ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ

ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ &
ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΙΤΛΟΣ

Χρήση της τεχνολογίας Ψηφιακών Διδύμων στις Παγίδες Εισβολών για την βελτίωση της Νοημοσύνης Απειλών στον τομέα της Κυβερνοασφάλειας

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

της

Νίντσιου Μαρίας

Επιβλέπων: Παναγιώτης Σαρηγιαννίδης

Αναπληρωτής καθηγητής

ΙΟΥΛΙΟΣ 2022, ΚΟΖΑΝΗ

HELLENIC DEMOCRACY
UNIVERSITY OF WESTERN MACEDONIA

FACULTY OF ENGINEERING
DEPARTMENT OF ELECTRICAL &
COMPUTER ENGINEERING

TITLE

Threat intelligence using Digital Twin Honeypots in Cybersecurity

DIPLOMA THESIS

by

Nintsiou Maria

Supervisor: Panagiotis Sarigiannidis

Associate professor

JULY 2022, KOZANI

ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ

ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ &
ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΙΤΛΟΣ

Χρήση της τεχνολογίας Ψηφιακών Διδύμων στις Παγίδες Εισβολών για την βελτίωση της Νοημοσύνης Απειλών στον τομέα της Κυβερνοασφάλειας

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

της

Νίντσιου Μαρίας

Επιβλέπων: Παναγιώτης Σαρηγιαννίδης

Αναπληρωτής καθηγητής

ΙΟΥΛΙΟΣ 2022, ΚΟΖΑΝΗ

HELLENIC DEMOCRACY
UNIVERSITY OF WESTERN MACEDONIA

FACULTY OF ENGINEERING
DEPARTMENT OF ELECTRICAL &
COMPUTER ENGINEERING

TITLE

Threat intelligence using Digital Twin Honeypots in Cybersecurity

DIPLOMA THESIS

by

Nintsiou Maria

***Supervisor**:* Panagiotis Sarigiannidis

Associate professor

JULY 2022, KOZANI

ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
& ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΔΗΛΩΣΗ ΜΗ ΛΟΓΟΚΛΟΠΗΣ ΚΑΙ ΑΝΑΛΗΨΗΣ ΠΡΟΣΩΠΙΚΗΣ ΕΥΘΥΝΗΣ

Δηλώνω ρητά ότι, σύμφωνα με το άρθρο 8 του Ν. 1599/1986 και τα άρθρα 2,4,6 παρ. 3 του Ν. 1256/1982, η παρούσα Διπλωματική Εργασία με τίτλο "**Χρήση της τεχνολογίας Ψηφιακών Διδύμων στις Παγίδες Εισβολών για την βελτίωση της Νοημοσύνης Απειλών στον τομέα της Κυβερνοασφάλειας**" καθώς και τα ηλεκτρονικά αρχεία και πηγαίοι κώδικες που αναπτύχθηκαν ή τροποποιήθηκαν στα πλαίσια αυτής της εργασίας και αναφέρονται ρητώς μέσα στο κείμενο που συνοδεύουν, και η οποία έχει εκπονηθεί στο Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών του Πανεπιστημίου Δυτικής Μακεδονίας, υπό την επίβλεψη του μέλους του Τμήματος κ. Παναγιώτη Σαρηγιαννίδη αποτελεί αποκλειστικά προϊόν προσωπικής εργασίας και δεν προσβάλλει κάθε μορφής πνευματικά δικαιώματα τρίτων και δεν είναι προϊόν μερικής ή ολικής αντιγραφής, οι πηγές δε που χρησιμοποιήθηκαν περιορίζονται στις βιβλιογραφικές αναφορές και μόνον. Τα σημεία όπου έχω χρησιμοποιήσει ιδέες, κείμενο, αρχεία ή / και πηγές άλλων συγγραφέων, αναφέρονται ευδιάκριτα στο κείμενο με την κατάλληλη παραπομπή και η σχετική αναφορά περιλαμβάνεται στο τμήμα των βιβλιογραφικών αναφορών με πλήρη περιγραφή. Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα. Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και μόνο.

Υπογραφή Φοιτητή: ΝΙΝΤΣΙΟΥ ΜΑΡΙΑ

Περίληψη

Την τελευταία πενταετία, έχουν αναφερθεί και καταγραφεί πολλές και κραυγαλέες περιπτώσεις κυβερνοεπιθέσεων σε εταιρείες, κυβερνητικές υπηρεσίες, ιστοσελίδες μέχρι και σχολές και πανεπιστήμια καθιστώντας ευάλωτες τις διαδικτυακές υπηρεσίες και την χρήση ψηφιακών συστημάτων. Έτσι, περισσότερο από ποτέ ο χώρος της Κυβερνοασφάλειας απαιτεί νέες λύσεις και εργαλεία για να προστατεύσει πιο αποτελεσματικά εμπιστευτικά δεδομένα, συστήματα και συσκευές όπως παραδείγματος χάριν οι Παγίδες εισβολών. Παρατηρήθηκε, ωστόσο, ότι η χρήση των Παγίδων εισβολών, παρ' ότι χρήσιμη και πιο σύγχρονη σε σχέση με άλλες μεθόδους, δεν αποτελεί λύση από μόνη της, καθώς οι στρατηγικές των επιθέσεων συνεχώς ανανεώνονται και τροποποιούνται ανάλογα με την περίσταση. Στο μεταξύ η ανάπτυξη στρατηγικής πρόληψης και αντιμετώπισης επιβραδύνεται από τη χρονοβόρα διαδικασία ανάλυσης των δεδομένων της επίθεσης. Αυτός ο χρόνος που χάνεται στην προσπάθεια ανάλυσης της πρώτης επίθεσης θα έπρεπε να δαπανάται κανονικά στην αποτροπή μιας δεύτερης. Η χρήση συνδυασμού μηχανισμών προστασίας δείχνει να είναι η καλύτερη πρακτική σε περιπτώσεις τέτοιων επιθέσεων. Μια πολύ καλή απάντηση σε αυτό το πρόβλημα αποτελεί η χρήση της τεχνολογίας των Ψηφιακών Διδύμων στις Παγίδες εισβολών για την βελτίωση της Νοημοσύνης Απειλών στον τομέα της Κυβερνοασφάλειας.

Σε αυτή τη διπλωματική εργασία, επιδιώκεται η γνωριμία με την πρωτοποριακή τεχνολογία των Ψηφιακών Διδύμων καθώς και η εφαρμογή της στον τομέα της Κυβερνοασφάλειας και πιο συγκεκριμένα πάνω σε Παγίδες εισβολών με σκοπό την συνεχή βελτίωσή τους. Στη συνέχεια προχωράει στην πρόταση ενός framework Ψηφιακού Δίδυμου βάση του οποίου θα γίνει η βελτίωση μιας Παγίδας εισβολών. Στο τέλος, παρουσιάζονται μέθοδοι και εργαλεία που είναι διαθέσιμα για το σχεδιασμό ενός μοντέλου Υπολογιστικών Δεδομένων και ενός μοντέλου Αναπαράστασης Δεδομένων για το framework.

Λέξεις-κλειδιά: Κυβερνοασφάλεια, Ψηφιακός Δίδυμος, Παγίδες εισβολών, κυβερνοεπιθέσεις, Ψηφιακός κλώνος

Abstract

During the last five years, many high-profile cyber-attacks on companies, government agencies, websites and even institutions and universities have been reported and recorded, making online services and digital systems vulnerable. Thus, more than ever, the field of Cybersecurity requires new solutions and tools to more effectively protect confidential data, systems and devices such as the Honeypots. It has been observed, however, that the use of Honeypots, although useful and more modern than other methods, is not a solution by itself, as attack strategies are constantly updated and modified according to the occasion. Meanwhile, developing a prevention and response strategy is being slowed down by the time-consuming process of analyzing attack data. This time lost in analyzing the first attack should normally be spent preventing a second one. Using a combination of protection mechanisms seems to be the best practice in cases of such attacks. A very good answer to this problem is using Digital Twins technology in Honeypots to improve Threat Intelligence in the domain of Cybersecurity.

This thesis aims at the acquaintance with the pioneering technology of Digital Twins as well as its application in the field of Cyber Security and, more specifically, on Honeypots to continuously improve them. In the end, methods and tools available for designing a Data Computational and a Representation model for the framework are presented.

Keywords: Cybersecurity, Digital Twin, Honeypot, cyber-attacks, Digital clone

Abbreviations

AI	Artificial Intelligence
API	Application Program Interface
APT	Advanced Persistent Threats
CDT	Cyber Digital Twin
CPDT	Cyber Physical Digital Twin
CPS	Cyber Physical System
DT	Digital Twin
HTTP	HyperText Transfer Protocol
ICS	Industrial Control System
IDS	Intrusion Detection System
IDPS	Intrusion Detection and Prevention System
IoT	Internet of Things
IPs	Internet Protocol addresses
IPS	Intrusion Prevention System
M2M	Machine to Machine
NASA	National Aeronautics and Space Administration
OPC	Open Platform Communications
PLC	Programmable Logic Controller
PLM	Product Lifecycle Management
SCADA	Supervisory Control and Data Acquisition
SMT	Surface Mount Technology
SMTP	Simple Mail Transfer Protocol

Acknowledgements

Firstly, I would like to express my deepest gratitude to my family for the extensive support during my last five years of university and this thesis. I am thankful to the University of Western Macedonia for the journey of knowledge that I experienced and the opportunities it provided for self-improvement in terms of engineering know-how and skills development that I acquired.

I would also like to express my sincere thanks to my supervisor, Assistant Professor Panagiotis Sarigiannidis, for his guidance and Mr. Paris-Alexandros Karypidis and Ms. Elisavet Grigoriou. Their assistance was invaluable in conducting the thesis work with their prompt, targeted and motivating comments.

Table of Contents

List of Tables

List of Figures

Chapter 1. Introduction

Cybersecurity or information technology security is a domain that aims to protect computers, network devices and data from unauthorized access. It usually combines two fundamental modules to defend against the “bad guys”:

- Detection
- Prevention

Digitization has been a turning point for every company and organization, but during the past decades, little preparation was done to protect all the information that went suddenly online. Over the last years, there has been an intense need for cybersecurity experts and tools to create a sturdy wall of defence for the facilities and detect potential threats due to all the cyber-attacks that took place on a large scale and shocked the global society.
This thesis proposes a new cybersecurity framework that combines modern technology to detect and prevent the cases above. Specifically, a Digital Twin (DT) is deployed to create a virtual twin of a honeypot that will assist in its optimization.

1.1 Thesis Objectives

- Research on State-of-the-Art DT technology in cybersecurity and Honeypot-related frameworks
- Propose a DT framework to support the Honeypot features and optimize it
- Suggest tools available to design Data Computational and Representation models for the development of the proposed framework

1.2 Thesis Structure

The thesis consists of 6 chapters. Chapter 1 includes the introduction of the topic, the thesis objectives and the thesis structure. Chapter 2 presents background on DT technology and cybersecurity threats, and previous detection and prevention measures. Chapter 3 shows related works that include DTs' theoretical background and frameworks that deploy DTs. In Chapter 4, a DT Honeypot framework is proposed. Chapter 5 includes suggested tools which will assist in

realizing the DT Representation and Computational models for the proposed framework. Chapter 6 concludes this thesis and presents ideas for the future development of the framework.

Chapter 2. Background

This chapter presents background on DT technology, its characteristics and current cybersecurity threats, detection and prevention measures.

2.1 Digital Twins Technology

In this section, fundamentals, theoretical background and definitions of DTs will be presented. Secondly, the components that compose the DT and types of DTs will be analyzed to get the gist of what a DT is and how this technology correlates or differentiates itself from other known technologies.

2.1.1 Fundamentals of Digital Twins

NASA first adopted the use of twins to rescue the Apollo 13 mission (1970) trapped in space due to damage to the main engine of their spacecraft. This twin, on Earth, served as a representative of the space counterpart and was utilized to test and simulate the condition of the spacecraft when the main engine was damaged. NASA had created various simulation machinery that had been used for training before the mission and also had a mock-up of the spacecraft on Earth [1]. When the problem appeared, they configured the existing simulators to match the condition of the damaged spacecraft and managed to recreate it digitally. By applying different simulation scenarios, NASA managed to provide a solution and salvage the mission. This concept of creating a twin and using it to simulate the conditions of its counterpart with real data is considered the precursor of the DT concept [2],[3].

- The idea of recreating the real world inside a virtual environment came from Gelernter (1991) as a description in his book ‘Mirror Worlds’ where the concept of twins was introduced [4].
- The DT concept was informally introduced in 2002 by **Michael Grieves** during his university presentation about product lifecycle management (PLM) with the title ``Conceptual Ideal for PLM" based on his work with John Vickers. The idea was to develop a digital model of a product to move from the primary and manual product data

[5] to constitute a digital base for life-cycle management. According to him, the DT consists of **three modules**: "a physical product in Real space, a virtual representation of that product in the Virtual space and the connections of data and information that tie the virtual and real products together" [6].

- In 2010, NASA initially defined DTs in a Roadmap report as "an integrated multi-physics, multi-scale, probabilistic simulation of a vehicle or system that uses the best available physical models, sensor updates, fleet history, etc., to mirror the life of its flying twin. It is ultra-realistic and may consider one or more important and interdependent vehicle systems"[7].

The definition of the DT concept has been debated, and many other attempts to define it have been made. Recent years (2015-2016) have included two different ideas. On the one hand, according to Abramovici et al. [8] and Schroder et al. [9], it is presented as a **digital clone** replicating a real-world object's characteristics and behaviours and recreates them as a final product in a virtual environment. On the other hand, Gabor et al. [10] and Rosen et al. [11] have stated that a **DT is the whole lifecycle environment of a production process** of an object which provides various functionalities such as monitoring, simulation and management in real-time.

Combining the two concept definitions into one, a DT can be described as a *product* and a *product's lifecycle*. According to Grieves and Vickers [12] in their newest paper (2017), DTs are real-time and remotely connected digital equivalents of physical objects, providing a rich representation and comprising their dynamic behaviours. Bochert and Rosen, in their paper [13], explain that using the term dynamic to define the DTs means that, apart from the **current state and behaviour**, the **simulation and prediction of future states and behaviours and the recollection of historical data of behaviours** are incorporated in the term as well. The term Digital Twin in its latest application fields does not only refer to the recreated model as a structure but also to the details and features of the model. Digital Twins are able to simulate the reactions to several tests and also "monitor the existing objects or products with their current states and processes" [14].

Since then, many authors have given definitions and descriptions similar to the previous ones. A most recent paper seems to be collecting all of the information of the past attempts into a single definition: "A Digital Twin is a set of **virtual information constructs** that **mimics the structure, context, and behaviour of an individual/unique physical asset**, is **dynamically updated** with data from its physical twin **throughout its lifecycle**, and **informs decisions that realize value**." [1].

A collective DT concept and definition timeline is presented in Figure 1 below.

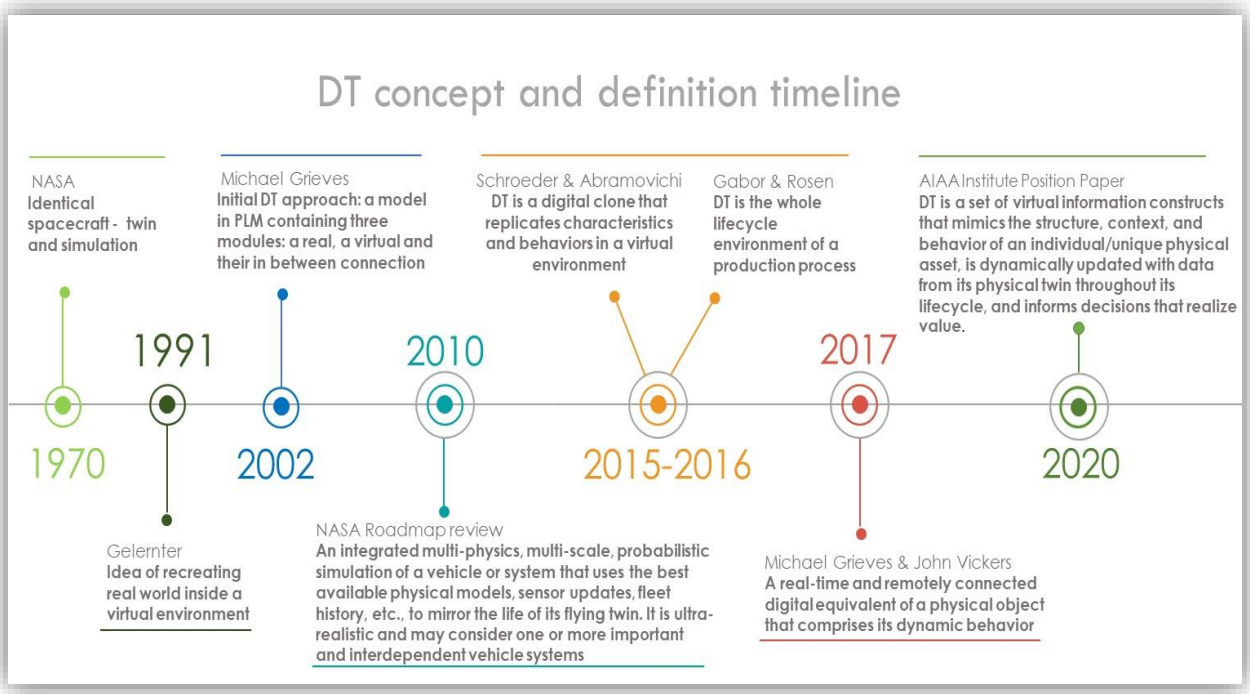

Figure 1. *DT concept and definition timeline*

2.1.2 Lifecycle of a DT and some necessary functions/tools

Digital Twins originated mainly from the industrial production area. They were deployed for the enhancement of a product's design phase and for simulating its behaviours in different conditions before the manufacture time, which otherwise would lead to the development of costly physical prototypes [15], [16]. After the **design phase**, a DT comes to life as a virtual object based on its design characteristics. A real-world product is created. A DT then moves to the **operational phase,** where it monitors the real-world product's current and historical state and behaviours with the use of sensors. Moving on to the **operational phase**, a DT can be utilized as an environment for management and simulations to prevent deviations from a product or malfunction. At the end, when a product is no longer needed, a DT passes on its last phase, the **disposal phase**. During this phase, a DT disconnects itself from the real counterpart and can either be preserved as a guide for newer versions of a product or be disposed of as it is no longer needed [17]. This phase can be extremely important for a company that decides to preserve the DT as mistakes in newer product versions can be traced back to their roots by retrieving older models and figuring out what went wrong through testing them. This lifecycle of a DT is demonstrated in Figure 2, where the different phases and the tools needed for each one are shown.

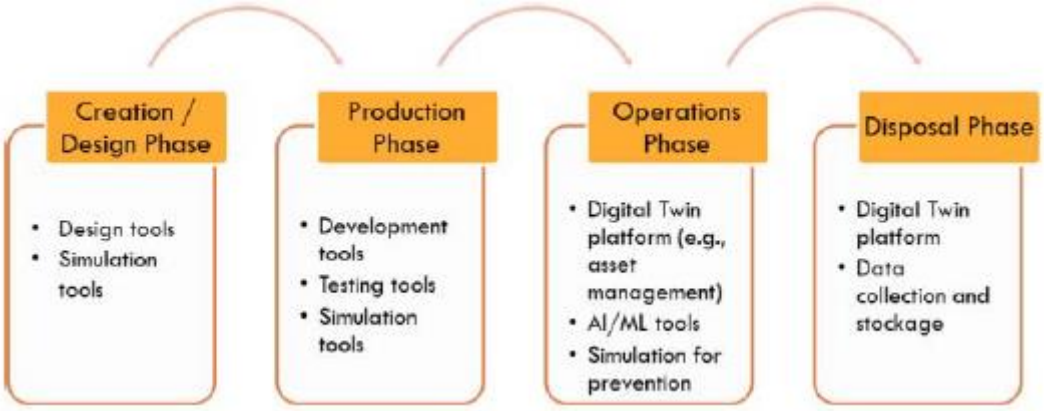

Figure 2. *A DT is being utilized in various manufacturing phases [17]*

2.1.3 Components of DT

To acknowledge the concept of a DT, it is necessary to go through the work of Grieves, who first introduced and analyzed its components by providing an explanation for each one. According to him, a DT consists of three components:

1. Physical product.
2. Virtual-digital part
3. The connection between the physical and virtual parts

As mentioned before, the **physical** product refers either to the product itself or its lifecycle. The **virtual part** implies the virtual counterpart that represents the physical one, and the **third part** refers to a bidirectional connection which serves as a way of data transferring between them. Those components discussed above are the ones that constitute a DT which means that if one of them is missing, the product cannot be characterized as a DT anymore.

However, additional components for a DT add functionality and are set according to the domain a DT is being applied to. Those components indicate the usefulness of a DT in various domains, which incorporate machine learning and the Internet of Things. One is the **DT performance evaluation**, which uses metrics such as accuracy, resilience, robustness and costs, evaluation methods and tests. A second one is machine learning to provide experts with predictions and feedback and propose mitigation strategies. On this occasion, a joint optimization feature is required for all DT subcomponents mentioned earlier. The last one is IoT devices that a DT uses to collect data from sub-components of the physical product. On this occasion, the high-fidelity

connection between the devices is required for an "Accurate and timely flow of information"[18].

One can understand that the DT definition is not affected by the absence of any of the additional components, but they need to be mentioned when used as they add to the overall performance of a DT.

2.1.4 Properties of a DT

A DT has its properties to work properly and results in a product that is not *only* a digital clone of an object. These properties can be categorized as either ***necessary*** or ***dynamic*** based on the type of assets they add to the digital clone. Necessary properties, as the adjective indicates, are fundamental for every DT to ensure that it runs properly in real-time situations. They are the following terms:

- **Real-time connection** with the physical object. As said before, this property defines a DT and is an integral part of it.
- **Self-evolution** is a property that is applied to enable self-adaptation and learning in real time. This can help a DT provide feedback on both ends, physical and digital objects, allowing for self-recreation and self-remodelling according to that feedback.
- **Availability** of time-continuous data for monitoring and as a machine learning input
- **Continuous machine learning analysis** makes space for more accurate, and real-time output forecasting as the data being analyzed are fresh and provide a better outlook about the current states of the physical object and the clone.
- **Domain-specific services** differentiate a DT from others by setting priorities or deciding to support the service according to the domain the DT is being used for.

Dynamic properties are the basis for creating a hierarchy for the DT. The way a hierarchy of a DT is built depends on the degree or the modes of two attributes: autonomy and synchronization. When it comes to setting a DT to provide information, process it and make a decision upon the outcome of that processing, there are different ways to do so.
According to the degree of autonomy given to the DT, there are three categories: ***autonomous, partly autonomous*** **and** ***not autonomous***. That simply means that a DT of the first category is set to act completely and make decisions based on its own 'intellect'. In the second category, a DT is observed by a human while working, giving results and improving itself and is assisted in

the decision-making part, whereas in the third category, it needs strictly human approval of each step of the processes and decisions to be made. This degree of freedom affects, of course, the self-evolution of a DT due to the extent of machine learning involvement. Self-evolution is more beneficial in the first category.

Synchronizing data continuously or at intervals can also influence the creation of the hierarchy of a DT. The difference in the data synchronization in the DT sub-components can result in different hierarchies based on the self-evolution, response and decision-making it will make. This usually depends on the resources available, the type of machine learning algorithm being utilized or the frequency of acquisition and the quantity requirements of data for this particular DT model. So, one understands that another reason might be the regularity of a DT update and the data storage. If a DT updates itself frequently and stores data continuously, then more resources are needed, and more components are added to the hierarchy of a DT in order to work properly [18].

2.1.5 Types of DTs

DTs can be categorized according to their production phases and the level of integration.

2.1.5.1 Based on production phases

Digital Twins are of three types and are depicted in Figure 3 below:

1. Digital Twin Prototype
2. Digital Twin Instance
3. Digital Twin Aggregate

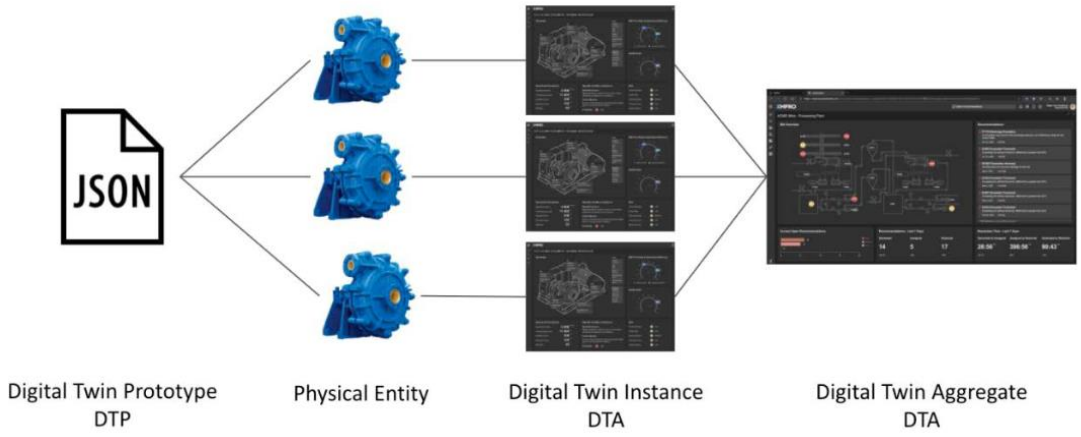

Figure 3. *Relationship between DT concepts [19]*

Grieves defined the following key terms to characterize the digital twin and indicate the difference between them:

1. **Digital Twin Prototype (DTP):** A DTP constitutes a representation model of the physical object that will be manufactured and thus contains a description and an information model about it. The DT can be implemented, before the existence of a real product, as a prototype and can be put through tests and simulations for the product model to meet the desired expectations. By creating a DTP, companies and organizations have an early product prototype and can apply changes without having extra prototype costs. In reality, it is usually the other way around, where a product prototype is already manufactured, and the DT is created afterwards for simulation and testing. For example, a DTP could contain a 3D model of a physical object depicting its assets and a description of how to manufacture it. A DTP is the design, analysis and information map for the first stage of the manufacturing process.

2. **Digital Twin Instance (DTI):** A DTI is about a particular physical instance of an object. It could include a list for enumerating individual object parts used in the production of this specific instance of an object and each and every process step followed during its production. The current operational state of the object instance could be included as well. Using one DTP is enough to create multiple physical objects, each of which can have its DTIs.

3. **Digital Twin Aggregate (DTA):** A DTA is an accumulation of multiple DTIs which enables querying information about a group of objects [19].

2.1.5.2 Based on the level of integration

The DT had precursors before reaching its final form, and because of them, there was confusion about their real identity. Those precursors are depicted in Figure 4 and are analyzed below.

Digital Model

The "Digital Model" was the first term in scientific literature. A digital model is a digital representation of a planned or existing physical entity. It is clear that its definition does not include the third fundamental module Grieves & Vickers mentioned in the DT definition; there is a connection between the digital and the physical entities and data transfer between them in both directions. The digital model concept suggests that if a slight alteration is done on the physical entity, then the digital model won't project the same alteration on itself but will remain the same, meaning that changes in the real world are not applied in the digital world and vice-versa.

Digital Shadow

"Digital Shadow" is the name of another precursor of the DT and is defined as a digital representation of a physical entity with a one-way flow of data: from the physical to the digital entity. A change in the real world will be visible in the digital but not the other way around.

Digital Twin

As mentioned in Section 2.1.1, the DT is a digital representation of a physical entity with a bidirectional connection between the digital and the physical entities. A change in both the real and the digital world will be visible in the other due to their connection [20].

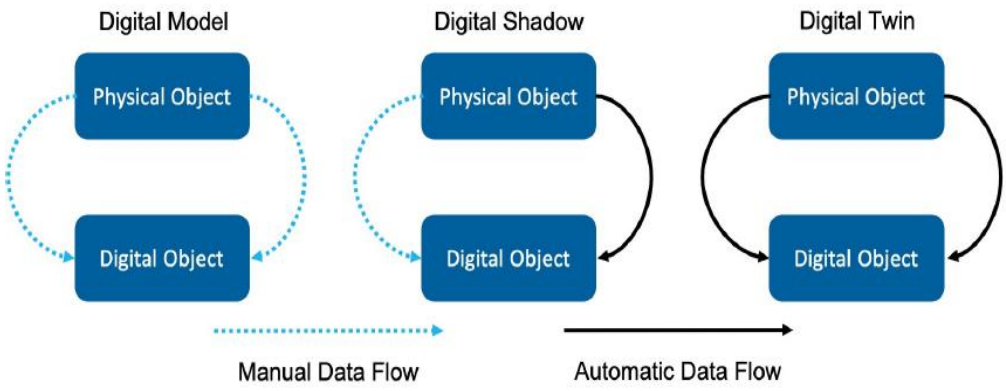

***Figure 4**. Digital Model, Shadow and Twin [20]*

2.1.6 DT typology

While the DTs are connected to their physical counterpart, they are useful for monitoring the actual state of the object, predicting future conditions and states, and remotely improving its condition. The existence of a DT, though, doesn't necessarily start with the manufacturing process of the physical object "to be twinned" and ends when it is finished. It can be created beforehand to define and simulate the different behaviours a physical twin might have and therefore notify and inform about them. After the manufacturing process is finished, a DT is still alive, and its tasks are a recollection of historical states and data from the physical twin and simulations to prevent deviant behaviours. According to the focus on each task, DTs can be of various types [21]:

- **Imaginary DT** is a conceptual entity that portrays a non-existent object. It contains all the information necessary to realize the object, including 3D models and specifications about materials and resources, and is available for simulations on those models.
- A **Monitoring DT** constitutes a virtual representation of the behaviours and state of an existing physical object. It is constantly connected to the object and monitors its condition, functioning and external environment.
- A **Predictive DT** estimates and computes an object's future states and behavioural characteristics with the assistance of predictive analytics, namely machine learning methods, statistical forecasting and simulation, based on real-time data acquired by the object.
- **Prescriptive DT** is a smart digital object that adds intelligence for recommendations and prevention measures based on optimization algorithms and expert heuristics. The output of Monitoring and Predictive twins is inputted to a Prescriptive DT to give suggestions on

which courses of action need to be taken. They generally assist humans in decision-making tasks and on either on-site or remote interventions.

- An **Autonomous DT** takes full control over the behaviours of the physical object and operates autonomously without human intervention. This type of DT can learn about the environment by becoming self-adaptive, conducting self-diagnosis on its own service needs and adapting to certain user preferences.
- **Recollection DT** is a memory hub that preserves the complete history of a physical object which doesn't exist anymore. The importance of those DTs lies in their ability to retrieve data from past states or versions of the object and, in this way, recreate the object in that specific state or version.

After seeing each type of DT individually, see Figure 5 it is quite noticeable that a DT is not exclusively categorized as predictive or monitoring, prescriptive or imaginary. During the different production and product maintenance stages, a DT combines most of the titles mentioned above as it is assigned various tasks.

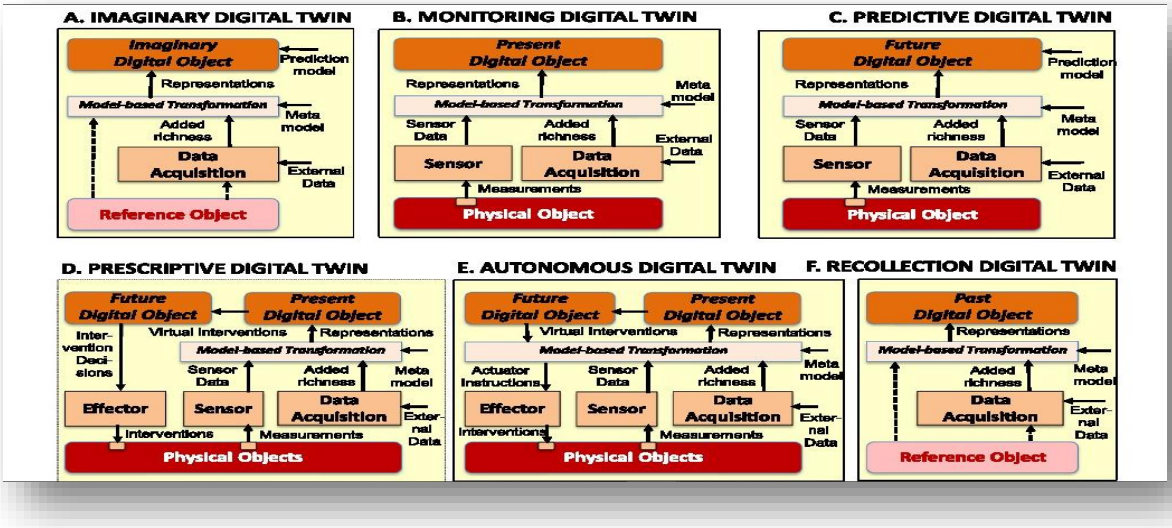

Figure 5. *DT Typology according to the different production stages [21]*

2.1.7 Technology stack for DT creation

Figure 6 collects all the different individual tools and methods that are combined for the creation of a DT. The tools and methods of ***Data management and connectivity*** are used to constitute the two-way communication of the DT between the physical and the virtual counterparts. The **Data representation task** is done using tools and methods such as ontologies and data formats.

Computational processing and machine learning methods and analytics build the **Data computation logic** of a DT and decision-making processes. Lastly, through validation, simulation and visualization of a **digital model, microservices** can be demonstrated with the help of 3D and virtual reality designs [22].

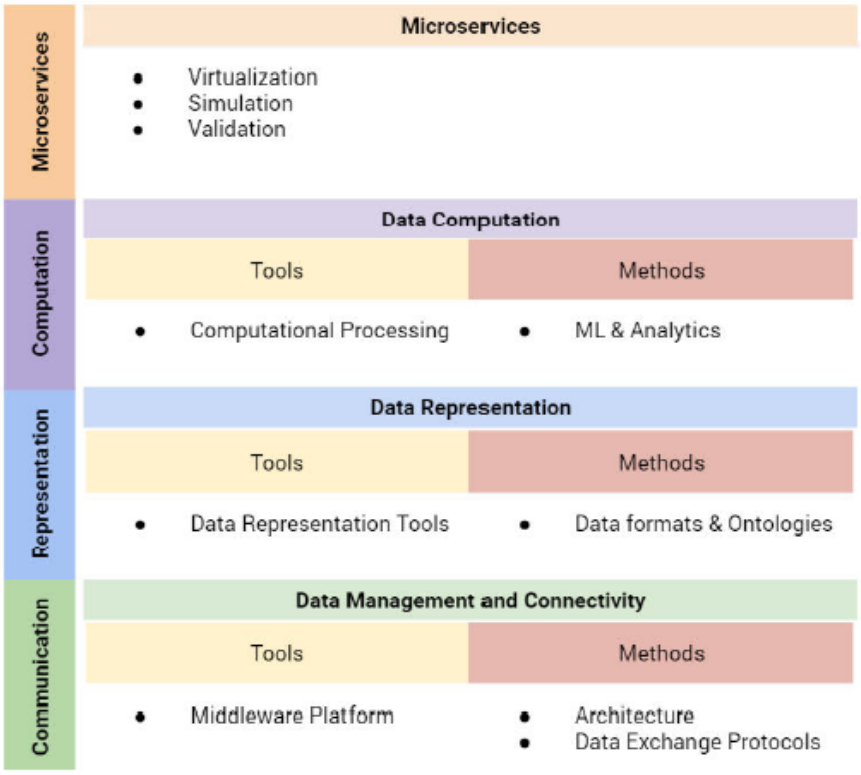

Figure 6. *Technology stack for DT creation [22]*

2.1.8 Enabling technologies for digital twin data management

In Figure 7, the core processes of a DT are depicted with specific parameters. A DT depends on **data collection** and **transmission** between the different components, **storage** of this data for processing and, in the end, a **fusion** of data with similar context and **visualization**. Thus, a DT depends on relative technologies that support the operations within its architecture. Examples of such technologies are shown for each operation included in a DT.

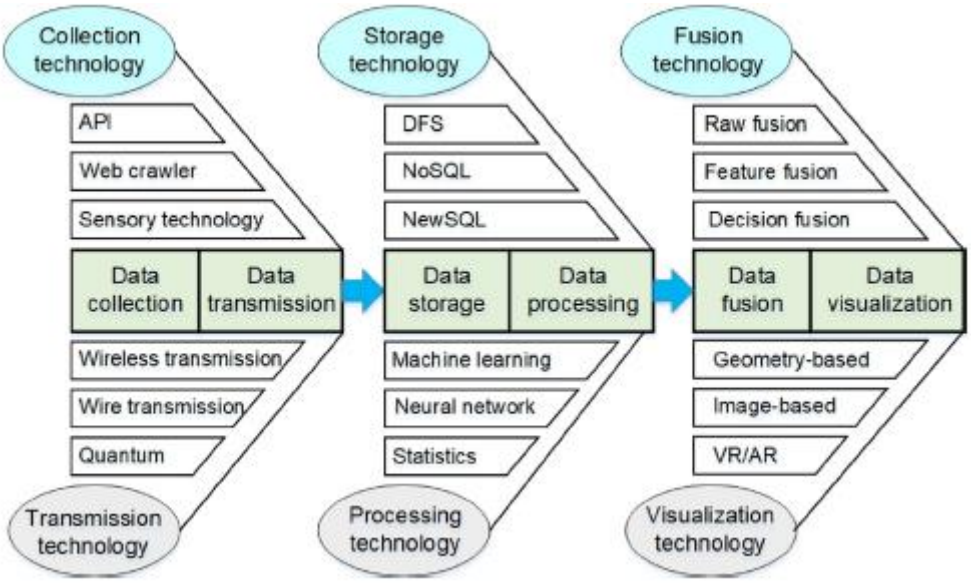

***Figure 7**. Core processes and technologies that enable DT data management [23]*

2.1.9 Digital Twin – Context Diagrams

Having seen the core technologies and processes that constitute the DT module structure, one can understand their traits, their relationship with the leading-edge technologies and the various use case scenarios that come as an advantage with their deployment.

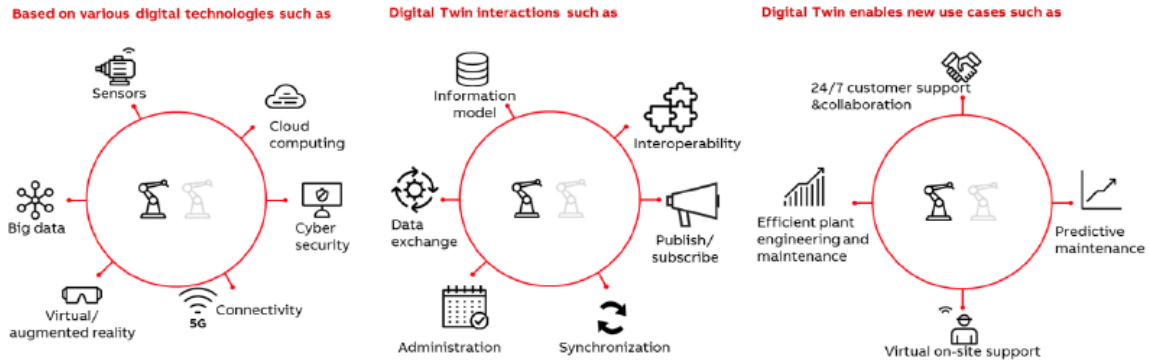

***Figure 8.** Depiction of DT's technological background, traits and benefits [24]*

As shown in Figure 8 above, DTs are growing based on the most recent digital technologies, can be interactive and enable real-time services and applications. The DT technology combines characteristics that are handy for efficiently predicting, planning and responding to tasks in real-time.

2.1.10 DT's Summary Table

Table 1. DT technology questions and answers [25]

Research Question	**Results**
'What is a Digital Twin?' ***DIGITAL TWIN DEFINITION***	"A set of adaptive models that emulate the behaviour of a physical system in a virtual system getting real-time data to update itself along its lifecycle. The digital twin replicates the physical system to predict failures and opportunities for changing and prescribe real-time actions for optimizing and/or mitigating unexpected events by observing and evaluating the operating profile system."
'Where is it appropriate to use a Digital Twin?' ***DIGITAL TWIN CONTEXTS and USE CASES***	1. Healthcare • Improving operational efficiency of healthcare operations 2. Maritime and Shipping • Design customization 3. Manufacturing • Product development and predictive manufacturing 4. City Management • Modelling and simulation of smart cities 5. Aerospace • Predictive analytics to foresee future problems
'Who is doing Digital Twins?' ***DIGITAL TWIN PLATFORMS***	GE Predix; SIEMENS PLM; Microsoft Azure; IBM Watson; PTC Thing Worx; Avera; Twin Thread; DNV-GL; Dassault 3D Experience; Sight Machine; Oracle Cloud

'When is it necessary for a Digital Twin to be developed?' ***DIGITAL TWIN LIFE CYCLE***	1. In design phase • The DT is used to help designers to configure and validate more quickly product development by accurately interpreting the market demands and the customer preferences. 2. In production phase • The DT shows great potential in real-time process control and optimization, as well as an accurate prediction. 3. In service phase • The DT can monitor the health of a product and perform diagnosis and prognosis.
'Why should a Digital Twin be used?' ***DIGITAL TWIN FUNCTIONS***	
'How to design and implement a Digital Twin?' ***DIGITAL TWIN ARCHITECTURE AND COMPONENTS***	1. The Physical layer involves various subsystems and sensory devices that collect data and working parameters. 2. The Network layer connects the physical to the virtual by sharing data and information. 3. The Computing layer consists of virtual models emulating the corresponding physical entities.

In Table 1, one can see the various fields in which DT technology is considered the most productive and suitable. DTs are used by well-known technology companies and organizations not only to be utilized during the creation process of a product or service but also for monitoring and testing various scenarios.

Testing a product-to-be or an already manufactured one gives the testers and developers the ability to portray possible malfunctions and dangerous behaviours and, in the case of cybersecurity, respond to security threats. Evaluation of security requirements and detection of vulnerabilities are also challenging, and time-consuming tasks accelerated when automation is deployed. The above reasons indicate why DTs have suddenly appealed to various fields of the virtual world.

DTs can be further employed for protecting critical infrastructure by duplicating the device or network that is prone to attacks which provide additional information for the security experts to conduct:

1. Detection
2. Investigation
3. Threat Prevention

The above actions can be implemented on the duplicate before an attack. This helps the security operators to gain time and give a thorough solution to the security leaks they find in their research. To better understand why DT technology is radically being applied in cybersecurity, one has to delve deeper into cybersecurity experts' challenges and acknowledge the tools available to do so.

2.2 Cybersecurity

In this section, cybersecurity threats, detection and prevention mechanisms will be described. This is done to fully grasp the current situation that cybersecurity experts have to deal with by showing the difficulties and the available tools.

2.2.1 Threats, detection and prevention measures

Cybersecurity challenges and pending threats have been raging due to the lack of information for professionals to study to address them properly. Moreover, cybersecurity experts usually face multiple threats during a single attack and need actual strategies to do that effectively. These threats vary from:

- **Malware** (malicious software): *"All software or firmware inserted into a system without the user's knowledge, allowing the theft of information, corrupting the functions of the equipment or evading the mechanisms implemented to control access to it."* [26]. Malware threats are generally known as worms, botnets and viruses, Trojan horses and ransomware and are usually the cause of network security and social network damages.
- **Denial of Service (DoS):** A cyber-attack floods the bandwidth or resources of a system, so it can’t respond to requests and is unable to function properly. A subclass of the DoS

attack is the Distributed DoS attack which is a DoS attack performed by multiple systems that are all targeting a specific victim at the same time [27].

- **Man in the Middle**: "An attack in which an attacker is positioned between two communicating parties to intercept and/or alter data travelling between them. It is a form of active wiretapping attack in which the attacker intercepts and selectively modifies communicated data to masquerade as one or more of the entities involved in a communication association." [28].
- **Phishing**: "Tricking individuals into disclosing sensitive personal information through deceptive computer-based means" [26]. This means it is "a technique for attempting to acquire sensitive data, such as bank account numbers, through a fraudulent solicitation in email or on a website, in which the perpetrator masquerades as a legitimate business or reputable person." [29].
- **SQL Injection**: "Attacks that look for websites that pass insufficiently-processed user input to database back-ends." [30] Also known as SQL insertion attack, it is a malicious code injection technique that exploits a device's weaknesses. It usually targets a website's SQL-based application software and injects malicious SQL queries [31].

Each of these attacks can be treated individually or with a combination of tools. Research [32] on them has shown that when prevention and detection measures are taken into account before such attacks try to compromise a device or network, the outcome lies on the defender's side, or at least less damage is done.

In order to protect against the most notorious ones, a few security tasks are mentioned. For malware attacks, it is advised to employ up-to-date anti-malware and anti-spyware systems along with firewalls and automated monitoring tools and not miss out on software updates that fix security issues. Firewalls monitor traffic inside a network by allowing or blocking it based on security rules; because of that, they cannot detect potential unknown dangers alone. Regarding DDoS attacks, there are also a few things that can be done to protect servers and infrastructures. Firstly, increasing the bandwidth can save the case of a server crashing and leveraging a CDN Solution will provide extra security. Secondly, switching to a hybrid or cloud-based solution can result in unlimited bandwidth and blocking the incoming traffic from outside the network can keep these attacks away and protect the hardware.

2.2.2 Intrusion Detection and Prevention system

Security experts know that a firewall does not protect facilities from network attacks such as Denial of Service attacks on “open” ports. An intrusion attack might be unauthorized access to files, data or privileges or destabilization of a network. IDPs are the cybersecurity answer when it comes to detecting abnormal activities and preventing similar situations to those mentioned above. An intrusion detection and prevention system can be either software or hardware configured to protect single systems or entire networks and is capable of detecting and attempting to prevent attacks [33].

An Intrusion Prevention System is an evolution of IDS that analyses, detect, and can take preventive measures on its own when an attack bypasses encryption and authorization protocols or methods that secure a system. An IDS is capable of monitoring, analyzing, logging system activity and informing security experts about anomalies detected in the system in real-time. According to scientific literature [34], Intrusion Detection Systems can monitor and analyze the system and the user activity. They are used together as a defence mechanism, as an IPS can be used for detection only, and thus the term Intrusion Detection and Prevention System (IDPs) includes them both and will be used to describe them [35]. Concluding, IDPs can identify possible threats and log information about them, creating reports for security experts while taking preventive measures to try and stop or avoid them simultaneously.

Mainly there are three parts to IDPS processing methodology, and those are: collection and preprocessing of information and data, classification analysis of them and protection against threats. An IDS and, consequently, an IDPS consists of three components to implement the above processing methodology:

- **Sensors-agents**: they are network modules that are responsible for the collection and preprocessing of information in a system
- **Analysis Engine**: the core component of an IDS, a module that analyses and classifies information and data, processes them and detects an incident of attack or malicious behaviour
- **Response Module**: the component that gets the information outcome from the Analysis engine and notifies the security experts. This module can perform limited actions, like activations of rules in a firewall, to diminish an attack and prevent intrusions of this kind on the system.

2.2.3 Honeypots

Honeypots are generally defined as “information system resource whose value lies in unauthorized or illicit use of that resource” [36]. As the name suggests, “honeypots” are created to serve as a mechanism to attract all the possible attackers who want to get the taste of the sweetness of intruding into a system. **Honeypots can mimic the behaviours and responses of a real system or device and trap attackers to block their way to the desired destination**. Their main goal is to deceive them and put them into a virtual surveillance box from which intelligence about their activities and their methods are collected and recorded.

The infrastructure of a honeypot, though, does not only rely on a virtual environment but also consists of real devices to seem more realistic to the attacker’s eye. Figure 9 shows a honeypot, which is, in fact, an extra server that draws to itself all abnormal internet traffic from the Firewall and protects the real server from attacks.

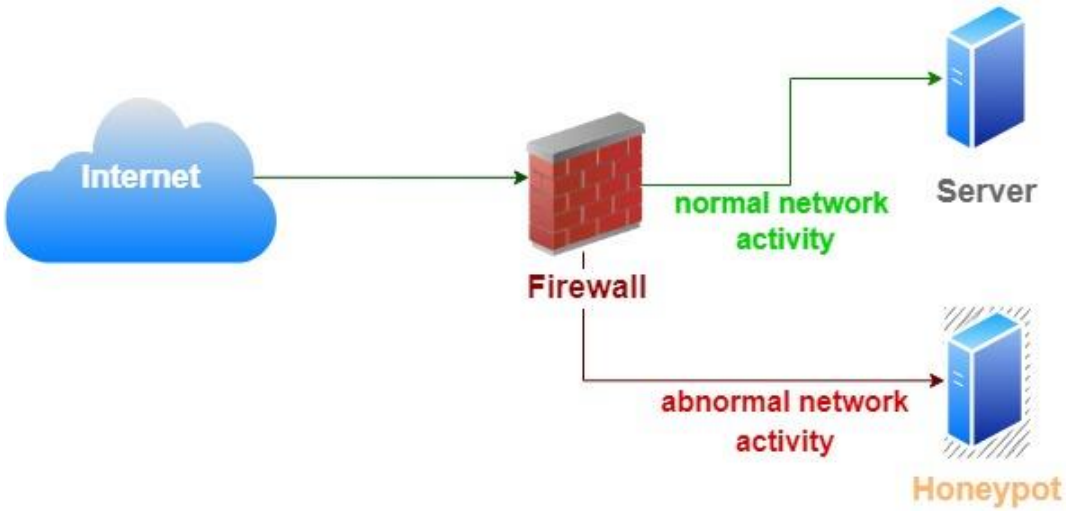

Figure 9. *The firewall sends the abnormal internet requests to a honeypot server*

It usually looks similar to a part of a network of devices, but underneath this cover, it is an isolated observation system. These honeypots are categorized as **physical,** while the other ones are called **virtual** honeypots. Their difference lies in the higher responsiveness that the virtual honeypots offer and the reliability the physical ones show, as they are tough to distinguish from a real target due to their physical existence in the network.

Honeypots can also be classified as ***production*** **and** ***research*** according to their design for different production or research purposes.

- **Companies mainly utilize production honeypot**s due to their ease of use. Their purpose is to ameliorate a company's security systems by imitating the reactions of a real company's assets (e.g. devices, services, operating systems) and gathering information about the attack simultaneously.
- **Research honeypots** have a complex implementation and are focused on collecting a mass of data compared to the poor data collection of the production ones. They are a strong intrusion detection tool that provides the defender with the strategies and the information of the attack to acknowledge existing weak points [37].

Honeypots can further be categorized based on **the level of their interaction** with the attacker in the next four categories:

- **Pure honeypots** are full-scale systems that imitate the production system and run on various servers. They have several sensors used to track and observe the attacker's activity. Data within a pure honeypot is made to look confidential in order to attract an attacker.
- **Low-interaction honeypots** simulate a device or service by providing little interaction to attract an adversary. Usually, they emulate some functions that seem realistic but can easily be recognized by an attacker when using other ways to infiltrate it than expected. Their main purpose is to collect statistical information about attacks.
- **Medium-interaction honeypots** offer attackers more ability to interact than low-interaction honeypots, but their main cause stays the same as the previous ones. Their difference is that they provide great functionality and trick the attackers into proceeding with their attack methods and unveiling their target.
- **High-interaction honeypots** are the whole package. They are the most advanced type, usually portraying a whole operating system. The adversary can highly interact with it, compromise it and utilize its functions at will while his actions, targets and motives are being recorded. Although this type of honeypot seems to be the most promising in terms of interaction, it can also be the most difficult to set up, exploited by seasoned hackers to reach the real system and needs constant monitoring to prevent such activity [38].

The different interaction levels of Honeypots' interaction levels [39] are depicted. Low interaction honeypots can operate as a fake server with limited connection abilities, medium interaction honeypots can send requests to the operating system and some other resources, and high interaction ones are operating systems the

According to their **placement inside a network,** they can be defined as:

- **Server honeypots** lure the potential intruder with the cover of several server services or client-side software ready to be compromised. This type of honeypot allows for better interaction for the intruder on the one hand and easier collection of data about his actions for the cyber experts on the other hand.

- **Client honeypots** answer malicious web servers that can compromise client machines from a single request to a website they are hosting. These honeypots are designed to interact with such servers by secluding the web browser from the honeypot system. This kind of server is detected by a client honeypot when there are changes to a list of files, directories and registry entries after an interaction with the server. So this honeypot initiates an interaction with remote servers and identifies whether the server is malicious or not.

  Server honeypots, though, are exposed, demand high resources and can be detected by the attackers. Client honeypots can also generate false alerts, and their performance speed is slow compared to the speed of the attacks [40]**.**

- **Hybrid honeypots** come to the rescue by combining both server and client sides. They can collect a large number of attacks by working with both sides, which provides a bigger scope for a better understanding of the attack incidents [41].

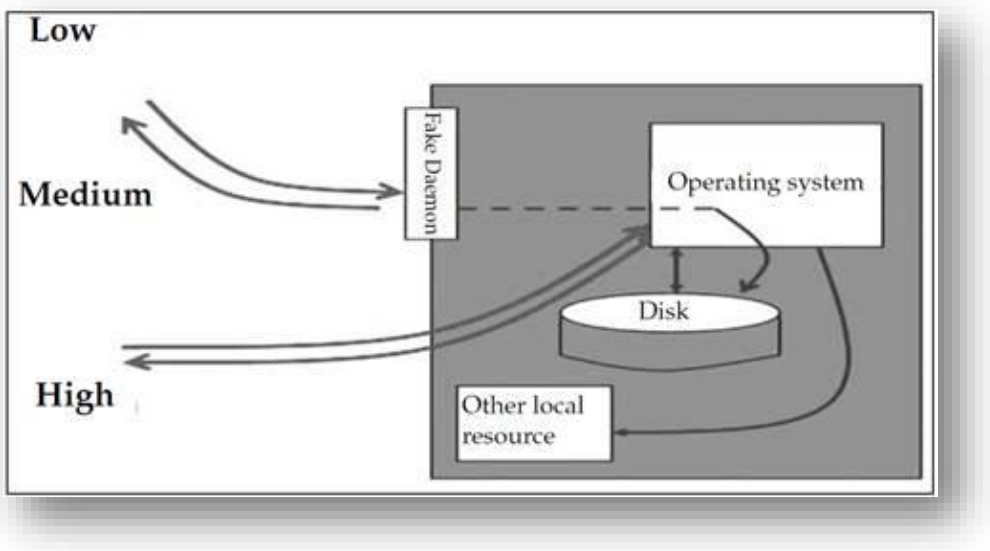

Figure 10. *Honeypot interaction levels [39]*

When compared to IDSs, honeypots are better at addressing challenges, such as reporting false positives and negatives, because honeypots can cope with network traffic detection on large systems and can work with a great volume of network traffic data. IDSs report false positives on normal network traffic when they are untuned and may not issue a report when attacks are too rapid for them to follow, too much traffic is flooding them, or the rule matching is causing many false positives [42].

2.2.3.1 Threat Hunting Loop

Threat hunting is a complicated process aiming to recognize cyber threats from various alerts. In Figure 11. Threat Hunting Loop [43] is the loop component of the threat hunting process, representing the various criteria on which the process depends. According to the diagram, a honeypot tasked with threat hunting creates hypotheses about a threat. Then it conducts an investigation using tools and techniques to find a pattern the threat uses to attack or manipulate the system. In this way, the honeypot improves its analytics and moves to a new cycle of threat hunting by creating a new hypotheses

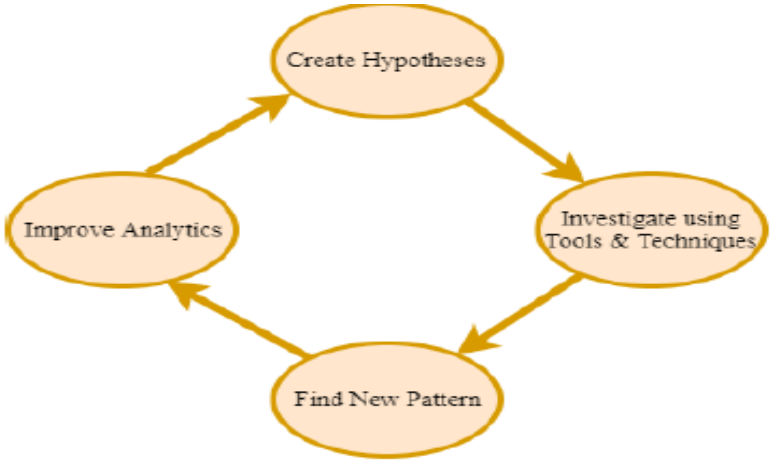

Figure 11. *Threat Hunting Loop [43]*

2.2.4 Cyber-Physical Systems

According to the literature [44], these systems are “integrations of computation and physical processes.” This means that such systems “control physical processes” through the utilization of “embedded computers and networks”. Integrating both computational and physical components in a network was a step forward in automating real-world tasks. CPSs have been

applied in various fields: transportation, defence and aviation, to name a few. These systems have been applied in industry automation and incorporate various devices and defence mechanisms to protect them. CPSs are deployed especially for large networks and systems to assist in managing and observing them as a whole.

Figure 12 Cyber-Physical System layering [45] shows that CPSs are the "system-of-systems". They have different layers, and human interaction with the system is possible through the device layer, where information about the system is found, and decision-making is done.

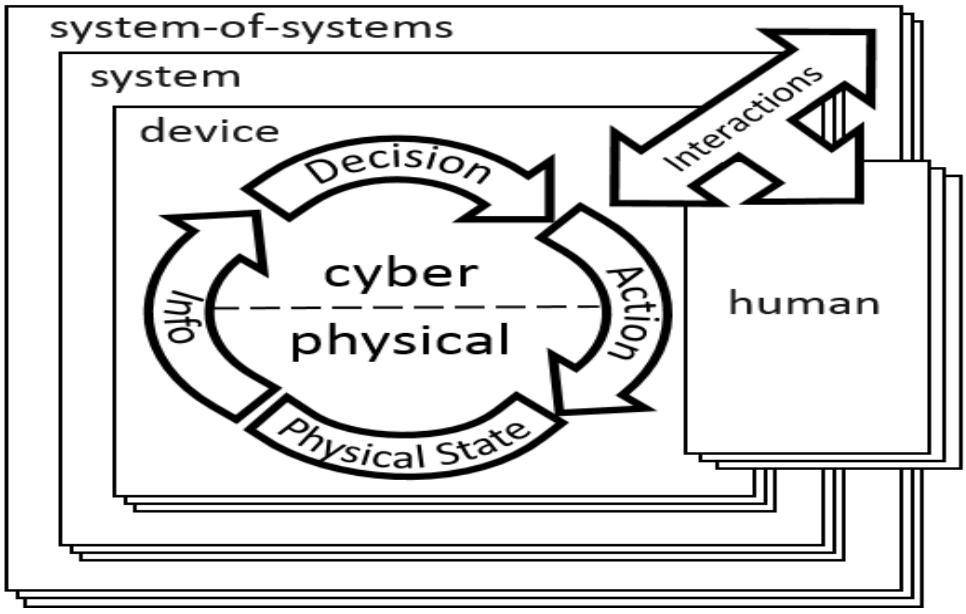

Figure 12. *Cyber-Physical System layering [45]*

Chapter 3. Related Work

This chapter presents related research and frameworks that use DTs for Cybersecurity. Seeing how DTs have been introduced in various fields has motivated many researchers to envision different applications and realize DT frameworks.

3.1 Digital Twins in Cybersecurity

Companies rapidly deploy DTs nowadays to support their security due to their adaptability to the existing rules and environment and the freedom in framework construction they provide.

3.1.1 Theoretical background

This section will present DT implementation for multiple purposes to fully understand the practical application of DTs in cybersecurity. In recent years, Digital twins have known an intense rise in utilization from the cybersecurity perspective. They are usually employed to help:

- monitor in real-time
- optimize
- predict

The life cycle and the various reactions to threats of a system or device. Simulation in real-time and real-life scenarios is the main attribute that makes them so popular among other tools.

Digital Twins may originate from the production domain, but recent approaches in cybersecurity prove that their application is imperative along with platforms and tools. Until now, DTs were depicted as virtual replicas connected to a physical product with the option of conducting simulations and keeping track of the behaviours and states of the object. Following, DTs are seen as detection and testing mechanisms. Specifically, in [46], various cases of SMS threat detection are noted down, and the contribution of DTs in each one of them is shown in the following table.

3.1.1.1 Types of attacks that a DT can detect

Table 2. Types of attacks that a DT can handle [46]

Type of attack	DT platform measures
Sensor attack	records and monitors historical data of the sensor and sends an alarm for deviant behaviour
Spoof attack	monitors the unique ID of the device continuously and avoids a possible spoof device
Hardware manipulation attack	runs a security process to check all hardware components whether they are connected or disconnected
Energy manipulation attack	monitors the historical consumption of each registered device, and if the consumption varies in an outlier value, a notification is sent to the user
Sniffing attack	monitors the network connections of all registered devices, both physical and virtual, records every new connection and notifies the user and uses RSA signatures to cipher the communication between the devices
DDOS attack	provides an embedded firewall that verifies network connections: only registered devices can connect. The platform records the frequency of the communication of each device and notifies the user of abnormal activity
Sensitive and data leakage (SDL) attack	only permits registered and authorized devices to view and export information and access sensitive data
Fault tolerance	connects with DT controllers as a backup, and when failure is detected, if an auxiliary DT controller exists, it switches it as the new DT controller and avoids the problem

3.1.1.2 How is DT different from existing technologies

DT combines individual technologies into a full product solution that clones and simulates a system or object, so it is obvious that it has some differences from existing technologies. To

begin with, digital twinning with a DT can be done in real-time in contrast with the previous simulation technologies and agent-based modelling, while machine learning cannot provide any twinning. Digital prototyping techniques require sensors and IoT components to provide a working model. On the contrary, a DT is not dependent on sensors or IoT components for data and other information related to the system or object it replicates. Optimization algorithms and applications cannot create simulations or provide security experts with real-time testing on the system as a DT does. Lastly, due to its real-time improving parameter to match the desired specifications of a system or object, a DT requires constant self-evolution to maximize the quality of its services when autonomous systems can skip this step and continue providing their services [18].

Honeypots, Decoys, and Deception

Deceiving an attacker and transferring the attack to an isolated place is done by honey potting, but maintaining the attacker's interest for a comparatively longer time while protecting the honeypot is difficult. Pauna et al. (2019) [47] researched Self-protecting honeypots, where they mentioned that this kind of honeypot could obtain real-world data to learn and protect itself under any circumstances while it also consumes the attacker's resources in its way to learn [48].

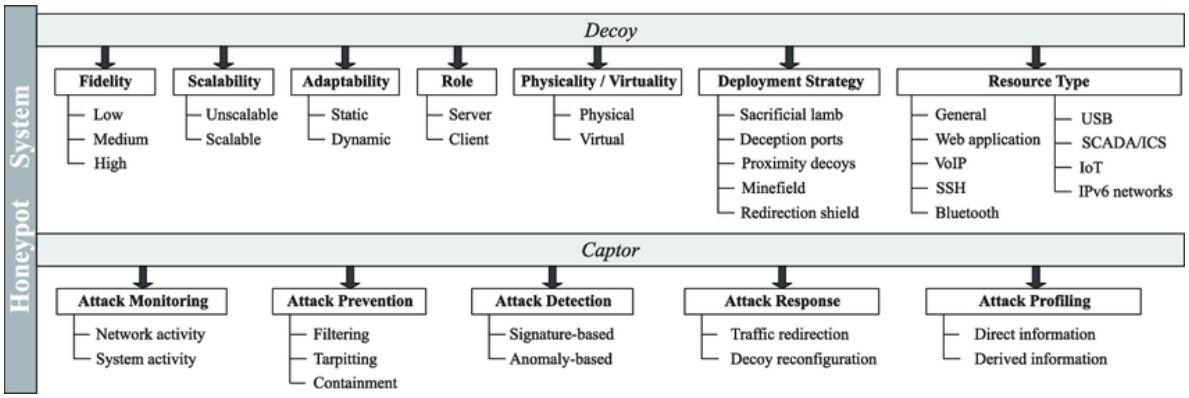

Figure 13. *Honeypot system providing Decoy and Captor capabilities [49]*

As shown in Figure 13. Honeypot system providing Decoy and Captor capabilities [49]. The Decoy that is created depends on high demanding resources while it tries to lure the attacker into thinking it is a real device with fully working capabilities.

3.1.1.3 Digital Twin in the Internet of Things

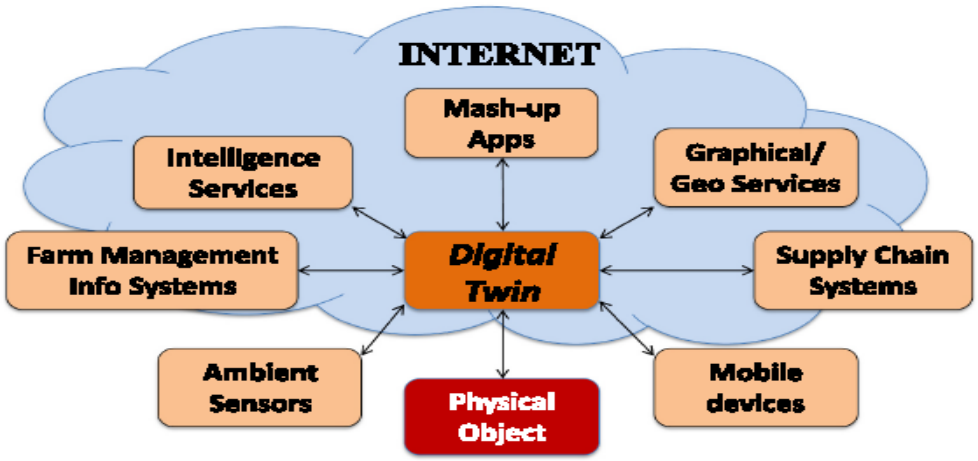

Figure 14. *DT correlation with the Internet of Things [21]*

An existing technology that can be a key to realizing Digital Twins is the Internet of Things, as it also "navigates" between the physical and the digital objects. IoT entities in the real world have digital counterparts which enable them to interact, communicate and exchange data with other physical objects connected in the same way. The digital counterparts are twins of the physical objects and are associated with them throughout their lifecycles [21]. The physical objects are linked with the Internet through virtual replicas while they store information online and communicate with other connected devices. Similarly, as shown in Figure 14, a DT takes the role of the Internet and stores object data and characteristics. Furthermore, it combines them with up-to-date information collected from other sources like Intelligence, geographical systems and sensors to bring more insight into the capabilities available in the product development and lifecycle.

3.1.1.4 Digital Twins in Fault and Incident Prediction

Intrusion Detection. CPS Twinning [50] and Cyber Situational Awareness Framework [51] are two frameworks that deal with intrusion detection with the utilization of DTs. CPS Twinning

framework proposed a new way of generating digital twins based on CPS specifications. This would assist intrusion detection by comparing real-time digital twin signals and physical device signals produced based on the specifications, thus recognizing any deviations. Cyber Situational Awareness Framework is an extension of the previous framework that added useful features to assist users with viewing visual feedback on detected intrusions and recovering past states, which would help trace the problem at its root.

Anomaly Detection. In [52], a digital twin is proposed that detects anomalies for transmission systems before they happen. It simulates current signals with historical data and compares them with the measured signals. This method proves successful and gives the user an advantage when aiming to prevent failures. Another work proposes a DT architecture [53]that uses signal temporal logic (STL) as specification rules and detects anomalies by checking if any process signals violate the specifications set.

Monitoring (Remote and On-site). The Digital Twin machining application developed by STEP Tools in [54] is a service-oriented digital twin that remotely monitors a CPS in real-time through Web-based applications. This digital twin model achieves on-site monitoring by becoming a virtual representation of the CPS and offering remote services such as state monitoring, prediction, fault diagnosis and scenario executions. Another approach to this theme is the tool called MTConnect-based Cyber-Physical Machine, which utilizes a DT to provide its users with nearly real-time remote monitoring of a physical machine.

Virtual Commissioning. The work in [55] discusses the deployment of certain layers of DTs for virtual commissioning. Having such a layer in their six-layered digital twin, the authors show that data flow between the digital and the real entities is not mandatory for controlling the order of process events, which proves useful when there is a need for fast prototyping and testing.

Autonomy. Digital twins benefit from the creation of autonomous systems. At the same time, they provide them with information about the lifecycle of an object and offer a simulation area to perform security and operational analysis, make predictions and make their own decisions based on analytics. Such a system is presented in [56], where operational decisions can be made autonomously by the Smart Car.

Predictive Analytics. A Five-dimensional digital twin model of tool system is introduced in [57], which performs data analytics. It discovers fault patterns through descriptive, diagnostic, predictive and prescriptive analyses and makes decisions based on the analyses result. By using predictive analytics on physical counterparts, future conditions and possible malfunctions can be forecasted and prevented.

Documentation and communication. The digital twin can prove useful for documentation and communication as it can report and visualize behaviours.

3.1.1.5 Digital Twins in Critical Infrastructure for Self-Protection

Just as the previous example of DT assisting in detection and protection, there is a new need for DT utilization to achieve self-protection and self-adaption for critical infrastructure. Facilities that are to be placed on the Internet are threatened by unpredictable cyber-attacks and thus demand a high degree of security. On the one hand, according to Danny Weyns [58], self-protection remains an open challenge, while, in the meantime, mostly machine learning approaches try to handle that matter. On the other hand, self-adaption has raised the interest of various researchers in twin runtime models and provides greater performance. These were created to cover, detect and deal with unprecedented situations. Such systems perform better when facing “What Ifs”, as Schluse et al. state in their paper [59]. They mention how DTs function together with a simulator, but their approach does not include self-adoption. As they don’t use a runtime synthesized model, it is difficult to change the modelled system when an unknown situation occurs.

Another work [60] proposes the creation of DTs based on modelling languages. The proposed architecture of a collection of components would cover many of the basic DT features, and in that way, it would achieve a better overall outcome. However, their approach would fail to achieve either self-protection or self-adaption because they built their models on a static model. This automatically means that if the physical system changes, the DT won’t be able to self-adapt and self-protect, as its model is static and cannot adapt to the change requested by the physical system.

3.1.1.6 Digital Twin benefits for cyber security

Cybersecurity can benefit from the utilization of Digital Twins and AR technologies. Having a digital representation with features and data, on the one hand, and being able to apply certain practices in an AR environment and seeing a difference in real life, on the other hand, is a promising collaboration patent for cybersecurity improvement. In a state-of-the-art review paper [61], the authors mention certain areas of application that would most benefit from this collaboration patent:

1. Contextualization of physical surroundings: Advanced Persistent Threats (APT) are security attacks that human operators cannot detect due to inconsistency. Such attacks remain unsuspected, and their target is not obvious as they remain silent for a long time until the next attempt.
2. Improving Cyber Situational Awareness
3. Integrating Domain Knowledge
4. Have the ability to change:
5. Manufacturing
6. Education and training
7. Cities, Transportation, and Energy Sector

3.1.1.7 Conceptual ideal of feature-based digital twin framework

In the following Figure 15, the most significant DT features are unveiled. An ideal conceptual DT model should consist of a data link feature, which is essential for a DT to connect with its physical counterpart and operational modules. In this paper [62], the term data link refers to a "hub for all information related to the physical twin". So after defining the data link, it is easy to understand that all information, which comes from the physical counterpart and other DT module sources such as data storage, analytics, security rules and human interaction through the user interface, is collected and communicated through this feature.

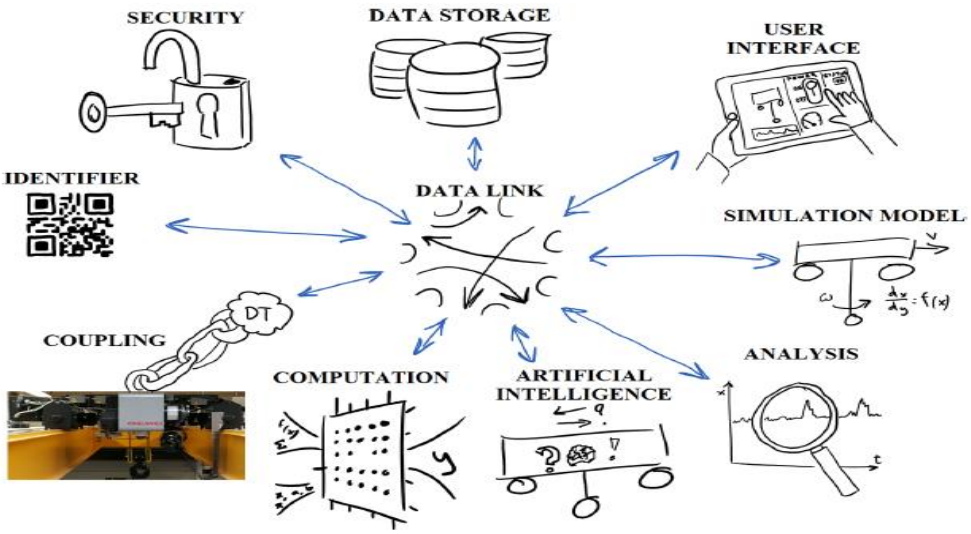

Figure 15. *The conceptual ideal of a feature-based framework [62]*

3.1.1.8 Main operations of a Digital Twin

When speaking about security, a DT offers a virtual, isolated environment that annihilates any previous risk during operations or simulations done on the physical counterpart. As described in Figure 16. The application of DTs to secure ICSs is discussed in the paper [63], where a DT framework is proposed. The authors point out that DT technology is superior to other security systems' technology, while DTs operate in multiple modes at the same time. Security incidents are more likely to be caught by DTs due to their direct connection with their physical counterpart. The framework proposed in the paper focuses on security management through various DT processes.

To begin with, specification data are collected through sensors and event logs into a Specification and a Historical/State database accordingly. The Specification database includes enough information to create the DT model, including security and safety rules predefined in the specifications obtained. The DT can perform basic tasks such as Emulation, Aggregation and Querying of data from the Historical/State database and Monitoring.

The DT has core abilities to run as a security operator:

- historical data analytics and optimization
- simulation
- replication

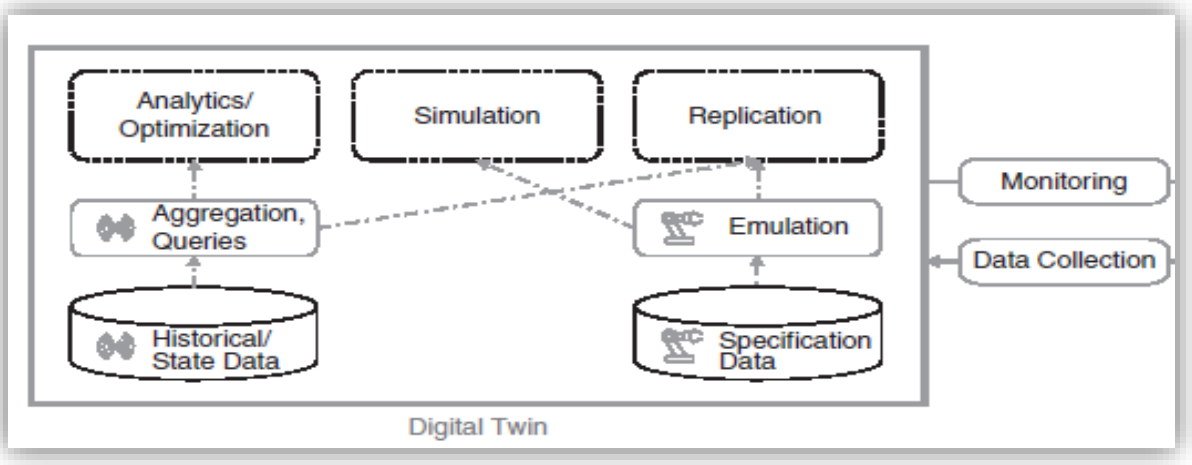

Figure 16. *Main DT operations [63]*

3.1.1.9 Digital Twins Simulations and their Use in Cyber Security

Cybersecurity requires new ways of monitoring systems in order to avoid potential attacks. Simulations of such security incidents can be beneficial in multiple ways:

- Enable repeatability
- Show the system's behaviour under a range of circumstances and configurations
- They run in a virtual environment, so the physical environment and the system are not affected by any unwelcome changes

The DT can serve as an attack detection mechanism providing simulations according to the preferred situation [64].

3.1.1.10 Cybersecurity Challenges created by Digital Twins

Cyber Digital Twins (CDT) provide an isolated space for cybersecurity professionals to perform security assessments and attack simulations without having the need to intervene or disrupt the functionality or set risks for the real system. So far, DTs are seen as a beneficial factor in the creation-manufacturing process of a product or as a simulation environment for testing, security analysis and monitoring. However, the deployment of DT technology depends on certain parameters to be assistive. A DT must be available at all times to support its physical counterpart before, during its lifecycle and after. Also, it should be mentioned that depending on the desired

usage and how the DT is implemented about the physical object or system, this technology might affect the availability of the object or system accordingly. This happens when implementation constraints or an attack compromises a DT. Because of the connection, the DT has with its physical counterpart, even a small malfunction of the DT could create a failure to the physical counterpart.

Moreover, a DT must preserve its *integrity* and not allow unauthorized destruction or modification of data during operations done throughout its lifecycle, as such actions would affect its security and its response capabilities. Having outdated or false information may lead to misunderstanding an incident, sounding false alarms and making a wrong decision. On the occasion of a situation like that can be caused even by a security operator. It is common practice for them to simulate activities and test different system configurations on CDTs to discover potential threats. A CDT constantly connected to the real system should have the same *security* configurations as the real system. Otherwise, there won't be accuracy in predictions or reflection of the real situation, and the CDT will make wrong decisions that will eventually affect the real one. To avoid such situations, it is best to have a reduced integration of a Digital Model for a CDT implementation so that tests and changes in configuration do not influence the real system.

From the attacker's perspective, if one manages to access the CDT, then, due to its connection with the real system, he/she automatically gains control over both the CDT and the real physical system. Even if the attacker has no way to manipulate the real system directly, he/she can test and learn its vulnerabilities from its virtual counterpart. An attacker could also have various motives rather than simply compromising and malfunctioning the CDT. Lots of security attacks target confidential information about system facilities or company data and manage to do their job by exploiting misconfigurations or default settings in unexpected parts of the security systems. When a DT reconstructs and replicates the real system, the security configurations must be transferred across the digital environment. In that way, important information is spread, which can result in serious danger if attackers have access to it. Although CPSs deploy extra security layers to prevent such threats, a DT requires all assets to be visible and accessible anytime. This makes them more vulnerable to threats. IPs and other data need to be acquired from the DT so that it can be able to replicate as accurately as possible the connections and interactions between different components of a system. So an attacker cannot only acquire IPs and data and recognize a company's patterns and business logic when compromising a DT. So *safety* and *security* play a significant role in implementing a CDT [65].

3.1.1.11 Cybersecurity Challenges solved by Digital Twins

DT technology can face various cybersecurity challenges thanks to its modelling and prediction capabilities. Applying a security patch on an Operational Technology infrastructure requires testing devices in isolation, which is time-consuming and expensive or would require a secondary system to do tests on it. Having a DT of the exact device or even the whole infrastructure and testing on a simulation area can eliminate the problem. Continuing with challenges, new advanced systems need security and thus require test cases and scenarios more often than traditional ones. Especially during the design and the development phase but also during their operational time, there is a demand for these systems to have fewer vulnerabilities which means more evaluation criteria and efficient automation of security tests are required. This situation again calls for DT deployment, as automation, on-time testing evaluation and self-development are its key values. Cyber threats cannot be avoided nowadays because serious information or operations are at risk. It should be mentioned that the newest methods and cybersecurity risks are evolving fast based on intelligent spyware and smart vulnerability and exploit discovery applications and are dynamic, which means that they pose an even greater danger to systems. DT can self-adopt to diverse situations, and their self-evolution allows them to respond immediately and with high accuracy [65].

3.1.1.12 Digital Twin Framework Requirements

In [66], Moyne et al. provide a condensed table of the required characteristics that a DT must include according to its definition, which was mentioned in the Introductory section of this thesis. To begin with, a DT **uses models** to mirror an aspect of a process, feature (asset), system or product and is capable of doing so because it has clearly defined modules. As stated previously, a DT has a certain relation to the physical system or object -to be twinned- and provides two-way communication between them. In order to realize the models that contain **computational units** and to be able to adapt to different application domains, **analytics and intelligence** have to be incorporated. A DT provides its services with **measurable accuracy and net value-add,** leading to measurable cost definition for every maloperation. This means that each and every operation of a DT has to be calculated financially for its costs and benefits to be clear.

DTs should distinguish themselves from relative technologies for their reusability, interoperability and interchangeability, maintainability, extensibility, capability and accuracy;

concepts are analyzed further in the table below. The vast majority of the requirements that must be met for a product to be called a DT are shown in Table 3.

Table 3. DT Requirements [67]

DT Requirements		
A DT must be able to use some form of narrow DT intelligence that allows it to provide its capability in a specified application domain		
Reusability	**Interoperability**	**Interchangeability**
DT solutions must be portable, re-usable and scalable	Multiple instances of the same DT class must be allowed to interact in a coordinated fashion	Interchangeability of different instances of the same DT class must be supported
Degree and method of re-usability (data translation, sub-set of metrics) must de definable	Integration of and coordination between instances of different DT classes must be supported. The same practice goes for DT and non-DT components relationships	The framework must support standardized definitions of DT structure, baseline minimum abilities, quantifiable capabilities metrics, exposed interfaces, services provided, and behaviour exhibited
Standardized, reusable and quantifiable verification and validation processes must be supported		
Maintainability	**Extensibility**	**Capability and accuracy**
The minimum required DT output quality to provide its intended capability must be quantifiable. Diagnosability of lack of sufficient quality of DT output should be identifiable in a time-critical fashion	The DT framework must be extensible to support DT solutions across the entire Smart Manufacturing ecosystem	DT solutions must be able to use evolving analytics techniques, including improvements on existing techniques and novel new techniques
The DT should be updated to continue to provide sufficient output quality if that level of maintenance is a requirement for the DT (as a form of scheduled maintenance)	The DT framework must address security requirements, including data partitioning and IP security required for DT operation across the entire Smart Manufacturing ecosystem	DT solutions must support structured and automated integration of analytics and Smart Manufacturing ecosystem information

Framework-specific
• **A DT framework must support an evolution rather than** a **revolution of capabilities, especially supporting the evolution of existing capabilities to align with the ultimate DT framework vision.**
• **A DT framework must support the entire DT lifecycle from envisioning and design through development, validation, deployment and maintenance**
• **A DT framework must provide a common DT definition, taxonomy and other mechanisms that allow the community to collaborate on DT technology, from DT fundamental research through applied research, development, deployment and maintenance**
• **A DT framework must support virtual counterparts across the entire subject-matter expertise (including the full supply chain) that can be used for detection, prediction, prescription and analysis of all aspects of** the **operation**
• **A DT framework must support an evolution from narrow intelligence toward more intelligence with fewer context restrictions**
• **A DT framework must support the union of subject-matter expertise and analytics as a continuing integral part of DT capability and evolution**

A DT needs to have specific integral parts and requirements. In order to realize a DT framework, it is important to apply most of the requirements mentioned before, at least, if not all, of them.

3.1.2 Frameworks

This section presents DT design methodologies and frameworks after going through the components, operations and requirements that a DT design has to be met. Analyzing the different frameworks and the methods used assists in understanding how they are built and how they provide their services through interacting with the environment they are deployed in.

3.1.2.1 Industry 4.0 Layer-Model

Industry 4.0 recommends a standard Layer Model to guide the design steps of a DT. The proposed building steps for DT development are shown below in the Layer Model. In Figure 17, the Industry 4.0 Layer Model's structure is depicted, and this model's correlation to the structure of an actual DT can be easily made. In the model, the connection between the real and the digital world is made through digitization. Having this layering in mind, someone notices that the DT structure works in the same way. Physical things exist in the real world, while their digitalized

counterparts exist in the digital world. Processes and communication, data transfer and functions commonplace in the real world need a respective representation in the digital world. This model demonstrates the various layers needed for digital counterparts to clone physical things [69].

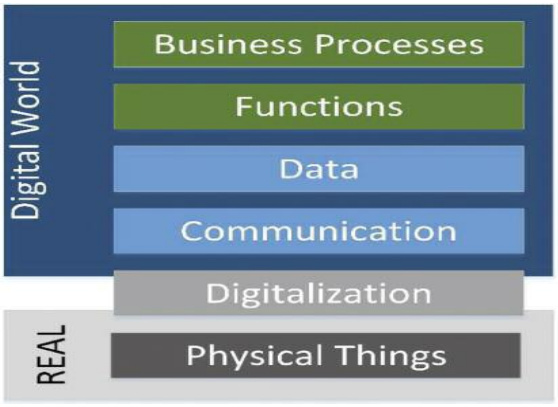

Figure 17. *Industry 4.0 Layer-Model [68]*

3.1.2.2 Design of Digital Twins for cybersecurity and safety

Every single one of the new technologies, including IoT, Cloud Computing, and Machine Learning, promises to provide the world with certain assets that were not yet explored or mentioned in previous decades. However, the combination of such technologies is demanded by companies and organizations as it is most assistive in fulfilling their digital transformation. Likewise, a DT combines various technologies to achieve its capability requirements which were analyzed in the previous section. Integration technologies and knowledge extraction methods, forecasting algorithms and real-time prediction are just a few of them required to make efficient decisions by collecting and considering Big Data from IoT devices and sensors. Representation and modelling methods are needed to create an accurate duplicate of a physical entity [70]. These capabilities of a DT call for appropriate assets to comprise a successful design. According to [71], three conditions have to be met for a technically sound DT design:

- Modelling of assets
- Decision-making methods and predictive analytics (to support decision-making)
- A knowledge base that is centered around its lifecycle and informs itself with historical and real-time sensor data and relative external information from other databases on the internet

Modelling of assets refers not only to the structure design of components and modules of the physical entity that will be twinned but also to the measurable parameters and the information concerning the production dates and history.

Decision-making is an integral DT component that enables it to take action after advice from predictive analytics such as regression, machine learning techniques, and statistical analysis.

Owning a **knowledge base** containing historical and real-time data differentiates the DT from other technologies as it can use them for developing and maintaining, and adapting itself in different situations. Sensor data and historical data or relative information obtained from other external sources, such as management systems or environmental parameters, comprise the knowledge base that assists a DT throughout its lifecycle.

3.1.2.3 Design Methodology for Digital Twins in IoT4CPS

According to the authors in [70], Digital Twins can serve as security and safety authenticators for IoT devices in CPSs. In order to construct suitable DTs for this task, they propose a certain design procedure. At first, by pinpointing and "modelling assets, security and safety objectives" and storing asset-related, historical lifecycle data and security requirements and permissions, a DT can collect and evaluate important information about the system. After that, a DT should design its own "relevant security and safety evaluation" measurements, which will later assist in understanding the system's general status. This step of the procedure combines identification, selection methods and analyses of existing measurements and risk assessment (finding the probability of occurrence of an unwanted incident) with the help of existing security requirements and configurations. Overall, this step is tricky and requires multiple measurements to be considered to output useful evaluation measurements. In the end, "threat identification and modelling", a thorough evaluation of the security, safety measurements and risk assessment mentioned before, builds the predictive (creating a model by learning from training data and using it to predict future results) and descriptive (using current and historical data to describe relationships) analytics methods of the DT.

3.1.2.4 Conceptual model of the Digital Twin prototype in IoT4CPS

The authors in the paper [70] tried to create a DT for security in Iot4CPS and presented it as a virtual honeypot. The conceptual model depicted in Figure 18 describes the DT prototype structure that the authors designed in order for it to conduct precise simulations and analyze and predict future states and events. Their DT has a "Virtualization Manager" and an "Interoperability Manager". The first one consists of three modules called "Data Manager",

"Models Manager", and "Services Manager". The second one consists of another three modules called "Monitoring Manager", Decision-Making Manager", and "Simulation Manager".

As explained by the authors, a DT comprises various services. So, a DT demands a huge collection of data from the real world as input for analyses and returns information and decision feedback. This task is done by the "Data Manager" with the components "Data Acquisition" and "Data Analytics". To portray the real counterpart lifecycle phases and states, it creates models that can either be computational or representational, which is covered by the "Models Manager". As a DT must be ready to monitor, simulate and make decisions, it needs to integrate those services. In this case, these tasks are done by the "Services Manager".

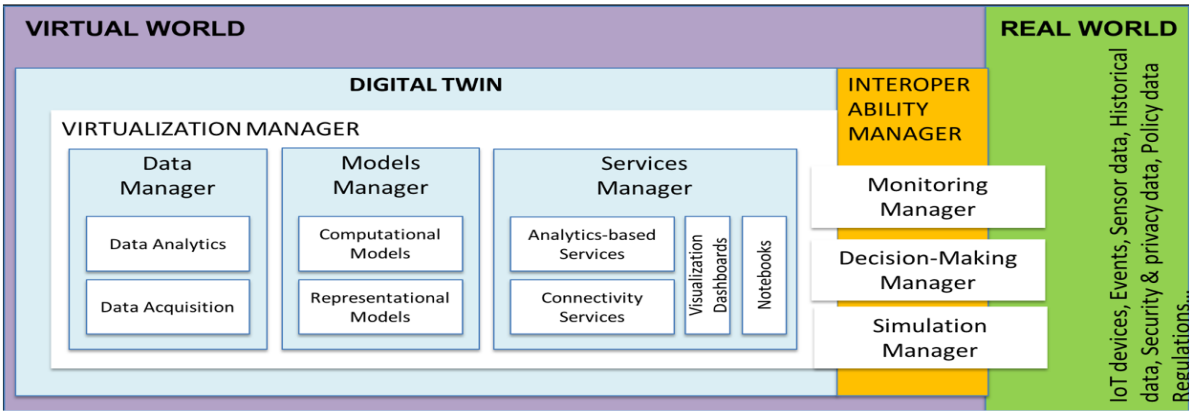

Figure 18. *Conceptual model of DT prototype in IoT4CPS [70]*

3.1.2.5 Cyber-physical Digital Twin Framework for Manufacturing

D. Lin and M.Low [72] propose a framework for a CPDT that aims to provide the manufacturing systems with simulation, predictive ability and intelligence for accurate analysis and decision-making. In their paper, they design a CPDT in three layers: the operation, the visualization and the intelligence. This cyber-physical system was prototyped for an SMT production line and was tasked to:

1. collect information for the physical assets that were located on the same production floor, which would make it easier for the system to process information about them and therefore be more accurate and faster in future tasks that require such information
2. visualize all the processes and physical assets in real-time by modelling them and synchronizing itself with them through obtaining real-time information about their current state

3. conducts real-time analysis on historical and current real-time data, which leads to accurate and on-time decision-making

Cybersecurity was also considered in the framework design as an adaptation of existing IT forensics investigation processes applied to SCADA systems. Specifically, they implemented data collection through sensors and IoT devices to demonstrate the proposed framework. The visualization process was done with the help of FlexSim, a simulation modelling platform combined with a SCADA database that provided real-time production information. This information was collected in an SQL Server and inputted in FlexSim to realize the models of the physical machines and the processes of the production line and be able to analyse them. However, they mentioned that there is still work for the intelligence layer. "More decision-making functions and in-depth data analytics", as well as more insight on the design of the interactions between the physical and the digital space, would be of assistance for future implementations [73].

3.1.2.6 Interoperability-context scenario of the digital twin

In the paper [74], the authors show a scenario of applying a DT to the production process for a distributed manufacturing system called "Connected micro smart factory". The activity diagram in Figure 19 depicts the scenario's physical and digital world components with the necessary processes between them. The physical world consists of the "Connected micro smart factory" and its IoT network, while the (digital) cyber world consists of a Manufacturing and a Cloud application. Specifically, taking a left-to-right approach, while the smart factory follows its production schedule, its manufacturing elements are constantly linked to IoT sensors/middleware, which collects data about the specific product, the process status and the general operation status of the factory. This data is then categorized and saved in a database. In the cyber world, the Manufacturing application is tasked with supervising the production schedule, and the Cloud application has the job of the middleman between the customer requests and the production machinery. When an order is received, the cloud application is tasked to verify the manufacturing bill of materials needed and create a supply chain plan pushed to the distributed manufacturing system. It also requests the Manufacturing application to extract the production schedule, which stores it in the database. In this scenario, a DT is applied to interoperate with the factory operation schedule when activated. It analyzes the current and past situations by requesting the related information from the sensors/middleware and the database

accordingly. The output of the analysis can be used from the cloud application for managing the production status by receiving manufacturing element data and historical data, assisting the creation of a DT model. The analysis output and the current data received from sensors/middleware assist with synchronization during the operational phase. When activating the DT, the human operators have the power to monitor and track current and historical status accordingly. The DT also assists in decision-making and future production planning as it can analyze current and historical data simultaneously while also having access to past production schedules.

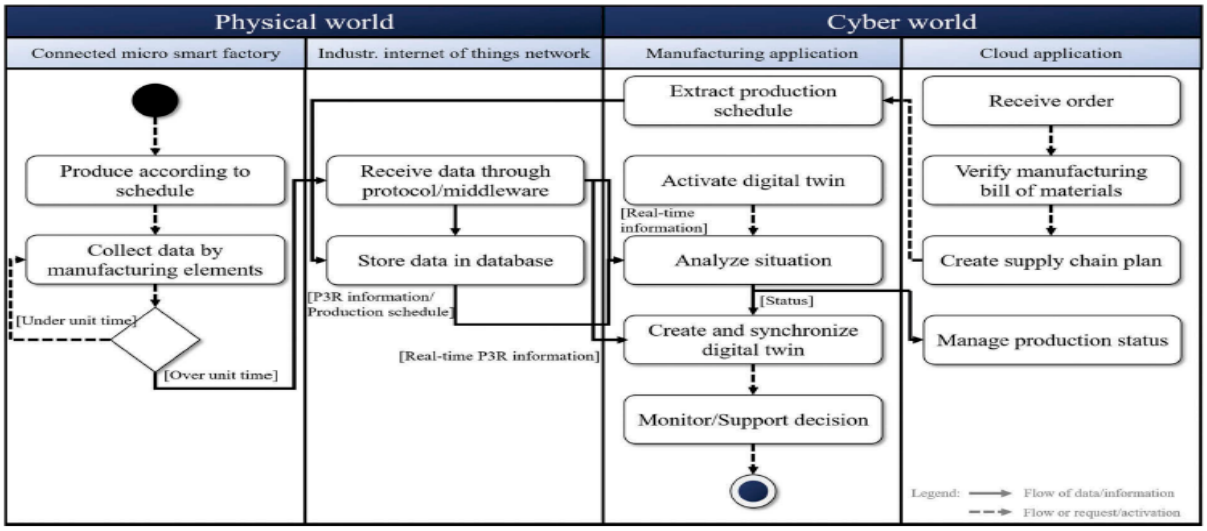

Figure 19. *Interoperability scenario of the DT application [74]*

3.1.2.7 Digital Twin-based conceptual simulation framework

In the framework depicted in Figure 20, a DT is deployed to assist in the real-time monitoring process of a factory's CPS based on Industry 4.0. The factory depends on PLCs and robot controllers, which are tasked with certain processes and help automate them. However, these devices have some disadvantages by design: they have limited storage space and computing capacity. Due to the volume and the variety of data collected from those devices, selecting the "right" data and storage this data in the right form to analyze it.

By simulating the CPS with DT deployment, the authors in [75] collect the data received from the various devices through a live and interactive connection between the physical and the digital counterparts. Specifically, they gather data from real devices in the virtual environment. This provides an advantage in terms of collection, analysis and processing time because the virtual

environment is equipped with enough storage and computing capacity. The virtual environment is the source of data input for the data collection, analysis and processing tasks. This means that the processing and the simulation tasks retrieve data from the DT. This data combines the collected data from real devices and descriptive data of the process events generated by the DT.

The DT components such as aggregation, classification, analytics and simulation methods and algorithms assist the CPS in supporting the decision-making process. After multiple analyses on and simulations of the data (evaluating deviations and predicting future states), the CPS can determine whether human intervention or machine intervention is required at that moment of the production phase. With the deployment of the DT, not only decision-making but "playback" is also possible. Logging data in the DT model can also be helpful in another way, as "errors and disturbances" recorded during a production phase can be analyzed and reproduced in simulation to optimize processes as well.

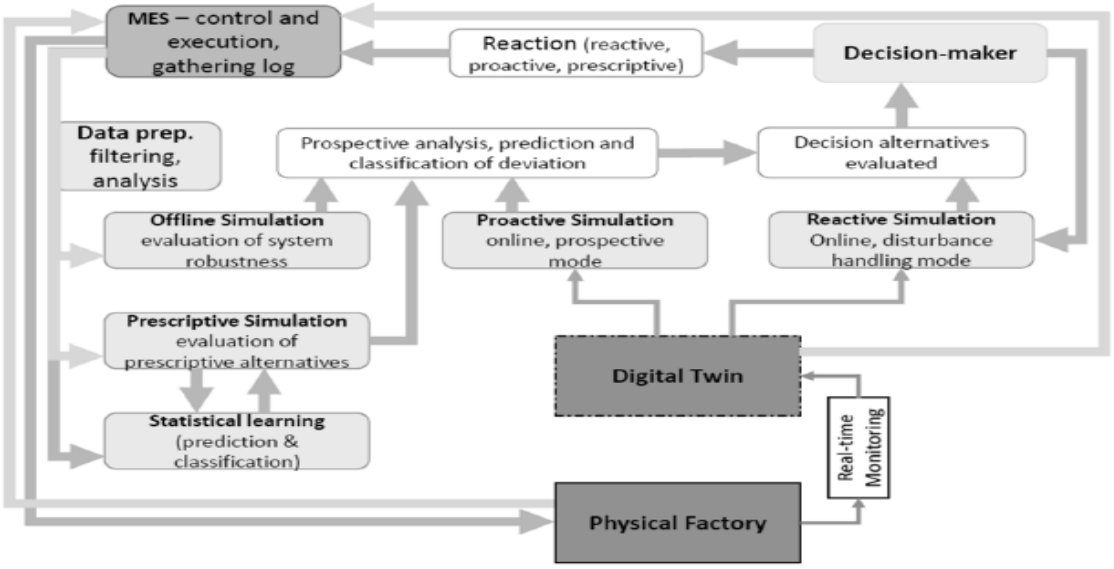

Figure 20. *Conceptual simulation network based on DT [75]*

3.1.2.8 Process-based Security Management Framework

The application of DTs to secure ICSs is discussed in [63], where a DT framework is proposed. The authors point out that DT technology is superior to other security systems' technology, while DTs operate in multiple modes at the same time. Security incidents are more likely to be caught by DTs due to their direct connection with their physical counterpart. The framework proposed in the paper focuses on security management through various DT processes depicted in Figure 21. To begin with, CPS specification data through sensors and event logs are collected into a

Specification and a Historical/State database accordingly. The Specification database includes enough information to create the DT model of the CPS, including security and safety rules predefined in the CPS specifications. The DT can perform basic tasks such as Emulation, Aggregation and Querying of data from the Historical/State database and Monitoring the CPS. As discussed in Section 3.1.1.8, a DT has core abilities to support security operations such as historical data analytics and optimization, simulation and replication.

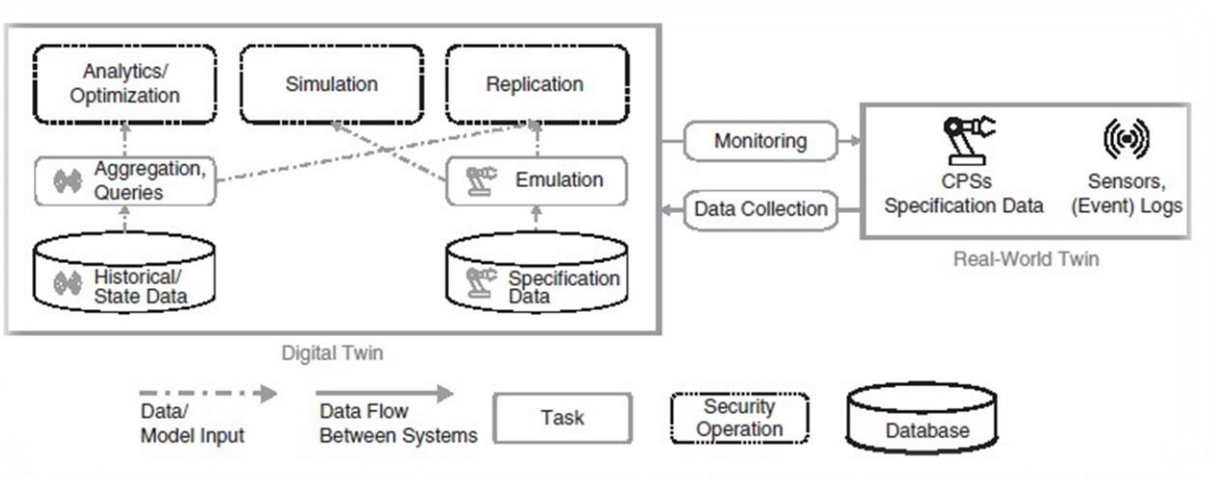

Figure 21. *DT security operations [63]*

A DT requires specification and environment data constantly flowing between itself, the physical counterpart and the environment it is located in, in order for it to create an accurate model. So, data collection is crucial for a DT and can be done with the application of sensors on real-world devices or by keeping logging files for registering their states and status. Having a specification database allows the DT to collect essential information about features or security rules and variables that are an integral part of the physical system. With the specification data already collected, a DT can proceed in emulation, simulation and replicating the physical counterpart. Simulation refers to vulnerability analysis and security testing on the physical counterpart through a DT model. For the creation of the model (replication), a DT also collects historical/state data stored in a database and, after aggregation, can be used along with specification data to accurately reflect the corresponding current status and condition of the physical counterpart. Threat detection and intrusion detection are performed through emulation, reproduction of stimuli and the application of differential algorithms. Historical data information is useful for ad hoc queries in the database, and aggregation of such data leads to better analytics and optimization of the DT model. Through statistical analyses, machine learning techniques

and data queries, the DT can conduct network traffic analyses and observations. Sending certain commands to the real counterpart also empowers the DT to monitor it.

3.1.2.9 CPS Twinning Framework

Digital Twins have been utilized for controlling CPSs and detecting potential weaknesses in them as well. Having a physical system mirror and its virtual counterpart allows security experts to investigate further. M. Eckhart and A. Ekelhart [76] proposed a framework for the design of a DT for a CPS, called CPS Twinning, that emphasizes security assessments for the CPS. Their objective was to provide cyber defence capabilities for CPS operations staff. The framework consists of two components of significant value: a **generator** that is tasked to automatically create a replica of a physical object and/or network topology and **virtual space**. When given a specification about a CPS, the generator's job is to extract the topology, the security rules and the devices that are parts of the network structure along with their configurations. To make the generation of the digital replica (twin) effective, automatic and swift, the authors took advantage of the existing CPS specifications and the information about the environment, the topology of the network and the individual components that constitute the network. The second component, the virtual space, can provide two different operation modes: replication and simulation. It is used to model the components of a cyber-physical system according to the topology specifications extracted earlier and represent them virtually with the use of digital twins. In the virtual space, simulations on the virtual topology can be executed, and replicas of data sources from the physical components can be reflected onto the virtual side. Simulations can be done to test and optimize the modelling of the virtual components. When it comes to testing the devices, monitoring processes and analyzing their security, other modules interact with digital twins. In the end, the DT is capable of handling:

- Intrusion Detection
- System testing and simulation
- Misconfigurations detection
- Penetration testing

3.1.2.10 DT cyber situational awareness framework for CPS

The authors of the previously mentioned DT framework, called CPS Twinning, extended their work in another paper to improve the existing framework by adding extra features. The feature of "state replication", which has been added to the framework, is essential for a cyber DT to keep

up with any latest changes and provide a virtual depiction of the physical devices. For the DT to replicate the condition and state of the physical devices, it collects data passively from the physical environment. In this way, monitoring of the DT and intrusion detection can be achieved, and at the same time, it makes it easy to spot any difference or deviation between its behaviour and the behaviour of the physical environment. This feature allows operators to inspect the DTs but restricts the framework in presenting only present states and behaviours, which is, in fact, a disadvantage because viewing past states is a capability highly required for a cyber DT framework. The authors mention that this problem is being solved by providing yet another feature called “record-and-replay”. It can reproduce DT states on demand by storing stimuli produced during a DT state's creation process. Users get a clear DT state outlook, can analyse various historical states and thus have a better view of the situation anytime, making the framework “situational awareness”. The feature of “visualization”, which has also been added to the framework, uses the information available from the two features mentioned previously, monitoring and intrusion detection, and delivers visual “security-relevant” information. The importance of this feature lies in supplying the human operator with visual changes or misbehaviour that could potentially impact the CPS's well-being [51].

Due to state mismatches, authors say that the feature “record-and-replay” is a work in progress, but in theory, it gives an advantage to the operators to apply in-depth analyses and recognize any inconsistencies immediately. Overall, the new features provided additional functionality to the existing framework by rendering it effective in dealing with, supplementary to the previous framework, tasks such as:

- Monitoring
- Risk assessment
- Incident handling

3.2 Digitally-Twinned Honeypots

This sub-section of the thesis shows how little research has been done during the last years on the specific topic. Digital Twinning of Honeypots is not yet “a thing”; however, it is becoming increasingly necessary to use digitally twinned honeypots against skilful attackers while honeypots alone cannot compare to the newest sophisticated attack systems.

As seen earlier in the thesis, honeypots may attract the attacker, but they could also be easily detected. The deployment of DT on honeypots is a new research domain which tries to answer that problem. In particular, DTs' capabilities to create high-interaction honeypots are mentioned in [77]. DTs can replicate a physical or virtual honeypot with high-level accuracy and mimic its behaviours and responses to attack incidents. With this idea in mind, developing multiple fake honeypots to surround the real one could save money and time consumed in finding and fixing vulnerabilities or faults in the real honeypot while providing information whenever there is an attack on the fake. For example, suppose a cybersecurity team of an organization has already deployed ten honeypots to filter its traffic and protect its systems against threats. Then, after some cyber-attacks. In that case, those honeypots are fingerprinted and avoided as machines, and the attack will transfer to another target. Then, the cybersecurity team will need extra ways and time to prevent another attack while changing the identity of those honeypots and getting them ready. Having those honeypots digitally twinned would mean that decoys would attract the attack traffic and act as a protective cover for the originals while functioning similarly. DT decoys could easily change values in certain parameters on demand to further engage the attacker, and this method can trick them into entering an observation environment.

Chapter 4. A framework for the Digital twin Honeypot Features

Considering how DTs can assist in creating solutions for security breaches in various systems such as CPSs and ICSs in related works, this thesis chapter will try to combine knowledge and strategies based on them to cover as many security tasks as possible into a single-solution framework. This framework will focus on DT cybersecurity multitasking around a honeypot while having an ultimate goal: optimization of the honeypot. The initial thought is to build a DT framework to support a honeypot during its runtime by replicating its features and functionality. The DT will be able to investigate the honeypot behaviour, performance and operation and provide a simulation area for testing, along with the visual output for the security operators simultaneously.

4.1 Basic criteria for the DT and which components are needed to support the Honeypot part

As described earlier in sections (2.1.3 and 3.1.1.8), there are basic functionalities that a DT must have in order for it to be a DT. Namely, a DT needs to be constantly connected to an object (or an idea) with its assets and features to copy. Also, the connection they have is directed on both parties, which assists the DT in improving itself and the object (especially when the DT is deployed for prototyping and there is not yet a physical existence of the object). A DT initially needs the acquisition of a plethora of data related to the system's assets to be twinned, the honeypot in this case. This data is passed from the system to the DT through sensors, datasets and external databases that might contain information relevant to the system that can help maintain or improve it. Data is also passed vice-versa, from the DT to the system for commands, monitoring and optimization.

The Honeypot itself has the following prebuilt features:

- Investigations processes
- Lists

The DT Honeypot will have an additional feature to the prebuilt ones:

+ An intelligent response to automatic actions based on observed activities

Investigations processes enable analysts to review and comprehend data collected from the honeypot. The Honeypot’s lists are updated based on the information of the investigation processes. As part of the Digital Twin honeypot development cycle, a collection of **data**

computational models and representation models will be applied to the Honeypot in order for it to acquire “intelligence”. The DT will perform analytics and processing during the honeypot lifecycle phases, and by using AI algorithms, inferred data acquired during the run time will be incorporated into the Digital Twin knowledge base. The Digital Twin honeypot will then **support customized analytics and will perform a series of cybersecurity tasks.**

4.2 DT framework overview

In this thesis, the main concern is finding a way to optimize the existing Honeypot during its runtime without interrupting it. After advising from the current scientific literature, DTs are, by far, the most suitable for this job. Digitally twinning the honeypot would benefit the honeypot itself and enable other operations by taking advantage of its features. Other cybersecurity tasks can be done parallel to the optimization, which will provide cybersecurity operators with an in-depth analysis of the honeypot and assist them in keeping up with it during its runtime. In order to visualize this concept, a DT framework for the honeypot is proposed. Looking at the DT framework shown in Fig.22, one can notice all of the tasks that the digitally twinned honeypot can accomplish, plus the security-specific ones.

There are certain design steps for the DT framework to be created. The process of digitally twinning the Honeypot requires data, so a model needs to be created with the data that describe the real Honeypot. Therefore, Data collection and storage of the data in a Storage point (a database) for later use is required. After that, the DT acquires the data from the Storage point and processing the data begins. The next step is moving to a Modeling-Replication phase, where a digital clone of the real Honeypot is being created. When the digital twin honeypot (the clone) is ready to run, Analytics-Optimization and State replication-replay modules are available. These extra features provide the digital twin honeypot with functionalities such as Monitoring and Decision-making. The role of the digital twin honeypot is not stopping in replicating the current condition, analyzing state and network traffic data, making decisions and monitoring. Although its main task is to optimize the real Honeypot, it also constitutes a virtual environment where cybersecurity operators can conduct Simulations and Testing without fearing damaging or occupying the real Honeypot during its runtime. What’s more, after the Monitoring processes are done, it informs the human operators with Visualization of the state it is in and its current configurations. This means that, in the end, cybersecurity operators have the power to simulate

various states and conditions in the virtual environment, test different configurations on them and visualize the results along with the outcome of the Monitoring of the network traffic data.

In the next section, a thorough explanation of each one of the components and functionalities of the digital twin honeypot framework depicted in Figure 22 is presented.

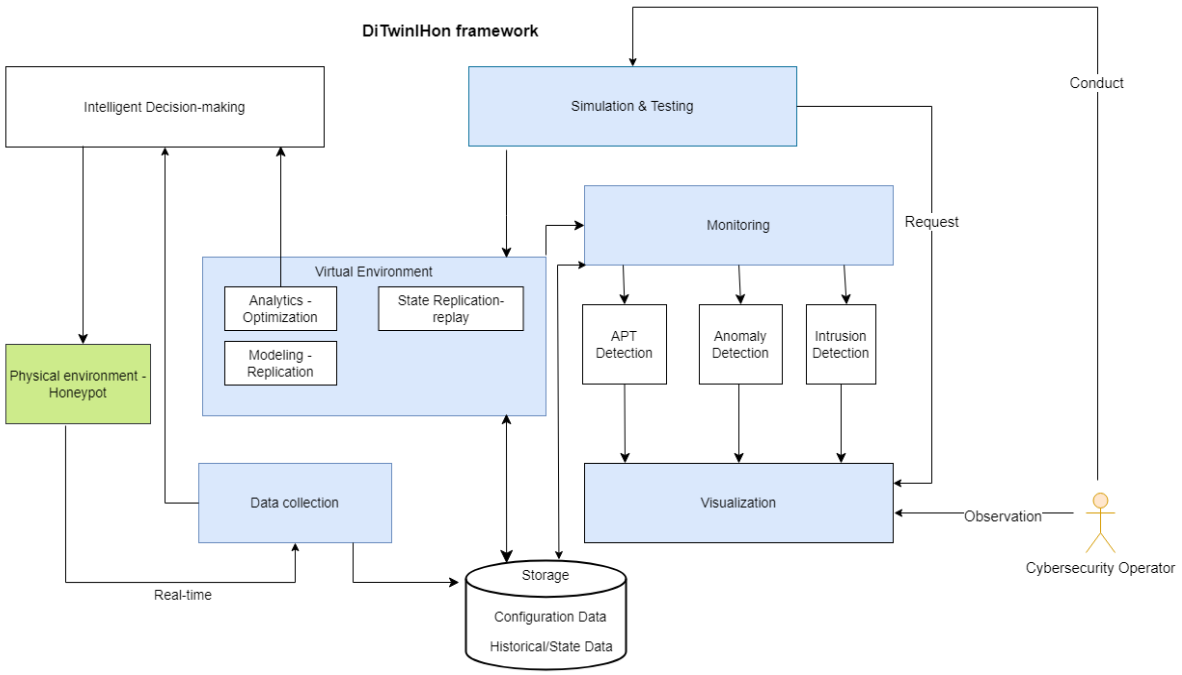

Figure 22. *DiTwinIHon framework*

4.3 DiTwinIHon framework (Digitally Twinned Intelligent Honeypot)

4.3.1 Data collection

First and foremost, the DT necessitates the collection of data in order for it to have accurate information reference to work with during the replication process. The source of information, in this case, is a real Honeypot with its features, configuration data and multiple functions which observes and logs the current network traffic. Configuration, current VM state, network traffic and other data that complete the description of the current condition of the real Honeypot is collected and stored in a Storage point (database). The process of collecting data is not done once and for all. The DT requires the real-time collection of data initially to replicate the Honeypot into its digital clone as accurately as possible. Collecting data in real-time also ensures that after

its creation, the DT will be able to keep up with any changes in the configuration of the real Honeypot. In this way, the DT is constantly 'fed' with new network traffic data for processing. The collected data is aggregated and inputted into machine learning algorithms, analytics, as well as representation models to provide knowledge and intelligence to the DT. Another source for data collection, which is not done in real-time, is the Decision-making outcome which contains information, such as pcap (network packet capture) files, about the best configuration for the Honeypot and state data of the best scenario that is worked out after the analysis and the decision-making processes. Data can be obtained from external sources, like databases of other DTs or cloud databases that contain relative information. This information either supports functions or provides assistive knowledge for certain tasks, which the DT did not initially obtain during the model creation stage.

4.3.2 Storage

With the utilization of databases, every piece of stored information can be taken into account, aggregated and used for analytics, computation and comparison. The "Storage" in the DT framework depicts the databases that store configuration and network traffic data and log files. Another type of data is stored in the Historical states of the DT, which were the previous and current states of the DT. It is important to store this amount of data, as machine learning and analytics methods require a great amount of data input for aggregation and comparison in order for them to achieve decent results.

After the cybersecurity operator requests the State replication-replay component, and after the recreation of the DT model based on the characteristics requested, there is also storing of the new DT model and the data that constitute the new model.

Last, the real twin is not the only source of information and data for Storage. A DT can associate and authorize other DTs or data storing points on the cloud, fetch information and store it in the Storage. In this thesis case, global cybersecurity attack definitions, characteristics and defence methods need to be stored for the DT as a knowledge base for the APT, anomaly and intrusion detection task. It is well known that the virus and malware definition and defence methods database must be constantly updated for the DT system to work with up-to-date information and learn to manage threats with cutting-edge techniques. Intelligent decision-making also benefits from relative external data storage. This data stored could be ML parameters and models that

match the current DT situation and enrich the knowledge base of the DT when comparisons or decisions are made.

4.3.3 Virtual Environment

During the replication phase, the DT adopts the same state and characteristics as the real Honeypot based on the collected data it obtains from it. For the digital twin honeypot to be able to provide a safe experimentation space and give the ability to analyze and make changes to the honeypot, a Virtual Environment is needed in which any changes in specifications or damage are done to the process will not affect the real Honeypot. The Virtual Environment extends the functionality of the DT, including analyses and optimization, which leads to intelligent Decision-making. Its advantage is that it can be used as a secure space for experimentation, where Simulation & Testing on the DT is feasible.

4.3.4 Simulation & Testing

Simulation of the current state and testing different configurations on the real Honeypot is a difficult and dangerous task during its runtime. When the Honeypot is running, it is not advised to make changes or experiment with it using malware and attacks. In many cases, though, it is necessary to experiment, especially when changes in configuration and testing on new parameters are needed to optimize the existing Honeypot. Cybersecurity operators are either forced to make decisions without being able to simulate and observe the results on the Honeypot or obliged to disconnect it from the network for a period to test and make additional changes. These methods are both dangerous and time-consuming because, in either case, the network will probably be in danger or unprotected, the log files kept during that period will be useless, and attempts will be in vain. Simulating and testing in an isolated environment gives a solution to this problem while it provides room for experimentation with real-time data, and even the case of destroying it would not cause a problem in the real world.

4.3.5 State Replication-replay

When simulating and testing various configurations on the real Honeypot, previous states are ignored as they cannot be recovered. During the runtime of the digital twin honeypot, though, this barrier can be overcome by storing every state stimuli in a database (Storage in Fig.22) and

reviving it whenever requested. Specifically, the digital twin honeypot can reproduce DT states on demand by storing stimuli produced during a DT state's creation process. This means that inside a database, previous states' stimuli are selected when a cybersecurity operator requires them. Cybersecurity operators can benefit from this feature as they can have a complete DT state outlook of a specific time or event that happened beforehand, whenever needed, can make analyses on those historical states and thus have a better view of the situation by combining previous knowledge.

4.3.6 Monitoring

Monitoring processes acquire previous Configuration and network traffic data to compare with the current data.

4.3.6.1 Intrusion Detection

Having an intrusion detected in real-time is not always the easiest task for a honeypot, as viewing log files and attempting to figure out mismatches or misbehaviour could take some time when new fast attack methods are on the rise. This task does not seem that complicated when the honeypot is equipped with previous data and can replicate historical states to compare during the investigation. What the honeypot is doing with this module is applying machine learning algorithms to detect matching patterns with malicious signatures from an attack signature database. The database contains all known attack signatures and thus can be utilized to find such activity and raise an alert.

4.3.6.2 Anomaly detection

When equipped with current and past configuration data and log files, the DT can detect any anomalies- misbehaviour or sudden unwanted changes in its current state- and can alert the operators and, simultaneously, provide them with a visual representation of the changes. Comparing the monitored activity and a baseline profile built within the honeypot's training phase and with a specific threshold set makes it easy to find any deviation in the monitored activity and consider it malicious.

4.3.6.3 APT detection

Advanced Persistent Threat is a cyber threat that aims to spy and extract valuable information from its target. This attack is a sophisticated descendant of the previously known Multi-Stage Attacks (MSAs). Those attacks are intended to obtain confidential information, intercept intelligence sent out by attacked computers, and enable the computers to automatically send related intelligence. While other types of attacks usually make their existence clear and hit the system "once and for all", APT attacks do not show up or uncover themselves, and they introduce themselves in several stages. They are underlying and thus manage not to raise suspicion for as long as the information needed is located or the damage is completed. In order to expose such attacks, it is best to keep previous states and log files to investigate the timing, the way they happened and the cause of the attacks [78].

4.3.6.4 Detection

APTs detection is a demanding task that cannot be done only by viewing previous and current log information. Detecting APTs may take years of logging misbehaviour, malfunctioning or unrequested changes in certain parameters and values and analyzing this information. Although APT detection is a new and not greatly explored area of research, [78] manages to give a solution by proposing a framework. This framework generates APT attack data and inputs it into a model. Then the model is trained with Hidden Markov models in order for it to recognize, learn and manage to detect the attack pattern. With this method, APT attack stage detection is possible. The model estimates the sequence of the APT stages and can achieve a high prediction accuracy of 91.8%. However, it is difficult to predict with such great accuracy when their model is not trained with 2-4 observations.

4.3.7 Visualization

This framework component uses the information available from previously mentioned features, intrusion detection, anomaly detection, and APT detection, and delivers visual "security-relevant" information. It is important as it provides the cybersecurity operator with visual changes or misbehaviour that could potentially impact the network or the Honeypot itself. With the help of this module, any unexpected changes in status or possible problematic behaviours are

made visible and can more easily be spotted when they can be viewed as charts and architectures instead of simple alerts and warnings.

4.3.8 Interaction

Cybersecurity operators can send commands to the DT for Simulation and Testing. They can control which states and data will be acquired from the databases through the State Replication-replay. Hence, they can bring the specific case scenario they want into life and make any tests or apply different configurations to it. Observing through the Visualization module is crucial in finding malfunctions or errors and analyzing the model specifications.

4.3.9 Intelligent Decision-making

After extensive analyses of historical and current data, monitoring network data, and predictive analyses for future states, the DT aims to update the real twin. Decision-making is an integral DT component that enables DT to fulfil that goal. With this component, the DT honeypot can identify the characteristics and the changes that need to be applied to them and can “make a decision” about the real honeypot for it to work more efficiently. After “making a decision”, the DT creates a configuration file and stores it in the Storage. The DT can apply this decision to the real honeypot by swapping its configuration file with the one created and ordering other appropriate changes in its working environment values.

Chapter 5. Methods and Tools available for Data Computational and Representation model creation

In this chapter, methods and tools to create the DT model are suggested based on the demands of the framework presented in the previous section.

5.1 DT Model creation

The first step in creating a DT model is to define how and which components will be utilized, what relationships should be formed between them and what connection measures should be taken into consideration (such as authorization of external applications or devices to access or provide data that is valuable during the various tasks or the processing phases). According to [79], there is no "one-way street" for creating a DT model. It can be created either by using a DT editor or by instantiating a DT based on already made akin DT models. Another way would be combining DT models or specific parts of them that are required. Analyses of an existing DT model and the DT characteristics could also provide the necessary information to create a new similar DT model. As described in sections 2.1 and 3.1.1, DTs require 'standard' and, in some cases, specific components to support their functions and tasks. There are also interrelations and data transfers between DT components and devices. These relationships between components and the connection mechanisms need to be described to create an accurate DT model. Depending on the nature of the problem, the creation of a DT model can be described and presented by providing data representation and computational models.

DT modelling, processing, querying and other basic functionalities, core components and relevant technologies that are fundamental for the DT model creation are shown in Figure 23 below.

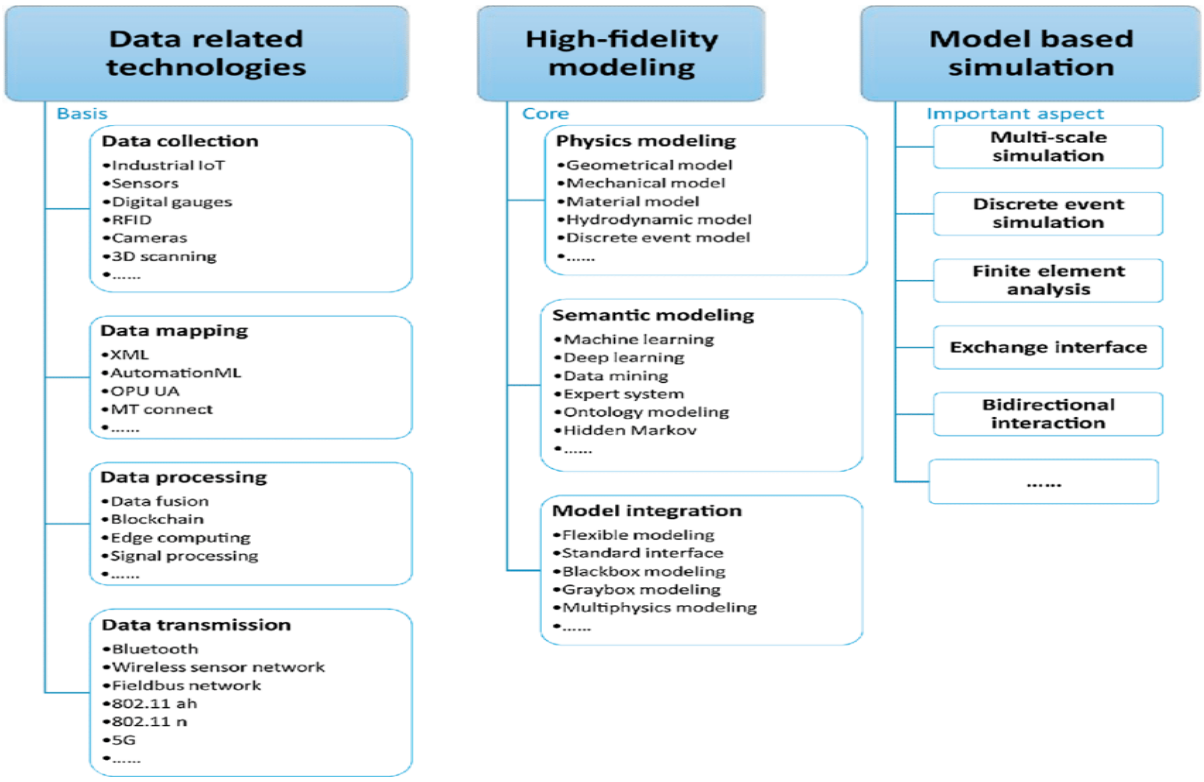

Figure 23. *Core components and related technologies required for a DT model creation [80]*

5.2 Data Representation model

In order to create a DT model, the first tasks required are the collection, exchange of data and search through data. Data representation models can represent the logical part behind those tasks and the related entities. Violeta Damjanovic and Behrendt & Wernher Behrendt [81] mention that when speaking about data representation models, we include the following types of models:

- Semantic data models
- XML-based models
- STEP model
- CAPEX model

Semantic data models are “high-level, user-oriented” data models designed to assist the user in viewing and interacting with the database [82]. XML-based models can encode documents in human and computer-readable formats [83]. A STEP model (Standard for Exchange of Product data) includes data that describe the components or entities completely by using a formal specification language [84]. CAPEX (which stands for computer-aided engineering exchange) is a “meta-model for the storage and exchange of engineering data models” [81].

Ontology models, according to this paper [85], contain rules created by the “concept definition”, can contain “conceptual knowledge of a DT”, and thus assist in restricting semantic concepts in domain-specific conceptual relationships and are commonly used for database mapping.

5.3 Data Computational model

After managing data collection, storage, and exchange, the DT needs to process and analyze the data. Analytics and processing during the lifecycle stages of the DT are highly demanded, especially to support the non-stop improvement they promise. The task of processing real-time and batch-oriented data collected from sensors is done with the help of computational data models [22].

By querying, aggregating, analyzing and processing real-time and batch-oriented data, DTs can optimize themselves, and by deploying ML and statistics on this data, analytics and decision-making are possible.

5.4 Tools for DT models creation

As mentioned in several sections in this thesis, and according to the framework presented in chapter 4, the DT model requires the following components:

- Representation
- Computation
- Communication
- Analysis
- Modelling
- Simulation
- Visualization
- Prediction
- Decision-making

In order to create the DT model, one needs to choose the relative tools to be able to do so. In this thesis, the easily accessible open-source tools that can be used to implement the DT tasks are included:

Table 4. Representation tools

Representation		
Semantics, Ontology, Modeling, XML-based and other data formats		
Tool	**Description**	**Source**
Microsoft Server SQL	Relational database management system	https://www.microsoft.com/en-us/sql-server/sql-server-downloads
MySQL	Open source relational database management system	https://www.mysql.com/
Apache Cassandra	Open source NoSQL database	https://cassandra.apache.org/_/quickstart.html
OntoSTEP	Open source ontology editor	https://www.nist.gov/services-resources/software/ontostep-plugin
Apache CouchDB	Database designed for the Web that stores data in JSON documents and supports powerful fault-tolerant storage	https://couchdb.apache.org/
OrientDB	A NoSQL database that stores a huge amount of documents/second and loads graphs equally fast	https://orientdb.org/
MongoDB	Cross-platform document-oriented NoSQL database	https://www.mongodb.com/
Neo4j	ACID-compliant transactional graph NoSQL database	https://neo4j.com/
Ontotext	Semantic graph database with text mining	https://www.ontotext.com/products/graphdb/
SQLite 3	Relational database management system	https://www.sqlite.org/download.html
SciGraph	Open source project to represent ontological data in Neo4j	https://www.springernature.com/gp/researchers/scigraph

Apache Flume	Distributed service that provides aggregation, and collection, can move massive amounts of data and is fault-tolerant	https://flume.apache.org/
InfluxDB	Open source time-series database that supports data transformation and prediction queries	https://www.influxdata.com/
AutomationML	Open XML-based and standardized data format	https://www.automationml.org/
STEP	Open format for systems to exchange design information	https://www.loc.gov/preservation/digital/formats/fdd/fdd000448.shtml

Table 5. Computation tools

Computation		
Search, analysis, processing and visualization of data		
Tool	**Description**	**Source**
Elasticsearch	Fast and scalable search and analytics engine	https://www.elastic.co/
Logstash	Open server-side data processing pipeline capable of ingesting and transforming data	https://www.elastic.co/logstash/
Kibana	Open user interface for visualization of data	https://www.elastic.co/kibana/
Elastic Stack	A platform for searching, analyzing and visualizing data in real-time that combines the previous three tools (Elasticsearch, Logstash, Kibana)	https://www.elastic.co/elastic-stack/
Matlab/Simulink	Data processing	https://www.mathworks.com/campaigns/products/trials.html

QFSM	A graphical tool for designing finite state machines	http://qfsm.sourceforge.net/
Beats	Open-source data shippers that capture data such as network traffic and metrics and send them to Elasticsearch	https://elastic.co/beats/
Apache Hadoop	The high-throughput system that can process large volumes of data using a distributed parallel processing paradigm and is used for batch queries	https://hadoop.apache.org/
HDFS (Hadoop Distributed File System)	A data storage system with cost-effective and reliable capability can handle both structured and unstructured data.	https://hadoop.apache.org/docs/r1.2.1/hdfs_design.html
Apache Spark	In-memory distributed data processing platform for large-scale data processing and batch analysis	https://spark.apache.org/

Table 6. Communication tools

Communication		
M2M connectivity, data exchange protocols		
Tool/Architecture / Protocol	**Description**	**Source**
J2EE (Java to platform, Enterprise Edition)	Standard platform for developing applications with SSH programming	https://www.oracle.com/tools/technologies/building-j2ee-web-applications.html
Master-Slave architecture RESTFul	The architectural style for an API that uses HTTP requests to access and use data	https://restfulapi.net/

Service-oriented architecture (SOA)	A software development model that relies on XML format, HTTP and SMTP and allows services to communicate across different platforms and languages to form applications	https://www.ibm.com/docs/en/rbd/9.5.1?topic=overview-service-oriented-architecture-soa#pegl_serv_overview__introsoa
Server-Client architecture	Computing model in which the server hosts, delivers and manages most of the resources and services to be consumed by the client	https://cio-wiki.org/wiki/Client_Server_Architecture
OPC Unified Architecture	M2M communication protocol	http://www.open62541.org/
MQTT (Message Queue Telemetry Transport)	M2M connectivity and messaging protocol which is optimized to connect physical devices with enterprise servers	https://github.com/mqtt/mqtt.org
XMPP (Extensible Messaging and Present Protocol)	Open XML technology for real-time communication	https://xmpp.org/
OpenDDS (Data Distribution Service)	Open source of DDS for real-time systems	https://opendds.org/downloads.html
OMA LWM2M (Lightweight M2M)	Open-source implementation for sensor networks and M2M communication	https://omaspecworks.org/
NTP (Network Time Protocol)	Protocol designed to synchronize the devices within a network	https://support.ntp.org/bin/view/Main/ExternalTimeRelatedLinks
PTP (Precision Time Protocol)	Ethernet or IP-based protocol for synchronization of time with high precision on a collection of devices within a network	https://github.com/ptpd/ptpd

TCP (Transmission Control Protocol)	Communications standard protocol for enabling two hosts to exchange data	https://tcpipmanager.sourceforge.io/download.html
UDP (User Datagram Protocol)	The communications protocol used to establish low-latency and loss tolerating connections between applications on the internet	https://github.com/nikhilroxtomar/UDP-Client-Server-Program-in-C
Eclipse Mosquitto	Open source message broker that implements the MQTT protocol	https://mosquitto.org/

Table 7. Machine Learning tools

Machine Learning		
Simulation, Analytics, Prediction and Decision-making		
Tool	**Description**	**Source**
Mworks software	Suite of open source applications and libraries for designing and running real-time experiments	https://mworks.github.io/
Tensorflow	Open-source library for dataflow and differentiable programming for machine learning applications	https://github.com/tensorflow/tensorflow
SciPy	Collection of open source software for scientific computing in Python	https://scipy.org/
R project	Statistical computing and graphics in R language	https://www.r-project.org/
ML and Deep Learning frameworks	Keras, Caffe, PyTorch, Torch Lua	http://keras.io/ https://caffe.berkeleyvision.org/ https://github.com/pytorch/pytorch https://github.com/torch/torch7 https://www.lua.org/

Apache MxNet	A powerful deep learning framework	https://mxnet.apache.org/versions/1.9.1/
Auto ML	Services that provide automated machine learning models	https://cloud.google.com/automl
OpenNN (Open Neural Networks)	Software Library that implements neural networks	https://www.opennn.net/
H2O	Open-source distributed in-memory machine learning platform with linear scalability, supporting the most widely used statistical & machine learning algorithms, including deep learning	https://h2o.ai/platform/ai-cloud/make/h2o/
CNTK (Microsoft Cognitive Toolkit)	A library that contains all of the blocks needed to build a neural network	https://github.com/microsoft/CNTK
Pydecisions library	Python library of management decision-making techniques	https://pypi.org/project/pydecisions/
MVNHMM (MultiVariate Nonhomogeneous Hidden Markov Model)	The toolbox that contains algorithms for modelling multivariate time series with hidden Markov models	http://www.datalab.uci.edu/resources/mvnhmm/
PyCaret	Python library that assists in ML Regression models creation	https://github.com/pycaret/pycaret

Chapter 6. Conclusion and Future work

In the last chapter, ideas about the future development of the framework are presented as well as the conclusion of the thesis.

6.1. Conclusion

This thesis presented the DT concept and components needed to create a model by referring to recent scientific literature. Frameworks that used DTs have been presented, mainly in the domain of Cybersecurity, and a new framework for a DT Honeypot has been proposed. Lastly, methods and tools currently available to visualize this framework have been suggested.

6.2 Future work

This framework is presented in the thesis as honeypot specific, which means that there are custom functionalities that support the tasks of a real honeypot, and therefore the main focus was its features and the network traffic logged by it. Moving on with the framework, it would be useful to extend its capabilities by digitally twinning the environment of the honeypot as well. Having a generator responsible for extracting the specifications of the physical environment (such as the network topology or the IoT devices connected) was an initial suggestion in the framework. At the same time, the Honeypot is a mimicking device and could mimic a CPS or an IoT device. Recreating the physical environment digitally gives an advantage to the security operators in these cases as they can observe how certain changes or attacks on one device (the Honeypot) can affect other devices connected to it. This thesis did not include this idea, though, as the Honeypot to-be-digitally-twinned controls the network traffic and does not include other IoT or CPS devices. This idea could be extended in future work. By doing so, security operators would have a clearer view of the whole network of devices that interact with the honeypot, and this would assist in getting feedback from those devices when testing or while a real attack is taking place. If the framework expands its “knowledge” to other devices (especially IoT devices), a great addition would be a surveillance component for their improvement, similar to the honeypot decision-making one. This component would retrieve alerts and log their current state and, after analysis, would send configuration suggestions to the device in order for it to work more efficiently and securely. Last but not least, due to the rise of new methods of honeypot

detection through ML, as stated in this paper [86], for reference, a component to prevent the detection mechanism would improve the honeypot's efficiency.

References

[1] H. Aydemir, U. Zengin, and U. Durak, "The Digital Twin Paradigm for Aircraft Review and Outlook," presented at the AIAA Scitech 2020 Forum, Orlando, FL, Jan. 2020. doi: 10.2514/6.2020-0553.

[2] H. Zhang, G. Zhang, and Q. Yan, "Digital twin-driven cyber-physical production system towards smart shop-floor," *J. Ambient Intell. Humaniz. Comput.*, vol. 10, no. 11, pp. 4439–4453, Nov. 2019, doi: 10.1007/s12652-018-1125-4.

[3] J. Guo and Z. Lv, "Application of Digital Twins in multiple fields," *Multimed. Tools Appl.*, Feb. 2022, doi: 10.1007/s11042-022-12536-5.

[4] D. H. Gelernter, *Mirror worlds, or, The day software puts the universe in a shoebox--: how it will happen and what it will mean*. New York: Oxford University Press, 1991.

[5] M. W. Grieves, "Product lifecycle management: the new paradigm for enterprises," *Int. J. Prod. Dev.*, vol. 2, no. 1/2, p. 71, 2005, doi: 10.1504/IJPD.2005.006669.

[6] E. VanDerHorn and S. Mahadevan, "Digital Twin: Generalization, characterization and implementation," *Decis. Support Syst.*, vol. 145, p. 113524, Jun. 2021, doi: 10.1016/j.dss.2021.113524.

[7] M. Grieves, "Origins of the Digital Twin Concept," 2016, doi: 10.13140/RG.2.2.26367.61609.

[8] M. Abramovici, J. C. Göbel, and P. Savarino, "Virtual Twins as Integrative Components of Smart Products," in *Product Lifecycle Management for Digital Transformation of Industries*, vol. 492, R. Harik, L. Rivest, A. Bernard, B. Eynard, and A. Bouras, Eds. Cham: Springer International Publishing, 2016, pp. 217–226. doi: 10.1007/978-3-319-54660-5_20.

[9] G. N. Schroeder, C. Steinmetz, C. E. Pereira, and D. B. Espindola, "Digital Twin Data Modeling with AutomationML and a Communication Methodology for Data Exchange," *IFAC-Pap.*, vol. 49, no. 30, pp. 12–17, 2016, doi: 10.1016/j.ifacol.2016.11.115.

[10] T. Gabor, L. Belzner, M. Kiermeier, M. T. Beck, and A. Neitz, "A Simulation-Based Architecture for Smart Cyber-Physical Systems," in *2016 IEEE International Conference on Autonomic Computing (ICAC)*, Wuerzburg, Germany, Jul. 2016, pp. 374–379. doi: 10.1109/ICAC.2016.29.

[11] 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-Pap.*, vol. 48, no. 3, pp. 567–572, 2015, doi: 10.1016/j.ifacol.2015.06.141.

[12] M. Grieves and J. Vickers, "Digital Twin: Mitigating Unpredictable, Undesirable Emergent Behavior in Complex Systems," in *Transdisciplinary Perspectives on Complex Systems*, F.-J. Kahlen, S. Flumerfelt, and A. Alves, Eds. Cham: Springer International Publishing, 2017, pp. 85–113. doi: 10.1007/978-3-319-38756-7_4.

[13] S. Boschert and R. Rosen, "Digital Twin—The Simulation Aspect," in *Mechatronic Futures*, P. Hehenberger and D. Bradley, Eds. Cham: Springer International Publishing, 2016, pp. 59–74. doi: 10.1007/978-3-319-32156-1_5.

[14] A. C. F. da Silva, S. Wagner, E. Lazebnik, and E. Traitel, "Using a Cyber Digital Twin for Continuous Automotive Security Requirements Verification." arXiv, Sep. 30, 2021. Accessed: Jun. 28, 2022. [Online]. Available: http://arxiv.org/abs/2102.00790

[15] J. V. Micheal Grieves, *Trandisciplinary Perspectives on Complex Systems*. Berlin,Germany: Springer International Publishing, 2017. [Online]. Available: https://www.springerprofessional.de/en/digitaltwin-mitigating-unpredictable-undesirableemergent-behav/10588328

[16] B. Schleich, N. Anwer, L. Mathieu, and S. Wartzack, "Shaping the digital twin for design and production engineering," *CIRP Ann.*, vol. 66, no. 1, pp. 141–144, 2017, doi: 10.1016/j.cirp.2017.04.040.

[17] R. Minerva, G. M. Lee, and N. Crespi, "Digital Twin in the IoT Context: A Survey on Technical Features, Scenarios, and Architectural Models," *Proc. IEEE*, vol. 108, no. 10, pp. 1785–1824, Oct. 2020, doi: 10.1109/JPROC.2020.2998530.

[18] A. Sharma, E. Kosasih, J. Zhang, A. Brintrup, and A. Calinescu, "Digital Twins: State of the Art Theory and Practice, Challenges, and Open Research Questions." arXiv, Dec. 04, 2020. Accessed: Jun. 28, 2022. [Online]. Available: http://arxiv.org/abs/2011.02833

[19] S. V. Nath, P. van Schalkwyk, and D. Isaacs, *Building industrial digital twins: design, develop, and deploy digital twins solutions for real-world industries using Azure Digital Twins*. Birmingham Mumbai: Packt, 2021.

[20] A. Fuller, Z. Fan, C. Day, and C. Barlow, "Digital Twin: Enabling Technologies, Challenges and Open Research," *IEEE Access*, vol. 8, pp. 108952–108971, 2020, doi: 10.1109/ACCESS.2020.2998358.

[21] C. Verdouw, B. Tekinerdogan, A. Beulens, and S. Wolfert, "Digital twins in smart farming," *Agric. Syst.*, vol. 189, p. 103046, Apr. 2021, doi: 10.1016/j.agsy.2020.103046.

[22] K. Y. H. Lim, P. Zheng, and C.-H. Chen, "A state-of-the-art survey of Digital Twin: techniques, engineering product lifecycle management and business innovation perspectives," *J. Intell. Manuf.*, vol. 31, no. 6, pp. 1313–1337, Aug. 2020, doi: 10.1007/s10845-019-01512-w.

[23] Q. Qi *et al.*, "Enabling technologies and tools for digital twin," *J. Manuf. Syst.*, vol. 58, pp. 3–21, Jan. 2021, doi: 10.1016/j.jmsy.2019.10.001.

[24] Dr. C. G. K. Eric Harper Dr. Somayeh Malakuti, "Journal of Innovation, Industrial Internet Consortium," *Digit. Twin Archit. Stand.*, Nov. 2019, [Online]. Available: https://www.iiconsortium.org/news-pdf/joi-articles/2019-November-JoI-Digital-Twins-in-Industrial-Applications.pdf

[25] C. Semeraro, M. Lezoche, H. Panetto, and M. Dassisti, "Digital twin paradigm: A systematic literature review," *Comput. Ind.*, vol. 130, p. 103469, Sep. 2021, doi: 10.1016/j.compind.2021.103469.

[26] M. Tracy, W. Jansen, K. Scarfone, and J. Butterfield, "Guidelines on Electronic Mail Security," p. 139.

[27] Z. Chao-yang, "DOS Attack Analysis and Study of New Measures to Prevent," in *2011 International Conference on Intelligence Science and Information Engineering*, Wuhan, China, Aug. 2011, pp. 426–429. doi: 10.1109/ISIE.2011.66.

[28] P. A. Grassi, M. E. Garcia, and J. L. Fenton, "Digital identity guidelines: revision 3," National Institute of Standards and Technology, Gaithersburg, MD, NIST SP 800-63-3, Jun. 2017. doi: 10.6028/NIST.SP.800-63-3.

[29] R. Kissel, A. Regenscheid, M. Scholl, and K. Stine, "Guidelines for Media Sanitization," National Institute of Standards and Technology, NIST SP 800-88r1, Dec. 2014. doi: 10.6028/NIST.SP.800-88r1.

[30] A. Regenscheid, G. Beier, S. Chokhani, P. Hoffman, J. Knoke, and S. Shorter, "Information system security best practices for UOCAVA-supporting systems," National Institute of Standards and Technology, Gaithersburg, MD, NIST IR 7682, 2010. doi: 10.6028/NIST.IR.7682.

[31] I. Lee, S. Jeong, S. Yeo, and J. Moon, "A novel method for SQL injection attack detection based on removing SQL query attribute values," *Math. Comput. Model.*, vol. 55, no. 1–2, pp. 58–68, Jan. 2012, doi: 10.1016/j.mcm.2011.01.050.

[32] S. U. M. Kamal, R. J. A. Ali, H. K. Alani, and E. S. Abdulmajed, "SURVEY AND BRIEF HISTORY ON MALWARE IN NETWORK SECURITY CASE STUDY: VIRUSES, WORMS AND BOTS," vol. 11, no. 1, p. 16, 2016.

[33] L. Pupillo, S. Fantin, A. Ferreira, C. Polito, and Centre for European Policy Studies, *Artificial intelligence and cybersecurity technology, governance and policy challenges: final report of a CEPS Task Force*. 2021. Accessed: Jun. 29, 2022. [Online]. Available:

https://www.ceps.eu/download/publication/?id=33262&pdf=CEPS-TFR-Artificial-Intelligence-and-Cybersecurity.pdf

[34] B. B. K. Swathi Pai M., "Big Data Security Analytic: A classification technique for Intrusion Detection System," presented at the IFERP International Conference, Bengaluru, Sep. 2015.

[35] P. I. Radoglou-Grammatikis and P. G. Sarigiannidis, "Securing the Smart Grid: A Comprehensive Compilation of Intrusion Detection and Prevention Systems," *IEEE Access*, vol. 7, pp. 46595–46620, 2019, doi: 10.1109/ACCESS.2019.2909807.

[36] L. Spitzner, "Honeypots: catching the insider threat," in *19th Annual Computer Security Applications Conference, 2003. Proceedings.*, Las Vegas, Nevada, USA, 2003, pp. 170–179. doi: 10.1109/CSAC.2003.1254322.

[37] L. Spitzner, "The Value of Honeypots, Part One: Definitions and Values of Honeypots," 2001. [Online]. Available: http://www.securityfocus.com/infocus/1492

[38] M. Hernandez y Lopez and C. Francisco Lerma Reséndez, "Honeypots: Basic Concepts Classification and Educational Use as Resources in Information Security Education and Courses," presented at the InSITE 2008: Informing Science + IT Education Conference, 2008. doi: 10.28945/3186.

[39] S. A. Joshi R.C., *Honeypots: A new paradigm to information security*. 2011.

[40] L. Zobal, D. Kolar, and R. Fujdiak, "Current State of Honeypots and Deception Strategies in Cybersecurity," in *2019 11th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, Dublin, Ireland, Oct. 2019, pp. 1–9. doi: 10.1109/ICUMT48472.2019.8970921.

[41] S. Kumar, R. Sehgal, and J. S. Bhatia, "Hybrid honeypot framework for malware collection and analysis," in *2012 IEEE 7th International Conference on Industrial and Information Systems (ICIIS)*, Chennai, India, Aug. 2012, pp. 1–5. doi: 10.1109/ICIInfS.2012.6304786.

[42] I. Mokube and M. Adams, "Honeypots: concepts, approaches, and challenges," in *Proceedings of the 45th annual southeast regional conference on - ACM-SE 45*, Winston-Salem, North Carolina, 2007, p. 321. doi: 10.1145/1233341.1233399.

[43] H. Almohannadi, I. Awan, J. Al Hamar, A. Cullen, J. P. Disso, and L. Armitage, "Cyber Threat Intelligence from Honeypot Data Using Elasticsearch," in *2018 IEEE 32nd International Conference on Advanced Information Networking and Applications (AINA)*, Krakow, May 2018, pp. 900–906. doi: 10.1109/AINA.2018.00132.

[44] E. A. Lee, "Cyber-Physical Systems - Are Computing Foundations Adequate?," p. 9.

[45] "Framework for Cyber-Physical Systems: Volume1, Overview," *NIST Spec. Publ. 1500-201*, doi: https://doi.org/10.6028/NIST.SP.1500-201.

[46] J. C. Olivares-Rojas, E. Reyes-Archundia, J. A. Gutierrez-Gnecchi, I. Molina-Moreno, J. Cerda-Jacobo, and A. Mendez-Patino, "Towards Cybersecurity of the Smart Grid using Digital Twins," *IEEE Internet Comput.*, pp. 1–1, 2021, doi: 10.1109/MIC.2021.3063674.

[47] A. Pauna, "Improved self adaptive honeypots capable of detecting rootkit malware," in *2012 9th International Conference on Communications (COMM)*, Bucharest, Romania, Jun. 2012, pp. 281–284. doi: 10.1109/ICComm.2012.6262612.

[48] Chris Anderson, "An Investigation of Self-Learning and Self-Protection for Adaptive Digital Twins," The University of Waikato, Research, 2021.

[49] W. Fan, Z. Du, D. Fernandez, and V. A. Villagra, "Enabling an Anatomic View to Investigate Honeypot Systems: A Survey," *IEEE Syst. J.*, vol. 12, no. 4, pp. 3906–3919, Dec. 2018, doi: 10.1109/JSYST.2017.2762161.

[50] M. Eckhart and A. Ekelhart, "A Specification-based State Replication Approach for Digital Twins," in *Proceedings of the 2018 Workshop on Cyber-Physical Systems Security and PrivaCy*, Toronto Canada, Jan. 2018, pp. 36–47. doi: 10.1145/3264888.3264892.

[51] M. Eckhart, A. Ekelhart, and E. Weippl, "Enhancing Cyber Situational Awareness for Cyber-Physical Systems through Digital Twins," in *2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, Zaragoza, Spain, Sep. 2019, pp. 1222–1225. doi: 10.1109/ETFA.2019.8869197.

[52] A. Kummerow *et al.*, "Challenges and opportunities for phasor data based event detection in transmission control centers under cyber security constraints," in *2019 IEEE Milan PowerTech*, Milan, Italy, Jun. 2019, pp. 1–6. doi: 10.1109/PTC.2019.8810711.

[53] E. C. Balta, D. M. Tilbury, and K. Barton, "A Digital Twin Framework for Performance Monitoring and Anomaly Detection in Fused Deposition Modeling," in *2019 IEEE 15th International Conference on Automation Science and Engineering (CASE)*, Vancouver, BC, Canada, Aug. 2019, pp. 823–829. doi: 10.1109/COASE.2019.8843166.

[54] Y. Lu, C. Liu, K. I.-K. Wang, H. Huang, and X. Xu, "Digital Twin-driven smart manufacturing: Connotation, reference model, applications and research issues," *Robot. Comput.-Integr. Manuf.*, vol. 61, p. 101837, Feb. 2020, doi: 10.1016/j.rcim.2019.101837.

[55] A. J. H. Redelinghuys, A. H. Basson, and K. Kruger, "A six-layer architecture for the digital twin: a manufacturing case study implementation," *J. Intell. Manuf.*, vol. 31, no. 6, pp. 1383–1402, Aug. 2020, doi: 10.1007/s10845-019-01516-6.

[56] A. Pokhrel, V. Katta, and R. Colomo-Palacios, "Digital Twin for Cybersecurity Incident Prediction: A Multivocal Literature Review," in *Proceedings of the IEEE/ACM 42nd International Conference on Software Engineering Workshops*, Seoul Republic of Korea, Jun. 2020, pp. 671–678. doi: 10.1145/3387940.3392199.

[57] Q. Qiao, J. Wang, L. Ye, and R. X. Gao, "Digital Twin for Machining Tool Condition Prediction," *52nd CIRP Conf. Manuf. Syst. CMS Ljubl. Slov. June 12-14 2019*, vol. 81, pp. 1388–1393, Jan. 2019, doi: 10.1016/j.procir.2019.04.049.

[58] D. Weyns, "Engineering Self-Adaptive Software Systems – An Organized Tour," in *2018 IEEE 3rd International Workshops on Foundations and Applications of Self* Systems (FAS*W)*, Trento, Sep. 2018, pp. 1–2. doi: 10.1109/FAS-W.2018.00012.

[59] M. Schluse, L. Atorf, and J. Rossmann, "Experimentable digital twins for model-based systems engineering and simulation-based development," in *2017 Annual IEEE International Systems Conference (SysCon)*, Montreal, QC, Canada, Apr. 2017, pp. 1–8. doi: 10.1109/SYSCON.2017.7934796.

[60] G. N. Schroeder, C. Steinmetz, R. N. Rodrigues, R. V. B. Henriques, A. Rettberg, and C. E. Pereira, "A Methodology for Digital Twin Modeling and Deployment for Industry 4.0," *Proc. IEEE*, vol. 109, no. 4, pp. 556–567, Apr. 2021, doi: 10.1109/JPROC.2020.3032444.

[61] F. Böhm, M. Dietz, T. Preindl, and G. Pernul, "Augmented Reality and the Digital Twin: State-of-the-Art and Perspectives for Cybersecurity," *J. Cybersecurity Priv.*, vol. 1, no. 3, pp. 519–538, Sep. 2021, doi: 10.3390/jcp1030026.

[62] J. Autiosalo, J. Vepsalainen, R. Viitala, and K. Tammi, "A Feature-Based Framework for Structuring Industrial Digital Twins," *IEEE Access*, vol. 8, pp. 1193–1208, 2020, doi: 10.1109/ACCESS.2019.2950507.

[63] M. Dietz and G. Pernul, "Unleashing the Digital Twin's Potential for ICS Security," *IEEE Secur. Priv.*, vol. 18, no. 4, pp. 20–27, Jul. 2020, doi: 10.1109/MSEC.2019.2961650.

[64] M. Dietz, M. Vielberth, and G. Pernul, "Integrating digital twin security simulations in the security operations center," in *Proceedings of the 15th International Conference on Availability, Reliability and Security*, Virtual Event Ireland, Aug. 2020, pp. 1–9. doi: 10.1145/3407023.3407039.

[65] D. Holmes, M. Papathanasaki, L. Maglaras, M. A. Ferrag, S. Nepal, and H. Janicke, "Digital Twins and Cyber Security – solution or challenge?," in *2021 6th South-East Europe Design Automation, Computer Engineering, Computer Networks and Social Media Conference (SEEDA-CECNSM)*, Preveza, Greece, Sep. 2021, pp. 1–8. doi: 10.1109/SEEDA-CECNSM53056.2021.9566277.

[66] J. Moyne, S. Mashiro, and D. Gross, "Determining a security roadmap for the microelectronics industry," in *2018 29th Annual SEMI Advanced Semiconductor Manufacturing Conference (ASMC)*, Saratoga Springs, NY, USA, Apr. 2018, pp. 291–294. doi: 10.1109/ASMC.2018.8373213.

[67] J. Moyne *et al.*, "A Requirements Driven Digital Twin Framework: Specification and Opportunities," *IEEE Access*, vol. 8, pp. 107781–107801, 2020, doi: 10.1109/ACCESS.2020.3000437.

[68] M. Tauber, F. Burgenland, and C. Schmittner, "Enabling Security and Safety Evaluation in Industry 4.0 Use Cases with Digital Twins," p. 2.

[69] D. Ariansyaha *et al.*, "Digital Twin Development: A Step by Step Guideline," *SSRN Electron. J.*, 2020, doi: 10.2139/ssrn.3717726.

[70] "IoT4CPS-Trustworthy IoT for CPS," IoT4CPS Consortium, Project No.863129. [Online]. Available: https://iot4cps.at/wp-content/uploads/2020/10/IoT4CPS_D5.5.3_v1.2_FINAL.pdf

[71] H.-M. Ríos, José Juan ,Oliva, Manuel, Mas, Fernando, "Product Avatar as Digital Counterpart of a Physical Individual Product: Literature Review and Implications in an Aircraft," Jul. 2015. doi: 10.3233/978-1-61499-544-9-657.

[72] W. D. Lin, Y. H. Low, Y. T. Chong, and C. L. Teo, "Integrated Cyber Physical Simulation Modelling Environment for Manufacturing 4.0," in *2018 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, Bangkok, Dec. 2018, pp. 1861–1865. doi: 10.1109/IEEM.2018.8607696.

[73] W. D. Lin and M. Y. H. Low, "Concept and Implementation of a Cyber-Physical Digital Twin for a SMT Line," p. 5, 2019.

[74] K. T. Park *et al.*, "Design and implementation of a digital twin application for a connected micro smart factory," *Int. J. Comput. Integr. Manuf.*, vol. 32, no. 6, pp. 596–614, Jun. 2019, doi: 10.1080/0951192X.2019.1599439.

[75] M. Putz and A. Schlegel, "The Role of Simulation in a Cyber-Physical Production Environment," p. 7.

[76] M. Eckhart and A. Ekelhart, "Towards Security-Aware Virtual Environments for Digital Twins," in *Proceedings of the 4th ACM Workshop on Cyber-Physical System Security*, Incheon Republic of Korea, May 2018, pp. 61–72. doi: 10.1145/3198458.3198464.

[77] D. Jones, C. Snider, A. Nassehi, J. Yon, and B. Hicks, "Characterising the Digital Twin: A systematic literature review," *CIRP J. Manuf. Sci. Technol.*, vol. 29, pp. 36–52, May 2020, doi: 10.1016/j.cirpj.2020.02.002.

[78] I. Ghafir *et al.*, "Hidden Markov Models and Alert Correlations for the Prediction of Advanced Persistent Threats," *IEEE Access*, vol. 7, pp. 99508–99520, 2019, doi: 10.1109/ACCESS.2019.2930200.

[79] L. Stojanovic *et al.*, "Methodology and Tools for Digital Twin Management—The FA3ST Approach," *IoT*, vol. 2, no. 4, pp. 717–740, Nov. 2021, doi: 10.3390/iot2040036.

[80] M. Liu, S. Fang, H. Dong, and C. Xu, "Review of digital twin about concepts, technologies, and industrial applications," *J. Manuf. Syst.*, vol. 58, pp. 346–361, Jan. 2021, doi: 10.1016/j.jmsy.2020.06.017.

[81] V. Damjanovic-Behrendt and W. Behrendt, "An open source approach to the design and implementation of Digital Twins for Smart Manufacturing," *Int. J. Comput. Integr. Manuf.*, vol. 32, no. 4–5, pp. 366–384, May 2019, doi: 10.1080/0951192X.2019.1599436.

[82] M. Hammer and D. McLeod, "The semantic data model: a modelling mechanism for data base applications," in *Proceedings of the 1978 ACM SIGMOD international conference on management of data - SIGMOD '78*, Austin, Texas, 1978, p. 26. doi: 10.1145/509252.509264.

[83] S. S. Choi, T. H. Yoon, and S. D. Noh, "XML-based neutral file and PLM integrator for PPR information exchange between heterogeneous PLM systems," *Int. J. Comput. Integr. Manuf.*, vol. 23, no. 3, pp. 216–228, Mar. 2010, doi: 10.1080/09511920903443234.

[84] "Procedure for the maintenance of the STEP Module and Resource Library. ISO TC 184/SC 4 N2538, 2009."

[85] S. Singh *et al.*, "Data management for developing digital twin ontology model," *Proc. Inst. Mech. Eng. Part B J. Eng. Manuf.*, vol. 235, no. 14, pp. 2323–2337, Dec. 2021, doi: 10.1177/0954405420978117.

[86] C. Huang, J. Han, X. Zhang, and J. Liu, "Automatic Identification of Honeypot Server Using Machine Learning Techniques," *Secur. Commun. Netw.*, vol. 2019, pp. 1–8, Sep. 2019, doi: 10.1155/2019/2627608.