineering Informatics*, vol. 55, p. 101876, 2023.

[29] D. Jones, C. Snider, A. Nassehi, J. Yon, and B. Hicks, "Characterising the digital twin: A systematic literature review," *CIRP Journal of Manufacturing Science and Technology*, vol. 29, pp. 36–52, 2020.

[30] F. Tao, H. Zhang, A. Liu, and A. Y. Nee, "Digital twin in industry: State-of-the-art," *IEEE Transactions on industrial informatics*, vol. 15, no. 4, pp. 2405–2415, 2018.

[31] B. Babayigit and M. Abubaker, "Industrial internet of things: A review of improvements over traditional scada systems for industrial automation," *IEEE Systems Journal*, 2023.

MANUSCRIPT 2

SCADA Systems

Zachary A. deWardener[1*], and Manbir S. Sodhi[1]

[1]Department of Industrial Engineering, University of Rhode Island, Kingston, Rhode Island, 02881

[*]Corresponding author: zdewardener@uri.edu

Drafted for submission, XXX–XXX.

ACM ISBN XXX-X-XXXX-XXXX-X/XX/XX

https://doi.org/XX.XXXX/XXXXXXX.XXXXXXX

2.1 Abstract

This article examines Supervisory Control and Data Acquisition (SCADA) systems, emphasizing their evolving architecture, applications across industries, and the integration of modern communication protocols. With the introduction of Industrial Internet of Things (IIoT) technologies, traditional SCADA systems have undergone transformative changes to meet the demands of modern industrial automation. Our study looks into the architectural nuances of SCADA systems, including point-to-point, mesh, and hybrid topologies, each pertinent to specific operational scopes from localized facilities to expansive, interconnected networks.

The versatility of SCADA systems is showcased through their wide-ranging applications in critical infrastructure, manufacturing, and utility management, highlighting their role in real-time monitoring, control, and predictive maintenance. We also address the security vulnerabilities inherent in SCADA systems, focusing on the challenges posed by the integration of cloud computing and IoT technologies. Through an analysis of current literature and case studies, the article identifies the potential for increased efficiency and resilience in SCADA systems while recognizing the need for robust security measures to mitigate emerging cyber threats.

Our findings suggest that while SCADA systems are pivotal in enhancing operational efficiency and reliability, the convergence with IIoT introduces complexities requiring advanced security solutions. We conclude with a call for future research to fortify SCADA systems against sophisticated cyber threats, ensuring their safe and sustainable integration into the fabric of industrial automation.

2.2 Introduction

2.2.1 Background of SCADA Systems

Supervisory Control and Data Acquisition (SCADA) systems are essential in today's industrial world. They act as the central hub for gathering, analyzing, and displaying data from automated or semi-automated processes. Additionally, they allow for the centralized control of machinery and equipment locally and remotely. Supervisory control focuses on overseeing Programmable Logic Controllers (PLCs), tracking system performance, and fine-tuning for optimal results, like improving throughput or takt time. On the other hand, data acquisition refers to the collection of real-time data and updates from various parts of a process. SCADA began in the 1960s as a basic computer monitoring system but has since transformed into an advanced, networked solution for real-time data collection and control. Nowadays, industries ranging from manufacturing and utilities to transportation and water treatment rely on SCADA systems.

2.2.2 Importance in Modern Industry

The advent of Industry 4.0 and the Industrial Internet of Things (IIoT) has accentuated the role of SCADA systems as a critical component in today's industrial ecosystem. With the increasing complexity of industrial operations and the growing need for real-time decision-making, SCADA systems offer a robust platform for digitally integrating various industrial processes. These systems facilitate the seamless interaction between human operators and machines, enhancing operational efficiency, safety, and reliability. Moreover, SCADA systems have proven invaluable in optimizing resource allocation, reducing downtimes, and improving the overall productivity of industrial setups.

2.2.3 Purpose and Structure of the Chapter

The primary objective of this chapter is to provide a comprehensive overview of SCADA systems, tracing their historical evolution, discussing their core components, and exploring their functionalities. The chapter will explore the various architectures that SCADA systems can adopt and examine their applications across different industries. Additionally, the chapter will discuss the current trends and future prospects of SCADA systems, including their integration with emerging state-of-the-art technologies like the Internet of Things (IoT) and cloud computing.

The chapter is structured to offer readers a logical flow of topics, beginning with foundational concepts and gradually moving toward more complex issues and future trends. Case studies will be presented to offer real-world insights into the implementation and impact of SCADA systems in specific industrial settings.

In summary, this chapter aims to serve as a comprehensive resource for industry professionals and academic researchers interested in the current landscape and future trajectory of SCADA systems in the industrial sector. By providing this multifaceted exploration into the concept of SCADA systems, the chapter seeks to clarify topics covered within subsequent chapters and contribute to the existing body of knowledge on process automation.

2.3 Historical Evolution of SCADA

2.3.1 Early control systems

In the era preceding the implementation of Supervisory Control and Data Acquisition (SCADA) systems, industrial operations were predominantly overseen by human operators who used basic mechanical apparatuses for plant floor management. As industrial enterprises started to extend their operations across various geographical locations, there was a gradual transition from these manual systems to the incorporation of relays and timers. This shift aimed to facilitate a mod-

icum of automated control, particularly in areas where immediate human oversight was impractical. However, these nascent control mechanisms, which were largely reliant on relay logic, presented several limitations. Notably, they necessitated considerable human interaction, lacked the capacity for real-time data collection and analysis, and were often restricted by geographical constraints.

2.3.2 Transition to computer-based systems

The 1960s marked a crucial turning point in the domain of industrial control systems as the concept of telemetry made its way into the industrial sector. With the proliferation of computers and advancements in electronic technology, industries began to see the potential of automating their monitoring and control processes. The first generation of SCADA systems emerged during this period, characterized by mainframe computers and proprietary software. Nonetheless, these systems, while revolutionary for their time, were still centralized and lacked the flexibility and scalability of modern SCADA solutions.

The subsequent decades witnessed rapid technological advancements. The late 1960s, leading through the 70s and 80s, saw the implementation of solid-state devices such as microprocessors and Programmable Logic Controllers (PLCs), which played a pivotal role in decentralizing SCADA operations. These systems were more resilient, could handle multiple tasks simultaneously, and offered better integration capabilities with other industrial systems.

Due to industry demand, technology vendors began building and selling SCADA applications as turn-key solutions for process control. As with all technological solutions, the demand for intelligent and efficient systems drove the evolution of SCADA architectures through three major "generations", the Monolithic SCADA, Distributed SCADA, and Networked SCADA [1].

Monolithic SCADA

As the first adaptation of this technology, Monolithic SCADA functioned as standalone mainframe-based systems that interchanged data from Remote Terminal Units (RTUs) to a central master computer over a Wide Area Network (WAN) architecture. The main downfall in this era of SCADA technology was the proprietary nature of the devices utilized. The exclusivity of the technology at the time created an issue where control systems were not able to interconnect RTUs with master computer systems from differing vendors. However, this technological shortcoming inspired the development of the second generation of SCADA system, the Distributed SCADA [1].

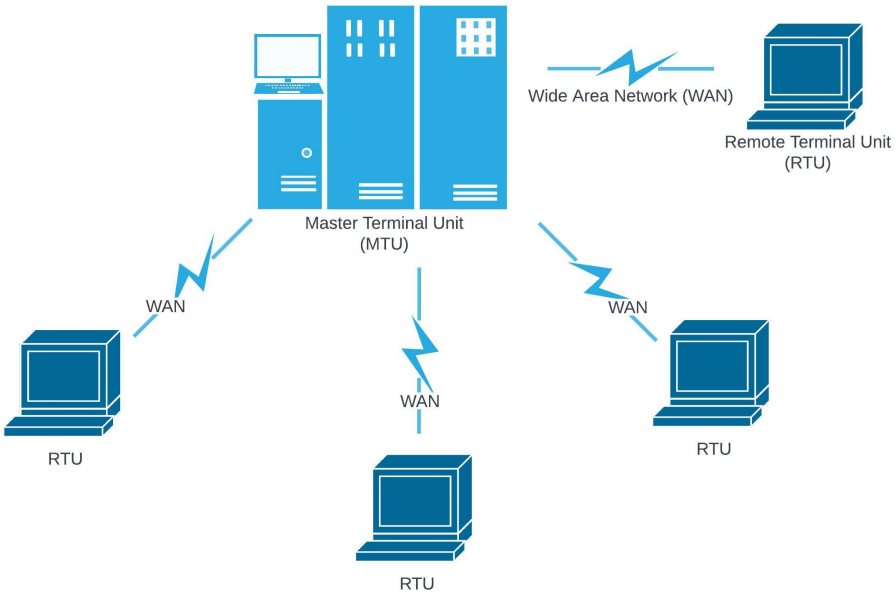


Figure 12. Monolithic SCADA system architecture

Distributed SCADA

The improvements made in the time of distributed SCADA systems involve decreasing the overall system footprint and implementing a Local Area Network (LAN) architecture, allowing the connection to multiple "operating stations".

These stations acted as lightweight, inexpensive, edge computers that allowed operators to interchange real-time data for the manipulation and observation of equipment in use. However, proprietary software, hardware, and peripheral devices were still prevalent in this rendition of SCADA technology. This issue prevailed until the third generation of the SCADA system was introduced to the market, the Networked SCADA [1].

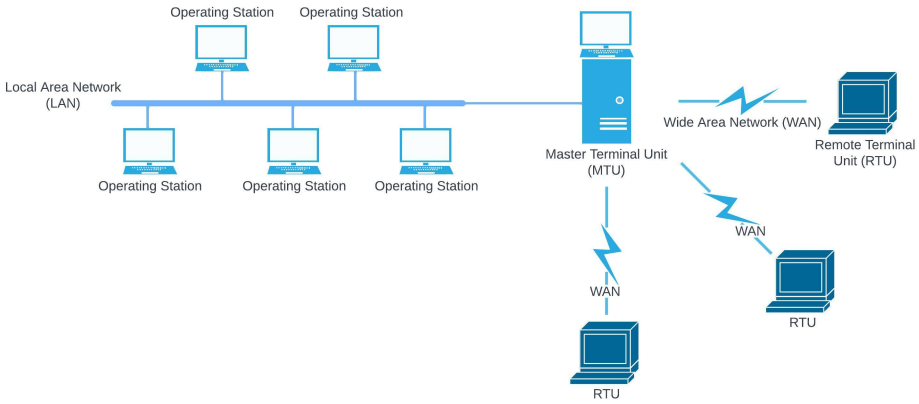


Figure 13. Distributed SCADA system architecture

Networked SCADA

Driven by customer demand for highly automated processes and an influx of industrial equipment vendors, Networked SCADA systems were established in the '90s. This rendition of SCADA provided the same basic functionality as the prior generation with the addition of an open system architecture in lieu of the hardware and software elements holding their proprietary nature. Data communications were transferred over WAN protocols such as Internet Protocol (IP) and ethernet connection rather than being restricted to a LAN. In the same decade, integrators, such as the OPC Foundation and OLE for Process Control, started developing and upholding standards for SCADA implementation across the industrial sector to ease the deployment of effective supervisory and control solutions for process

automation [1].

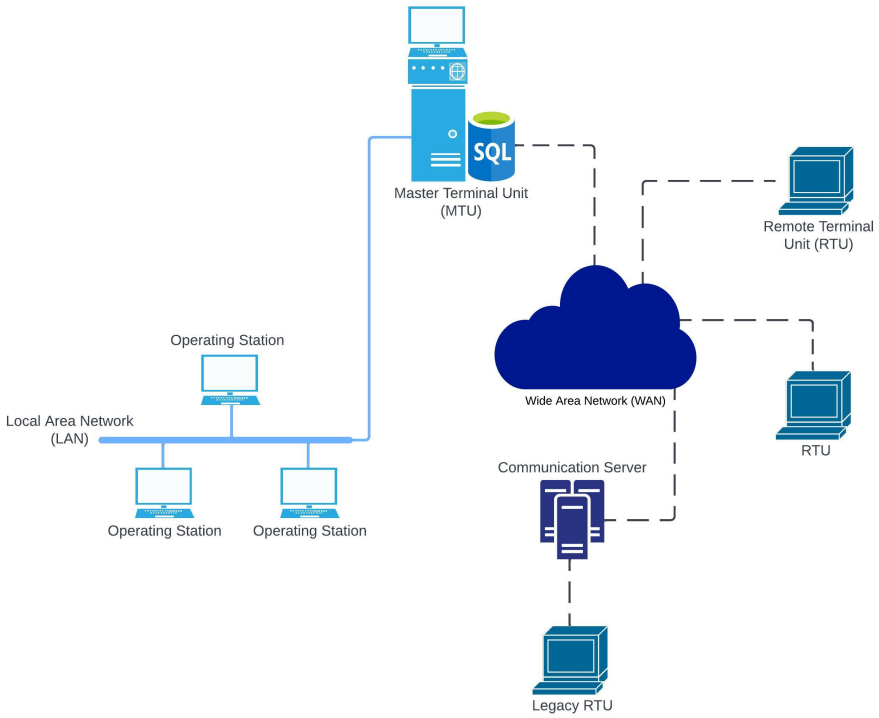


Figure 14. Networked SCADA system architecture

2.3.3 Modern SCADA systems (IoT SCADA)

With the rapid progression of Information Technology (IT) and web-based applications in the 21st century, SCADA systems have been elevated from a localized system to a widely distributable asset for businesses to automate their operations from geographically remote locations. Modern SCADA systems are characterized by their ability to integrate with many devices, enhanced data analytics capabilities, and resilience against system failures. Furthermore, the integration of SCADA with other emerging technologies, such as the Internet of Things (IoT), artificial intelligence (AI), cloud computing, and database management systems such as SQL, has further expanded its capabilities to allow for the application of lightweight production planning toolsets. Today's SCADA systems are not just about mon-

itoring and control; they are about predictive maintenance, real-time analytics, and optimizing operations using data-driven insights [2]. Figure 15 depicts the layers at which automated systems are formatted. The lowest level represents integrated field devices such as sensors, actuators, and physical devices that generate raw data and handle the tangible operations within a process. The second level encompasses the Programmable Logic Controllers (PLCs) and Remote Terminal Units (RTUs) within the system (both of which are covered later in this paper). These elements are responsible for controlling and operating the field devices based on the input data and predetermined logic. Our SCADA systems reside at the supervisory level, providing remote access to the system's components along with the collection, analysis, and visualization of data sourced from the control level. The planning stage employs a digital platform termed the manufacturing execution system (MES), which oversees the entire production cycle, from the utilization of raw materials to the punctual delivery of products. Positioned at the apex of the automation hierarchy is the management tier, utilizing a software-embedded management framework called the enterprise resource planning (ERP) system, orchestrating and harmonizing all business processes within the factory. [3]

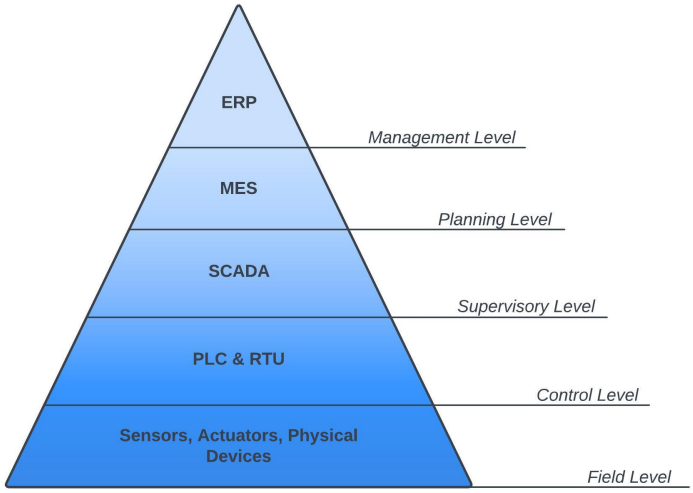


Figure 15. Automation Layer Pyramid [3]

2.4 Fundamental Components of SCADA

The efficiency and versatility of Supervisory Control and Data Acquisition (SCADA) systems can be attributed to their composite structure, which consists of several integral components. Each component plays a distinct role, ensuring that the system can monitor, gather, and analyze data and then act upon that data in real-time. This section delves into the core components of a SCADA system, elucidating their functions and significance in the broader framework. Figure 16 Provides a general overview of the fundamental components of SCADA systems.

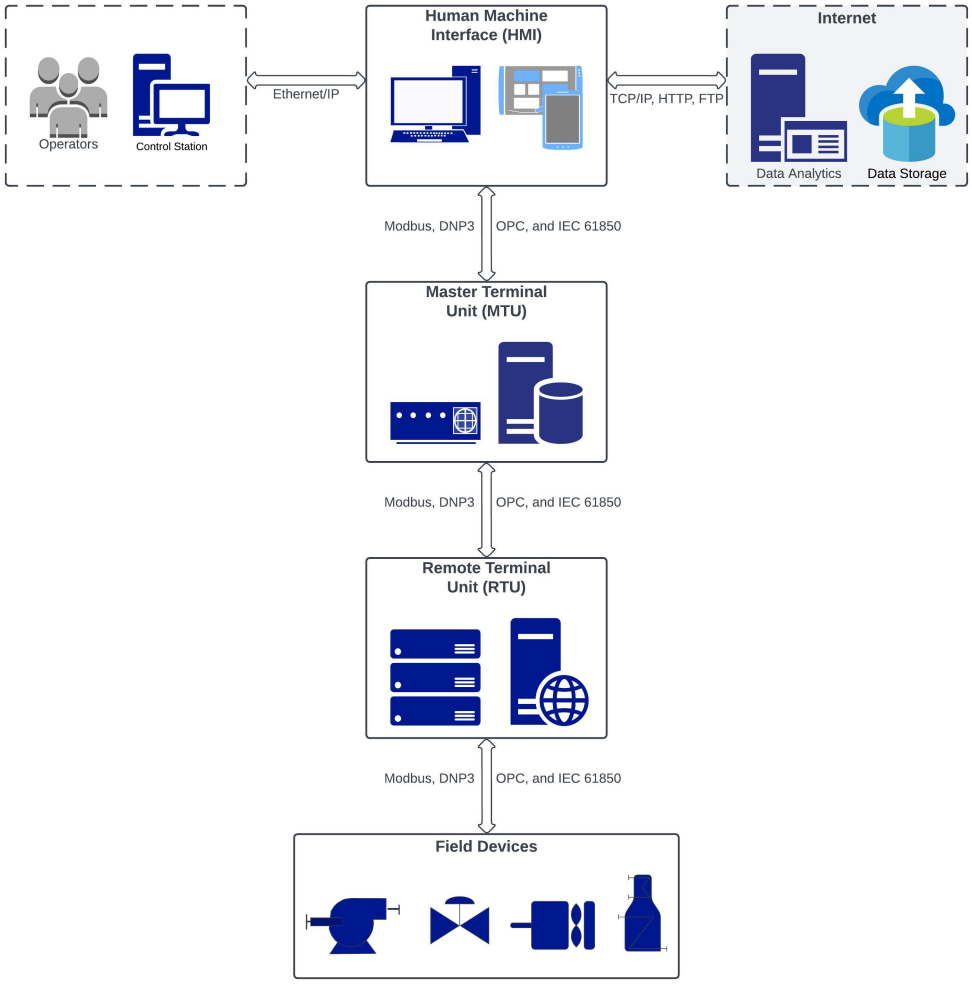


Figure 16. General SCADA architecture [4]

2.4.1 Human Machine Interface (HMI)

The main component of a SCADA system is the Human-Machine Interface (HMI), a graphical representation that provides operators with a visual overview of the system's status and performance. The HMI serves as the primary interaction point between human operators and the equipment or machines in the overall system, allowing for real-time monitoring, control adjustments, and troubleshooting. It displays data intuitively, often using diagrams, charts, and alarms, ensuring that operators can quickly identify and respond to any anomalous behavior.

2.4.2 Programmable Logic Controllers (PLCs) and Remote Terminal Units (RTUs)

PLCs and RTUs are the backbone of a SCADA system. These devices are responsible for interfacing with the Master Terminal Unit (MTU) [4] and field devices, such as sensors and actuators, to collect data and execute control actions. While both PLCs and RTUs perform similar functions, they differ in their complexity and application:

- **Programmable Logic Controllers (PLCs):** PLCs, while similar to RTUs, are capable of performing more sophisticated control functions. They are considered robust, flexible devices that can execute simple and complex logic operations both locally and remotely. They are typically used in environments that require high-speed control and processing.

- **Remote Terminal Units (RTUs):** RTUs, on the other hand, are more streamlined devices designed for remote operations via a telemetry system. They are often deployed in faraway locations and are optimized for data acquisition and transmission over long distances. These devices can also contain basic microcontrollers, providing the ability to handle simple boolean instructions. [1]

2.4.3 Communication Infrastructure

Selecting an appropriate communication infrastructure is an integral part of the SCADA system, ensuring seamless data transmission between the MTU and its distributed components. This infrastructure can encompass a variety of communication mediums, including wired networks (i.e., ethernet), wireless technologies, and even satellite communications. The choice of communication mechanism is often dictated by the system's geographical spread, the volume of data to be transmitted, and the required data transmission speed. Some examples of commonly

used communication protocols include Modbus, Profibus, DNP3, and MQTT.

Modbus

Initially devised by Gould-Modicon and currently owned by Schneider Electric, Modbus is an open-source protocol that's commonly used to exchange data between MTUs and RTUs within a SCADA system. Modbus supports both serial and TCP/IP communication (controller-responder, formerly known as "master-slave" and client-server architectures, respectively) and supports integrity checks within its architecture but lacks encryption and other security measures that may be needed for certain use cases. [5]

Profibus

Follows a similar controller-responder architecture as Modbus with a slightly different method of deterring cyber threats. The responders are initiated in a safe state and must be connected to the controller in a specific sequence based on an internal timer. If the controller does not talk to a certain responder in an allotted time frame, the system is forced to renew the start-up sequence, adding an additional layer of security. [5]

DNP3

Is an open-source protocol, allowing secure and flexible communication between various devices. Due to its error checking, multiplexing, and data fragmentation capabilities, this method is commonly used in high-risk industries such as oil and gas, power and water distribution, and transportation. However, this protocol has also been criticized for not providing sufficient authentication. [5]

MQTT

Otherwise known as Message Queuing Telemetry Transport, is a lightweight publish-subscribe protocol commonly used for networks which may perform with

a certain level of latency or one which contains limited bandwidth resources. The main security concerns revolving MQTT architectures are that of improperly configured clients, but the capabilities for authentication, encryption, and access control are available for implementation. [6][7]

2.4.4 Supervisory system

The core component of the SCADA structure is the management or supervisory system. It gathers information from different PLCs and RTUs, analyzes this information, and then displays it on the HMI. Additionally, it oversees advanced control procedures, handles alerts, and archives past data for subsequent evaluation. This management system frequently collaborates with other organizational platforms, like Enterprise Resource Planning (ERP) or Manufacturing Execution Systems (MES), to offer a comprehensive perspective of the whole process. [2]

2.5 Key Functionalities of SCADA

Supervisory Control and Data Acquisition (SCADA) systems, while diverse in their applications across various industries, share a core set of functionalities that underpin their utility. These functionalities not only define the operational capabilities of SCADA systems but also highlight their transformative impact on modern industrial processes. This section delves into these key functionalities, elucidating their significance and the benefits they confer on industries.

2.5.1 Data acquisition

Data acquisition is the foundation upon which the SCADA systems functionalities are built. It provides real-time insights into system performance, enabling operators to monitor conditions, detect anomalies, and make informed decisions. Process data is acquired from PLCs or RTUs using built-in sensors and predetermined logic to collect the current state of the system. That data is then transmitted

to the system's MTU and operating stations for storing and visualizing the incoming data, respectively, allowing operators to easily understand the system's status and make data-driven decisions. [5]

2.5.2 Real-time control

Real-time control ensures that industrial processes run smoothly, efficiently, and safely. It allows for immediate intervention in case of anomalous behavior, minimizing downtimes and potential damages. From simple tasks, such as turning a device on or off, to complex operations like adjusting process parameters based on real-time data, real-time control is a crucial aspect of SCADA systems.

2.5.3 Alarm management

Alarm management enhances the safety and reliability of industrial operations. It ensures that potential issues are promptly identified and addressed, preventing minor anomalies from escalating into major incidents. SCADA systems are adept at monitoring thresholds and triggering alarms when predefined conditions are breached. These alarms can be visual (using differing colors to represent the severity of the alarm), auditory, or even transmitted as SMS notifications to designated personnel when anomalous events occur within the system. [8]

2.5.4 Data storage and analysis

Due to the advent of IoT technology, sharing vast amounts of process data from sensors and actuators embedded within the system's components, modern SCADA systems are equipped with robust data storage capabilities, archiving historical real-time data for future reference. A common data storage method used in industry is Structured Query Language (SQL) databases, used to store and extract time stamped data [5]. Coupled with advanced analytics tools, SCADA can transform this data into actionable insights, identifying trends, predicting failures,

and optimizing operations. The ability to store and analyze data extends the utility of SCADA beyond real-time monitoring and control. It paves the way for predictive maintenance, process optimization, and continuous improvement in industrial operations.

2.6 SCADA System Architecture

The architecture of a Supervisory Control and Data Acquisition (SCADA) system plays a pivotal role in determining its efficiency, scalability, and resilience. As industries have evolved and technological advancements have burgeoned, SCADA architectures have undergone significant transformations to cater to diverse operational needs. This section provides an in-depth exploration of the various architectures that SCADA systems can adopt, elucidating their characteristics, advantages, and potential applications.

2.6.1 Point-to-point

Ideal for small-scale operations where only a few devices need to communicate, point-to-point architecture represents the most basic form of SCADA design. In this topology, individual devices or components communicate directly with each other without any intermediary, as seen in Figure 17. For example, think of the link between a computer (being the controller or MTU) and a printer (being the responder or PLC) via a wired connection [8]. Due to its simplicity, point-to-point is often found in localized systems, such as a single manufacturing line or a small water treatment facility.

Figure 17. Point-to-Point Topology

2.6.2 Star

Star architectures, commonly used in medium-scale operations where centralized control is desired, such as in building management systems or localized energy grids, entails all devices or nodes being connected to a central hub or controller (Figure 18). This central hub is responsible for managing communications, processing data, and relaying information. While offering a centralized control mechanism makes it easier to manage and monitor traffic, it also provides a single point of failure, which can simplify troubleshooting and adversely, can be seen as a target in cyber attacks, leading to the failure of the entire network. [9]

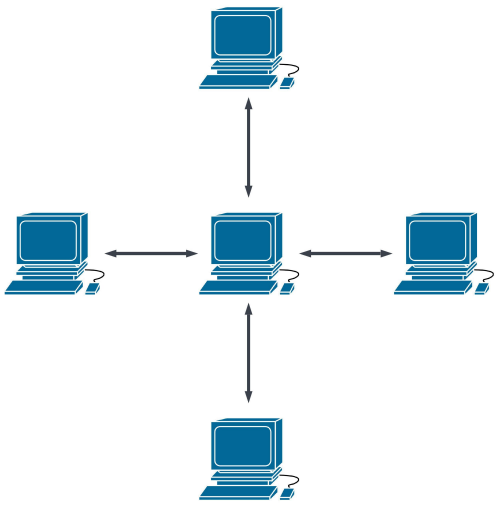


Figure 18. Star Topology

2.6.3 Mesh

Mesh architectures are characterized by multiple interconnections between devices or nodes. Each device is connected to several others, allowing for multiple communication pathways (Figure 19). The primary advantage of a mesh architecture is its inherent redundancy. If one communication pathway fails, data can still be transmitted through alternative routes, ensuring uninterrupted operation. Ideal for large-scale, geographically dispersed operations where resilience and scalability are paramount, mesh topologies are commonly utilized in complex systems such as national energy grids or vast transportation networks. [9]

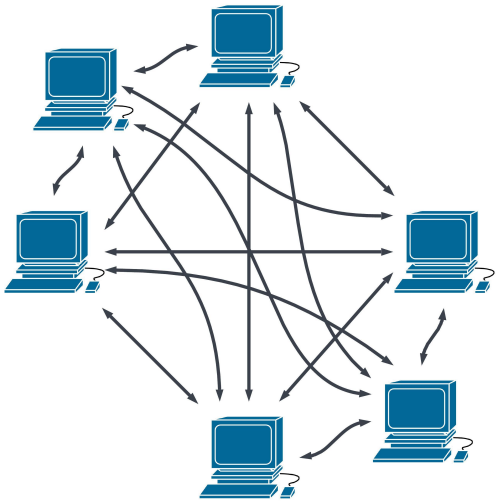


Figure 19. Mesh Topology

2.6.4 Hybrid

As the name suggests, hybrid architectures combine elements from the aforementioned architectures to create a tailored solution that meets specific operational needs. Hybrid architectures (as seen in Figure 20) offer flexibility, allowing industries to leverage the strengths of multiple architectures while mitigating their weaknesses. Hybrid architectures are often found in complex operations that have

diverse requirements, such as smart cities or multi-facility manufacturing conglomerates.

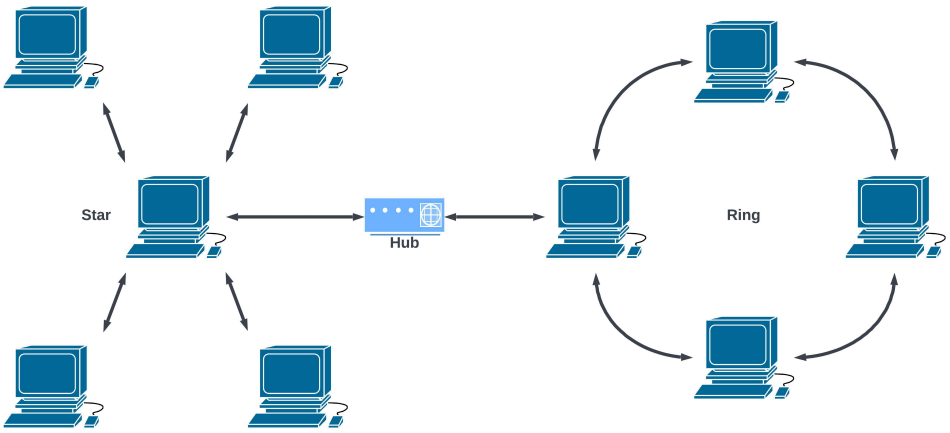


Figure 20. Hybrid Topology (Star and Ring)

2.7 Applications of SCADA in Various Industries

The versatility and adaptability of Supervisory Control and Data Acquisition (SCADA) systems have rendered them invaluable across a plethora of industries. From ensuring the uninterrupted flow of electricity to optimizing intricate manufacturing processes, SCADA systems have revolutionized the way industries operate. This section provides a comprehensive exploration of the diverse applications of SCADA across various sectors, elucidating the challenges they address and the benefits they confer.

2.7.1 Energy and utilities

SCADA systems play a pivotal role in the energy sector, particularly in power generation, transmission, and distribution. They monitor and control substations, transformers, and circuit breakers, ensuring the efficient and reliable delivery of electricity. With the increasing demand for energy and the challenges posed by renewable integration, SCADA systems provide real-time data and control capa-

bilities, optimizing grid performance, enhancing fault detection, and facilitating predictive maintenance. One example of SCADA system implementation in the energy sector is National Grid's deployment of Swiss automation company ABB's SCADA technology within their US East Coast headquarters. Their system allows operators to visualize the network status across multiple states efficiently, increasing grid reliability and security against potential cyber threats.

Figure 21. National grid SCADA system [10]

2.7.2 Manufacturing

In the context of the manufacturing sector, SCADA systems play a pivotal role in the digital transformation of traditional production environments. These systems, when integrated with advanced analytics and machine learning algorithms, can predict potential equipment failures, optimize maintenance schedules, and enhance overall operational efficiency. By analyzing real-time data from sensors and Industrial IoT equipment, SCADA systems can provide actionable insights that lead to proactive decision-making [4]. This not only minimizes unplanned downtimes but also optimizes resource allocation and energy consumption. Furthermore, with the integration of SCADA with enterprise resource planning (ERP) and manufacturing execution systems (MES), manufacturers can achieve a seamless flow of information across all levels of the organization. This interconnectedness ensures that the manufacturing processes are agile, responsive, and aligned with market demands, solidifying the role of SCADA as an indispensable tool in the modern manufacturing paradigm.

Moreover, the versatility of SCADA systems in the manufacturing sector extends beyond mere equipment monitoring and process control. With the increasing emphasis on sustainability and green manufacturing, SCADA systems are being leveraged to monitor and manage energy consumption, emissions, and waste production. By providing granular data on energy usage patterns and waste generation, these systems enable manufacturers to identify inefficiencies and implement sustainable practices, thereby contributing to both environmental stewardship and cost savings. Additionally, the fusion of SCADA with augmented reality (AR) [11] and virtual reality (VR) has ushered in an innovative phase of instruction and problem-solving. Service professionals can now interact with digital models of equipment and infrastructures, facilitating distant assessments, directed maintenance, and in-depth training sessions [12]. Such advancements not only expedite the resolution of challenges but also guarantee that employees remain updated with contemporary expertise and capabilities, boosting the operational efficacy and competitive edge of manufacturing processes.

In essence, the integration and expansion of SCADA systems in the manufacturing sector are not just about automation and efficiency; they represent a holistic approach to modern manufacturing that encompasses both sustainability and workforce development.

2.7.3 Transportation

With the growing complexity of transportation networks, SCADA systems ensure safety, punctuality, and efficient resource allocation. They play a crucial role in preventing accidents, managing traffic congestion, and optimizing transportation schedules. Modern SCADA systems find applications in transportation infrastructure, including railways, airports, and traffic control systems. They monitor and control signaling systems, track vehicle movements, and manage logistics in a cen-

tralized manner. Regarding the railroad transport industry, SCADA systems can be used to monitor and control operations from junction switch positions, power supplies, signals, and crossing lights down to the status of the train's positioning sensors, all in real time from miles away. [13]

2.7.4 Water and wastewater management

Ensuring a consistent supply of clean water and efficient wastewater treatment is paramount for urban areas. SCADA systems are integral to water treatment plants, distribution networks, and sewage systems. They monitor water quality, control pumps and valves, and manage treatment processes ensuring water quality, reducing leakages, and enhancing the efficiency of treatment plants in an optimized manner. Researchers in [14] highlight the advantages of utilizing SCADA technology for the processes within a water treatment plant (WTP) with the purpose of efficiently and safely managing the disbursement and treatment of drinking water to the population of India. Their findings state that the manual operation of such plants are very inefficient and dangerous, pointing out that there is a considerable delay between critical situations arising and corrective action taking place. Whereas SCADA systems being employed in the process would lead to higher productivity and a safer environment for the downstream consumer.

2.7.5 Others

While the aforementioned sectors represent some of the primary applications of SCADA, its versatility extends to numerous other industries, including agriculture (for irrigation and crop monitoring), mining (for equipment monitoring and safety protocols), and healthcare (for facility management and equipment monitoring).

2.8 Advancements and Trends in SCADA

The dynamic nature of technological advancement ensures that no system remains static, and SCADA systems are of no exception. As industries evolve and process autonomy becomes more prevalent, SCADA systems have witnessed a plethora of advancements, adapting to the ever-changing demands of the industrial landscape. This section delves into the latest trends and innovations shaping the future of SCADA, offering insights into their implications and potential impacts.

2.8.1 Integration with Internet of Things (IoT)

The proliferation of IoT devices has paved the way for SCADA systems to integrate with a myriad of different sensors, actuators, smart devices, and databases, facilitating real-time data collection from a vast array of sources. In their previous applications, SCADA systems have proven to be a robust and dependable means for process automation, making them the face of industry 3.0 (otherwise known as the third industrial revolution), characterized by logic-based automation and advancements in Information Technology (IT). These systems along with their relative, Distributed Control Systems (DCS), have improved business value by reducing costs and increasing profitability. However, these legacy systems have always faced inherent limitations when it comes to device intelligence. This is where IoT devices thrive. The complementary nature of these technologies allows processes already utilizing legacy SCADA systems to build upon their monitoring and control operations, allowing for more data-centric activities such as intelligent manufacturing and production capabilities [4]. The convergence of SCADA and IoT enhances the granularity and scope of data acquisition. It enables more precise monitoring, predictive maintenance, and the optimization of operations, especially in geographically dispersed setups.

2.8.2 Cloud-based SCADA

Modern SCADA systems are increasingly migrating to cloud platforms, leveraging the scalability, flexibility, and cost-efficiency of cloud computing. Cloud-based SCADA systems offer enhanced data storage capabilities, advanced analytics tools, and improved accessibility. They also facilitate easier integration with other enterprise systems and reduce the overheads associated with on-premises infrastructure. However, with all the benefits of cloud-based SCADA systems comes the heightened risk of malicious actors infiltrating these systems through cyber attacks. As cited in [15], the proliferation of these complex network environments has opened the door for cyber threats to critical infrastructures, such as the Stuxnet virus disrupting an Iranian nuclear system in 2010. Although there are many benefits that come with cloud-based SCADA systems, close attention must be paid to system security - especially when integrating new technologies with classical SCADA systems.

2.8.3 Mobile SCADA solutions

With the ubiquity of mobile devices, SCADA solutions are now being designed to be accessible via smartphones and tablets, allowing operators and technicians to monitor and control systems remotely. [16] elaborates on this advancement by expressing that the proliferation of Java-based applications allowed any Java-enabled browser to remotely view and control the SCADA system from any terminal within the control center via TCP/IP network connection. Mobile SCADA solutions enhance the agility of operations, allowing for real-time interventions even when personnel are off-site. They also facilitate quicker response times during emergencies and provide a platform for real-time collaboration.

2.8.4 Predictive maintenance and analysis

Predictive maintenance minimizes unplanned downtimes, extends equipment lifespan, and reduces maintenance costs. It ensures that interventions are timely and based on data-driven insights rather than reactive responses to failures. With the advent of historian services, collecting and storing vast amounts of real-time process data through IoT device integration, paired with modern machine learning capabilities, allowing for the timely inference of anomalous machine behavior, operators can quickly be notified when a machine may need maintenance.

2.8.5 Cybersecurity Enhancements

As SCADA systems control critical infrastructure, ensuring their security is crucial to prevent potential disruptions, data breaches, and other malicious activities. Enhanced cybersecurity measures not only protect the integrity of SCADA systems but also ensure the safety and reliability of the operations they control. With the increasing digitization of industrial systems, cybersecurity has become paramount. Modern SCADA systems are being equipped with advanced security protocols, encryption techniques, and intrusion detection systems (IDS).

2.9 Challenges and Limitations

While SCADA systems have revolutionized industrial operations with their multifaceted functionalities and adaptability, they are not without their challenges. As with any technology, SCADA systems come with inherent limitations and face external challenges that can impact their efficiency and reliability. This section provides a comprehensive exploration of these challenges, offering insights into their implications and potential mitigation tactics.

2.9.1 Technical challenges

SCADA systems, especially those that are expansive and complex, can encounter technical issues related to integration, data synchronization, and communication disruptions. Technical challenges can lead to data inaccuracies, delayed responses, and even system downtimes. Addressing these issues requires continuous monitoring, regular system updates, and the integration of fail-safe mechanisms.

2.9.2 Economic considerations

The implementation and maintenance of SCADA systems can be capital-intensive, especially for large-scale operations or those requiring advanced functionalities. Economic challenges can deter smaller enterprises from leveraging the full potential of SCADA due not only to the cost of the software but also the time needed for securing the system's components, thorough security testing, and operator training involved. It's essential to conduct a thorough cost-benefit analysis, considering both immediate costs and long-term operational savings.

2.9.3 Security Concerns

As SCADA systems become more interconnected and accessible, they become vulnerable to cyber-attacks, data breaches, and unauthorized access. Security breaches can lead to operational disruptions, data loss, and even pose safety risks. Ensuring robust cybersecurity measures, continuous monitoring, and employee training are paramount to safeguarding SCADA systems. The following three figures elaborate on some of the potential cyber threats faced by SCADA systems (Figure 22 and Figure 23) along with where those threats commonly take place. (Figure 24)

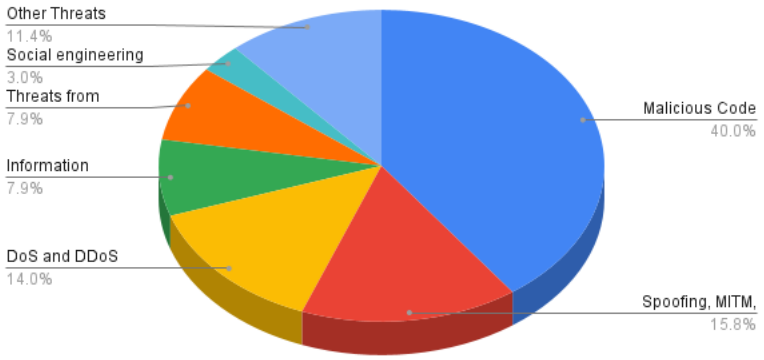


Figure 22. A statistical view of threats facing SCADA systems. [17]

1.Authorization Violation	9.Information leakage	17.Sabotage	25.Traffic Analysis
2.Bombs (Logic or Time)	10.Intercept/ Alter	18.Scavenging	26.Trap Door/ Back Door
3.Browsing	11.Interference Database Query Analysis	19.Spying	27.Trojan Horse
4.Bypassing Controls	12.Masquerade	20.Service Spoofing	28.Tunneling
5.Data Modifications	13.Physical Intrusion	21.Sniffers	29.Unauthorized Access Violations of Permission
6.Denial of Service	14.Replay	22.Substitution	30.Unauthorized Access Piggybacking
7.Eavesdropping	15.Repudiation	23.Terrorism	31.Virus
8.Illegitimate Use	16.Resource Exhaustion	24.Theft	32.Worm

Figure 23. Common attack types experienced in a SCADA system [18]

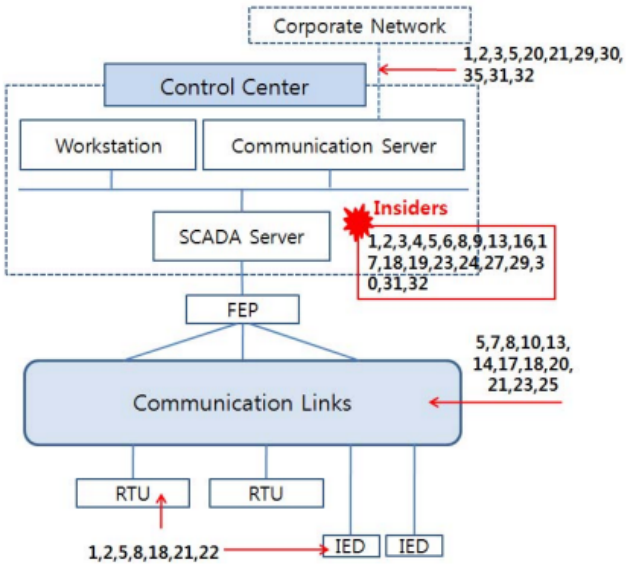


Figure 24. Attack types applied to common SCADA architecture [18]

2.9.4 Human factors

The successful operation of SCADA systems is not solely dependent on technology but also on the human operators managing them. Resistance to change, inadequate training, and human errors can pose significant challenges. Another consideration that may cause issues down the line is the design of the HMI. If the procedures within the control panel are not laid out in a logical manner, employees may be more susceptible to making mistakes or delaying response time in their attempts to find certain functions. Human-related challenges can lead to operational inefficiencies, increased error rates, and even safety concerns. Continuous training, user-friendly interfaces, and the integration of automated checks can help mitigate these challenges.

2.9.5 Scalability and Adaptability

As industries evolve and expand, SCADA systems need to scale and adapt accordingly. However, not all systems are designed with easy scalability in mind, leading to potential bottlenecks. Limitations in scalability can hinder growth, lead to inefficiencies, and necessitate costly system overhauls. Future-proofing SCADA systems through modular designs and cloud integration can address these challenges.

2.10 Conclusion

2.10.1 Recap of Key Points Discussed

The concept of industrial automation has been enriched by the exploration of SCADA systems, which have emerged as a crucial aspect of modern industrial operations. These systems, characterized by their hierarchical structure, are exemplified by the "Automation Layer Pyramid," which delineates the various components and their interrelationships within an industrial setting.

A deep dive into the SCADA architecture revealed its multifaceted nature.

Central to this architecture is the Human-Machine Interface (HMI), which serves as the crux between operators and the system, providing a visual representation of system status and facilitating user interaction. Complementing the HMI are the Programmable Logic Controllers (PLCs) and Remote Terminal Units (RTUs), which function as the operational backbone, ensuring real-time data acquisition and control.

The document also sheds light on the evolution of SCADA systems, tracing their trajectory from rudimentary control systems to sophisticated, interconnected networks capable of overseeing complex industrial processes. This evolution has been marked by technological advancements, integration with emerging technologies such as the Internet of Things (IoT), and a heightened emphasis on security in the face of increasing cyber threats.

In essence, the discussion provided a panoramic view of SCADA systems, capturing their essence, evolution, and enduring significance in the realm of industrial automation.

2.10.2 Implications for Industry Professionals

The advancements in SCADA systems offer a plethora of opportunities for industry professionals. The integration of IoT devices and the potential of machine learning capabilities can revolutionize how industries operate, shifting from reactive responses to proactive insights. For researchers, the evolving landscape of SCADA presents a rich ground for further exploration, especially in enhancing system security and integrating newer technologies.

The case studies provided earlier offer tangible evidence of the transformative power of SCADA in various sectors, from electrical grid management to manufacturing. These real-world applications underscore the importance of SCADA in driving efficiency, reliability, and innovation in industrial processes.

2.10.3 Recommendations for Future Research

In the contemporary domain of industrial automation, SCADA systems have ascended to a prominent position, exemplifying unmatched efficiency and control. As the industrial sector advances, several research trajectories become evident. The nexus between SCADA and the rapidly expanding Industrial Internet of Things (IIoT) presents promising opportunities, anticipated to revolutionize real-time data processing and decision-making paradigms. Concurrently, the persistent challenges posed by cyber threats underscore the imperative of fortifying security protocols within SCADA systems, making a comprehensive approach to enhancing security measures paramount to safeguarding critical infrastructure.

The advent of machine learning techniques introduces the potential for predictive maintenance within SCADA frameworks. Such advancements aim to proactively identify and rectify system anomalies, thereby mitigating potential disruptions. Innovations in the realm of user experience are on the horizon, particularly concerning the Human-Machine Interface (HMI). With the progression of augmented and virtual reality technologies, there exists potential for significant enhancements in HMI design and functionality.

As the global industrial community gravitates towards sustainable practices, the amalgamation of SCADA with eco-friendly technologies emerges as a prospective benchmark. Such integrations could catalyze the shift towards more energy-efficient industrial operations.

In conclusion, while SCADA systems have come a long way, the journey of innovation and improvement is continuous. By focusing on these recommended areas, researchers can pave the way for a more efficient, secure, and versatile SCADA ecosystem in the future.

List of References

[1] A. Ujvarosi, "Evolution of scada systems," *Bulletin of the Transilvania University of Brasov. Series I: Engineering Sciences*, pp. 63–68, 2016.

[2] I. Automation, "The new scada," Jun 2014. [Online]. Available: https://inductiveautomation.com/resources/article/old-scada-vs-the-new-scada

[3] B. Babayigit and M. Abubaker, "Industrial internet of things: A review of improvements over traditional scada systems for industrial automation," *IEEE Systems Journal*, 2023.

[4] A. Nechibvute and H. D. Mafukidze, "Integration of scada and industrial iot: Opportunities and challenges," *IETE Technical Review*, p. 1–14, 2023.

[5] M. Alanazi, A. Mahmood, and M. J. M. Chowdhury, "Scada vulnerabilities and attacks: A review of the state-of-the-art and open issues," *Computers & Security*, p. 103028, 2022.

[6] S. N. Deshpande and R. M. Jogdand, "A novel scheduling algorithm development and analysis for heterogeneous iot protocol control system to achieve scada optimization: a next generation post covid solution," *International Journal of Information Technology*, pp. 1–9, 2023.

[7] E. T. Inc., "7 essential things to know about mqtt security 2023," Mar 2023. [Online]. Available: https://www.emqx.com/en/blog/essential-things-to-know-about-mqtt-security

[8] O. AKANDE, *Industrial Automation from scratch: A hands-on guide for using sensors, actuators, plc, HMI,... and SCADA to Automate Industrial Processes.* PACKT PUBLISHING LIMITED, 2022.

[9] Z. Hasnain, "Pcs scada network prototype," Ph.D. dissertation, Murdoch University, 2012.

[10] S. E. International, "Grid automation: National grid adopts abb's scada system," Jun 2015. [Online]. Available: https://www.smart-energy.com/regional-news/north-america/grid-automation-national-grid-adopts-abbs-scada-system/

[11] T. G. Kukuni, B. Kotze, W. Hurst, and L. Lepekola, "Augmented reality in smart manufacturing: A user experience evaluation," *Webology*, vol. 19, no. 3, pp. 2405–2423, 2022.

[12] G. F. Cyrino, J. O. Neto, D. A. De Lima, A. Cardoso, E. A. Lamounier, G. F. De Lima, D. P. Campos, and L. F. Queiroz, "Optimizing hvdc maintenance and training through virtual and augmented reality: A methodology

proposal," in *2023 18th Iberian Conference on Information Systems and Technologies (CISTI)*. IEEE, 2023, pp. 1–4.

[13] I. Lopez, M. Aguado, C. Pinedo, and E. Jacob, "Scada systems in the railway domain: enhancing reliability through redundant multipathtcp," in *2015 IEEE 18th International Conference on Intelligent Transportation Systems*. IEEE, 2015, pp. 2305–2310.

[14] S. L. Andhare and P. J. Palkar, "Scada a tool to increase efficiency of water treatment plant," *Asian Journal of Engineering and Technology Innovation*, vol. 2, no. 4, pp. 7–14, 2014.

[15] A. Sajid, H. Abbas, and K. Saleem, "Cloud-assisted iot-based scada systems security: A review of the state of the art and future challenges," *Ieee Access*, vol. 4, pp. 1375–1384, 2016.

[16] M. Karacor and E. Ozdemir, "Mobile phone-based scada automation," *Measurement and Control*, vol. 37, no. 9, pp. 268–272, 2004.

[17] H. M. Al Hamadi, C. Y. Yeun, and M. J. Zemerly, "A novel security scheme for the smart grid and scada networks," *Wireless personal communications*, vol. 73, pp. 1547–1559, 2013.

[18] D.-J. Kang, J.-J. Lee, S.-J. Kim, and J.-H. Park, "Analysis on cyber threats to scada systems," *2009 Transmission amp; Distribution Conference amp; Exposition: Asia and Pacific*, 2009.

MANUSCRIPT 3

Data-Driven Defense for Smart Manufacturing: A Novel Anomaly Detection Dataset and Comparative Algorithmic Study

Zachary A. deWardener[1*], Marwan F. Abdelatti,and Manbir S. Sodhi[1]

[1]Department of Industrial Engineering, University of Rhode Island, Kingston, Rhode Island, 02881

[*]Corresponding author: zdewardener@uri.edu

Drafted for submission, XXX–XXX.

ACM ISBN XXX-X-XXXX-XXXX-X/XX/XX

https://doi.org/XX.XXXX/XXXXXXX.XXXXXXX

3.1 Abstract

The Industrial Internet of Things (IIoT) seeks to optimize the control of industrial systems by facilitating the exchange of vast amounts of data among various components. However, the extensive and interconnected nature of IIoT communication networks renders most present industrial structures susceptible to significant security vulnerabilities. Thus, there is an urgent requirement for anomaly detection solutions as a critical post-attack process in IIoT cybersecurity. In this study, we present a systematic literature review of IIoT cybersecurity research over the past decade, with a particular emphasis on anomaly detection techniques. We summarize our review results using advanced visualization techniques and establish correlations between different IIoT cybersecurity research domains. We then delve into our case study, elaborating on the creation of our own intrusion detection system (IDS) trained on artificial data derived from high fidelity simulation modeling. After generating the simulated machine process data for our model-based manufacturing system (i.e., the LabFab system), we subsequently investigate several supervised machine learning (ML) algorithms and develop anomaly detection systems that enable decision-makers to detect potential attacks promptly and take appropriate action. Specifically, we discuss our development of a novel dataset for Industrial IoT that concentrates on the application layer. We then train four ML models on this database and evaluate their performance using standard metrics. Our findings pave the way for future research in the field of IIoT cybersecurity and provide a roadmap for the development of new, intelligent anomaly detection systems to safeguard smart manufacturing systems against potential cyber-attacks.

3.2 Introduction

Modern industrial control systems (ICS) are leveraging the capabilities of Industry 4.0 and the Internet of Things (IoT) to improve their performance through the Industrial Internet of Things (IIoT). These systems possess the ability to make autonomous decisions and establish remote connections, empowered by intelligent sensors and actuators [1]. The IIoT has garnered substantial attention in both research and commercial spheres in recent years, with a projected global market worth of approximately $950 billion by 2025 [2]. However, the vast amount of data transfer required by these systems necessitates a high level of security to safeguard against cyberattacks capable of disrupting the entire process. In order to combat these malicious events, companies worldwide are committing vast amounts of capital towards programmatic solutions (i.e., Intrusion Detection Systems (IDS)) backed by Machine Learning (ML) algorithms used to spot malicious behavior. However, these algorithms must be trained on data from the system both at normal working conditions and when an attack scenario is taking place. One prominent issue with this is that collecting and labelling data from a physical system can be time consuming, expensive, and potentially dangerous when dealing with cyber security applications. For this reason our work covers the generation of synthetic data for an IDS to accurately detect and label anomalous behavior with the goal of providing companies a cost-effective way to build, train, and test their own IDS in-house.

This paper primarily discusses the development of an anomaly detection dataset and investigates the application of standard ML algorithms for detecting anomalies in IIoT systems. The dataset is drawn from a modular smart manufacturing system (LabFab) designed by [3], where robot operating system (ROS) and MQTT protocol are incorporated to simulate an Industry 4.0 factory of multiple

machinery stations. The dataset comprises historical sensory data collected from the system during both normal operation and under seven different types of attacks, obtained from the application layer, where smart controllers communicate with smart sensors and actuators integrated into the system. The ML models are trained to learn behavioral patterns of the system processes and to identify potential patterns of various attacks. Despite the apparent simplicity of this task, the large number of system devices and the high degree of variability between different components increase the complexity of the problem.

The remainder of the paper is organized as follows: a literature review of IIoT cybersecurity with a focus on anomaly detection and synthetic dataset generation methods is presented in Section 3.3. The following section 3.4 covers the development and application of our LabFab simulation, used for the creation of our synthetic dataset. Section 3.5 explains the benchmark Distributed Smart Manufacturing System (DSMS) dataset, including a description of the main features and their meanings, as well as the procedures we followed to prepare the dataset for training and testing on the models. The algorithms and metrics considered for evaluation are discussed in Section 3.6. Experimental results are reported and discussed in Section 3.7, followed by our future work and conclusions in Section 3.8.

3.3 Related Works

3.3.1 Intrusion Detection Systems

The realization of Industry 4.0's complete potential may be hindered by cybersecurity vulnerabilities. In an attempt to establish a dependable security framework for Industrial Internet of Things (IIoT) process automation, numerous research efforts have been conducted. In this regard, our study examines the current research trends in IIoT security by scrutinizing nearly 500 relevant works in the literature. The process we followed for our systematic literature review is depicted

in Figure 25. While adhering to the four main stages typical of such reviews, we tailored certain activities to align with the unique needs of our study. Our approach encompassed the following steps:

1. *Objective Definition:* Our goal was to identify and critically assess research published in the last ten years on cybersecurity in industrial Internet of Things (IoT) systems.

2. *Information Source Selection:* We conducted a comprehensive search using two extensive databases: Scopus (`www.scopus.com`) and Google Scholars (`scholars.google.com`).

3. *Keyword and Query Selection:* Our search was guided by key terms relevant to the field, including "cybersecurity", "information security", "industrial internet of things", "IIoT", "Industry 4.0", "smart manufacturing", and "smart factory". The search query formulated was: ("cybersecurity" OR "security") AND ("industrial internet of things" OR "IIoT" OR "Industry 4.0" OR "smart manufacturing" OR "smart factory"), resulting in 538 articles published or accepted for publication as of July 2022.

4. *Filtering and selection:* We refined our search by excluding articles not written in English and those that were merely poster presentations.

5. *Qualitative Content Selection Mechanism:* To assure the quality of the articles selected, we implemented a qualitative mechanism. This involved a thorough screening of each article's abstract. Articles deemed irrelevant or not meeting our criteria were excluded from further consideration.

The acquired research articles have been systematically categorized into 30 distinct categories. To facilitate a thorough understanding of the findings obtained,

we have created a Sankey diagram, which presents the quantitative results and their interrelationships in an input/output format. This visualization is particularly useful since a considerable number of the articles under study cover multiple categories and a comprehensive overview is necessary.

Figure 26 depicts the Sankey diagram of the survey results, where the nodes of specific colors represent the 30 categories and subcategories, with their sizes reflecting the reported article numbers. The figure comprises three distinct areas: the left area delineates the six core categories found in the reported articles, namely data security, authentication, anomaly detection, resource management, datasets/testbeds, and surveys.

The middle area illustrates the subcategories that either branch out from the core categories or represent additional features commonly found therein. This area encompasses 14 subcategories, including data storage, authorization, encryption algorithms, data sharing, threat analysis, behavioral anomaly, intrusion detection, security framework, threat analysis, security framework, asset management, computing reservation, network resources, and testbeds.

Lastly, the area on the right showcases the ten main techniques used in the adopted approaches, such as artificial intelligence (AI)/neural networks (NN)-based algorithms, post-quantum or quantum cryptography, parallel GPU acceleration, and blockchain, among others. The links between the nodes signify the number of articles discussing the respective topics.

Based on the Sankey diagram depicted in Figure 26, the three primary topics that have been widely explored in the literature are data security, surveys, and anomaly detection, accounting for 138, 123, and 81 publications, respectively. Data security encompasses data storage techniques, data sharing methods, and data encryption. Among the publications relevant to anomaly detection, 17 utilized AI or

NN approaches, while six employed blockchain techniques at both the network and application layers. 14 publications addressed behavioral anomalies on the application layer, while 23 publications focused on intrusion detection, with one article utilizing evolutionary algorithms and heuristic techniques. Anomaly detection via network traffic analysis was the subject of 11 publications, nine introduced security frameworks, and one addressed pre-attack threat analysis for anomaly detection. Table 2 summarizes relevant publications selected by the authors showcasing the main category of the research focus (as presented in column 1), the specific point of research (column 2), list of selected papers (column 3), and the technique the researchers used to deliver their approach (column 4).

Our research specifically delves into machine learning-based [5, 6, 7, 8, 9, 10, 11] and deep learning-based [12, 13, 14] approaches. These methodologies can be classified into three categories: supervised [5, 6, 7], semi-supervised [8, 9], and unsupervised [10, 11].

In [8], a semi-supervised learning approach using auto-encoders on individual clients and a global neural network was proposed for anomaly detection in Industrial Internet of Things (IIoT) device data. However, the utilization of local training data resulted in considerable communication overhead, and the random selection of clients led to increased training time and communication costs for unselected clients. In [9], a deep learning-based semi-supervised scheme that employed auto-encoders and deep neural networks was suggested for training on data collected from gas pipeline remote telemetry unit streams. However, the authors did not provide any comparative analysis of their algorithm in terms of speed or accuracy with other state-of-the-art approaches. A Gaussian Mixture-based model was introduced in [10] to identify zero-day attacks in IIoT edge networks, which was evaluated on **NSL-KDD** and **UNSW-NB15** datasets. The proposed

method exhibited higher accuracy and faster processing in comparison to five other anomaly detection techniques. However, the authors did not assess the system's performance against other types of attacks. In [11], a multi-feature data clustering optimization model was proposed for intrusion detection in industrial networks. The algorithm was evaluated on the **NSL-KDD** dataset and six different attack methods, but it required a greater number of features to outperform other algorithms in the literature.

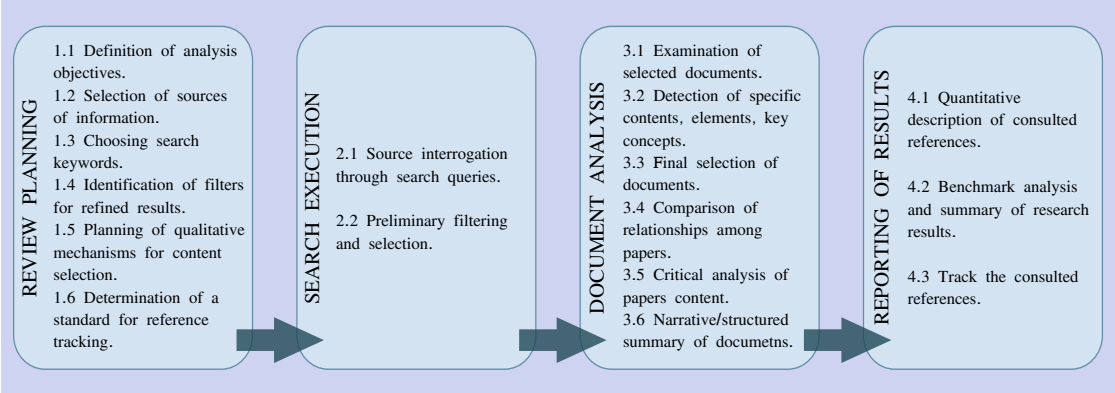


Figure 25. Typical steps and activities in a systematic literature review [4]

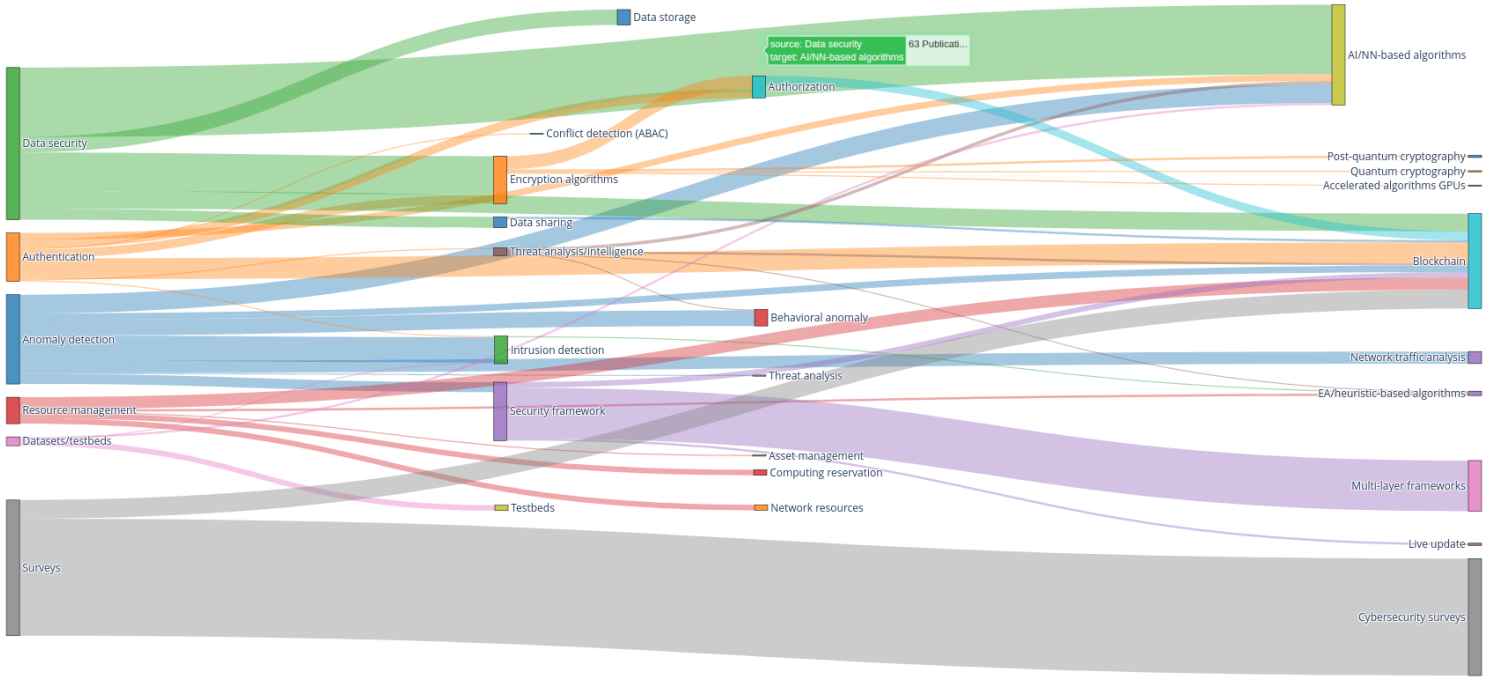


Figure 26. Sankey diagram of reported literature [15]

Table 2. Selected publications

Category	**Subcategory**	**Related papers**	**Used Technique**
Anomality detection	Intrusion detection	[16, 10, 17, 18, 12, 19, 9, 20]	AI/NN-based algorithms
		[5]	EA/heuristic-based algorithms
Data security	Device authentication	[21, 22]	Cryptographic Protocols
	Encryption algorithms	[23]	Structural analysis
	Data storage	[24]	Integration of fog and cloud computing
		[25]	Cryptographic Data Structures
	Data sharing	[26, 27, 28, 29]	Cryptographic Algorithms
Datasets/Testbeds	N/A	[30, 31, 32, 33]	N/A
Resource management	Asset management	[34]	Security Content Automation Protocol
	Edge computing	[35]	EA/heuristic-based algorithms
	Source location privacy	[36]	Phantom node, fake path.
	Remote attestation process	[37]	Machine to machine communication
Surveys	N/A	[2, 38, 39, 40, 41, 42, 43, 44, 45, 46]	N/A

3.3.2 Synthetic Dataset Generation

The generation of artificial data for cyber security development and testing has emerged as a subject of significant interest in contemporary academic research. This is partially due to the fact that the deployment of effective IDSs requires a substantial amount of data to train and test the ML models that make for accurate and timely inference. In order to create truly effective IDSs, it is necessary to have datasets that are representative of real-world scenarios. However, collecting and labeling datasets from real processes is time-consuming, expensive, and potentially hazardous as there is also a risk of compromising sensitive data in the process.

One approach to overcoming this challenge is to generate artificial data using machine learning techniques. A number of studies have proposed the use of generative models, such as Generative Adversarial Networks (GANs), to create synthetic datasets that mimic anomalous circumstances while taking into account real-world behavior. For example, *Wen Xu et al.* [47] proposed an extended Bi-GAN (Bi-directional GAN)-based approach to generate simulated network traffic data that can be used to train and evaluate one-class classification models more efficiently. They demonstrated that their generated datasets had similar statistical characteristics to real network traffic, and the trained IDS achieved a high detection rate on both the generated and real datasets while requiring less training overhead and computational complexity. Their work however differs from ours in the fact that we do not utilize this form of data generation, nor are we observing network-level traffic.

Another approach is to use synthetic data generation techniques, such as rule-based and model-based methods. Rule-based methods involve the creation of rules and constraints that govern the generation of data, while model-based methods use statistical models to generate data that is similar to the real-world data. *Jeske*

et al. [48] proposed their architecture for a synthetic data generation tool that can expedite the process of building test cases for Information Discovery and Analysis Systems (IDAS), which are commonly applied in many applications including homeland security, where these systems are used to correlate data and identify potential threats or significant events. the authors focus primarily on the development of their IDAS Data Set Generator (IDSG), for handling the issue of generating synthetic baseline datasets in which hypothetical future scenarios can be overlain while maintaining the key relationships to real data behavior. For the proposed IDSG, the authors utilized statistical and rule-based methods along with semantic graphs to produce datasets more efficiently with the goal of testing the effectiveness of IDASs. The work presented in [48] also differs from our study as we look to perform anomaly inference through an IDS and dataset generation through simulation methods rather than rule-based.

In addition to these approaches, there has also been research on the use of simulation environments as a means for generating synthetic datasets for cyber security testing. For example, *belenko et. al* [49], utilizes a network simulator (NS-3) to generate synthetic data to be used for network intrusion detection on a Vehicular Adhoc NETwork (VANET), which handles vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2X) communication for enhanced safety and convenience on the road. the simulator takes a number of input parameters such as Location, Scenario of Threat (5 attack scenarios), Number of Hosts, and Number of Malicious Hosts to generate three synthetic datasets containing different network-level insights. Figure 27 illustrates the schema of the VANET dataset generation used in this study. The work of [49] differs from our proposed dataset generation method due to the fact that the researchers within this study focus on network-level traffic, while we focus primarily on events occuring at the application-layer.

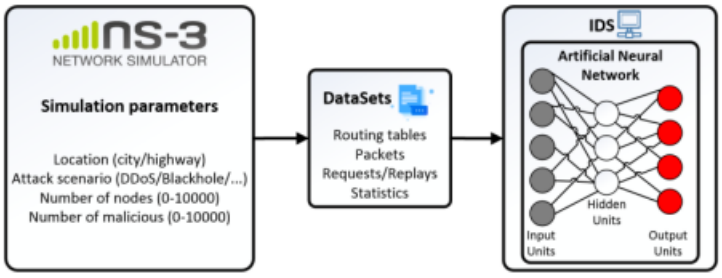


Figure 27. Schema of VANET dataset generation.

In summary, these approaches offer a promising solution to the challenges associated with collecting and labeling real-world datasets. However, further research is needed to explore their effectiveness and limitations in different application domains, along with their applicability to different types of cyber security scenarios. In the following sections we will demonstrate our utilization of the simulation environment method to generate synthetic data that follows almost exactly that of the real system's behavior.

3.4 Simulating the Manufacturing System

As highlighted in Section 3.2, our research employed a model-based manufacturing system, LabFab, as a testing ground for developing our Intrusion Detection System (IDS). The choice of this system over a real factory environment was driven by two primary factors: [i] the academic nature of our research, with LabFab being a student-constructed system designed to impart interdisciplinary skills relevant to Industry 4.0 concepts, and [ii] the practicality of model-based systems in professional settings for obtaining critical system performance insights without disrupting production or risking damage to costly equipment. In aligning our research with industrial applications, we approached LabFab as a surrogate for a real factory environment, which necessitated the creation of a substantial dataset for training and testing our IDS.

To address this need, we generated a synthetic dataset using MATLAB's Simulink, a widely-used block-diagram environment for multi-domain simulation and model-based design, in conjunction with the SimEvents library for discrete event simulations. Our simulation aimed to closely replicate the LabFab manufacturing system, incorporating various components such as Entity Generators, Servers, Input/Output Switches, and Simulink Function Blocks. The comprehensive simulation encompassed five key processes: [i] the part ordering process, [ii] part routing, [iii] data logging, [iv] attack scenario process, and [v] a simulated process for each operation in the physical system. The detailed structure and interactions of these processes are depicted in Figure 28.

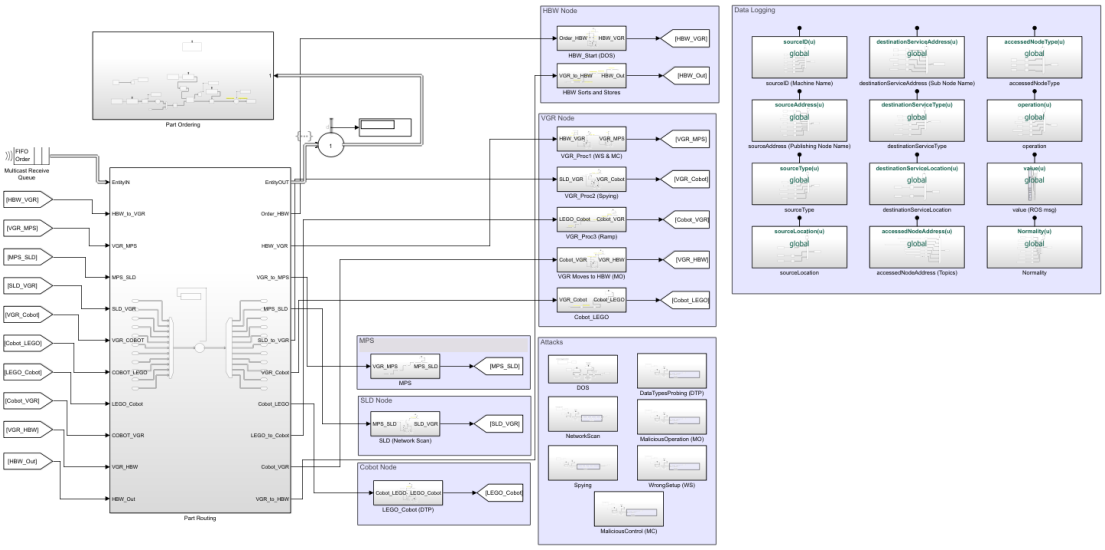


Figure 28. LabFab system simulation (full-view).

3.4.1 Part Ordering Process

Figure 29 presents a detailed view of the part ordering process within our simulation. This process begins with the creation of nine distinct parts, referred to as entities, by the "PartOrder" Entity Generator block. These entities are then directed to an Entity Store block, symbolizing the High-Bay Warehouse (HBW),

where they are temporarily stored. Subsequently, a second Entity Generator block, named "Order", generates an entity endowed with a Part_ID attribute. This attribute is randomly assigned a value from a set range (Part_ID = 1,2,...,9), representing the diversity of parts available in the system.

The simulation then moves to the Entity Selector block, where the Part_ID attribute of the order entity is matched with one of the nine initially created parts. This step effectively simulates the placement of an order within the system. Following this, the entities proceed through the Entity Gate block, which is configured to permit passage at the start of the simulation, leading them to the first data-logging Entity Server block. This server block is programmed to send messages (highlighted in yellow) to the workspace, triggered by the "entry" and "exit" event actions. The entire data logging process is further explained in Section 3.4.3 and illustrated in Figure 31 and Figure 32, providing a comprehensive understanding of the simulation's part ordering and tracking mechanism.

3.4.2 Part Routing Process

After leaving the Part Ordering Process, entities travel through the Part Routing process as seen in Figure 30 which updates the entity's *CurrentRoute* and *CurrentStep* attributes based on the predetermined *ServiceProcess* attribute. After every operation, entities pass through this process to determine the parts next target location. Inspiration for the setup of this portion was drawn from the MATLAB example model found in [50].

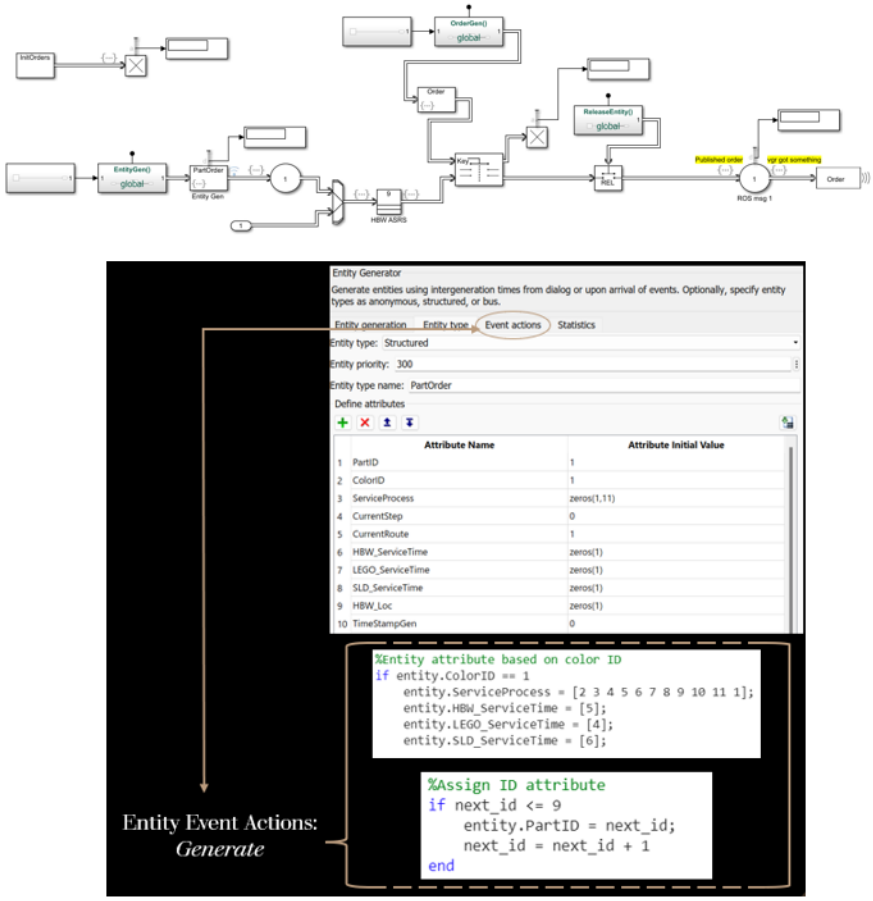


Figure 29. Internal view of the LabFab simulation part ordering process.

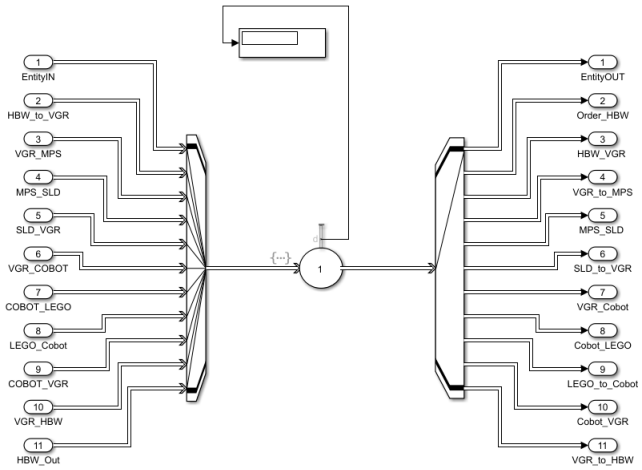


Figure 30. Internal view of LabFab simulation entity routing process.

3.4.3 Data Logging Process

Data logging and the generation of datasets are pivotal components in any simulation, serving as the foundation for analysis and model validation. In our simulation, data was meticulously captured through event action signals emanating from Entity Server blocks. These signals were then channeled to Simulink Function blocks located in the Data Logging section, as depicted in Figure 31. The journey of these signals continued through an Entity Output Switch, where their path was determined by a specific port number value, such as Normality(**1**), illustrated in Figure 32.

Once the signals traversed this stage, they were converted into string constants and relayed to the MATLAB workspace in a timeseries format. This format facilitated easy access and manipulation within MATLAB's workspace environment. The data underwent a series of transformations for enhanced clarity and usability. Finally, the simulation data was exported to an Excel file, organized by feature columns in separate sheets, ready for pre-processing via Python scripting. The process script, *ExportData2Excel.mlx*, is available at `https://github.com/Zdewardener/LabFab_DS2OS_Simulink`. This repository serves as a valuable resource for those interested in replicating or building upon our simulation and data handling methodologies.

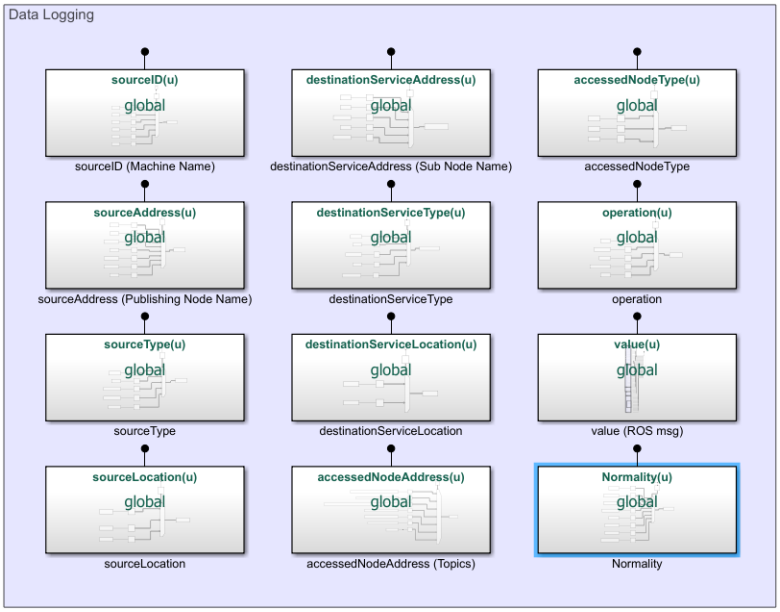


Figure 31. Data Logging area (external-view).

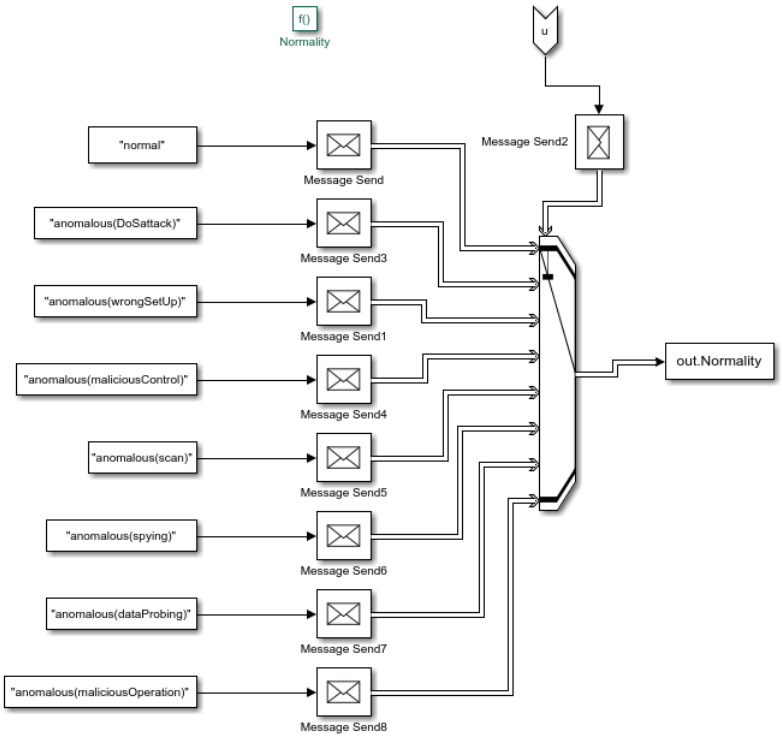


Figure 32. Inside the data logging Simulink Function blocks (Normality).

3.4.4 Simulated Attack process Flow

The Process Flowchart depicted in Figure 36 outlines our methodology for simulating a Denial Of Service (DOS) attack within the LabFab system using Simulink. This framework was adapted to simulate six additional attack patterns (Figure 33), incorporating minor modifications in data logging to align with the specific characteristics of each attack type, described in Section 3.5.2. the upper section of the Flowchart presents a streamlined depiction of the process, akin to what is shown in Figure 34. Conversely, the lower section of the Flowchart corresponds to the Simulink subsystems that manage the attack entities, as illustrated in Figure 35.

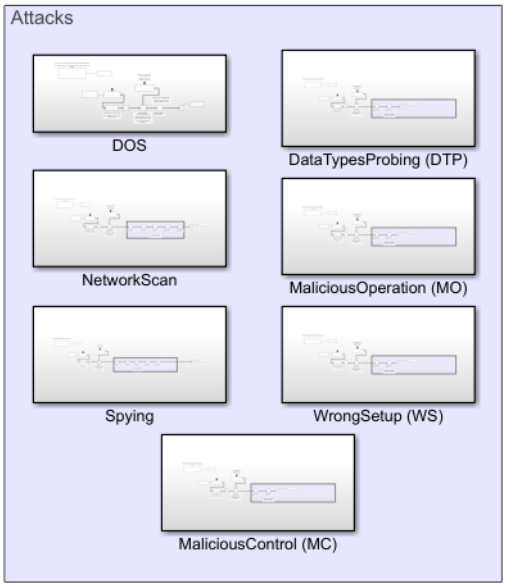


Figure 33. Simulink Subsystems containing cyber attack scenarios (external-view).

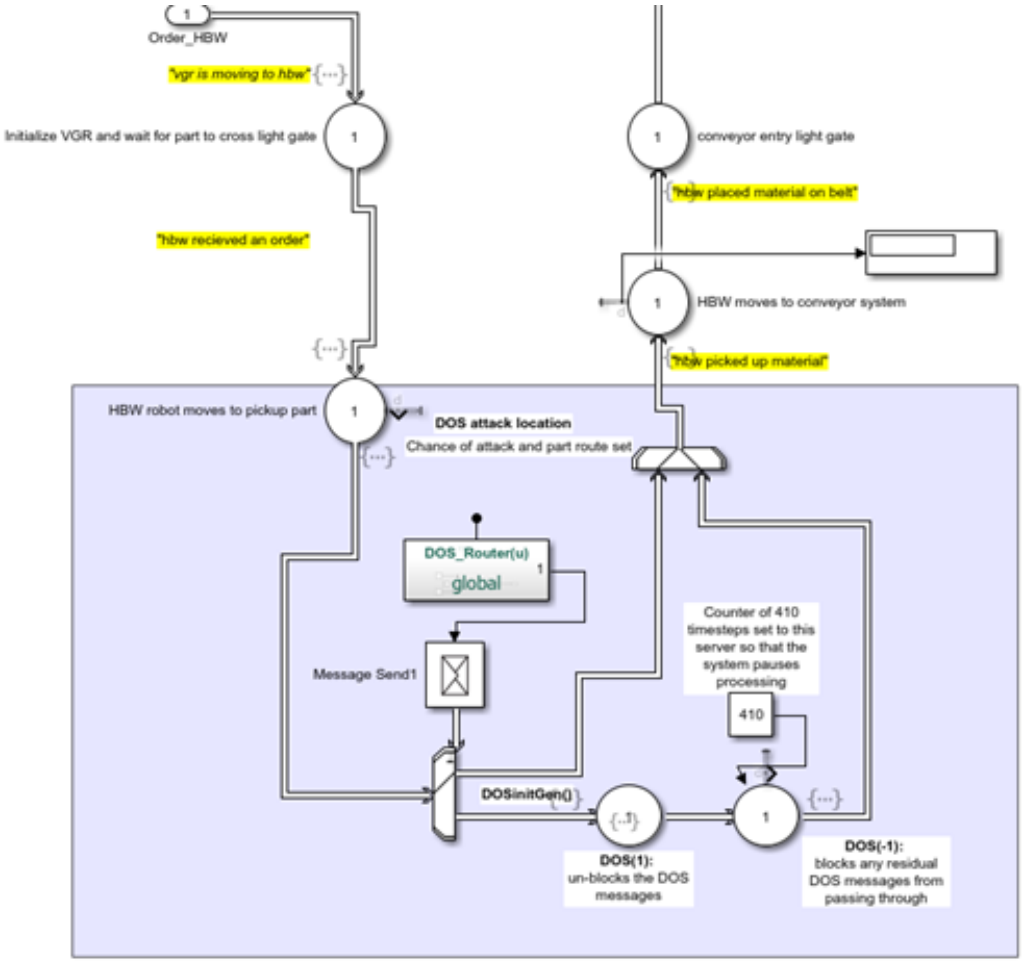


Figure 34. DOS attack routing found in HBW_Start subsystem (Initializing DOS attack).

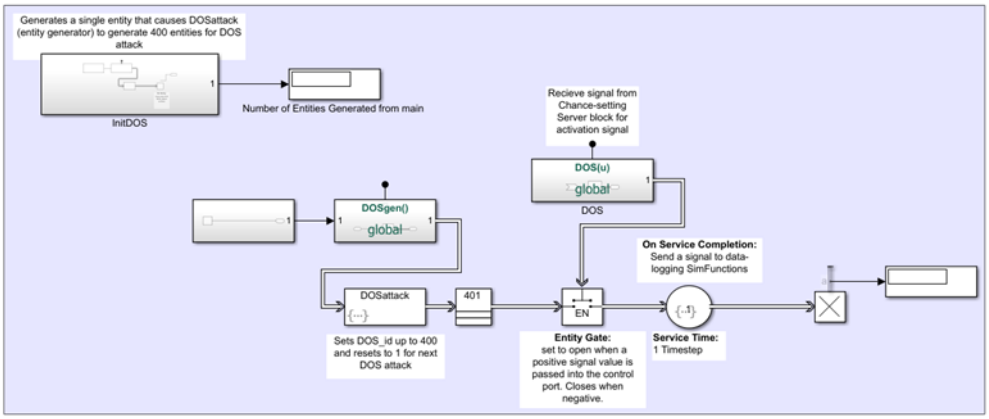


Figure 35. DOS attack Subsystem structure (Generating DOS messages).

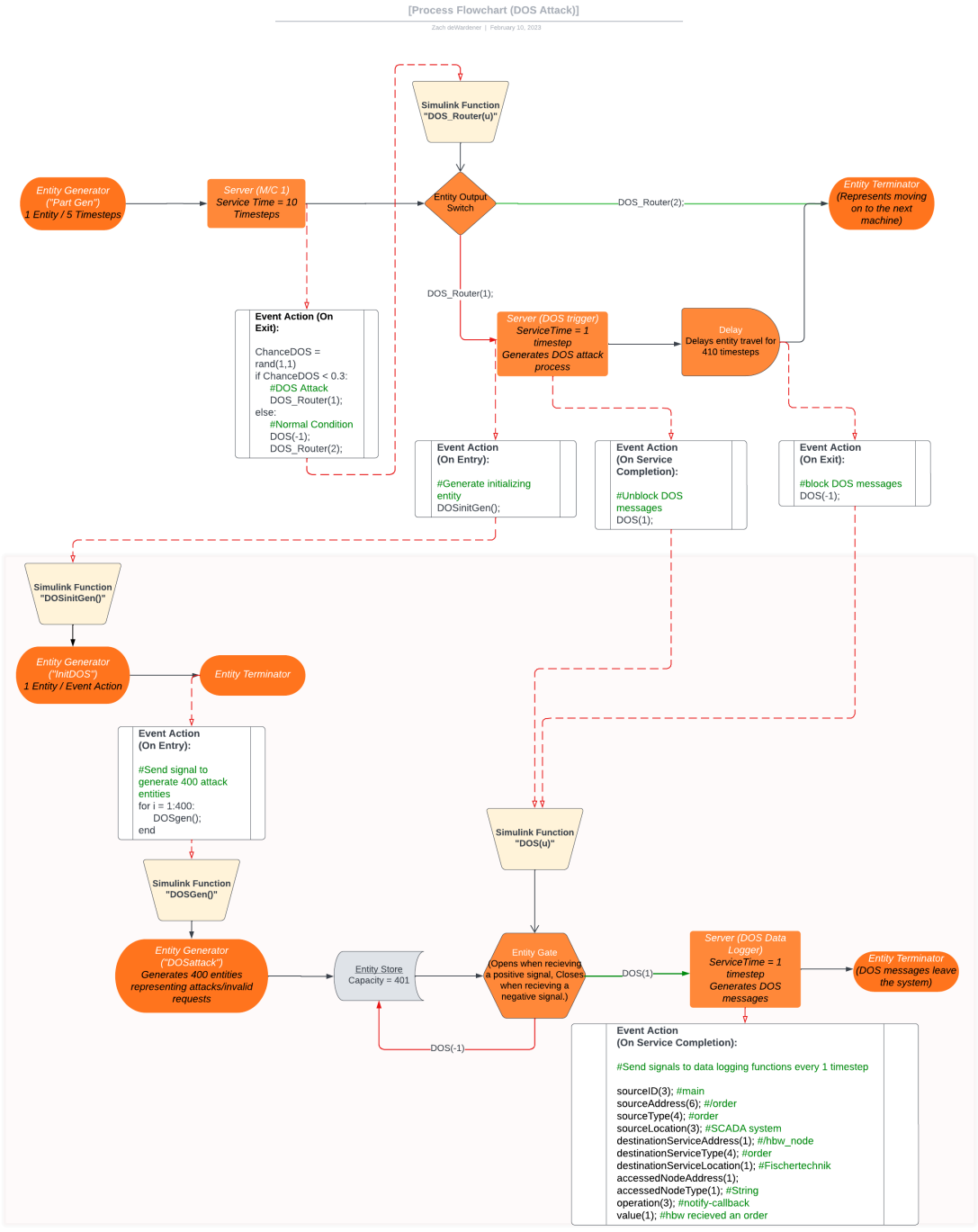


Figure 36. Process Flowchart for a DOS attack within the simulated model.

3.4.5 Simulated Dataset

After the simulation ran for a varying number of time steps (dependant on the attack type), we exported the resulting data from MATLAB's workspace to be further processed into 300,000+ rows of observational messages with accompanying column-features as seen in Figure 37, showing the state of normal behavior and Figure 38, showing anomalous behavior (DoS attack). The feature labels atop of these datasets are the same as those found in the DS2OS dataset [51] and information regarding their meanings in the case of our research are discussed further in Section 3.5 below.

Time	SourceID	SourceAddress	SourceType	SourceLocation	DestinationServiceAddress	DestinationServiceType	DestinationServiceLocation	AccessedNodeAddress	AccessedNodeType	Operation	Value	Normality
1676246400	main	/Order	Order	SCADA system	/hbw_node	Order	FischerTechnik	/new_order, /rosout	String	notify-callback	Published Order	normal
1676246470	main	/Order	Order	SCADA system	/vgr_node	Order	FischerTechnik	/diagnostics, /new_order, /prov	String	notify-callback	VGR got something	normal
1676246470	main	/Order	Next Task	FischerTechnik	/vgr_node	Next Task	FischerTechnik	/diagnostics, /new_order, /prov	String	notify-callback	VGR is Moving to HBW	normal
1676246670	main	/Order	Order	SCADA system	/hbw_node	Order	FischerTechnik	/diagnostics, /hbw_pub, /new_	String	notify-callback	HBW Recieved an Order	normal
1676247670	hbw	/hbw_node	Reference Switch	FischerTechnik	/hbw_node	Next Task	FischerTechnik	/diagnostics, /hbw_pub, /new_	String	notify-callback	HBW Picked Up Material	normal
1676253670	hbw	/hbw_node	Motion Sensor	FischerTechnik	/hbw_node	Next Task	FischerTechnik	/diagnostics, /hbw_pub, /new_	String	notify-callback	HBW Placed Material on Belt	normal
1676256681	hbw	/hbw_node	Motion Sensor	FischerTechnik	/vgr_node	Next Task	FischerTechnik	/diagnostics, /new_order, /prov	Boolean	notify-callback	VGR Picked Up Material from HBW	normal
1676256781	vgr	/vgr_node	Reference Switch	FischerTechnik	/vgr_node	Next Task	FischerTechnik	/diagnostics, /new_order, /prov	String	notify-callback	VGR is Moving to MPS	normal
1676258281	hbw	/hbw_node	Next Task	FischerTechnik	/hbw_node	Next Task	FischerTechnik	/diagnostics, /hbw_pub, /new_	String	notify-callback	HBW Material is Picked by VGR	normal
1676260281	hbw	/hbw_node	Motion Sensor	FischerTechnik	/hbw_node	Next Task	FischerTechnik	/diagnostics, /hbw_pub, /new_	String	notify-callback	Empty Box is Ready for HBW Pick Up	normal
1676261511	hbw	/hbw_node	Next Task	FischerTechnik	/hbw_node	Next Task	FischerTechnik	/diagnostics, /hbw_pub, /new_	String	notify-callback	HBW Picked up the Empty Box from the Belt	normal
1676272011	vgr	/vgr_node	Next Task	FischerTechnik	/vgr_node	Motion Sensor	FischerTechnik	/diagnostics, /new_order, /prov	String	notify-callback	VGR Dropped Material at MPS	normal
1676287511	vgr	/vgr_node	Next Task	FischerTechnik	/vgr_node	Reference Switch	FischerTechnik	/diagnostics, /new_order, /prov	String	notify-callback	VGR is Set to Home	normal
1676288111	hbw	/hbw_node	Next Task	FischerTechnik	/status_update (main)	Next Task	FischerTechnik	/diagnostics, /hbw_pub, /new_	String	notify-callback	HBW Empty Box is Back on the Shelf	normal
1676288111	hbw	/hbw_node	Next Task	FischerTechnik	/status_update (main)	Next Task	FischerTechnik	/provide_empty_loc, /rosout, /r	String	notify-callback	Status Updated Successfully	normal
1676302711	hbw	/hbw_node	Reference Switch	FischerTechnik	/status_update (main)	Next Task	FischerTechnik	/diagnostics, /hbw_pub, /new_	String	notify-callback	HBW Robot is Set to Home	normal
1676323913	sld	/sld_node	Motion Sensor	FischerTechnik	/vgr_node	Next Task	FischerTechnik	/diagnostics, /new_order, /prov	Boolean	notify-callback	VGR Received an Order at the BLUE Topic	normal
1676338913	vgr	/vgr_node	Next Task	FischerTechnik	/vgr_node	Motion Sensor	FischerTechnik	/diagnostics, /new_order, /prov	String	notify-callback	VGR Dropped Material at DPS	normal
1676345913	vgr	/vgr_node	Next Task	FischerTechnik	/vgr_node	Reference Switch	FischerTechnik	/diagnostics, /new_order, /prov	String	notify-callback	VGR is Set to Home	normal
1676345913	cobot	/listener (Cobot)	Motion Sensor	FischerTechnik	/status_update (main)	Next Task	Expansion	/lego_blue_pub, /lego_red_put	Boolean	notify-callback	Order is ready for picking	normal
1676346013	vgr	/vgr_node	Motion Sensor	FischerTechnik	/listener (Cobot)	Next Task	Expansion	/diagnostics, /new_order, /prov	String	notify-callback	Calling Cobot for Picking	normal
1676401013	lego	/lego_node	Motion Sensor	Expansion	/listener (Cobot)	Next Task	Expansion	/lego_blue_pub, /lego_red_put	Boolean	notify-callback	Blue order is ready for pick up	normal
1676401013	lego	/lego_node	Motion Sensor	Expansion	/listener (Cobot)	Next Task	Expansion	/diagnostics, /lego_blue_pub, /	String	notify-callback	Calling Cobot for blue Pick up	normal
1676418013	vgr	/vgr_node	Motion Sensor	FischerTechnik	/vgr_node	Next Task	FischerTechnik	/diagnostics, /new_order, /prov	Boolean	notify-callback	VGR is Picking Up Material from in-Processir	normal
1676429013	vgr	/vgr_node	Next Task	FischerTechnik	/vgr_node	Next Task	FischerTechnik	/diagnostics, /new_order, /prov	String	notify-callback	VGR is Moving to DPS Color Sensor	normal
1676434013	vgr	/vgr_node	Color Sensor	FischerTechnik	/status_update (main)	Next Task	FischerTechnik	/diagnostics, /new_order, /prov	String	notify-callback	DPS Detected Blue Object	normal
1676434213	main	/status_update (main)	Order	FischerTechnik	/vgr_node	Next Task	FischerTechnik	/provide_empty_loc, /provide_	String	read	Published Return Position	normal
1676437213	main	/Order	Next Task	FischerTechnik	/vgr_node	Next Task	FischerTechnik	/diagnostics, /new_order, /prov	String	notify-callback	VGR is Moving to HBW	normal
1676446213	hbw	/hbw_node	Reference Switch	FischerTechnik	/hbw_node	Next Task	FischerTechnik	/diagnostics, /hbw_pub, /new_	String	notify-callback	HBW Picked Empty Box from The Shelf	normal
1676452213	hbw	/hbw_node	Motion Sensor	FischerTechnik	/hbw_node	Next Task	FischerTechnik	/diagnostics, /hbw_pub, /new_	String	notify-callback	HBW Placed Box on the Belt	normal
1676452224	hbw	/hbw_node	Motion Sensor	FischerTechnik	/vgr_node	Next Task	FischerTechnik	/diagnostics, /new_order, /prov	Boolean	notify-callback	VGR Dropped Material at HBW	normal
1676458224	hbw	/hbw_node	Next Task	FischerTechnik	/hbw_node	Next Task	FischerTechnik	/diagnostics, /hbw_pub, /new_	String	notify-callback	HBW Material is Dropped by VGR	normal
1676458235	hbw	/hbw_node	Motion Sensor	FischerTechnik	/hbw_node	Next Task	FischerTechnik	/diagnostics, /hbw_pub, /new_	Boolean	notify-callback	Material is Ready for HBW Pick Up	normal
1676458735	hbw	/hbw_node	Motion Sensor	FischerTechnik	/hbw_node	Next Task	FischerTechnik	/diagnostics, /hbw_pub, /new_	Boolean	notify-callback	HBW Picked up Material from the Belt	normal
1676460435	vgr	/vgr_node	Next Task	FischerTechnik	/vgr_node	Reference Switch	FischerTechnik	/diagnostics, /new_order, /prov	String	notify-callback	VGR is Set to Home	normal
1676460935	hbw	/hbw_node	Next Task	FischerTechnik	/status_update (main)	Next Task	FischerTechnik	/diagnostics, /hbw_pub, /new_	String	notify-callback	HBW Placed Material on the Shelf	normal

Figure 37. Simulated Dataset (normal behavior).

1677636610	hbw	/hbw_node	Reference Switch	FischerTechnik	/status_update (main)	Next Task	FischerTechnik	/diagnostics, /hbw_pub, /new_	String	notify-callback	HBW Robot is Set to Home	normal
1677636610	main	/Order	Order	SCADA system	/hbw_node	Order	FischerTechnik	/new_order, /rosout	String	notify-callback	Published Order	normal
1677636680	main	/Order	Order	SCADA system	/vgr_node	Order	FischerTechnik	/diagnostics, /new_order, /prov	String	notify-callback	VGR got something	normal
1677636680	main	/Order	Next Task	FischerTechnik	/vgr_node	Next Task	FischerTechnik	/diagnostics, /new_order, /prov	String	notify-callback	VGR is Moving to HBW	normal
1677636880	main	/Order	Order	SCADA system	/hbw_node	Order	FischerTechnik	/diagnostics, /hbw_pub, /new_	String	notify-callback	HBW Recieved an Order	normal
1677639880	main	/Order	Order	SCADA system	/hbw_node	Order	FischerTechnik	/diagnostics, /hbw_pub, /new_	String	notify-callback	HBW Recieved an Order	anomalous(DoSattack)
1677640880	main	/Order	Order	SCADA system	/hbw_node	Order	FischerTechnik	/diagnostics, /hbw_pub, /new_	String	notify-callback	HBW Recieved an Order	anomalous(DoSattack)
1677641880	main	/Order	Order	SCADA system	/hbw_node	Order	FischerTechnik	/diagnostics, /hbw_pub, /new_	String	notify-callback	HBW Recieved an Order	anomalous(DoSattack)
1677642880	main	/Order	Order	SCADA system	/hbw_node	Order	FischerTechnik	/diagnostics, /hbw_pub, /new_	String	notify-callback	HBW Recieved an Order	anomalous(DoSattack)
1677643880	main	/Order	Order	SCADA system	/hbw_node	Order	FischerTechnik	/diagnostics, /hbw_pub, /new_	String	notify-callback	HBW Recieved an Order	anomalous(DoSattack)
1677644880	main	/Order	Order	SCADA system	/hbw_node	Order	FischerTechnik	/diagnostics, /hbw_pub, /new_	String	notify-callback	HBW Recieved an Order	anomalous(DoSattack)
1677645880	main	/Order	Order	SCADA system	/hbw_node	Order	FischerTechnik	/diagnostics, /hbw_pub, /new_	String	notify-callback	HBW Recieved an Order	anomalous(DoSattack)
1677646880	main	/Order	Order	SCADA system	/hbw_node	Order	FischerTechnik	/diagnostics, /hbw_pub, /new_	String	notify-callback	HBW Recieved an Order	anomalous(DoSattack)
1677647880	main	/Order	Order	SCADA system	/hbw_node	Order	FischerTechnik	/diagnostics, /hbw_pub, /new_	String	notify-callback	HBW Recieved an Order	anomalous(DoSattack)
1677648880	main	/Order	Order	SCADA system	/hbw_node	Order	FischerTechnik	/diagnostics, /hbw_pub, /new_	String	notify-callback	HBW Recieved an Order	anomalous(DoSattack)
1677649880	main	/Order	Order	SCADA system	/hbw_node	Order	FischerTechnik	/diagnostics, /hbw_pub, /new_	String	notify-callback	HBW Recieved an Order	anomalous(DoSattack)
1677650880	main	/Order	Order	SCADA system	/hbw_node	Order	FischerTechnik	/diagnostics, /hbw_pub, /new_	String	notify-callback	HBW Recieved an Order	anomalous(DoSattack)
1677651880	main	/Order	Order	SCADA system	/hbw_node	Order	FischerTechnik	/diagnostics, /hbw_pub, /new_	String	notify-callback	HBW Recieved an Order	anomalous(DoSattack)
1677652880	main	/Order	Order	SCADA system	/hbw_node	Order	FischerTechnik	/diagnostics, /hbw_pub, /new_	String	notify-callback	HBW Recieved an Order	anomalous(DoSattack)
1677653880	main	/Order	Order	SCADA system	/hbw_node	Order	FischerTechnik	/diagnostics, /hbw_pub, /new_	String	notify-callback	HBW Recieved an Order	anomalous(DoSattack)
1677654880	main	/Order	Order	SCADA system	/hbw_node	Order	FischerTechnik	/diagnostics, /hbw_pub, /new_	String	notify-callback	HBW Recieved an Order	anomalous(DoSattack)
1677655880	main	/Order	Order	SCADA system	/hbw_node	Order	FischerTechnik	/diagnostics, /hbw_pub, /new_	String	notify-callback	HBW Recieved an Order	anomalous(DoSattack)
1677656880	main	/Order	Order	SCADA system	/hbw_node	Order	FischerTechnik	/diagnostics, /hbw_pub, /new_	String	notify-callback	HBW Recieved an Order	anomalous(DoSattack)
1677657880	main	/Order	Order	SCADA system	/hbw_node	Order	FischerTechnik	/diagnostics, /hbw_pub, /new_	String	notify-callback	HBW Recieved an Order	anomalous(DoSattack)
1677658880	main	/Order	Order	SCADA system	/hbw_node	Order	FischerTechnik	/diagnostics, /hbw_pub, /new_	String	notify-callback	HBW Recieved an Order	anomalous(DoSattack)
1677659880	main	/Order	Order	SCADA system	/hbw_node	Order	FischerTechnik	/diagnostics, /hbw_pub, /new_	String	notify-callback	HBW Recieved an Order	anomalous(DoSattack)
1677660880	main	/Order	Order	SCADA system	/hbw_node	Order	FischerTechnik	/diagnostics, /hbw_pub, /new_	String	notify-callback	HBW Recieved an Order	anomalous(DoSattack)
1677661880	main	/Order	Order	SCADA system	/hbw_node	Order	FischerTechnik	/diagnostics, /hbw_pub, /new_	String	notify-callback	HBW Recieved an Order	anomalous(DoSattack)
1677662880	main	/Order	Order	SCADA system	/hbw_node	Order	FischerTechnik	/diagnostics, /hbw_pub, /new_	String	notify-callback	HBW Recieved an Order	anomalous(DoSattack)
1677663880	main	/Order	Order	SCADA system	/hbw_node	Order	FischerTechnik	/diagnostics, /hbw_pub, /new_	String	notify-callback	HBW Recieved an Order	anomalous(DoSattack)
1677664880	main	/Order	Order	SCADA system	/hbw_node	Order	FischerTechnik	/diagnostics, /hbw_pub, /new_	String	notify-callback	HBW Recieved an Order	anomalous(DoSattack)
1677665880	main	/Order	Order	SCADA system	/hbw_node	Order	FischerTechnik	/diagnostics, /hbw_pub, /new_	String	notify-callback	HBW Recieved an Order	anomalous(DoSattack)
1677666880	main	/Order	Order	SCADA system	/hbw_node	Order	FischerTechnik	/diagnostics, /hbw_pub, /new_	String	notify-callback	HBW Recieved an Order	anomalous(DoSattack)
1677667880	main	/Order	Order	SCADA system	/hbw_node	Order	FischerTechnik	/diagnostics, /hbw_pub, /new_	String	notify-callback	HBW Recieved an Order	anomalous(DoSattack)
1677668880	main	/Order	Order	SCADA system	/hbw_node	Order	FischerTechnik	/diagnostics, /hbw_pub, /new_	String	notify-callback	HBW Recieved an Order	anomalous(DoSattack)

Figure 38. Simulated Dataset (DoS attack).

3.5 The Smart Manufacturing Data set

As prior mentioned , one significant component of this study involved obtaining dependable simulation data that accurately portray real-world situations that may affect Industrial Internet of Things (IIoT) devices within the targeted system. As our investigation concentrates mainly on the communication that occurs between machines within a manufacturing system, we emulated the targeted system through a simulation environment created using MATLAB's Simulink software. The objective was to produce synthetic data that closely approximates the messaging output of the real system.

In light of the string-based messaging structure that the physical system proposed in [3] employs, we endeavored to replicate these patterns as closely as possible in our simulated system. To accomplish this objective, we leveraged Simulink's SimEvents software in our simulation environment to model the discrete events that transpire within the genuine system. When specific events occur within the simulated environment (such as when an entity crosses a light-gate sensor or traverses a server at the end of the process, indicating that the subsystem's process has concluded), all relevant data is transmitted into MATLAB's workspace, accompanied by the time at which it occurred and the message itself. The resulting data was then ready for further pre-processing, and eventual exportation into a .xlsx file.

Subsequent to the exportation of the simulated data from MATLAB's workspace, we imported the collection of data into a Python script. This script merged the data into a single sheet, with message data values corresponding to simulated timestamps. After creating a complete dataset, we translated the simulated time values into UNIX time and exported the information into a .csv file for utilization in the anomaly detection machine learning (ML) model described in Section 3.6

Our successfully developed dataset has been made publicly available on GitHub at the link: `https://github.com/MarwanAbdelatti/DSMS`. We cordially invite interested researchers to access and utilize our dataset to advance their research in this field. Our objective in making the dataset readily available is to facilitate the development and evaluation of novel ML algorithms and promote collaborative research. We believe that our dataset can serve as a valuable resource for the broader research community, and we encourage interested parties to take full advantage of this opportunity.

3.5.1 Description of the Data

The distributed smart manufacturing system **DSMS** dataset is an anomaly detection dataset designed for a lab-scale smart manufacturing system introduced in [3]. The dataset designed in this study is meant to be a benchmark for IIoT systems. The dataset comprises features from the application layer of the system that utilizes service IDs and node identifiers instead of IP addresses and protocol packets. It provides network flow information about the communication among all endpoints in the system, which includes numerous smart sensors, such as lighting sensors, motion detectors, and temperature sensors, distributed across 21 locations. The dataset encompasses 12 attribute features that characterize the current operation taking place in the system followed by a single feature that labels the anomalous nature of the events occurring:

- **sourceID:** a unique identifier assigned to each accessing service controller that reflects the hostname or machine name in the context of MQTT and robot operating system (ROS) protocols that are used in the system of study.
- **sourceAddress:** refers to the address from which an accessing service sends its requests. In the publish-subscribe communication system model on which

the dataset is based, this attribute is associated with the publishing node name.

- **sourceType:** defines the category of each registered service and determines its type in the system (e.g., "motion sensor").
- **sourceLocation:** refers to the physical location of the registered accessing service, such as a room number.
- **destinationServiceAddress:** refers to the address to which requests are sent and corresponds to the subscribing node name in a publish-subscribe communication system model.
- **destinationServiceType:** defines the type of the registered destination service.
- **destinationServiceLocation:** refers to the physical location of the registered destination service.
- **accessedNodeAddress:** refers to the topic name that holds the message being communicated between publishing and subscribing services.
- **accessedNodeType:** defines the type of message supported by the topic, which can be a standard type like "Boolean" or a custom type like "\lightController" (the same definition as the message type in the ROS context).
- **operation:** refers to the type of operation being performed, such as read/write, subscribe/unsubscribe, notify-callback, register/unregister service, and lock the entire subtree of nodes.

- **value:** refers to the value being read or written between services (e.g., temperature value).

- **timestamp:** refers to the time at which a particular service has access to another (e.g., a request to switch off the light).

- **normality:** refers to the attack status being applied to the system. When the system is exhibiting normal behavior, the data within this column reads "normal". Otherwise, if the system is experiencing an attack scenario, the data will read "anomalous_(*attack_type*)" as seen in the following sections of this article.

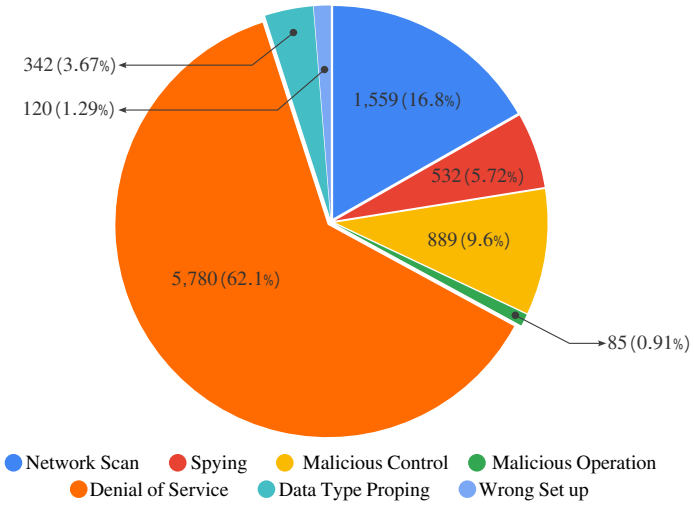


Figure 39. Distribution of anomalies in **DSMS**

The dataset comprises 357,952 record indices that encompass benign behaviors and is divided into seven attack scenarios, illustrated in Figure 39 [52]. The attack scenarios are delineated as follows:

3.5.2 Attack Scenarios

Net Scan

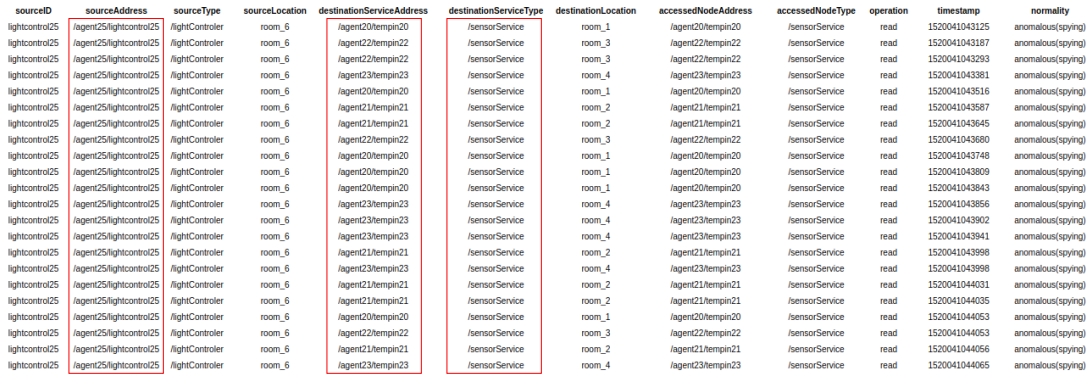
The present dataset demonstrates an attack scenario wherein the perpetrator conducts a scan of active services within the system. This type of attack is characterized by a significant number of requests emanating from a single source service to multiple destination services, as illustrated in Figure 40. In the context of smart manufacturing, it is necessary to implement a service, such as `/sld_node`, targets a variety of destination services at random, including `/status_update, /lego_node, /hbw_node, /listener(cobot node), and /vgr_node`..

sourceID	sourceAddress	sourceType	sourceLocation	destinationServiceAddress	destinationServiceType	destinationLocation	accessedNodeAddress	accessedNodeType	operation	timestamp	normality
washingmachine1	/agent6/washingmachine1	/washingService	Bathroom	/agent1/lightcontrol1	/lightControler	BedroomChildren	/agent1/lightcontrol1/lightOn	/derived/boolean	write	1520032371775	anomalous(scan)
washingmachine1	/agent6/washingmachine1	/washingService	Bathroom	/agent2/lightcontrol2	/lightControler	BedroomParents	/agent2/lightcontrol2/lightOn	/derived/boolean	write	1520032372093	anomalous(scan)
washingmachine1	/agent6/washingmachine1	/washingService	Bathroom	/agent2/tempin2	/sensorService	BedroomParents	/agent2/tempin2	/sensorService	read	1520032373151	anomalous(scan)
washingmachine1	/agent6/washingmachine1	/washingService	Bathroom	/agent3/tempin3	/sensorService	Dinningroom	/agent3/tempin3	/sensorService	read	1520032373984	anomalous(scan)
washingmachine1	/agent6/washingmachine1	/washingService	Bathroom	/agent4/tempin4	/sensorService	Kitchen	/agent4/tempin4	/sensorService	read	1520032374695	anomalous(scan)
washingmachine1	/agent6/washingmachine1	/washingService	Bathroom	/agent3/lightcontrol3	/lightControler	Dinningroom	/agent3/lightcontrol3/lightOn	/derived/boolean	write	1520032374980	anomalous(scan)
washingmachine1	/agent6/washingmachine1	/washingService	Bathroom	/agent4/lightcontrol4	/lightControler	Kitchen	/agent4/lightcontrol4/lightOn	/derived/boolean	write	1520032375402	anomalous(scan)
washingmachine1	/agent6/washingmachine1	/washingService	Bathroom	/agent5/tempin5	/sensorService	Garage	/agent5/tempin5	/sensorService	read	1520032375631	anomalous(scan)
washingmachine1	/agent6/washingmachine1	/washingService	Bathroom	/agent5/lightcontrol5	/lightControler	Garage	/agent5/lightcontrol5/lightOn	/derived/boolean	write	1520032376262	anomalous(scan)
washingmachine1	/agent6/washingmachine1	/washingService	Bathroom	/agent6/lightcontrol6	/lightControler	Bathroom	/agent6/lightcontrol6/lightOn	/derived/boolean	write	1520032376832	anomalous(scan)
washingmachine1	/agent6/washingmachine1	/washingService	Bathroom	/agent6/tempin6	/sensorService	Bathroom	/agent6/tempin6	/sensorService	read	1520032377750	anomalous(scan)
washingmachine1	/agent6/washingmachine1	/washingService	Bathroom	/agent1/lightcontrol1	/lightControler	BedroomChildren	/agent1/lightcontrol1/lightOn	/derived/boolean	write	1520032384024	anomalous(scan)
washingmachine1	/agent6/washingmachine1	/washingService	Bathroom	/agent2/lightcontrol2	/lightControler	BedroomParents	/agent2/lightcontrol2/lightOn	/derived/boolean	write	1520032384653	anomalous(scan)
washingmachine1	/agent6/washingmachine1	/washingService	Bathroom	/agent1/tempin1	/sensorService	BedroomChildren	/agent1/tempin1	/sensorService	read	1520032385397	anomalous(scan)
washingmachine1	/agent6/washingmachine1	/washingService	Bathroom	/agent2/tempin2	/sensorService	BedroomParents	/agent2/tempin2	/sensorService	read	1520032385688	anomalous(scan)
washingmachine1	/agent6/washingmachine1	/washingService	Bathroom	/agent3/tempin3	/sensorService	Dinningroom	/agent3/tempin3	/sensorService	read	1520032386417	anomalous(scan)
washingmachine1	/agent6/washingmachine1	/washingService	Bathroom	/agent4/lightcontrol4	/lightControler	Kitchen	/agent4/lightcontrol4/lightOn	/derived/boolean	write	1520032386853	anomalous(scan)
washingmachine1	/agent6/washingmachine1	/washingService	Bathroom	/agent3/lightcontrol3	/lightControler	Dinningroom	/agent3/lightcontrol3/lightOn	/derived/boolean	write	1520032386902	anomalous(scan)
washingmachine1	/agent6/washingmachine1	/washingService	Bathroom	/agent4/tempin4	/sensorService	Kitchen	/agent4/tempin4	/sensorService	read	1520032387204	anomalous(scan)
washingmachine1	/agent6/washingmachine1	/washingService	Bathroom	/agent5/lightcontrol5	/lightControler	Garage	/agent5/lightcontrol5/lightOn	/derived/boolean	write	1520032389069	anomalous(scan)
washingmachine1	/agent6/washingmachine1	/washingService	Bathroom	/agent5/tempin5	/sensorService	Garage	/agent5/tempin5	/sensorService	read	1520032389351	anomalous(scan)
washingmachine1	/agent6/washingmachine1	/washingService	Bathroom	/agent6/tempin6	/sensorService	Bathroom	/agent6/tempin6	/sensorService	read	1520032389526	anomalous(scan)

Figure 40. Network scan attack

Spying

The **DSMS** dataset exhibits a particular form of attack, in which the attacker attempts to extract operational values from active services. Unlike the scan attack, this type of offensive maneuver manifests itself in a large volume of *read* requests emanating from a single source service to multiple destination services of the same kind, as demonstrated in Figure 41. An illustration of an attack of this nature on the smart manufacturing system would involve the `/hbw_node`, an unassociated service with the SLD station, accessing the data of the `/sld_red_pub, /sld_white_pub, /sld_blue_pub, and /vgr_node` services, for instance.



sourceID	sourceAddress	sourceType	sourceLocation	destinationServiceAddress	destinationServiceType	destinationLocation	accessedNodeAddress	accessedNodeType	operation	timestamp	normality
lightcontrol25	/agent25/lightcontrol25	/lightControler	room_6	/agent20/tempin20	/sensorService	room_1	/agent20/tempin20	/sensorService	read	1520041043125	anomalous(spying)
lightcontrol25	/agent25/lightcontrol25	/lightControler	room_6	/agent22/tempin22	/sensorService	room_3	/agent22/tempin22	/sensorService	read	1520041043187	anomalous(spying)
lightcontrol25	/agent25/lightcontrol25	/lightControler	room_6	/agent22/tempin22	/sensorService	room_3	/agent22/tempin22	/sensorService	read	1520041043293	anomalous(spying)
lightcontrol25	/agent25/lightcontrol25	/lightControler	room_6	/agent23/tempin23	/sensorService	room_4	/agent23/tempin23	/sensorService	read	1520041043381	anomalous(spying)
lightcontrol25	/agent25/lightcontrol25	/lightControler	room_6	/agent20/tempin20	/sensorService	room_1	/agent20/tempin20	/sensorService	read	1520041043516	anomalous(spying)
lightcontrol25	/agent25/lightcontrol25	/lightControler	room_6	/agent21/tempin21	/sensorService	room_2	/agent21/tempin21	/sensorService	read	1520041043587	anomalous(spying)
lightcontrol25	/agent25/lightcontrol25	/lightControler	room_6	/agent21/tempin21	/sensorService	room_2	/agent21/tempin21	/sensorService	read	1520041043645	anomalous(spying)
lightcontrol25	/agent25/lightcontrol25	/lightControler	room_6	/agent22/tempin22	/sensorService	room_3	/agent22/tempin22	/sensorService	read	1520041043680	anomalous(spying)
lightcontrol25	/agent25/lightcontrol25	/lightControler	room_6	/agent20/tempin20	/sensorService	room_1	/agent20/tempin20	/sensorService	read	1520041043748	anomalous(spying)
lightcontrol25	/agent25/lightcontrol25	/lightControler	room_6	/agent20/tempin20	/sensorService	room_1	/agent20/tempin20	/sensorService	read	1520041043809	anomalous(spying)
lightcontrol25	/agent25/lightcontrol25	/lightControler	room_6	/agent20/tempin20	/sensorService	room_1	/agent20/tempin20	/sensorService	read	1520041043843	anomalous(spying)
lightcontrol25	/agent25/lightcontrol25	/lightControler	room_6	/agent23/tempin23	/sensorService	room_4	/agent23/tempin23	/sensorService	read	1520041043856	anomalous(spying)
lightcontrol25	/agent25/lightcontrol25	/lightControler	room_6	/agent23/tempin23	/sensorService	room_4	/agent23/tempin23	/sensorService	read	1520041043902	anomalous(spying)
lightcontrol25	/agent25/lightcontrol25	/lightControler	room_6	/agent23/tempin23	/sensorService	room_4	/agent23/tempin23	/sensorService	read	1520041043941	anomalous(spying)
lightcontrol25	/agent25/lightcontrol25	/lightControler	room_6	/agent21/tempin21	/sensorService	room_2	/agent21/tempin21	/sensorService	read	1520041043998	anomalous(spying)
lightcontrol25	/agent25/lightcontrol25	/lightControler	room_6	/agent23/tempin23	/sensorService	room_4	/agent23/tempin23	/sensorService	read	1520041043998	anomalous(spying)
lightcontrol25	/agent25/lightcontrol25	/lightControler	room_6	/agent21/tempin21	/sensorService	room_2	/agent21/tempin21	/sensorService	read	1520041044031	anomalous(spying)
lightcontrol25	/agent25/lightcontrol25	/lightControler	room_6	/agent21/tempin21	/sensorService	room_2	/agent21/tempin21	/sensorService	read	1520041044035	anomalous(spying)
lightcontrol25	/agent25/lightcontrol25	/lightControler	room_6	/agent20/tempin20	/sensorService	room_1	/agent20/tempin20	/sensorService	read	1520041044053	anomalous(spying)
lightcontrol25	/agent25/lightcontrol25	/lightControler	room_6	/agent22/tempin22	/sensorService	room_3	/agent22/tempin22	/sensorService	read	1520041044053	anomalous(spying)
lightcontrol25	/agent25/lightcontrol25	/lightControler	room_6	/agent21/tempin21	/sensorService	room_2	/agent21/tempin21	/sensorService	read	1520041044056	anomalous(spying)
lightcontrol25	/agent25/lightcontrol25	/lightControler	room_6	/agent23/tempin23	/sensorService	room_4	/agent23/tempin23	/sensorService	read	1520041044065	anomalous(spying)

Figure 41. Spying attack

Data Types Probing

This type of attack involves sending data of a different type than the one assigned to the target destination service. Unlike the previous two types of attacks, this attack involves a single destination service accessed by a single source for *write* operations. The attack involves writing data of a type different from the original destination service type, as depicted in Figure 42, where the normal behavior involves data of types `/basic/text` or `/basic/number`, while the attack involves data of another different type, namely, `/basic/composed`. For instance, the smart manufacturing system could be a victim of this attack when the `/place_order` service is found to write a numerical value or a random string to the `/new_order` service instead of the pre-set values like "red", "white", and "blue".

sourceID	sourceAddress	sourceType	sourceLocation	destinationServiceAddress	destinationServiceType	destinationLocation	accessedNodeAddress	accessedNodeType	operation	value	timestamp	normality
washingmachine2	/agent11/washingmachine2	/washingService	Watterroom	/agent11/battery4	/batteryService	Watterroom	/agent11/battery4/charging	/basic/text	read	true	1520080046007	normal
washingmachine2	/agent11/washingmachine2	/washingService	Watterroom	/agent11/battery4	/batteryService	Watterroom	/agent11/battery4/charge	/basic/composed	write		1520080047708	anomalous(dataProbing)
washingmachine2	/agent11/washingmachine2	/washingService	Watterroom	/agent11/battery4	/batteryService	Watterroom	/agent11/battery4/charge	/basic/number	read	9	1520080066681	normal
washingmachine2	/agent11/washingmachine2	/washingService	Watterroom	/agent11/battery4	/batteryService	Watterroom	/agent11/battery4/charging	/basic/text	read	true	1520080066732	normal
washingmachine2	/agent11/washingmachine2	/washingService	Watterroom	/agent11/battery4	/batteryService	Watterroom	/agent11/battery4/charge	/basic/composed	write		1520080077942	anomalous(dataProbing)
washingmachine2	/agent11/washingmachine2	/washingService	Watterroom	/agent11/battery4	/batteryService	Watterroom	/agent11/battery4/charge	/basic/number	read	10	1520080087396	normal
washingmachine2	/agent11/washingmachine2	/washingService	Watterroom	/agent11/battery4	/batteryService	Watterroom	/agent11/battery4/charging	/basic/text	read	false	1520080087447	normal
washingmachine2	/agent11/washingmachine2	/washingService	Watterroom	/agent11/battery4	/batteryService	Watterroom	/agent11/battery4/charge	/basic/number	read	10	1520080087509	normal
washingmachine2	/agent11/washingmachine2	/washingService	Watterroom	/agent11/battery4	/batteryService	Watterroom	/agent11/battery4/charge	/basic/number	write	9	1520080087550	normal
washingmachine2	/agent11/washingmachine2	/washingService	Watterroom	/agent11/battery4	/batteryService	Watterroom	/agent11/battery4/charge	/basic/number	read	9	1520080087642	normal
washingmachine2	/agent11/washingmachine2	/washingService	Watterroom	/agent11/battery4	/batteryService	Watterroom	/agent11/battery4/charge	/basic/composed	write		1520080108054	anomalous(dataProbing)
washingmachine2	/agent11/washingmachine2	/washingService	Watterroom	/agent11/battery4	/batteryService	Watterroom	/agent11/battery4/charging	/basic/text	read	false	1520080108193	normal
washingmachine2	/agent11/washingmachine2	/washingService	Watterroom	/agent11/battery4	/batteryService	Watterroom	/agent11/battery4/charge	/basic/number	read	9	1520080108244	normal
washingmachine2	/agent11/washingmachine2	/washingService	Watterroom	/agent11/battery4	/batteryService	Watterroom	/agent11/battery4/charge	/basic/number	write	8	1520080108255	normal
washingmachine2	/agent11/washingmachine2	/washingService	Watterroom	/agent11/battery4	/batteryService	Watterroom	/agent11/battery4/charge	/basic/number	read	8	1520080108367	normal
washingmachine2	/agent11/washingmachine2	/washingService	Watterroom	/agent11/battery4	/batteryService	Watterroom	/agent11/battery4/charging	/basic/text	read	false	1520080129093	normal
washingmachine2	/agent11/washingmachine2	/washingService	Watterroom	/agent11/battery4	/batteryService	Watterroom	/agent11/battery4/charge	/basic/number	read	8	1520080129441	normal
washingmachine2	/agent11/washingmachine2	/washingService	Watterroom	/agent11/battery4	/batteryService	Watterroom	/agent11/battery4/charge	/basic/number	write	7	1520080129687	normal
washingmachine2	/agent11/washingmachine2	/washingService	Watterroom	/agent11/battery4	/batteryService	Watterroom	/agent11/battery4/charge	/basic/number	read	7	1520080129923	normal
washingmachine2	/agent11/washingmachine2	/washingService	Watterroom	/agent11/battery4	/batteryService	Watterroom	/agent11/battery4/charge	/basic/composed	write		1520080138064	anomalous(dataProbing)
washingmachine2	/agent11/washingmachine2	/washingService	Watterroom	/agent11/battery4	/batteryService	Watterroom	/agent11/battery4/charging	/basic/text	read	false	1520080150474	normal
washingmachine2	/agent11/washingmachine2	/washingService	Watterroom	/agent11/battery4	/batteryService	Watterroom	/agent11/battery4/charge	/basic/number	read	7	1520080150525	normal

Figure 42. Data Types Probing (DTP) attack

Malicious Operation

This attack occurs when a source service attempts to access a destination service with an improper operation type, such as trying to write to a sensor service that is only meant for reading. Figure 43 illustrates an example where the `/lightcontrol` service attempts to write values to the `/movementSensor` service. In the smart factory, a malicious operation may occur when a service tries to subscribe (i.e., read) to the `/place_order` topic, which is only meant for writing to other services to initiate a new order process.

sourceID	sourceAddress	sourceType	sourceLocation	destinationServiceAddress	destinationServiceType	destinationLocation	accessedNodeAddress	accessedNodeType	operation	value	timestamp	normality
lightcontrol14	/agent14/lightcontrol14	/lightControler	Showerroom	/agent14/movement14	/movementSensor	Showerroom	/agent14/movement14/movement	/derived/boolean	read	0	1520039030354	normal
lightcontrol14	/agent14/lightcontrol14	/lightControler	Showerroom	/agent14/movement14	/movementSensor	Showerroom	/agent14/movement14/movement	/derived/boolean	read	0	1520039051459	normal
lightcontrol14	/agent14/lightcontrol14	/lightControler	Showerroom	/agent14/movement14	/movementSensor	Showerroom	/agent14/movement14/movement	/derived/boolean	read	0	1520039072287	normal
lightcontrol14	/agent14/lightcontrol14	/lightControler	Showerroom	/agent14/movement14	/movementSensor	Showerroom	/agent14/movement14/movement	/derived/boolean	read	0	1520039092941	normal
lightcontrol14	/agent14/lightcontrol14	/lightControler	Showerroom	/agent14/movement14	/movementSensor	Showerroom	/agent14/movement14/movement	/derived/boolean	write	1	1520039093554	anomalous(malitiousOperation)
lightcontrol14	/agent14/lightcontrol14	/lightControler	Showerroom	/agent14/movement14	/movementSensor	Showerroom	/agent14/movement14/movement	/derived/boolean	write	1	1520039111486	anomalous(malitiousOperation)
lightcontrol14	/agent14/lightcontrol14	/lightControler	Showerroom	/agent14/movement14	/movementSensor	Showerroom	/agent14/movement14/movement	/derived/boolean	read	0	1520039113574	normal
lightcontrol14	/agent14/lightcontrol14	/lightControler	Showerroom	/agent14/movement14	/movementSensor	Showerroom	/agent14/movement14/movement	/derived/boolean	write	1	1520039130256	anomalous(malitiousOperation)
lightcontrol14	/agent14/lightcontrol14	/lightControler	Showerroom	/agent14/movement14	/movementSensor	Showerroom	/agent14/movement14/movement	/derived/boolean	read	0	1520039134320	normal
lightcontrol14	/agent14/lightcontrol14	/lightControler	Showerroom	/agent14/movement14	/movementSensor	Showerroom	/agent14/movement14/movement	/derived/boolean	write	1	1520039148077	anomalous(malitiousOperation)
lightcontrol14	/agent14/lightcontrol14	/lightControler	Showerroom	/agent14/movement14	/movementSensor	Showerroom	/agent14/movement14/movement	/derived/boolean	read	0	1520039154912	normal

Figure 43. Malicious operation attack

Wrong Setup

This type of attack involves an attempt by a source service to access a destination service in an incorrect location. An instance of this attack is illustrated in Figure 44, where the `/lightcontrol` service in the parents' room is attempting to read sensory data from the children's room. In the context of a smart factory, this scenario can occur when the `/new_order` service attempts to access empty locations on the high-bay warehouse station through the `/what_is_empty` topic.

sourceID	sourceAddress	sourceType	sourceLocation	destinationServiceAddress	destinationServiceType	destinationLocation	accessedNodeAddress	accessedNodeType	operation	value	timestamp	normality
lightcontrol11	/agent11/lightcontrol11	/lightControler	Watterroom	/agent11/movement11	/movementSensor	Watterroom	/agent11/movement11/movement	/derived/boolean	read	0	1520047536135	normal
lightcontrol1	/agent1/lightcontrol1	/lightControler	BedroomChildren	/agent1/movement1	/movementSensor	BedroomChildren	/agent1/movement1/movement	/derived/boolean	read	0	1520047536279	normal
lightcontrol27	/agent27/lightcontrol27	/lightControler	room_8	/agent27/movement27	/movementSensor	room_8	/agent27/movement27/movement	/derived/boolean	read	1	1520047536597	normal
lightcontrol21	/agent21/lightcontrol21	/lightControler	room_2	/agent21/movement21	/movementSensor	room_2	/agent21/movement21/movement	/derived/boolean	read	0	1520047536648	normal
lightcontrol3	/agent3/lightcontrol3	/lightControler	Dinningroom	/agent3/movement3	/movementSensor	Dinningroom	/agent3/movement3/movement	/derived/boolean	read	0	1520047536996	normal
lightcontrol23	/agent23/lightcontrol23	/lightControler	room_4	/agent23/movement23	/movementSensor	room_4	/agent23/movement23/movement	/derived/boolean	read	1	1520047538001	normal
lightcontrol6	/agent6/lightcontrol6	/lightControler	Bathroom	/agent6/movement6	/movementSensor	Bathroom	/agent6/movement6/movement	/derived/boolean	read	1	1520047538636	normal
lightcontrol7	/agent2/lightcontrol7	/lightControler	BedroomParents	/agent1/movement1	/movementSensor	BedroomChildren	/agent1/movement1/movement	/derived/boolean	read	1	1520047540938	anomalous(wrongSetUp)
lightcontrol4	/agent4/lightcontrol4	/lightControler	Kitchen	/agent4/movement4	/movementSensor	Kitchen	/agent4/movement4/movement	/derived/boolean	read	0	1520047543403	normal
lightcontrol25	/agent25/lightcontrol25	/lightControler	room_6	/agent25/movement25	/movementSensor	room_6	/agent25/movement25/movement	/derived/boolean	read	1	1520047544715	normal
lightcontrol10	/agent10/lightcontrol10	/lightControler	Livingroom	/agent10/movement10	/movementSensor	Livingroom	/agent10/movement10/movement	/derived/boolean	read	1	1520047544817	normal
lightcontrol24	/agent24/lightcontrol24	/lightControler	room_5	/agent24/movement24	/movementSensor	room_5	/agent24/movement24/movement	/derived/boolean	read	0	1520047544920	normal

Figure 44. Wrong set up attack

Malicious Control

This type of attack occurs when a source service sends a request to an irrelevant destination service. The malicious control attack type is a broader version of the wrong setup attack, where the irrelevance is not necessarily related to the source/destination location. An example of this type of attack is when a `/washingmachine` service attempts to open the `/entranceDoor`, as depicted in Figure 45. In a smart factory setting, this scenario could occur when the `/vgr_node` attempts to activate the Cobot unit.

sourceID	sourceAddress	sourceType	sourceLocation	destinationServiceAddress	destinationServiceType	destinationLocation	accessedNodeAddress	accessedNodeType	operation	value	timestamp	normality
tempin13	/agent13/tempin13	/sensorService	Bedroom	/agent13/tempin13	/sensorService	Bedroom	/agent13/tempin13	/sensorService	write	20.0421	1520112672379	normal
tempin14	/agent14/tempin14	/sensorService	Showerroom	/agent14/tempin14	/sensorService	Showerroom	/agent14/tempin14	/sensorService	write	20.4965	1520112675238	normal
tempin13	/agent13/tempin13	/sensorService	Bedroom	/agent13/tempin13	/sensorService	Bedroom	/agent13/tempin13	/sensorService	write	19.6981	1520112693258	normal
tempin14	/agent14/tempin14	/sensorService	Showerroom	/agent14/tempin14	/sensorService	Showerroom	/agent14/tempin14	/sensorService	write	19.6647	1520112695964	normal
tempin13	/agent13/tempin13	/sensorService	Bedroom	/agent13/tempin13	/sensorService	Bedroom	/agent13/tempin13	/sensorService	write	20.3708	1520112714506	normal
tempin14	/agent14/tempin14	/sensorService	Showerroom	/agent14/tempin14	/sensorService	Showerroom	/agent14/tempin14	/sensorService	write	19.8126	1520112716546	normal
tempin13	/agent13/tempin13	/sensorService	Bedroom	/agent12/doorlock3	/doorLockService	Entrance	/agent12/doorlock3/open	/basic/text	write	true	1520112730129	anomalous(malitiousControl)
tempin14	/agent14/tempin14	/sensorService	Showerroom	/agent12/doorlock3	/doorLockService	Entrance	/agent12/doorlock3/open	/basic/text	write	true	1520112730170	anomalous(malitiousControl)
tempin13	/agent13/tempin13	/sensorService	Bedroom	/agent13/tempin13	/sensorService	Bedroom	/agent13/tempin13	/sensorService	write	20.2831	1520112735693	normal
tempin14	/agent14/tempin14	/sensorService	Showerroom	/agent14/tempin14	/sensorService	Showerroom	/agent14/tempin14	/sensorService	write	20.3814	1520112737128	normal
tempin13	/agent13/tempin13	/sensorService	Bedroom	/agent13/tempin13	/sensorService	Bedroom	/agent13/tempin13	/sensorService	write	20.1684	1520112757105	normal
tempin14	/agent14/tempin14	/sensorService	Showerroom	/agent14/tempin14	/sensorService	Showerroom	/agent14/tempin14	/sensorService	write	19.5005	1520112758048	normal
tempin13	/agent13/tempin13	/sensorService	Bedroom	/agent13/tempin13	/sensorService	Bedroom	/agent13/tempin13	/sensorService	write	20.2968	1520112777636	normal
tempin14	/agent14/tempin14	/sensorService	Showerroom	/agent14/tempin14	/sensorService	Showerroom	/agent14/tempin14	/sensorService	write	20.4676	1520112780372	normal

Figure 45. Malicious control attack

Denial of Service

A Denial-of-Service (DoS) attack is characterized by a single source service inundating a target destination service with an overwhelming volume of requests, rendering the latter unable to respond effectively. Figure 46 illustrates a representative instance of a DoS from the dataset, wherein the timestamp column exhibits an extremely high-frequency rate of requests, occurring nearly every second. A DoS attack may target the smart manufacturing system when an excessive number of requests are transmitted from the `/sld_node`, for example, to `/sld_red_pub`. Such an attack has the potential to inflict damage on the pneumatic valves, leading to a catastrophic halt of the entire system.

sourceID	sourceAddress	sourceType	sourceLocation	destinationServiceAddress	destinationServiceType	destinationLocation	accessedNodeAddress	accessedNodeType	operation	timestamp	normality
lightcontrol1	/agent1/lightcontrol1	/lightControler	BedroomChildren	/agent1/movement1	/movementSensor	BedroomChildren	'agent1/movement1/movement	/derived/boolean	read	1520040030051	anomalous(DoSattack)
lightcontrol1	/agent1/lightcontrol1	/lightControler	BedroomChildren	/agent1/movement1	/movementSensor	BedroomChildren	'agent1/movement1/movement	/derived/boolean	read	1520040030052	anomalous(DoSattack)
lightcontrol1	/agent1/lightcontrol1	/lightControler	BedroomChildren	/agent1/movement1	/movementSensor	BedroomChildren	'agent1/movement1/movement	/derived/boolean	read	1520040030053	anomalous(DoSattack)
lightcontrol1	/agent1/lightcontrol1	/lightControler	BedroomChildren	/agent1/movement1	/movementSensor	BedroomChildren	'agent1/movement1/movement	/derived/boolean	read	1520040030054	anomalous(DoSattack)
lightcontrol1	/agent1/lightcontrol1	/lightControler	BedroomChildren	/agent1/movement1	/movementSensor	BedroomChildren	'agent1/movement1/movement	/derived/boolean	read	1520040030056	anomalous(DoSattack)
lightcontrol1	/agent1/lightcontrol1	/lightControler	BedroomChildren	/agent1/movement1	/movementSensor	BedroomChildren	'agent1/movement1/movement	/derived/boolean	read	1520040030057	anomalous(DoSattack)
lightcontrol1	/agent1/lightcontrol1	/lightControler	BedroomChildren	/agent1/movement1	/movementSensor	BedroomChildren	'agent1/movement1/movement	/derived/boolean	read	1520040030059	anomalous(DoSattack)
lightcontrol1	/agent1/lightcontrol1	/lightControler	BedroomChildren	/agent1/movement1	/movementSensor	BedroomChildren	'agent1/movement1/movement	/derived/boolean	read	1520040030059	anomalous(DoSattack)
lightcontrol1	/agent1/lightcontrol1	/lightControler	BedroomChildren	/agent1/movement1	/movementSensor	BedroomChildren	'agent1/movement1/movement	/derived/boolean	read	1520040030059	anomalous(DoSattack)
lightcontrol1	/agent1/lightcontrol1	/lightControler	BedroomChildren	/agent1/movement1	/movementSensor	BedroomChildren	'agent1/movement1/movement	/derived/boolean	read	1520040030060	anomalous(DoSattack)
lightcontrol1	/agent1/lightcontrol1	/lightControler	BedroomChildren	/agent1/movement1	/movementSensor	BedroomChildren	'agent1/movement1/movement	/derived/boolean	read	1520040030060	anomalous(DoSattack)
lightcontrol1	/agent1/lightcontrol1	/lightControler	BedroomChildren	/agent1/movement1	/movementSensor	BedroomChildren	'agent1/movement1/movement	/derived/boolean	read	1520040030063	anomalous(DoSattack)
lightcontrol1	/agent1/lightcontrol1	/lightControler	BedroomChildren	/agent1/movement1	/movementSensor	BedroomChildren	'agent1/movement1/movement	/derived/boolean	read	1520040030064	anomalous(DoSattack)
lightcontrol1	/agent1/lightcontrol1	/lightControler	BedroomChildren	/agent1/movement1	/movementSensor	BedroomChildren	'agent1/movement1/movement	/derived/boolean	read	1520040030065	anomalous(DoSattack)

Figure 46. Denial of service (DoS) attack

3.5.3 Data Visualization

In order to acquire further insights into the dataset, we opt to reduce its dimensionality to enable plotting. To achieve this, we utilize a principal component analysis (PCA) procedure that is based on singular value decomposition (SVD) to project the data onto a lower dimensional space. We categorize the features into three groups and visualize them in a three-dimensional scatter plot, as displayed in Figure 47. In the plot, each color corresponds to a particular type of attack, as indicated by the label column in the dataset. The visualization explicitly demonstrates that the data is naturally separable, indicating that machine learning models can swiftly identify anomalies and achieve high evaluation scores unless overfitting occurs (as exhibited in Section 3.7).

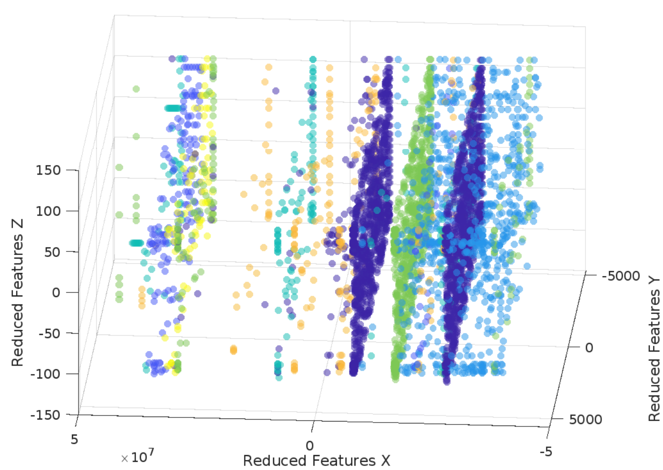


Figure 47. Visualization of data

3.5.4 Data Pre-Processing

The dataset consists of twelve columns that include features such as source and destination IDs, addresses, locations, data types, and a column designating the attack type. These features are predominantly strings, with the exception of the "timestamp" column. Several of the features contain missing data, including 2,050 empty entries in the "value" feature and 148 in the "accessedNodeType" feature. These features underwent necessary adjustments as described in subsequent sections of this paper.

To establish a pristine test set at this early stage, we employed stratified sampling, which entails dividing the data into homogeneous subgroups, or "strata", and drawing a specified number of instances from each subgroup to ensure that the test set accurately represents the entire population. In this study, we performed stratified sampling to extract 20% of the overall data based on the attack type. The code used to execute this stratification process and the corresponding percentages of each category in the test set and the original dataset are presented in Figure 48. The percentages demonstrate that each category is represented with precision in the test set, reflecting ratios similar to those in the initial dataset.

```
from sklearn.model_selection import StratifiedShuffleSplit

split = StratifiedShuffleSplit(n_splits=1, test_size=0.2, random_state=1)
for train_index, test_index in split.split(encoded_text,
                                           encoded_text[normality_col_head]):
  strat_train_set = encoded_text.iloc[train_index]
  strat_test_set = encoded_text.iloc[test_index]

# Convert 'Timeline' header to int value
strat_train_set.rename({'Time': 11}, axis=1, inplace=True)
strat_test_set.rename({'Time': 11}, axis=1, inplace=True)

print('Frequency of each category in the test data:\n{}'.format(
    strat_test_set[normality_col_head].value_counts()/len(strat_test_set)))
print('\nFrequency of each category in the original data:\n{}'.format(
    encoded_text[normality_col_head].value_counts()/len(encoded_text)))

strat_train_set.describe()
```

```
Frequency of each category in the test data:
0    0.476184
7    0.475129
5    0.029687
4    0.009500
6    0.002375
3    0.002375
2    0.002375
1    0.002375
Name: Normality, dtype: float64

Frequency of each category in the original data:
0    0.476175
7    0.475134
5    0.029688
4    0.009500
6    0.002378
2    0.002375
1    0.002375
3    0.002375
Name: Normality, dtype: float64
```

Figure 48. Stratification of data

The majority of machine learning algorithms exhibit suboptimal performance when confronted with missing features. Given that our dataset is primarily comprised of string types, utilizing the average or median to fill in empty values is not feasible. As a result, we elect to remove data rows that contain missing entries. To further enhance the accuracy of machine learning models, we employ "ordinal-encoding" to transform all text data in the features into numerical values. This method assigns a unique number to each text element and constructs a list of unique text entries for each feature, with each element represented by an integer [53]. The class labels are also encoded using eight numbers, ranging from 0 to 7, to represent the seven potential attacks and normal conditions.

Upon statistical analysis of the encoded attributes, we observe varying scales among the class labels. Table 3 displays the statistics of each attribute, including the number of records, mean, standard deviation, minimum and maximum values, and the three quartiles. Our findings reveal that some attribute values fall within the range of 0 to 10 or less, while others range in the hundreds or thousands. This inconsistency in the numeric data has the potential to significantly impede the performance of machine learning models. To mitigate this issue, we apply "min-max" normalization to each feature to ensure that each feature has a range between 0 and 1. The following equation is utilized to achieve this goal:

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{1}$$

where x is the original value, x' is the resulted normalization, and x_{min} and x_{max} are the minimum and maximum values, respectively [53].

Table 3. Attribute statistics before scaling

	mean	std	min	25%	50%	75%	max
sourceID	38.19	29.98	0	15	29	71	83
sourceAddress	45.14	27.79	0	16	45	73	88
sourceType	2.74	2.21	0	2	2	4	7
sourceLocation	8.61	5.51	0	5	8	12	20
destination Service Address	42.00	26.58	0	15	42	71	83
destination Service Type	2.27	1.77	0	0	3	4	7
destination Service Location	8.69	5.27	0	5	7	12	20
accessedNode Address	84.35	52.68	0	31	84	140	166
accessedNode Type	3.48	2.89	0	0	3	7	10
operation	0.92	1.38	0	0	0	3	3
value	5,168	4,482	0	2	5,187	10,594	10,622
normality	6.88	0.84	0	7	7	7	7

3.5.5 Cross-Validation of Data

To enhance the accuracy of our models and prevent overfitting, we partition our training data into sections, or "folds", whereby some sections are used to train the models and others to evaluate them. This technique, referred to as "cross-validation", is employed iteratively until the entire training set has been utilized. The primary purpose of cross-validation is to evaluate the performance of the ML models and, hence, improve their accuracy [53]. In our study, we utilize the "k-fold" cross-validation algorithm, which involves randomly shuffling the training set and dividing it into $k = 5$ folds. We used five folds to balance computational cost and variance reduction. It was found that using more folds (e.g., 10-fold cross-validation) slightly reduced the variance of the estimate, but it significantly increased the computational cost of running multiple iterations.

3.6 Training Models

In our study on detecting and classifying IIoT attacks using the **DSMS** dataset, we evaluate the performance of various machine learning models using metrics such as precision, recall, and F1-score. While accuracy is a commonly used metric for evaluating machine learning models, it may not be ideal for classifiers, especially when there is an imbalance in the class distribution (such as the normal behavior versus other attacks in our dataset) [53]. This is because accuracy is computed as the ratio of correctly classified samples (i.e., "True Positive" and "True Negative") to the total number of samples. In the case where TP, TN, FP, and FN represent the number of "True Positive", "True Negative", "False Positive", and "False Negative" classifications, respectively, the accuracy metric is given by:

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2}$$

Precision is a metric that represents the accuracy of the positive predictions made by a classification model. It is defined as the ratio of true positive (TP) classifications to the total number of positive predictions, as expressed by the following equation:

$$precision = \frac{TP}{TP + FP} \tag{3}$$

The recall, also known as sensitivity or true-positive rate (TPR), is a metric that measures the ability of a classifier to identify all positive instances. It is the ratio of "True Positive" classifications to the total number of actual positives. The recall can be computed using the following equation:

$$recall = \frac{TP}{TP + FN} \tag{4}$$

The F1-score is a commonly used metric to evaluate the overall performance of a classifier, as it represents the harmonic mean of the precision and the recall. A high F1-score can only be achieved when both precision and recall are high. It is computed using the following formula:

$$F1 = 2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{5}$$

While precision may provide more specificity than accuracy, it neglects the "False Negative" (FN) attacks, which are critical in evaluating the performance of a classifier in systems like ours. When there is a trade-off between precision and recall, recall is generally more desirable because it considers FN, even if it results in an increased number of "False Positive" (FP) alarms.

3.6.1 ML Models

We utilize the following machine learning models:

K-Nearest Neighbors

The k-nearest neighbor (kNN) algorithm is a supervised classification machine learning algorithm that leverages labeled training data to predict the class labels of a test set based on the labels of its k nearest neighbors [54]. The nearest neighbors are determined by a distance metric, such as Euclidean distance, and the number of nearest neighbors (k) is a pre-defined hyperparameter.

Decision trees

The decision tree (DT) algorithm builds a tree model based on the given attribute features. It splits a basic node into multiple sub-nodes based on correlations between the features until it reaches the class labels (leaves). This split operation continues until specific criteria, such as maximum tree depth, are met.

Random Forest

The random forest (RF) algorithm is an ensemble meta-algorithm that involves multiple decision trees. The RF algorithm trains its model on different subsets of the features for n training iterations to build n decision trees, which helps to prevent overfitting.

Artificial Neural Network

Artificial neural networks (ANNs) are the fundamental component of deep learning algorithms that are used in complex machine learning tasks, such as computer vision (e.g., Google Images) and speech recognition services (e.g., Apple's Siri). The multi-layer perceptron (MLP) is the simplest form of an ANN that learns an approximation function that maps between input features and output labels. The MLP can have one or more nonlinear layers between the input and output layers, which are called hidden layers.

3.7 Experimental Results

The machine learning algorithms outlined in Section 3.6 are trained using pre-processed features that describe specific system properties by which behavior is classified as normal or abnormal, with seven attack types. The training data contains classification labels of the system behavior for the models to learn from. As detailed in Section 3.5.4, the training set is obtained by separating the testing set from the data. The cross-validation step, discussed in Section 3.5.5, divides the training set into five folds to train and validate the models on each. Once the models are trained, the normalized test set, based on eq. (1), is evaluated using the trained models. Table 4 reports the evaluation results in both the training and testing stages. The first five rows represent the precision, recall, F1 score, accuracy, and execution speed during the training process, while the following five

Table 4. Evaluation metrics and speeds

	Model	ANN	RF	DT	kNN
Training	**Precision**	0.9968	0.9996	0.9996	0.9992
	Recall	0.9967	0.9996	0.9985	0.9992
	F1	0.9967	0.9996	0.9996	0.9992
	Accuracy	0.9874	0.9984	0.9984	0.9968
	Speed (sec)	31.313	48.398	1.93	0.888
Testing	**Precision**	0.9969	0.9419	0.9923	0.9551
	Recall	0.9968	0.8930	0.8930	0.9312
	F1	0.9968	0.8690	0.8700	0.9255
	Accuracy	0.9874	0.9984	0.9984	0.9968

rows represent their counterparts during the testing stage. Columns 3–7 represent the five utilized machine learning algorithms.

The data presented in the table indicate that the algorithms exhibit similar accuracy values while presenting varying precision, recall, and F1 metrics, particularly during the testing phase. These findings support the argument presented in Section 3.6 that accuracy is not a precise metric for our specific application. While the remaining algorithms can achieve rapid anomaly detections (approx. 1 second on the test set), they display significant disparities in their training times. Specifically, the random forest (RF) and artificial neural networks (ANN) algorithms require extensive training periods of 48 and 31 seconds, respectively, rendering them less desirable for applications demanding prompt training, despite their high recall scores. In contrast, the decision tree (DT) and k-nearest neighbor (kNN) algorithms deliver outstanding performance, with kNN demonstrating high accuracy metric scores.

3.8 Conclusions and Future Work

Industrial Internet of Things (IIoT) systems are composed of smart devices, including sensors and controllers, with the aim of enhancing system performance

and optimizing industrial processes. These devices communicate with each other continuously, exchanging large volumes of data. While this provides numerous benefits, it also creates vulnerabilities to severe security attacks due to the widespread communication and potential inconsistency between devices within the same network. Therefore, to operate safely and avoid service disruptions, IIoT systems require anomaly detection systems. In terms of future work on the simulation side of this project, some additional features could be added in order to bring a further depth to the model. For instance, adding in sensor readings from all light gates and motor outputs to provide the system administrator and IDS algorithm a dataset with a higher level of complexity. Another possible route for this project could be implementation using a comparable simulation software that allows for ROS communication. Since Simulink's ROS library would not allow us to publish string messages in a reliable way, we were forced to rely on String Constant blocks containing the messages that are meant to be transmitted via ROS connection. In this study, we conducted a two-stage investigation. First, we performed a comprehensive literature review of cybersecurity research for IIoT systems, focusing on anomaly detection approaches. We employed a standardized and systematic approach, utilizing advanced Sankey illustrations to present our review's conclusions and illustrate the correlations between different fields of IIoT cybersecurity research.

In the second stage of our study, we focused on developing a cyberattack dataset for a modular IIoT system and subsequently evaluated the performance of various supervised machine learning algorithms for training and testing purposes. To optimize the models' effectiveness, we pre-processed the dataset, which entailed removing empty fields, encoding text entries, and scaling features. We assessed the performance of four ML models, namely k-nearest neighbor (kNN), decision

tree (DT), random forest (RF), and artificial neural network (ANN). Our analysis revealed that the ANN algorithm showed high precision and the fastest detection time during testing, it took the longest time to train, rendering it less suitable for applications that require fast training. The DT algorithm, on the other hand, displayed the most favorable performance in terms of precision metrics and training and testing speeds.

Based on the results of the data visualization plot, it is evident that the data points were naturally separable, thereby providing a plausible explanation for the excellent performance of basic machine learning (ML) algorithms. However, this may not necessarily hold true for more intricate Industrial Internet of Things (IIoT) systems that involve the communication of several hundred or thousand devices and more sophisticated processes. To gain a deeper insight into such complex systems, we intend to conduct a realistic case study using a digital twin of an actual factory. Our objective is to test various ML algorithms on this system and possibly propose novel algorithms to enhance its performance.

List of References

[1] J. C. Silva, M. Saadi, L. Wuttisittikulkij, D. R. Militani, R. L. Rosa, D. Z. Rodríguez, and S. Al Otaibi, "Light-field imaging reconstruction using deep learning enabling intelligent autonomous transportation system," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 2, pp. 1587–1595, 2021.

[2] V. R. Kebande, "Industrial internet of things (iiot) forensics: The forgotten concept in the race towards industry 4.0," *Forensic Science International: Reports*, vol. 5, p. 100257, 2022.

[3] M. Abdelatti and M. Sodhi, "Lab-scale smart factory implementation using ros," in *Robot Operating System (ROS) The Complete Reference (Volume 7)*. Springer, 2023, pp. 119–143.

[4] A. Corallo, A. M. Crespino, V. Del Vecchio, M. Lazoi, and M. Marra, "Understanding and defining dark data for the manufacturing industry," *IEEE Transactions on Engineering Management*, 2021.

[5] S. M. Kasongo, "An advanced intrusion detection system for iiot based on ga and tree based algorithms," *IEEE Access*, vol. 9, pp. 113 199–113 212, 2021.

[6] M. Zolanvari, Z. Yang, K. Khan, R. Jain, and N. Meskin, "Trust xai: Model-agnostic explanations for ai with a case study on iiot security," *IEEE Internet of Things Journal*, 2021.

[7] V. K. Nguyen, M. N. Q. Truong, Q. T. Le, T. H. Nguyen, *et al.*, "A novel approach for data collection and network attack warning," in *2019 11th International Conference on Knowledge and Systems Engineering (KSE)*. IEEE, 2019, pp. 1–6.

[8] O. Aouedi, K. Piamrat, G. Muller, and K. Singh, "Federated semi-supervised learning for attack detection in industrial internet of things," *IEEE Transactions on Industrial Informatics*, 2022.

[9] M. Al-Hawawreh, E. Sitnikova, and F. den Hartog, "An efficient intrusion detection model for edge system in brownfield industrial internet of things," in *Proceedings of the 3rd International Conference on Big Data and Internet of Things*, 2019, pp. 83–87.

[10] N. Moustafa, M. Keshk, K.-K. R. Choo, T. Lynar, S. Camtepe, and M. Whitty, "Dad: a distributed anomaly detection system using ensemble one-class statistical learning in edge networks," *Future Generation Computer Systems*, vol. 118, pp. 240–251, 2021.

[11] W. Liang, K.-C. Li, J. Long, X. Kui, and A. Y. Zomaya, "An industrial network intrusion detection algorithm based on multifeature data clustering optimization model," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 3, pp. 2063–2071, 2019.

[12] S. Latif, Z. Idrees, Z. Zou, and J. Ahmad, "Drann: A deep random neural network model for intrusion detection in industrial iot," in *2020 International Conference on UK-China Emerging Technologies (UCET)*. IEEE, 2020, pp. 1–4.

[13] B. Susilo and R. F. Sari, "Intrusion detection in iot networks using deep learning algorithm," *Information*, vol. 11, no. 5, p. 279, 2020.

[14] B. Li, Y. Wu, J. Song, R. Lu, T. Li, and L. Zhao, "Deepfed: Federated deep learning for intrusion detection in industrial cyber–physical systems," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 8, pp. 5615–5624, 2020.

[15] M. F. Abdelatti, "Empowering industrial iot systems with gpu-powered optimization algorithms and cybersecurity tools," Ph.D. dissertation, University of Rhode Island, 2022.

[16] H. Mittal, A. K. Tripathi, A. C. Pandey, M. D. Alshehri, M. Saraswat, and R. Pal, "A new intrusion detection method for cyber–physical system in emerging industrial iot," *Computer Communications*, vol. 190, pp. 24–35, 2022.

[17] Y. Wu, "Basic intrusion technology of industrial internet of things—based on machine learning," in *Journal of Physics: Conference Series*, vol. 1738, no. 1. IOP Publishing, 2021, p. 012094.

[18] T. Vaiyapuri, Z. Sbai, H. Alaskar, and N. A. Alaseem, "Deep learning approaches for intrusion detection in iiot networks–opportunities and future directions," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 4, 2021.

[19] H. Yao, P. Gao, P. Zhang, J. Wang, C. Jiang, and L. Lu, "Hybrid intrusion detection system for edge-based iiot relying on machine-learning-aided detection," *IEEE Network*, vol. 33, no. 5, pp. 75–81, 2019.

[20] M. Al-Hawawreh, E. Sitnikova, and N. Aboutorab, "X-iiotid: A connectivity-agnostic and device-agnostic intrusion data set for industrial internet of things," *IEEE Internet of Things Journal*, vol. 9, no. 5, pp. 3962–3977, 2021.

[21] G. Rathee, S. Garg, G. Kaddoum, and B. J. Choi, "Decision-making model for securing iot devices in smart industries," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 6, pp. 4270–4278, 2020.

[22] Z. Yang, J. He, Y. Tian, and J. Zhou, "Faster authenticated key agreement with perfect forward secrecy for industrial internet-of-things," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 10, pp. 6584–6596, 2019.

[23] H. Emil, A. Vulfin, K. Mironov, A. Frid, M. Guzairov, and A. Kirillova, "Secure data exchange in the industrial internet of things network of the fuel and energy complex," in *2020 International Conference on Electrotechnical Complexes and Systems (ICOECS)*. IEEE, 2020, pp. 1–6.

[24] J.-S. Fu, Y. Liu, H.-C. Chao, B. K. Bhargava, and Z.-J. Zhang, "Secure data storage and searching for industrial iot by integrating fog computing and cloud computing," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 10, pp. 4519–4528, 2018.

[25] J. Wang, B. Wei, J. Zhang, X. Yu, and P. K. Sharma, "An optimized transaction verification method for trustworthy blockchain-enabled iiot," *Ad Hoc Networks*, vol. 119, p. 102526, 2021.

[26] Y. Gao, Y. Chen, X. Hu, H. Lin, Y. Liu, and L. Nie, "Blockchain based iiot data sharing framework for sdn-enabled pervasive edge computing," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 7, pp. 5041–5049, 2020.

[27] G. Rathee, F. Ahmad, R. Sandhu, C. A. Kerrache, and M. A. Azad, "On the design and implementation of a secure blockchain-based hybrid framework for industrial internet-of-things," *Information Processing & Management*, vol. 58, no. 3, p. 102526, 2021.

[28] W. Liang, M. Tang, J. Long, X. Peng, J. Xu, and K.-C. Li, "A secure fabric blockchain-based data transmission technique for industrial internet-of-things," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 6, pp. 3582–3592, 2019.

[29] F. Martin-Tricot, C. Eichler, and P. Berthomé, "An enrolment gateway for data security in heterogeneous industrial internet of things," in *2020 IEEE 29th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE)*. IEEE, 2020, pp. 199–202.

[30] M. Zubair Islam, Shahzad, R. Ali, A. Haider, and H. Kim, "Iotactilesim: a virtual testbed for tactile industrial internet of things services," *Sensors*, vol. 21, no. 24, p. 8363, 2021.

[31] M. Al-Hawawreh and E. Sitnikova, "Developing a security testbed for industrial internet of things," *IEEE Internet of Things Journal*, vol. 8, no. 7, pp. 5558–5573, 2020.

[32] B. Craggs, A. Rashid, C. Hankin, R. Antrobus, O. Şerban, and N. Thapen, "A reference architecture for iiot and industrial control systems testbeds," in *Living in the Internet of Things (IoT 2019)*. IET, 2019, pp. 1–8.

[33] S. Pfrang, J. Kippe, D. Meier, and C. Haas, "Design and architecture of an industrial it security lab," in *Testbeds and Research Infrastructures for the Development of Networks and Communities: 11th International Conference, TRIDENTCOM 2016, Hangzhou, China, June 14-15, 2016, Revised Selected Papers*. Springer, 2017, pp. 114–123.

[34] N. Matsumoto, J. Fujita, H. Endoh, T. Yamada, K. Sawada, and O. Kaneko, "Asset management method of industrial iot systems for cyber-security countermeasures," *Information*, vol. 12, no. 11, p. 460, 2021.

[35] Y. Zhao, N. Hu, Y. Zhao, and Z. Zhu, "A secure and flexible edge computing scheme for ai-driven industrial iot," *Cluster Computing*, vol. 26, no. 1, pp. 283–301, 2023.

[36] Z. Xiong, H. Wang, L. Zhang, T. Fan, and J. Shen, "A ring-based routing scheme for distributed energy resources management in iiot," *IEEE Access*, vol. 8, pp. 167 490–167 503, 2020.

[37] M. Conti, P. Kaliyar, and C. Lal, "Secure machine to machine communication in industrial internet of things," *Security and Privacy Trends in the Industrial Internet of Things*, pp. 199–219, 2019.

[38] A. Corallo, M. Lazoi, M. Lezzi, and A. Luperto, "Cybersecurity awareness in the context of the industrial internet of things: A systematic literature review," *Computers in Industry*, vol. 137, p. 103614, 2022.

[39] R. J. Raimundo and A. T. Rosário, "Cybersecurity in the internet of things in industrial management," *Applied Sciences*, vol. 12, no. 3, p. 1598, 2022.

[40] P. Arora, B. Kaur, and M. A. Teixeira, "Security in industrial control systems using machine learning algorithms: An overview," *ICT Analysis and Applications*, pp. 359–368, 2022.

[41] S. Pal and Z. Jadidi, "Analysis of security issues and countermeasures for the industrial internet of things," *Applied Sciences*, vol. 11, no. 20, p. 9393, 2021.

[42] S. F. Tan and A. Samsudin, "Recent technologies, security countermeasure and ongoing challenges of industrial internet of things (iiot): A survey," *Sensors*, vol. 21, no. 19, p. 6647, 2021.

[43] L. L. Dhirani, E. Armstrong, and T. Newe, "Industrial iot, cyber threats, and standards landscape: Evaluation and roadmap," *Sensors*, vol. 21, no. 11, p. 3901, 2021.

[44] N. Abosata, S. Al-Rubaye, G. Inalhan, and C. Emmanouilidis, "Internet of things for system integrity: A comprehensive survey on security, attacks and countermeasures for industrial applications," *Sensors*, vol. 21, no. 11, p. 3654, 2021.

[45] M. Serror, S. Hack, M. Henze, M. Schuba, and K. Wehrle, "Challenges and opportunities in securing the industrial internet of things," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 5, pp. 2985–2996, 2020.

[46] D. Hamouda, M. A. Ferrag, N. Benhamida, and H. Seridi, "Intrusion detection systems for industrial internet of things: a survey," in *2021 International Conference on Theoretical and Applicative Aspects of Computer Science (ICTAACS)*. IEEE, 2021, pp. 1–8.

[47] W. Xu, J. Jang-Jaccard, T. Liu, F. Sabrina, and J. Kwak, "Improved bidirectional gan-based approach for network intrusion detection using one-class classifier," *Computers*, vol. 11, no. 6, p. 85, 2022. [Online]. Available: https://www.mdpi.com/2073-431X/11/6/85

[48] D. R. Jeske, B. Samadi, P. J. Lin, L. Ye, S. Cox, R. Xiao, T. Younglove, M. Ly, D. Holt, and R. Rich, "Generation of synthetic data sets for evaluating the accuracy of knowledge discovery systems," in *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, 2005, pp. 756–762.

[49] V. Belenko, V. Krundyshev, and M. Kalinin, "Synthetic datasets generation for intrusion detection in vanet," in *Proceedings of the 11th international conference on security of information and networks*, 2018, pp. 1–6.

[50] MATLAB, "Distributing multi-class jobs to service stations - matlab & simulink." [Online]. Available: https://www.mathworks.com/help/simevents/ug/distributing-multi-class-jobs-to-service-stations.html

[51] F. Aubet and M. Pahl, "Ds2os traffic traces," 2018.

[52] M.-O. Pahl and F.-X. Aubet, "All eyes on you: Distributed multi-dimensional iot microservice anomaly detection," in *2018 14th International Conference on Network and Service Management (CNSM)*. IEEE, 2018, pp. 72–80.

[53] A. Géron, *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems.* " O'Reilly Media, Inc.", 2019.

[54] S. Khare and M. Totaro, "Ensemble learning for detecting attacks and anomalies in iot smart home," in *2020 3rd International Conference on Data Intelligence and Security (ICDIS)*. IEEE, 2020, pp. 56–63.

MANUSCRIPT 4

Digital Twin validation

Zachary A. deWardener[1*], and Manbir S. Sodhi[1]

[1]Department of Industrial Engineering, University of Rhode Island, Kingston, Rhode Island, 02881

[*]Corresponding author: zdewardener@uri.edu

Drafted for submission, XXX–XXX.

ACM ISBN XXX-X-XXXX-XXXX-X/XX/XX

https://doi.org/XX.XXXX/XXXXXXX.XXXXXXX

4.1 Abstract

This paper investigates the challenges and potential of applying machine learning techniques to validate digital twins, thoroughly analyzing the current state of research and practice. Through extensive literature review, this study identifies critical obstacles in the path to successful implementation, such as system complexity, the need for diverse similarity measures, and the influence of external factors like functional behavior and computational efficiency. The research highlights the discrepancy in validation approaches for different systems and underscores the importance of considering these differences when testing for visual and functional similarity.

We propose using machine learning models to establish similarity measures between digital twins and their physical counterparts, recognizing the nuanced patterns that such technologies can uncover. Despite the theoretical promise of these methods, the article acknowledges the infancy of practical applications and the absence of a standardized validation framework. The findings from the study reveal a 50% accuracy rate in differentiating between real and simulated datasets, with interesting implications for the training and interpretation of artificial neural networks (ANNs) in this context. Concluding with a reflection on the implications for digital twin technology, emphasizing the need for future research to address the highlighted challenges and develop robust validation methods to enhance digital twins' reliability and security in various domains.

4.2 Introduction

In the realm of quality control, the processes of validation and verification, often abbreviated as V&V, are critical for ensuring the trustworthiness of any quantitative analysis. These steps are essential for confirming the dependability of solutions before they are applied to real-world scenarios. Without a reliable method to assess the accuracy of outcomes, the results remain unverified and potentially unreliable. This principle is particularly pertinent in simulations involving high-value and intricate systems. Digital Twins, which are frequently employed in such simulations, must adhere to this standard as well. Despite the broad application of Digital Twin technology across various sectors, such as healthcare and manufacturing, there has been a tendency in research to focus on defining the concept rather than establishing a universal method for its validation.

A consensus on validation techniques for Digital Twins is critical for their effective implementation and for ensuring that these simulations perform as expected across different industries. A standardized approach to validation would provide a benchmark for evaluating the performance of Digital Twins, allowing for more consistent and reliable use in various applications. It would also facilitate a shared understanding and methodology that could be beneficial for interdisciplinary research and development.

As the use of Digital Twins expands, the need for a standardized validation framework becomes more pressing. Future research directions may benefit from a focus on creating flexible yet rigorous validation protocols that can be adapted to the unique requirements of different Digital Twin applications. Such efforts would likely involve collaboration across academic, industrial, and regulatory domains to establish practical guidelines that uphold the integrity and utility of Digital Twins in complex system management. This collaborative approach could help to ensure

that Digital Twins live up to their potential as a transformative technology in various fields.

4.3 Literature Review

The studies reviewed in this paper were mainly derived from the Google Scholar database using the following combination of keywords: Measures of Similarity + Digital Twin. This search resulted in a total of 75,700 publications prior to further refining to show the most relevant articles from 2017-2023 which still provided a staggering 17,100 results.

4.3.1 Previous research on DT model validation

Noticing a vast gap in research on digital twin validation, the goal of this paper is to provide a frame of reference for our proposed validation method and bring to light some of the most recent developments both supporting and opposing our view on the matter of validating digital twins.

Digital twins – Batty (2018)

The author of [1] delves into the intricate definitions of digital twins, challenging the conventional notion that a virtual model should be a complete mirror image of its physical counterpart. The author argues that when a digital twin becomes an exact replica of the physical system, it loses its separate identity and merges with the system itself. This raises a pertinent question: how can a "true digital twin", which runs in real-time and is indistinguishable from the physical system, be employed to derive insights about the system and test new configurations for optimization?

While exploring applications of digital twins in the smart city and railway management implementations both in literature and within their own lab, the author leaves certain questions unanswered, such as how to accurately measure

the similarity between digital models and real-world systems. This gap in the research opens avenues for further exploration and investigation. As we delve deeper into the realm of digital twins, it is crucial to recognize that as the digital representation inches closer to the real system, the line between the two begins to blur. This necessitates a reevaluation of the concept of digital twins as perfect mirrors of physical entities, challenging us to redefine and refine our understanding of this innovative technological paradigm.

Online validation of digital twins for manufacturing systems – Lugaresi (2023)

Lugaresi et al. [2] focused their study on formulating a method of online digital twin validation using small sets of data, measuring the similarity between real system data and digital model data through sequence comparison techniques. They proposed the use of three sequence comparison formulas to derive similarity indicator values between 0 (completely dissimilar) and 1 (perfect replication), the Longest Common Sub-sequence (LCSS), the Modified Longest Common Subsequence (mLCSS), and Dynamic Time Warping (DTW). With the goal of validating a given digital model in two parts:

- **logic validation** - focusing mainly on the model's process flows and assumptions
- **input validation** – concerning the accuracy of the input data supplied to the model.

All the while keeping in mind that a simulation model should never reach a similarity indicator level of 1 since simulations are abstractions of real systems, and it is not possible to achieve the perfect modelling of that system. To test their hypothesized methods, the authors formulated a statistical design of experiment on a discrete event simulation model of a five-station closed system using

Rockwell Arena Simulation Software with the goal of measuring two referenced KPIs: System Time and Inter-departure Time. By changing aspects of the digital model such as mean station processing times and threshold values within their comparison formulas at both bottleneck and non-bottleneck stations, they proved the effectiveness of their formulas in detecting deviations from the simulated data values when compared to the physical process data.

Digital Twins in Architecture, Engineering, Construction and Operations. A Brief Review and Analysis – Al-Sehrawy (2021)

Al-Sehrawy et al. [3] along with their work in reviewing the overall concept of the digital twin as it applies to different industry applications, proposed a novel approach to measuring the similarity of digital models to real applications. This approach entailed an adaptation of Grieves Tests of Virtuality (GTV) as proposed in [4] and more recently updated in [5]. Their hypothetical methodology entailed a three-step analysis, first evaluating the Visual Fidelity of the digital twin model, measuring the level of geometric detail and the granularity of graphical information in terms of reflecting the appearance of the real system. Followed by Reflective Fidelity in gauging the level of detail pertaining to non-graphical information, and lastly Performance Fidelity, assessing the level of accuracy in predicting the true and real behavior of the physical system. While the author's reasoning is sound and generally aligns with the approach for validating digital twins that is proposed later in this paper, the authors do not supply an example of its use, requiring further research in this strategy.

How to tell the difference between a model and a digital twin – Wright (2020)

The authors of [6] delve into the intricate distinctions between a model and its digital twin, highlighting the recognized benefits and potential of digital twin

models, while also suggesting directions for future research to fully tap into this innovative approach. In their methodology, the authors align with the consensus that a digital twin must include a physical entity, which allows the model to extract real-time data for various applications such as process monitoring, control, and predictive operations. They acknowledge the ideal scenario where digital twin behavior is derived entirely from physics-based modeling, while also highlighting the practicality of surrogate models or metamodels. These data-driven models, due to their higher level of abstraction, require less computational complexity compared to their physics-based counterparts.

The authors do not prescribe a specific validation method, instead advocating for a statistical approach to validation considering the inherent uncertainties. They direct readers to existing publications that offer guidance on uncertainty evaluation for complex and computationally demanding problems. The significance of data trustworthiness is emphasized, with the authors recommending metadata analysis as a means to ensure data reliability. However, the paper stops short of pinpointing the most effective validation method, thereby opening a pathway for future research in this domain.

Correspondence measure: a review for the digital twin standardization – Khan (2023)

Khan et al., [7] put forth a unique approach to standardizing Digital Twins by introducing a "correspondence measure" designed to quantitatively assess the accuracy and reliability of a digital twin in relation to its physical counterpart. This measure is crucial in establishing a standardized metric that can be universally applied across various domains. To build a solid foundation for their claims, the authors meticulously curated an extensive literature review, drawing from a rich pool of academic articles and case studies.

Through their comprehensive analysis, the authors unearthed a myriad of challenges and limitations that could potentially impede the successful implementation of their proposed approach. These challenges include the varying complexities inherent in different systems, each with its unique characteristics and behavior that may necessitate distinct similarity measures. For instance, while a similarity measure based on physical attributes may be apt for a digital twin of a manufacturing machine, it may prove inadequate for a digital twin of a human organ, where biological and physiological factors play a crucial role. This discrepancy raises questions about the necessity of testing visual similarity. Additionally, the authors identified external factors such as functional behavior, performance under varying conditions, and response to external stimuli, as well as computational efficiency, privacy, and security concerns as potential hurdles.

Interestingly, the authors also explored the potential of leveraging machine learning techniques to calculate similarity measures. These models, once trained, can decipher complex patterns and relationships within the data, thereby facilitating the computation of a similarity score between the digital twin and its physical counterpart. This approach is particularly advantageous when dealing with intricate systems where a straightforward comparison may not yield accurate results. Despite these valuable insights, it is important to note that the correspondence measure proposed by the authors is still in its infancy, existing primarily in the theoretical realm with no practical applications developed to date.

Toward a Digital Twin for real-time geometry assurance in individualized production – Söderberg (2017)

In [8] the authors delve into the potential integration of digital twins in geometry assurance, with the ultimate goal of synthesizing variation simulation and quality control research to pave the way for a fully autonomous production line.

Their methodology is rooted in a thorough review of pertinent literature, complemented by a practical example that illustrates the potential implementation of this system in a sheet metal assembly process.

The authors' findings underscore the importance of strategically positioning a digital twin inspection model early in the product lifecycle, specifically during the pre-production phase. This proactive approach is crucial in circumventing the need for costly rectifications that would be inevitable if errors were identified during the production stage. While the paper doesn't delve deeply into the intricacies of digital twin model validation, it does highlight the potential of artificial intelligence (AI) and machine learning as valuable tools for capturing and addressing non-modeled effects in a data-driven manner, as exemplified in their case study.

Using trace alignments for measuring the similarity between a physical and its digital twin – Muñoz (2022)

Muñoz et al., [9] present an innovative approach for aligning data traces between a digital twin and its physical counterpart, utilizing an adapted Needleman-Wunsch algorithm, traditionally used in bioinformatics for sequence alignment, to compare sequences of snapshots that detail the state of a system at specific moments, capturing the intricate web of object relationships and values. Their methodology involves an abstraction function to extract comparable data points and the Frechet distance to measure alignment, with a focus on the relationship between matched points and tolerance levels. The findings suggest that a high-fidelity digital twin should have a high percentage of matched points with low tolerance and distance. Future work aims to test the algorithm under heavier loads and incorporate real-world uncertainties. The researchers concede that while a perfectly accurate digital twin is unattainable, striving for the highest possible fidelity is essential. This is particularly true in applications where the digital twin's

behavior, such as that of a robotic arm, must closely mirror the physical twin to prevent errors and potential damage.

Multidimensional data modeling and model validation for digital twin workshop – Qian (2021)

In their research, the authors of [10] delve into the complexities of a multidimensional spatiotemporal model, which aims to encapsulate the production factor variables within a Digital Twin Workshop (DTW). They employ this model to represent the production process and engage in its verification through higher order singular value decomposition, a sophisticated mathematical technique. Their study underlines the importance of validation as a critical process in determining how well a simulation mirrors the real world within its intended scope of application.

Their findings reveal that the model is not only capable of accurately capturing the explicit and implicit characteristics of the workshop system, but also serves as a robust representation of the system's multifaceted nature. However, the authors acknowledge that their study opens the door for future research to investigate alternative methods of model validation or to apply the model to various workshop environments, expanding the scope and applicability of their findings.

Digital Twins: Past, Present, and Future – Grieves (2023)

In the realm of digital twin technology, Grieves, the author of [5], is widely regarded as a pioneering figure, often referred to as the father of digital twin technology due to his early adoption and comprehensive analysis of its benefits, which he has presented extensively to the academic community. His study delineates the growth of the Digital Twin Model from its inception in 2002, noting the exponential increase in related academic papers to over 52,000 by 2023. Grieves categorizes Digital Twins into three types aligned with the product lifecycle phases, emphasizing the role of Digital Twin Prototypes (DTP) for virtual development, Digital

Twin Instances (DTI) for tracking individual products, and Digital Twin Aggregates (DTA) for collective data analysis and predictive applications.

Employing Grieves' Tests of Virtuality (GTV), a set of criteria developed to assess the fidelity of a digital twin in relation to its physical counterpart. These tests are designed to ensure that the digital twin can not only replicate the appearance and behavior of the physical object but also predict its future state and reflect its real-time status. Supplied below is a more in-depth look at each of the four tests:

- **Visual Test:** This test is concerned with the digital twin's visual congruence with the physical object. An observer is presented with two displays: one showing the physical object and the other its digital representation. The observer can request any spatial manipulation, such as rotation or zoom, to view the object from different angles. If the observer cannot distinguish the digital twin from the physical object based on visual criteria alone, the digital twin passes the visual test.

- **Behavioral Test:** The behavioral test goes beyond static appearance to evaluate dynamic congruence. Here, the observer witnesses both the physical object and the digital twin subjected to the same external forces or conditions. The observer then examines the responses of both. If the reactions and behaviors under these conditions are indistinguishable, the digital twin passes the behavioral test.

- **Reflectivity Test:** This test examines the digital twin's ability to reflect the real-time status of the physical object. The observer compares the real-time data and state of the physical object with that of the digital twin. If the digital twin accurately reflects the current state of the physical object without any discernible lag or discrepancy, it passes the reflectivity test.

- **Prediction Test:** The newest addition to Grieves' tests, the prediction test, evaluates the digital twin's ability to forecast the future state of the physical object. The observer requests that the digital twin simulate a future state after a certain period. Once that time elapses, the observer compares the actual state of the physical object with the predicted state from the digital twin. If the two states match closely, the digital twin passes the prediction test.

Within his study, Grieves also challenges the prevalent misconception that a DT must have a concurrent physical object, asserting instead the necessity of strong resemblance and the intent for the digital model to eventually materialize physically. His findings confront the "Digital Twin Fallacy", advocating for a broader understanding of DTs' role across the product lifecycle and solidifying his status as a seminal voice in the field.

Defining a Digital Twin: A Data Science-Based Unification – Emmert (2023)

The authors of [11] embark on a study to distill a formal, theoretical definition of Digital Twins that transcends the application-driven interpretations commonly found in scholarly discourse. They draw a line in the sand between the Digital Twin (DT), a mathematical model dedicated to the generation of data, and the Digital Twin System (DTS), which extends beyond to encompass data analysis and decision-making, often leveraging machine learning.

The crux of their proposition lies in the validation of a DT. They propose that a model earns the title of a DT when the divergence between its output and the empirical data from its physical twin falls below a pre-set threshold. This divergence can be quantified through statistical means such as the Kolmogorov-Smirnov test or by computing the Mean Squared Error, with the threshold tailored

to the context at hand.

They elucidate the roles of DT and DTS; the former is a near-perfect data mimic of its physical counterpart, while the latter is an overarching system that processes this data into actionable decisions. The authors recognize the impracticality of continuous data updates, especially in sensitive domains like healthcare, suggesting a balance between update frequency and system accuracy.

4.4 Case Study: Digital Twin Validation in the LabFab System

The following section highlights an in-house study performed with the goal of proposing a novel digital twin validation method, by leveraging the pattern recognition capabilities of Machine Learning principles, that could be applied across various industries. Before delving into our study, it is important to note that this implementation has not yet been applied outside of academic research as this is a theoretical concept that has the potential of standardizing digital twin practices across multiple domains.

4.4.1 Description of the case
Overview of the LabFab System

The LabFab system refers to a model-based manufacturing system built as a means for students to gain multi-disciplinary experience in working with industry 4.0 concepts to realize process automation on a manageable scale. This system, in its current evolutionary state, is comprised of physical components sourced from German-based robotics company Fischertechnik combined with a six-axis collaborative robot (cobot) from Elephant Robotics and a conveyor system built with LEGO components. These physical elements are connected and controlled by Raspberry Pi 4 boards (RasPi) - as a means of networking, and FTduino boards - responsible for motor and sensor operations. The RasPi-based networking infrastructure follows similarly to a star-topology with the “MainPi” being the central

hub and MQTT broker, handling communication to-and-from connected RasPis via ROS publish/subscribe messaging format. For more in-depth information on the LabFab system setup, readers are encouraged to review the work of Abdelatti in [12]. The physical autonomous processes within the LabFab system are performed with the goal of carrying colored circular pieces (alias: parts) from the High-Bay Warehouse (HBW), representing raw material storage, through various subsequent processes to eventually return to storage as finished goods. A full description of these processes accompanied by video demonstration can be found on YouTube at `https://www.youtube.com/watch?v=4biX9enHBuU&t=4s`.

Creation of the Digital Twin

The digital twin of our LabFab system was created using MATLAB's simulation software, Simulink, in conjunction with their discrete events simulation software package, SimEvents. Further information regarding the modelling of internal processes within the digital twin are covered extensively in chapter 3, section 4, of this manuscript. The only difference being, the probability of cyber threats occurring is adjusted to zero and server processing times are set as a uniform normal distribution acquired from real-system behavioral data. An important aspect to note is our focus on modelling events occurring within the application layer of the system. This means our data output for both the real system and digital twin system is revolving around the ROS messages themselves, accompanied by metadata explaining certain permissions (i.e., read/write), service types, topic names, timestamps, and more (all of which are also explained thoroughly in chapter 3, section 5.

4.4.2 Method of Validation Used

Our proposed validation methodology loosely aligns with Grieves Tests of Virtuality (GTV) and, in turn, that of the Turing Tests of Intelligence (i.e., Imitation game) proposed by Allen Turing in 1950. Allen Turing formulated these groundbreaking tests to measure the level at which computerized systems can imitate human intelligence. These tests both indicate the need for a human-in-the-loop when measuring the similarity between a real and simulated process, whether it be observing the response to a specific question being asked to a human and computer (Imitation Game) or attempting to differentiate the current status of a simulated system compared to that of its physical counterpart (GTV). Our rendition leverages the pattern recognition capabilities of Machine Learning (ML) algorithms to differentiate between real and simulated datasets, negating the requirement of having a human observer in the validation process.

Data Collection Methods

The data collection process from the real system is carried out within the MainPi by opening an RQT-console window, allowing us to initiate the recording of ROS data prior to running parts through the LabFab system. Once a single run of a red, white, or blue colored part is complete, the 39 lines of recorded data are then exported into a .csv file and are ready for further processing. For the time being, this data is formatted partially by hand to exclude certain information considered not pertinent to the validation testing, described in the following section. The resulting RQT data only captures four features from the real system, being the timestamp of events occurring (Alias: RunTime) measured in UNIX time initially and later converted to seconds starting at t = 0 for the publishing of an order, the ROS message itself (Alias: msg), the node from which the message is published (Alias: node), and the topics to which the message is published (Alias: topics). It is

crucial to highlight that the LabFab system is a continuously evolving project built by students of the Industrial and Systems Engineering program at the University of Rhode Island. So, while the current data streams from the physical system only capture ROS-based messages, future renditions look to encapsulate more of the process data similar to that of an industrial manufacturing system.

As for the data collection process on the digital twin side, simulation data is initially assembled in MATLAB's workspace, then formatted and exported to an Excel document for further processing via Python script. Since the simulation outputs a higher dimensional dataset than our physical system, the non-essential features are dropped from the set, and columns are renamed to match our physical process data.

Once the simulated and physical datasets are formatted correctly, an additional column of data is added to both called "Type", which denotes whether the dataset is real (Type = 1) or simulated/fake (Type = 0).

Data Analysis Methods

To measure the similarity between collected sets of real and simulated data, we utilized the pattern recognition capabilities of artificial neural networks (ANN) with a binary cross-entropy loss function. The goal of this implementation is to confuse the ML algorithm, specifically designed to classify data into one of two binary groups. The theory behind our implementation is that if the algorithm cannot detect the difference between the datasets to a certain extent, the simulated model can considered a digital twin. One could use the F-score to quantify the level of similarity reached by the digital twin's data compared to real data. F-score is a commonly used statistical measure of accuracy when dealing with binary classification, representing the precision and recall of data with a measure between 0 and 1. Regarding the F-measure threshold that must be reached to consider

the model a digital twin, we selected a level of 0.5 to be considered equivalent - meaning the model could not differentiate between real and simulated datasets.

As a proof of concept, we tested multiple configurations of ANN, from layout changes concerning the number of hidden layers to varying parameters such as batch size, epochs, layer sizes, learning rates, optimizers, and dropout values. For parameter optimization, we utilized a tool called "sweep" from Weights and Biases (WandB), which trains and evaluates models with randomly selected parameters set prior to network definition. An example of the selected parameters used to train the models iteratively is provided in Figure 49. Once the sweep is completed, the performance of each model can be visualized within the WandB user interface in the form of line plots, scatter plots, parallel coordinate plots, and more. An example of a parallel coordinate plot used for the parameter sweep in a neural network (NN) comprised of two hidden layers using a Bayes search method can be seen in Figure 50. Once the best-performing model parameters are realized, they can be transferred into a neural network separate from the one used for WandB sweep with the desired metrics, for example, in Figure 51.

```
# sweep configuration
sweep_configuration = {
    'method': 'bayes',
    'name': 'sweep',
    'metric': {
        'goal': 'maximize',
        'name': 'f1_score'
    },
    'parameters': {
        'batch_size': {'values': [10,15,25,32,50,64,75,100]},
        'epochs': {'values': [50,75,100,125,150,200,250,300]},
        'fc_layer_size': {'values': [7,9,11,13,15,20,25,30,50]},
        'sc_layer_size': {'values': [7,9,11,13,15,20,25,30,50]},
        'lr': {'max': 0.1, 'min': 0.0001},
        'optimizer':{'values': ['adam','SGD','RMSprop','nadam','Adadelta','Adagrad','Adamax','Ftrl','AdamW']},
        'dropout': {'values': [0, 0.3, 0.5]},
     },

}
```

Figure 49. WandB sweep parameters used.

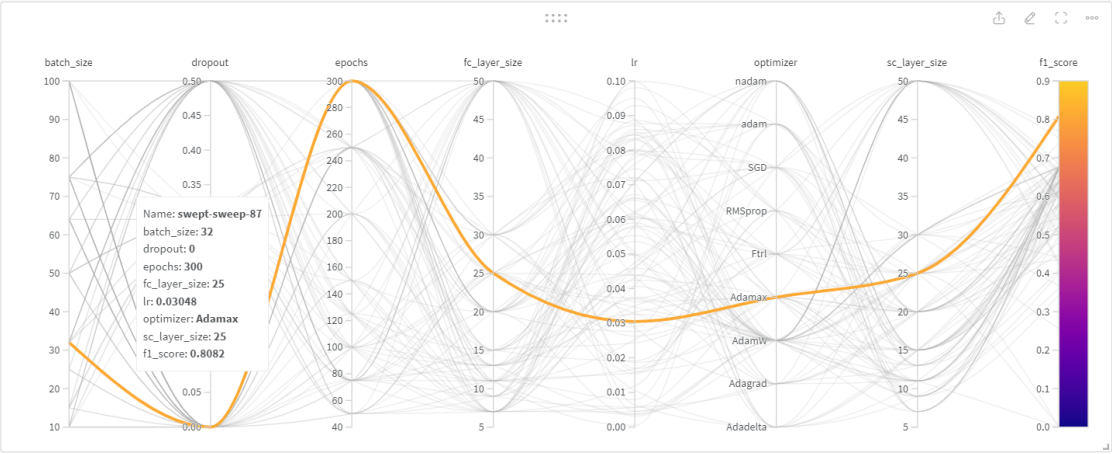


Figure 50. WandB sweep parallel coordinate plot.

```
#Build the model
def Neural_Net():

  # initiate Network
  network = Sequential()

  # input layer that is fully connected with ReLU activation
  network.add(Dense(4, input_dim=4, activation="relu",use_bias=True,bias_initializer="glorot_normal", kernel_initializer="glorot_normal"))
  network.add(Dropout(0))
  #hidden layers
  network.add(Dense(25, activation="relu", use_bias=True, bias_initializer="glorot_normal", kernel_initializer="glorot_normal"))
  network.add(Dropout(0))
  network.add(Dense(25, activation="relu", use_bias=True, bias_initializer="glorot_normal", kernel_initializer="glorot_normal"))

  #Output layer that is fully connected with sigmoid activation
  network.add(Dense(1, activation="sigmoid",kernel_initializer="glorot_normal"))


  # Compile network model.
  network.compile(optimizer=get_optimizer(0.03048, 'Adamax'),
              loss="binary_crossentropy",
              metrics=["binary_accuracy",
              tfa.metrics.F1Score(num_classes=2,
                                  average='micro',
                                  threshold=0.5),
              tf.keras.metrics.AUC(from_logits=False)
                      ])


  #return completed network
  return network

# Wrap Keras model so it can be used by scikit-learn
neural_network = KerasClassifier(
                                model=Neural_Net,
                                epochs=300,
                                batch_size=32,
                                verbose=0,
                                callbacks=[WandbCallback()]
                                )
```

Figure 51. Neural network configuration for WandB sweep.

4.4.3 Results and Implications

The whole concept of attempting to trick an ML algorithm into failing at classifying groups of data successfully is, admittedly, wildly confusing. However, when put visually in the form of a confusion matrix, it begins to make more sense. The performance of the NN Turing Test, when exposed to a portion of simulated and real data that are unlabelled, clearly has a difficult time differentiating between the two sets of data as seen in Figure 52. This phenomenon can be shown in two cases:

Case 1: The model classifies the majority of data as simulated (type = 0), including those that are real (type = 1) or vice-versa, as seen in the previously mentioned figure.

Case 2: The model can only correctly classify the data 50% of the time, as seen in Figure 53

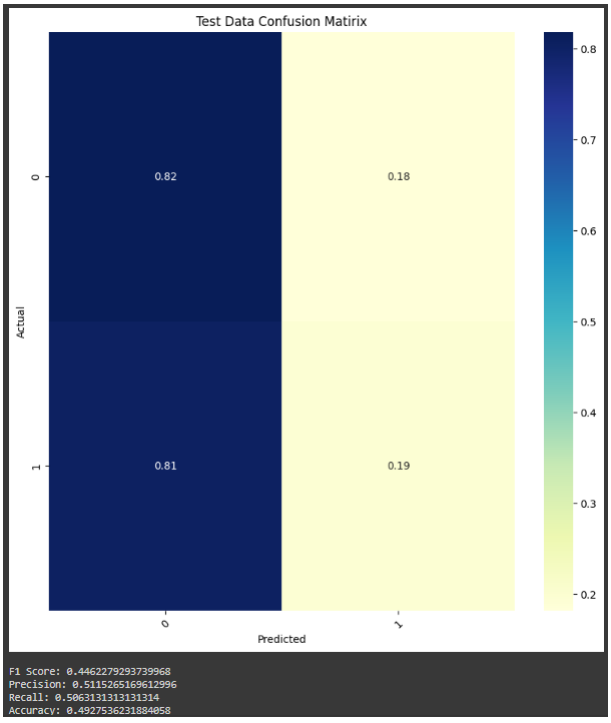


Figure 52. Test data confusion matrix Case 1.

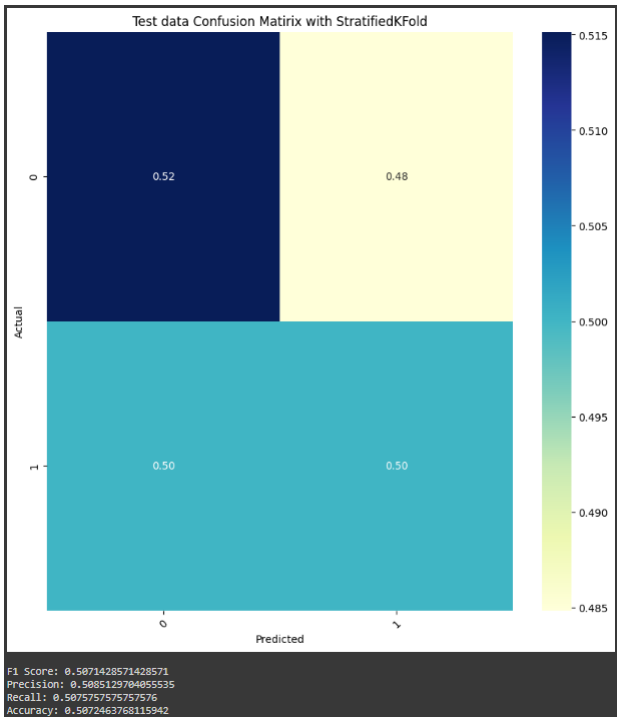


Figure 53. Test data confusion matrix Case 2.

To further test our model, we exposed the trained NN to three separate situations to see how it would perform. The first of which was using two complete datasets of real and simulated data that were outside of the model training process. The main difference between these datasets and those used for training is their time values, and the results can be seen in Figure 54. The next test was to expose the trained NN model to simulated data only, and the last test was to use real data only. These two tests specifically gave fascinating results, as seen in Figure 55 and Figure 56, where even while being exposed to only one class of data, the trained model still classifies some instances as the opposing class.

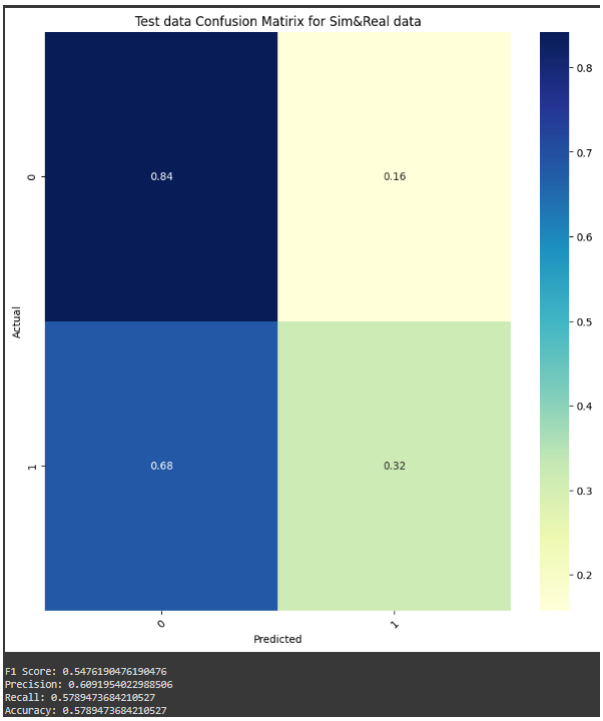


Figure 54. External Sim and Real test data confusion matrix.

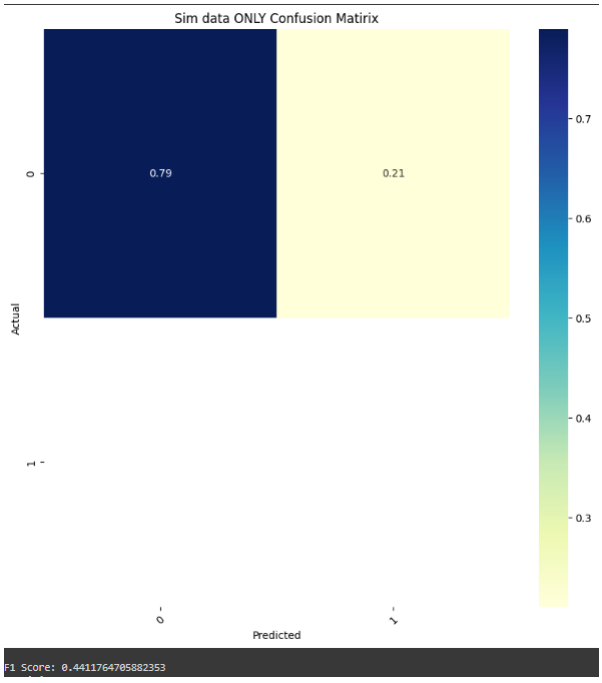


Figure 55. Sim data only test confusion matrix.

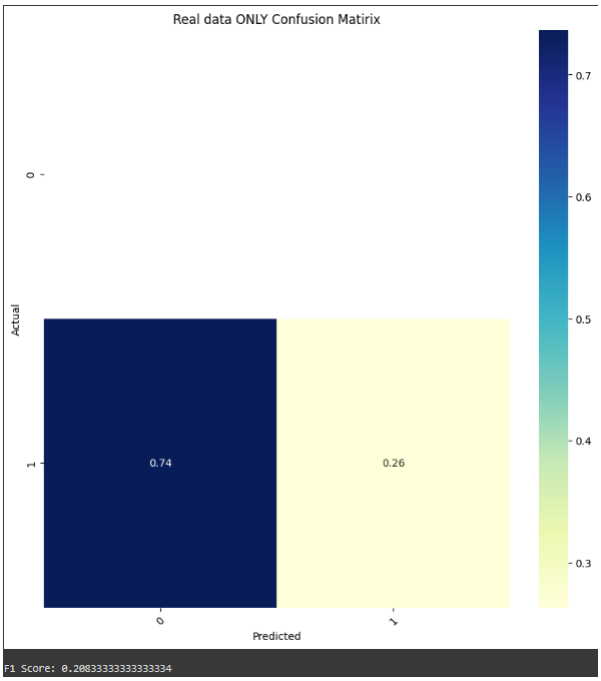


Figure 56. Real data only test confusion matrix.

Implications for the Digital Twin Concept

The implications of creating a standardized method for digital twin validation are of tremendous proportions. Most notably, the reliability of the digital twin system itself is of enormous concern. Ensuring that digital twin models meet a certain threshold of accuracy in portraying their physical counterparts builds trust among users and stakeholders. Should the fidelity of a digital twin to its physical equivalent fall below an established benchmark, the credibility of its predictive capabilities for tasks such as maintenance scheduling or production planning is compromised.The development of a standardized validation approach for digital twin models may also enhance interoperability and scalability. A consensus within the academic community on a validation technique that ensures high behavioral fidelity of baseline digital twin systems would pave the way for establishing, assessing, and integrating dependable digital twin applications across various sectors. Moreover, such standardization would facilitate the expansion of digital twin

models from limited-scope projects to comprehensive enterprise applications with greater ease.

4.4.4 Challenges in Digital Twin Validation

The challenges associated with digital twin validation are similar to those regarding the multiple definitions of digital twins flooding academic research today. Much of the implementations in the literature attempt to solve digital twin validation in their unique methods to address models constructed for sector-specific applications rather than focusing on aspects that are concurrent throughout all digital twin models, the data. While visual and predictive fidelity may be beneficial in specific applications, the driving force behind the benefits of digital twinning is the behavioral fidelity supported by physics and data-driven responses. With our proposal of using a ML-based Turing Test for digital twin validation, the relationships between simulated and physical systems spatiotemporal data can be realized across multiple domains, facilitating the widespread usage of digital twin technology.

4.4.5 Limitations of the Study & Future Research

Our investigation encountered certain constraints, notably the application of our research to a model-based system that yields only Robot Operating System (ROS) level data. To thoroughly evaluate the advantages of integrating machine learning in digital twin validation, future research should extend these methodologies to systems that offer detailed, process-oriented data, such as sensor outputs and spatial-environmental information. Additionally, we acknowledge our limited expertise in executing machine learning algorithms, which may affect the appropriateness of the algorithm we employed, despite the theoretical soundness of using machine learning for spatiotemporal pattern analysis. A more robust approach

would involve collaboration between experts in machine learning and those with deep knowledge of the system processes under study. Furthermore, while our focus was on manufacturing systems, we suggest that subsequent studies explore the application of this validation technique in diverse sectors, including but not limited to healthcare, construction, and railway management, to enhance the generalizability and utility of our findings.

4.5 Conclusion

Our research offers a comprehensive exploration into the current validation methods applied to digital twins while proposing the usage of machine learning algorithms. Embarking on a novel approach to assess the fidelity of digital twins, this framework aims to establish a standardized validation protocol that could be universally applied, enhancing the reliability and credibility of digital twins across various domains.

The results of our internal case-study underscore the complexity of creating a reliable digital twin. The challenge of distinguishing between real and simulated datasets was addressed by setting an F-score threshold level of 0.5 for equivalence, which denotes the model's inability to differentiate between the two. This threshold was pivotal in establishing a baseline for comparison and further testing the model's classification capabilities under varied conditions.

Our findings reveal that while the model demonstrated a 50% accuracy rate in classifying data, it also exhibited the intriguing ability to misclassify data when exposed solely to one type of dataset, either real or simulated. This outcome not only highlights the intricacies involved in training ANNs but also the nuanced understanding required to interpret their performance.

The implications of our research are profound for the concept of digital twins. Establishing a standardized validation method is paramount to ensuring the cred-

ibility and reliability of digital twin systems. The trust placed in digital twins by users and stakeholders is contingent upon their ability to accurately mirror their physical counterparts. A digital twin that fails to meet established accuracy benchmarks may jeopardize its utility for predictive tasks such as maintenance scheduling and production planning.

Despite the promising advancements, it is crucial to acknowledge the nascent stage of the proposed correspondence measure and the absence of practical applications to date. Our limited expertise in executing machine learning algorithms may have influenced the outcomes, suggesting the need for further research with a more robust approach and interdisciplinary collaboration.

In conclusion, the study contributes to the ongoing discourse on digital twin validation by providing a proof of concept for the use of ANNs in this domain. Future research should aim to refine these methodologies and extend them to more complex, data-rich systems. The pursuit of a standardized, adaptable validation framework will likely be a collaborative effort across various domains, necessitating a shared understanding and commitment to rigorous testing protocols.

List of References

[1] M. Batty, "Digital twins," *Environment and Planning B: Urban Analytics and City Science*, vol. 45, no. 5, pp. 817–820, 2018. [Online]. Available: https://doi.org/10.1177/2399808318796416

[2] G. Lugaresi, S. Gangemi, G. Gazzoni, and A. Matta, "Online validation of digital twins for manufacturing systems," *Computers in Industry*, vol. 150, p. 103942, 2023.

[3] R. Al-Sehrawy and B. Kumar, "Digital twins in architecture, engineering, construction and operations. a brief review and analysis," in *Proceedings of the 18th International Conference on Computing in Civil and Building Engineering: ICCCBE 2020*. Springer, 2021, pp. 924–939.

[4] M. Grieves and J. Vickers, "Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems," *Transdisciplinary perspectives on complex systems: New findings and approaches*, pp. 85–113, 2017.

[5] M. W. Grieves, "Digital twins: Past, present, and future," in *The Digital Twin*. Springer, 2023, pp. 97–121.

[6] L. Wright and S. Davidson, "How to tell the difference between a model and a digital twin," *Advanced Modeling and Simulation in Engineering Sciences*, vol. 7, no. 1, pp. 1–13, 2020.

[7] T. H. Khan, C. Noh, and S. Han, "Correspondence measure: a review for the digital twin standardization," *The International Journal of Advanced Manufacturing Technology*, pp. 1–21, 2023.

[8] R. Söderberg, K. Wärmefjord, J. S. Carlson, and L. Lindkvist, "Toward a digital twin for real-time geometry assurance in individualized production," *CIRP annals*, vol. 66, no. 1, pp. 137–140, 2017.

[9] P. Muñoz, M. Wimmer, J. Troya, and A. Vallecillo, "Using trace alignments for measuring the similarity between a physical and its digital twin," in *Proceedings of the 25th International Conference on Model Driven Engineering Languages and Systems: Companion Proceedings*, 2022, pp. 503–510.

[10] W. Qian, Y. Guo, K. Cui, P. Wu, W. Fang, and D. Liu, "Multidimensional data modeling and model validation for digital twin workshop," *Journal of Computing and Information Science in Engineering*, vol. 21, no. 3, p. 031005, 2021.

[11] F. Emmert-Streib, "Defining a digital twin: A data science-based unification," *Machine Learning and Knowledge Extraction*, vol. 5, no. 3, pp. 1036–1054, 2023.

[12] M. Abdelatti and M. Sodhi, "Lab-scale smart factory implementation using ros," in *Robot Operating System (ROS) The Complete Reference (Volume 7)*. Springer, 2023, pp. 119–143.