AI-enabled Cybersecurity using Synthetic Data

Fabrizio Baiardi
Dipartimento di Informatica, Università di Pisa, Italy
0000-0001-9797-2380
fabrizio.baiardi@unipi.it

Salvatore Ruggieri
Dipartimento di Informatica, Università di Pisa, Italy
0000-0002-1917-6087
salvatore.ruggieri@unipi.it

Vincenzo Sammartino
Dipartimento di Informatica, Università di Pisa, Italy
0009-0002-4632-1179
vincenzo.sammartino@phd.unipi.it

***Abstract*—Historical data is not always adequate to train AI models to secure ICT infrastructures due to the dynamic risk landscape. This paper introduces a novel methodology that combines AI-driven adversary simulation with digital twin technology to generate synthetic data to train AI models. A security twin extends an infrastructure inventory with information on current vulnerabilities and attacks. By describing threat agents through other twins, we simulate their attack strategies to discover how they exploit the infrastructure' vulnerabilities and implement their intrusions. A Monte Carlo approach is adopted that runs multiple independent simulations, capturing alternative intrusion scenarios. This method addresses the challenges of data shifts in cybersecurity by producing synthetic data to faithfully describe rapidly evolving environments. This results in more accurate risk management and better resilience. Initial experimental results demonstrate the effectiveness of security twins in assessing and managing the risk due to intrusions. An extension of the digital twin technology to proactive cybersecurity offers significant implications for smart industries, healthcare, and critical infrastructure defence.**

***Index Terms*—security twin, adversary simulation, data shift, synthetic data, AI in cybersecurity, Monte Carlo simulations**

I. Introduction

The development of critical ICT infrastructures has led to significant economic growth and improved social connectivity. However, it has also increased the risk of malicious misuse, with social and economic impacts [1]–[4].

As critical information infrastructures, or simply infrastructures, become more and more interconnected and pervasive, several threat actors, or simply actors, attempt to orchestrate intrusions to control and manipulate infrastructure components, to damage public and private institutions. These intrusions exploit various vulnerabilities in software, hardware, and even human elements such as users and administrators of the infrastructure. To amplify their impact, actors can also employ physical attacks or phishing techniques.

The overall risk scenario is highly dynamic, with frequent statistical distribution shifts that reduce the reliability of historical data to estimate probabilities of interest and to train AI models [5] to proactively identify intrusions. This dynamic nature, combined with the increasing size and complexity of infrastructures, exacerbates the challenges of effectively managing security, particularly in the absence of formal models and robust methodologies.

Synthetic data generation through adversary simulations of intrusions offers a practical solution to this challenge. Inspired by digital twin technology [6], these simulations use executable models, or *digital twins*, of both the target infrastructure and the actor. We denote the twin of the infrastructure as a *security twin*, focusing on security and safety issues rather than providing a detailed virtual replica of the infrastructure, and the twin of the actor as an *attacker twin*, simulating adversary behavior and his strategies.

Moreover, exploiting AI methods in the design of security twins significantly improves the simulation of the behavior of actors in intrusions. The adoption of a Monte Carlo method that runs multiple independent simulations of intrusions, results in a comprehensive exploration of alternative intrusions and produces the proper amount of synthetic data to train robust models specialized on an infrastructure.

Sect. II of this paper briefly describes intrusions and the information of interest on intrusions. Then Sect. III discusses the issues in predicting these intrusions. The following sections describe the proposed solution. Sect. IV discusses the forecasting of intrusions using twins, and Sect. V focuses on the analysis of synthetic data. Sect. VI presents preliminary experimental results.

II. Executing and Describing Intrusions

We introduce some definitions regarding actors and intrusions. We consider intelligent actors with a predefined goal, a set of privileges, i.e., access rights to infrastructure resources they aim to acquire. An actor is assumed to have only one goal, and actors do not cooperate in an intrusion.

An intrusion is a sequence of actions an actor executes to reach its goal and control some physical or logical resources [7]–[9]. The alternative actions of an actor can be classified in terms of tactics, techniques, and procedures (TTPs) as defined in the MITRE ATT&CK Matrix [10] that defines 12 classes of actions in an intrusion. Some actions collect information to discover the infrastructure components and identify the potential weaknesses and related attacks. A possible example is the execution of a vulnerability scan. One class of the ATT&Ck Matrix includes attacks that exploit the weaknesses of the target infrastructure. Other attacks target (human) users modeled as additional infrastructure components. These are the actions that the actor executes to acquire some access rights. Lastly, some actions manipulate some components to produce an impact, i.e. damage. An action may fail because of the environment or for other reasons. The failure probability is

known for each action. Each action is paired with properties that include the success probability, the execution time, the noise it generates (the probability of being detected by an IDS), and pre- and post-conditions that describe, respectively, the rights to execute the action and the information and the rights a successful action grants. These conditions are deduced from several vulnerability databases [11], [12]. An intrusion is successful if, after its last action, the actor reaches its goal so that it can lock, encrypt, delete, or steal the resources it controls. To estimate the overall risk, we need a proper sample of the intrusions actors can implement and their success probability.

We describe each action through a tuple with six elements:

<A, IPs, IPd, precond, postcond, information>

where:

1. *A* is the specific action taken by the actor, corresponding to a technique or subtechnique in the ATT&CK matrix [10]. If the action is an attack, A also includes details about any enabling vulnerabilities.
2. *IPs*, the IP address of the source node where *A* is initiated.
3. *IPd*, the IP address of the target node of *A*.
4. *precond*, the access rights the actor needs to execute *A*.
5. *postcond*, the access rights granted to the actor after performing *A*.
6. *information*, the knowledge acquired by the actor as a result of *A*.

As an example, suppose *IPs* is the address of a node executing an active scanning technique (*A*) targeting *IPd*. In this case, the *information* field includes the vulnerabilities of *IPd* that have been discovered. The *precond* field is empty because no prior access rights are needed to scan. If *A* is a successful attack, the *postcond* field includes the access rights the actor acquires.

Actions that fail do not return information or rights. Hence, the corresponding fields in the associated tuple are empty.

Each actor has a strategy that, in an intrusion, chooses the action to execute at a given step according to its current state, i.e., the access rights and the information it has previously acquired. An action *A* is *enabled* in a state if the actor owns the rights in the precondition of *A*. Regardless of this strategy, an actor can execute an enabled action only. As a consequence, a sequence of actions *S* is an intrusion of an actor *Ag* if it satisfies the following:

1) the first action in *S* belongs to the *attack surface* of *Ag*, i.e., it is enabled by the legal access rights of *Ag*,
2) the union of the legal access rights of *Ag* and all those returned by the actions in *S* preceding to *A* includes the precondition of *A*.

An intelligent actor repeats failed actions only when conditions change (e.g., new information is obtained, vulnerabilities are reassessed, or the action is attempted on a different target node).

An intrusion *S* is *successful* if, at the end of *S*, *Ag* owns the access rights of its goal. Otherwise, *S* fails. *A* is *useless* if *Ag* does not need the privileges or information *A* grants to achieve its goal. This happens if the postcondition of *A* do not intersect the precondition of any action in *S* after *A*. *Ag* executes a useless action because it has not collected enough information on the target.

Different intrusions can reach the same goal, and the number of alternatives influences the overall risk and shows the degree of freedom of *Ag* when building an intrusion.

III. Intrusion and Data Shifts

Effective risk management requires information on alternative intrusions, the vulnerabilities they exploit, and their success probabilities. This information cannot always be discovered using data collected on past intrusions against the same or similar infrastructures because of data shifts, namely because the underlying probability distributions change too quickly.

As an example, historical data for evaluating the success probability of intrusions by *Ag* quickly become obsolete if:

- *Ag* adopts new, previously unseen strategies.
- New vulnerabilities are disclosed, altering intrusion dynamics.
- Infrastructure nodes deploy new applications, changing the attack surface.
- Logical or physical connections are modified, affecting network paths.
- Nodes are added or removed, impacting overall risk assessments.

The reason for obsolescence is that these events may result in new intrusions and/or change the intrusion success probability. In other words, these events produce a data shift in the distribution of the data that affects the probabilities of interest in the training of an AI model [13], [14]. In more detail, the new behaviours of *Ag* change either the actions it selects or the sequence of actions in its intrusion. For example, *Ag* may increase the time it spends collecting information before selecting the vulnerability to exploit. This implies *Ag* has more accurate information available, and this reduces both useless attacks and attack failures. We can update *Ag*'s success probability after observing several intrusions, but we cannot ask *Ag* to repeat a successful intrusion to compute more accurate probability values. Furthermore, methods for online data shift detection, while are still applicable, may result in an unacceptable delay in an accurate estimate of success probabilities.

The discovery of a new vulnerability, the deployment of a new application, or a new connection may enable new attacks that, in turn, result in new intrusions. New intrusions cannot always be predicted using data on previous intrusions because novel vulnerabilities increase in a non-linear way the number of intrusions.

Any framework for discovering intrusions or their success probability should handle data shifts that affect the risk scenario. These shifts are very frequent and prevent the collection of a sufficient amount of data on real intrusions. Lack of information may result in a black swan or a perfect storm

disaster [15], [16], that is, intrusions that are unknown in advance or known but unexpected because their probability cannot be estimated with the required precision.

A practical alternative is to adopt synthetic data produced by adversary simulations using security twins and actor twins. By simulating the behaviour of intelligent actors and generating synthetic data through adversary simulations, we gain a comprehensive understanding of potential intrusions. The use of Monte Carlo methods that run multiple independent simulations can cover diverse attack scenarios and provide valuable insight into intrusions and their success probabilities. Security twins provide a virtual representation of ICT infrastructures, enabling detailed and dynamic simulations of potential intrusions. By timely updating security and actor twins, we can generate relevant and up-to-date synthetic data, which allow for covering data shifts. Such synthetic data can be used to train robust AI models without relying on historical data. This improves both the prediction of intrusions and the accuracy of their detection by leveraging advancements in intrusion detection systems and the effective use of datasets to model evolving threats [17]–[20].

IV. Forecasting Intrusion using Twins

This section presents first the security twin and the actor twin. Next, we discuss adversary simulations.

According to [21], a digital twin is

> a virtual model for a physical entity in the digital form to simulate entity behaviors, monitor the ongoing status, recognize internal and external complexities, detect abnormal patterns, reflect system performance, and predict future trends.

This definition stresses that both the security twin and the actor one are formal models of, respectively, the infrastructure and the actor. Both models abstract the entity they model and represent only some of their attributes. Their common goal is to accurately model the behaviors of, respectively, an actor and the infrastructure in an intrusion.

A. The Security Twin

A security twin is an enriched inventory of hardware and software infrastructure modules, it describes:

a) the infrastructure (hw) nodes and their connections,
b) the infrastructure (sw) modules with their operations,
c) the mapping of modules onto nodes and their configurations,
d) the accounts on each node, and the rights they own,
e) any vulnerability of each module or node, the attacks it enables, and the properties of these attacks,
f) routing and filtering rules,
g) the logical connections among modules the previous rules determine,
h) intrusion sensors, the subnet, or the endpoint each sensor monitors, and the probability it detects an intrusion,
i) hierarchical relations, i.e., the one between a hypervisor and the virtual machines it runs,
j) information flows among the modules, i.e., one from a web server to a database.

The twin includes information on the module configuration because alternative configurations result in distinct vulnerabilities. The operations of a module determine the access rights that an actor can acquire since a distinct right exists for each operation.

Pre- and post-conditions exist also for actions, and they determine how an actor strategy maps the current state of an actor into the actions and the attacks it can execute. The mapping is also constrained by the information on the system the actor has acquired. Hierarchical relations and information flows determine dependencies among modules. For example, access rights on a hypervisor result in the control of the virtual machines it manages. Access rights on the source of an information flow enable the manipulation of the values transmitted to the receiving module(s).

A security twin also describes (human) user classes and pairs each class with access rights and the probability that a user in the class is the victim of an attack implemented by stealing authentication information or by phishing.

Alternative security twins of the same infrastructure differ in the number of details on the modules, the granularity of operations and the correspoinding access rights. Further details concern the behavior of the components. For example, a twin that neglects intrusion sensors returns worst-case results because it cannot model intrusion detection. A more accurate twin can model the detection and the resulting failure of an intrusion. However, to minimize the simulation overhead, a security twin never describes in full detail the system, its inputs, and its computations.

B. The Actor Twin

This twin describes the strategy of the actor, its initial access rights and information, and its goal. The attack surface of an infrastructure for an actor includes any attack enabled by the access rights of the actor before starting an intrusion.

The strategy of an actor [7] maps the current state of the actor, its goal, and the target infrastructure to the action to execute. The status of an actor includes the access rights and the information that the actor has collected. If the strategy returns an action that is an attack, it also returns a target module on a target node and the vulnerability to exploit. Several alternative strategies are possible. For example, a strategy may prefer actions to collect information and select an attack only when it cannot collect further information. Instead, another strategy may execute an attack as soon as the actor owns the privileges in the attack precondition. This may speed up the escalation at the expense of useless attacks. Some strategies include social engineering attacks, while others only exploit vulnerabilities in the infrastructure modules.

The state of an actor also includes a memory that records the last actions and their result (success or failure). A small memory implies that the actor forgets its failures quickly and may repeat an action even after many failures.

C. Adversary Simulations

A simulation results from the coevolution of the security twin and the actor twin, starting from the actor's attack surface. A coevolution is a sequence of steps, in discrete time, executed with real-time semantics. Each step applies the actor strategy, executes the action it returns, and determines its success or failure using the corresponding success probability in the security twin. Each step updates the actor's status according to the success or failure of the action and incorporates the real-time dynamics of the system. Each step also considers the reaction of the infrastructure to the action that has been executed. For example, an intrusion sensor that detects an attack in real-time results in the immediate failure of the action and potentially of the intrusion.

A simulation may also update the security twin because some attacks change the infrastructure. The corresponding step updates some twin attributes to account for the change. For example, the actor can update routing or filtering rules, changing the logical topology in both the system and the security twin. A physical attack on a connection changes both the logical topology and the physical one.

A simulation is successful when the actor reaches a goal. It fails when no action can be selected, or a predefined number of steps have been executed without reaching the goal, or a predefined threshold on the simulated time has been reached, or the infrastructure detects an attack. The sequence of actions of a successful simulation leads to a successful intrusion, as defined in Sect. II.

D. Using Twins and AI for Intrusion Planning

The security and the actor twins support the exploration of alternative planning strategies the actor may apply to reach its goal, allowing defender teams to anticipate actor strategies.

A significant body of knowledge exists on classic AI planning [22], [23], which forms the foundation for simulating adversary strategies. Recent research has further explored the effectiveness of leveraging large language models (LLMs) to plan and execute complex tasks, including cybersecurity intrusions. For example, [24] demonstrates that LLM agents can exploit zero-day vulnerabilities, showcasing the potential of AI-driven simulations to uncover previously unknown attack vectors [25]–[28].

These AI-driven strategies, trained using the synthetic data produced by twin simulations, enable infrastructure defenders to anticipate high-risk scenarios posed by powerful adversaries. By running simulations where actors apply sophisticated AI-driven planning strategies, defenders can proactively deploy proper countermeasures, even against the most advanced and resourceful attackers.

V. Intrusion Data Analysis and AI

The simulation of intrusions can identify and mitigate critical vulnerabilities before attackers exploit them. This proactivity is fundamental to preserving resilence.

A successful intrusion returned by a simulation that is not possible in the target infrastructure is a *false positive*. A successful intrusion against the target infrastructure that is not discovered by any simulation is *false negative*. The probability that a set of independent simulations does not produce a false negative decreases as the number of simulations increases. Independence is crucial, as it ensures that the occurrence of an action in a simulation does not affect the actions of other simulations.

By the Law of Large Numbers, we estimate the probability of an event of interest in the ratio of the number of simulations where the event occurs over the total number of simulations. More refined approaches than frequency counting are described in [29]–[33].

Events of interest include:

1) The actor reaches a goal.
2) A module is attacked.
3) The actor executes a given intrusion.
4) An intrusion takes less than a specified time limit.

Events can also be considered under conditioning assumptions, e.g., if the actor passes through a specific node then it reaches a goal.

Low-probability, high-impact events require many simulations for robust estimation. They can be executed in parallel (due to the independence assumption) and without disturbing the operation of the target activity.

Intrusions are a fundamental input to devise countermeasures and improve the robustness and resiliency of the target infrastructure. The intrusions generated by simulations can be analysed using AI and Data Mining techniques, such as the frequent pattern approach of [33], to identify the most dangerous intrusion patterns and then to improve the infrastructure.

VI. Preliminary Experimental Results

This section describes how to implement and update a security twin and an adversary simulation framework. It focuses on creating and managing three databases: the security twin database, its vulnerability database, and the actor database. We also describe how simulations use these databases to generate synthetic intrusion data.

A. Security Twin

The security twin is built by a reverse engineering process that extracts information on the vulnerabilities of each module and the attacks they enable. This information populates a database [34] of the system inventory with information on nodes, connections, and configurations built using inventory management tools. Information about vulnerabilities can be obtained as the output of vulnerability scans.

B. Vulnerability Database

The vulnerability database describes vulnerabilities in infrastructure components, including preconditions, postconditions, execution times, and noise levels. It is built using data from sources such as the NVD and CVE databases [11], [12]. Such a database is also part of the security twin, but making it separate from the security twin database simplifies some tasks in the adversary simulations and enhances manageability. As

an example, the information on a vulnerability is stored once, even if it appears in several nodes.

C. Actor Database

This database stores the data about the twin of each actor with information on its strategy, TTPs and goal. Each twin drives the simulation of the corresponding actor. Data to build the twins are gathered from threat intelligence reports and security analysis.

D. Implementation

According to the framework, experiments run multiple independent simulations using the security twin, the vulnerability database, and some profiles from the actor database. These simulations produce data that describe the dynamic interactions between actors and the target infrastructure. Such data, including intrusions, are collected in other databases for further analysis.

Strategies and Intrusion Planning: The current prototype offers four actor strategies:

- **Random**: less sophisticated actors select the next action randomly.
- **Privilege First**: the actor speeds up the acquisition of access rights.
- **Success First**: the actor prefers actions with the highest likelihood of success.
- **Noise First**: the actor prefers actions that produce the lowest noise to avoid detection.

Future versions of the prototype will implement AI-enabled planning strategies and compare to the above ones.

Data Shift Analysis: Data shifts may occur for any change in the infrastructure or in the actor behavior. A further reason for an expected and positive shift is the patching of vulnerabilities chosen through some simulations. The impacts of all these shifts can be estimated through new simulations.

Simulation Outputs: The outputs of simulations to be used as the input for AI and Data Mining analyses include:

- **Attack Paths and Sequences**: Detailed sequences of actions in intrusions,
- **Success Rates**: Estimated intrusion success probabilities, True Positive Rate (Recall), True Negative Rate (Specificity),
- **Time Metrics**: Time to reach goals.
- **Impact Assessment**: Evaluations of attack impacts, the weighted average of the damages caused by individual intrusions based on a cost model. This model depends on the business processes using the infrastructure.

The outputs of multiple simulations support an estimation of the probabilities of the events of interest as well as the computation of centrality measures for the infrastructure nodes weighted w.r.t. successful intrusion probabilities. The latter is important information for scheduling patch deployment.

E. A First Example

Consider a simplified infrastructure where components include servers, workstations, network devices, and users. The table below illustrates a subset of this system:

Node	Type	Connections
Node A	Server	Node B, Node C
Node B	Workstation	Node A, Node D, Node E, User 1
Node C	Router	Node A, Node D, Node E
Node D	Workstation	Node B, Node C, Node F, User 2
Node E	DB Server	Node B, Node C, Node F
Node F	Firewall	Node D, Node E
User 1	User	Node B
User 2	User	Node D

The system architecture consists of 60 nodes and 15 users, featuring a diverse array of components including servers, workstations, a router, a firewall, and a database server.

We assume the infrastructure nodes are affected by a total of 50 vulnerabilities. Despite efforts to maintain security through regular patching, around 15 vulnerabilities remain unaddressed.

Our simulation software is implemented in Python, using NumPy for numerical computations and NetworkX for network analysis.

We have run simulations to analyze alternative scenarios to investigate the baseline attack paths, the selection of countermeasures, and the existence of new actors.

A first simulation, consisting of 100 runs, was conducted, resulting in the following initial scenario:

- **Common Attack Path**: Node A to Node B to Node E.
- **Success Probability**: 65% for a phishing attack targeting User 1.
- **Average Time to Compromise**: 2 hours.

After patching vulnerabilities and improving firewall rules, a second simulation of 100 runs was performed, showing a new scenario:

- **Altered Attack Path**: Node A to Node C to Node E.
- **Success Probability**: Reduced to 30%.
- **Average Time to Compromise**: Increased to 4 hours.

The deployment of countermeasures resulted in a data shift. The actor has now chosen a distinct path as a reaction to the closure of preferred paths due to the patching of some vulnerabilities. The ability to predict the reaction of an actor to a countermeasure is fundamental for risk management.

The simulation of an advanced strategy, specifically aimed at minimizing noise, results in the following scenario:

- **New Attack Path**: Node A to Node B to Node C.
- **Success Probability**: 75%.
- **Average Time to Compromise**: 1.51 hours.

This scenario demonstrates how a sophisticated attacker can adapt to existing security controls, modifying their paths to optimize success probability and minimize time to compromise.

To further examine the effect of emerging vulnerabilities, a final simulation introduced an additional vulnerability on node C, resulting in a significant change in the attack path:

- **Updated Attack Path**: Node A to Node C to Node E.
- **Success Probability**: 90%.
- **Average Time to Compromise**: 1.04 hour.

Adding a single vulnerability markedly increased the success probability and reduced the time to compromise, emphasizing how even minor changes in the vulnerability landscape heavily influence attack paths. This also confirms that rapid exploitation of new vulnerabilities [35] deeply changes the risk scenario and results in large data shifts.

VII. Conclusion

Critical information infrastructures face significant threats as actors exploit vulnerabilities in software, hardware, and human elements, often using physical or phishing attacks. Rapidly evolving risk scenarios and frequent data shifts reduce the reliability of historical data for intrusion prediction and AI training.

Merging digital twins with adversary simulation offers a solution by generating synthetic data to counter data shifts. This approach simulates vulnerabilities and attacker behaviors, enabling the creation of realistic intrusion scenarios. Integrating AI into digital twins enhances their ability to emulate intelligent threat actors and train advanced intrusion detection models.

Future work will refine these methods, expand applications across infrastructures, and leverage AI to optimize synthetic data, strengthening defenses against evolving cyber threats.

References

[1] S Furnell *et al.*, "Understanding the full cost of cyber security breaches," *Computer fraud & security*, 2020.

[2] M Bada *et al.*, "The social and psychological impact of cyberattacks," *Emerging Cyber Threats and Cognitive Vulnerabilities*, V Benson *et al.*, Eds., USA, 2020.

[3] A Dinicu *et al.*, "The multidimensional impact on society of cyber attacks targeting the energy critical infrastructure sector," *Land Forces Academy Review*, vol. 26, 2021.

[4] K Thakur *et al.*, "Impact of cyber-attacks on critical infrastructure," *BigDataSecurity/HPSC/IDS*, IEEE, 2016.

[5] D Amodei *et al.*, "Concrete problems in AI safety," *arXiv preprint arXiv:1606.06565*, 2016.

[6] C McLean *et al.*, "Modeling and simulation of critical infrastructure systems for homeland security applications," *US Nat. Inst. Standard Technol., Gaithersburg, MD, USA, Tech. Rep. NISTIR*, vol. 7785, 2011.

[7] A Applebaum *et al.*, "Attack flows — beyond atomic behaviors," *MITRE Engenuity*, 2022.

[8] F Baiardi *et al.*, "Twin based continuous patching to minimize cyber risk," *European Journal for Security Research*, vol. 6, 2021.

[9] M Ryan, *Ransomware Revolution: The Rise of a Prodigious Cyber Threat*. Berlin: Springer Cham, 2021.

[10] B Strom *et al.*, *Mitre att&ck™: Design and philosophy*.

[11] The MITRE Corporation, *Cve*.

[12] NIST, *National vulnerability database*.

[13] J Quinonero-Candela *et al.*, *Dataset Shift in Machine Learning*. USA: MIT Press, 2009.

[14] J Moreno-Torres *et al.*, "A unifying view on dataset shift in classification," *Pattern Recognition*, vol. 45, 2012.

[15] T Aven, "On the meaning of a black swan in a risk context," *Safety science*, vol. 57, 2013.

[16] E Paté-Cornell, "On "black swans" and "perfect storms": Risk analysis and management when statistics are not enough," *Risk Analysis: An International Journal*, vol. 32, 2012.

[17] Y Mirsky *et al.*, "Kitsune: An ensemble of autoencoders for online network intrusion detection," *arXiv*, 2018.

[18] F Shahid *et al.*, "Cyber threat intelligence using deep learning: An overview," *IEEE Access*, vol. 8, 2020.

[19] A Subbaswamy *et al.*, "From development to deployment: Dataset shift, causality, and shift-stable models in health ai," *Biostatistics*, vol. 21, 2020.

[20] A Storkey, "When training and test sets are different: Characterizing learning transfer, dataset shift," *Machine Learning*, 2009.

[21] Q Qi *et al.*, "Digital twin and big data towards smart manufacturing and industry 4.0: 360-degree comparison," *IEEE Access*, vol. 6, 2018.

[22] Z Zhao *et al.*, "Large language models as commonsense knowledge for large-scale task planning," *ArXiv*, vol. abs/2305.14078, 2023.

[23] Z Wu *et al.*, "Embodied task planning with large language models," *ArXiv*, vol. abs/2307.01848, 2023.

[24] R Fang *et al.*, "Teams of llm agents can exploit zero-day vulnerabilities," *arXiv preprint arXiv:2406.01637*,

[25] M Ghallab *et al.*, *Automated Planning: Theory and Practice*. Elsevier, 2004.

[26] J Portillo *et al.*, "Digital twins in cyber-physical systems: Concepts, applications, and research challenges," *IEEE Access*, vol. 8, 2020.

[27] S Bayer *et al.*, "Cyber threat intelligence using machine learning: A systematic review," *IEEE Transactions on Information Forensics and Security*, vol. 16, 2021.

[28] A Faruque *et al.*, "Defensive techniques using ai for real-time detection and response to cyber threats," 2019.

[29] DW Scott, *Multivariate density estimation: Theory, practice, and visualization*. 2015.

[30] A Gelman *et al.*, *Data Analysis Using Regression and Multilevel/Hierarchical Models*. 2006.

[31] A Esuli *et al.*, *Learning to Quantify*. Springer, 2023.

[32] JA Miller *et al.*, "A survey of deep learning and foundation models for time series forecasting," *CoRR*, vol. abs/2401.13912, 2024.

[33] M D'Andreagiovanni *et al.*, "Sequential pattern mining for ICT risk assessment and management," *J. Log. Algebraic Methods Program.*, vol. 102, 2019.

[34] S Pore *et al.*, "A comparative study of SQL and NoSQL databases with MongoDB," *IJARCET*, vol. 6, 2017.

[35] C Charrier *et al.*, *How low can you go? an analysis of 2023 time-to-exploit trends*, 2024.