CISSAN

Collective intelligence supported by security aware nodes

D1.1 Detection and analysis of weak signals

Editors: Kurt Tutschku (BTH), Jianguo Ding (BTH), Alexey Kirichenko (University of Jyväskylä)
contact: kurt.tutschku@bth.se

Abstract

Deliverable D1.1 discusses the state-of-the-art (SotA) for selected technology areas of the CISSAN project. Its objective is to outline, structure, and summarize potentially relevant approaches and techniques for improving cybersecurity of IoT networks, in particular, through enabling collective intelligence (CI) of security aware network nodes. Other key objectives of D1.1 are to identify research and technology gaps within the project and to help in prioritizing project efforts.

Key areas considered in D1.1 include CI approaches for IoT network security, Generative Adversarial Networks (GANs) and their applications to producing synthetic data for improving IoT network security, blockchain-based approaches for data integrity protection and traceability, distributed log files management, use of AI for attack detection in IoT networks, use of AI for malicious purposes, honeypots for cyberattacks targeting IoT devices and networks.

Participants in project CISSAN are:

- University of Jyväskylä
- Bittium Wireless Ltd
- Bittium Biosignals Ltd
- Geodata ZT GmbH
- Mattersoft
- Mint Security Ltd
- Netox Ltd
- Nodeon Ltd
- Scopesensor Ltd
- Wirepas Ltd
- Councilbox Ltd
- Affärsverken Karlskrona AB
- Arctos Labs
- Clavister AB
- Blekinge Tekniska Högskolan
- Blue Science Park
- Savantic AB
- Techinova AB

CISSAN – Collective intelligence supported by security aware nodes

D1.1 Detection and analysis of weak signals

Editors: Kurt Tutschku and Jianguo Ding (Blekinge Tekniska Högskola), Alexey Kirichenko (University of Jyväskylä)

Project coordinator: Alexey Kirichenko (University of Jyväskylä)

CELTIC published project result

Executive Summary

Deliverable D1.1 presents the state-of-the-art (SotA) for selected technology areas of the CISSAN project. Its objective is to outline, structure, and summarize potentially relevant approaches and techniques for improving cybersecurity of IoT networks, in particular, through enabling collective intelligence (CI) of security aware network nodes. Other key objectives of D1.1 are to identify research and technology gaps within the project and to help in prioritizing project efforts.

Key areas considered in D1.1 include CI approaches for IoT network security, Generative Adversarial Networks (GANs) and their applications to producing synthetic data for improving IoT network security, blockchain-based approaches for data integrity protection and traceability, distributed log files management, use of generative AI for attack detection in IoT networks, several topics around threat intelligence, communication, information sharing, trust management and authentication schemes, and use of AI for malicious purposes. D1.1 maps the reviewed essential research findings in those areas to the project needs and the initial CISSAN architectural choices, as considered by the project partners, helping support the project plan relevance by reflecting and adapting to such findings dynamically.

In the section on CI methods for IoT network security (Section 3.1), several concepts and paradigms are reviewed: Artificial Intelligence (including hierarchical and federated learning), Crowdsourcing, Multi-Agent Systems, and Self-Organizing Systems. Potential challenges and risks of applying corresponding techniques to IoT cybersecurity and their diversity in the level of required central control are considered.

The need of high-quality data for the development of Intrusion Detection Systems (IDS) is articulated in Section 3.2 and leveraging Generative Adversarial Networks (GANs) for addressing this need through generating synthetic data, primarily for training and testing detection models and algorithms, is introduced and discussed. The potential of GANs in enhancing the efficiency of IDS is evaluated, including key requirements of reflecting the correlation and distribution of real-world data and providing interpretability in data feature representation.

A use case / application driven analysis of blockchain techniques is presented in Section 3.3. Starting with a general introduction, common blockchain types, identity models, and the trilemma of blockchain technology (in achieving security, decentralization, and scalability) are reviewed, followed by the analysis of blockchain technology limitations and its use in providing evidence and traceability. Potential risks for CISSAN partners in proposing blockchain-based solutions and potential solutions to the blockchain technology trilemma are considered. The section is concluded by presenting popular blockchain platforms and key suppliers of those as part of the state-of-the-practice and by mapping key requirements to most significant technologies to investigate in CISSAN.

Section 3.4 presents several secure log file management algorithms and techniques for IoT networks, including the DistLog system, probabilistic logging, Transparent IoT (T-IoT) framework, the LogSafe System, the DASLog System, Optimal Data Witnessing, the LogStack System, and the Integrity Protocol. These algorithms and techniques are considered for potential application in CISSAN, especially for supporting CI by security information exchange among network nodes.

In Section 3.5, we review recent publications discussing generative AI applications to intrusion detection and threat hunting in IoT networks and in other potentially relevant contexts and cybersecurity tasks. Since this is a truly vibrant research domain at the time of writing, our goal in this section was to briefly introduce the current capabilities of quickly evolving generative AI-based models – primarily Large Language Models (LLMs) and GANs – and certain research directions that can influence CISSAN plans.

Section 3.6 discusses several topics that CISSAN partners consider possibly relevant for specific use cases: use of LLMs for building and operating honeypots as a way of collecting threat intelligence (unlike Section 3.5, the papers reviewed here have no focus on IoT); communication and information sharing between resource constrained network nodes; peer-to-peer trust management and authentication schemes in intelligent transportation systems and ad-hoc networks.

Finally, Section 3.7 reviews the use of AI for enabling and facilitating cyberattacks (malicious use of AI). While numerous techniques were proposed in this domain, the real-world evidence seems to

indicate that the greatest concerns in the short to medium term are perhaps the use of AI in social engineering and reconnaissance. Countering social engineering is not part of CISSAN's agenda, but we will analyse whether AI-assisted reconnaissance should be prioritized by the project. Two other potentially interesting research directions to consider in the project are (i) attackers targeting our ML models (exploiting AI techniques) and (ii) attackers building and operating AI-powered IoT botnets.

D1.1 is produced in the first stage of the project, and D1.2 will present an update, collected, analysed, and taken into use throughout the project timeframe. Enabling techniques, methods, and algorithms for collective intelligence in IoT networks will be considered in greater detail, such as collective intelligence communication protocols, security task delegation (run-time) and security functionality distribution (design-time) algorithms, node trust management, aggregation of AI decisions. Other areas of attention will likely include explanations of detected anomalies, support of response actions, and data quality verification methods.

List of Authors (in alphabetical order)

- Dure Adan Ammara, BTH, Sweden.
- Niko Candelin, Netox, Finland.
- Jianguo Ding, BTH, Sweden.
- Alberto Doval, Councilbox, Spain.
- Alexey Kirichenko, University of Jyväskylä, Finland.
- Mikko Lehkonen, University of Jyväskylä, Finland.
- Veikko Markkanen, University of Jyväskylä, Finland.
- Rodrigo Martinez, Councilbox, Spain.
- Sara-Päivi Paukkeri, University of Jyväskylä, Finland.
- Martin Rubio, Councilbox, Spain.
- Ilgin Safak, University of Jyväskylä, Finland.
- Kurt Tutschku, BTH, Sweden.

Table of Contents

List of Figures

List of Tables

1 Introduction

CISSAN WP1 focuses on the current state of technology – including concepts, approaches, and experimental prototypes – for the CISSAN project. This work package collects, summarizes, organizes, and clarifies technologies relevant for the project. The title of the two WP1 deliverables (D1.1 and D1.2), *Detection and analysis of weak signals*, should be understood broadly as mapping "essential research findings" to the project needs and supporting the project plan relevance by reflecting and adapting to such findings dynamically, possibly when their potential impact and value for CISSAN are not obvious yet. The title is also meant to refer to a specific challenge, important in the project scope, of dealing with weak local observations that might not be sufficient to reliably detect and understand complex cyberattacks. One way of addressing this challenge is through the use of intelligent and collaborative detection techniques, such as AI-based methods, blockchain-based technologies, and collective decision-making, which are key research and innovation directions in CISSAN.

D1.1 is produced in the first stage of the project, and D1.2 will present an update, collected, analysed, and taken into use throughout the project timeframe.

Objective of this Document

The CISSAN project leverages multiple paradigms and approaches to address security challenges and threats that IoT and OT systems and networks face, such as collective intelligence (CI), artificial intelligence (AI), and distributed ledger technologies (DLT). CISSAN will identify and analyse security challenges and threats in the three project use cases and more generally in the IoT and OT domains and will propose a conceptual CISSAN-powered platform and framework that integrate security solutions and technologies to prevent and mitigate cyberattacks.

Furthermore, CISSAN is a highly collaborative project and encompasses diverse use cases in multiple domains (public transportation, smart grids, tunnelling and construction). This approach enables its technical solutions to be reuseable and applicable broadly, but also leads to a high diversity of approaches in research and technology development employed by the project partners. Hence, it is important to identify main areas which are currently considered by the CISSAN partners, matching them with the capabilities of state-of-the-art solutions which can be enhanced and adapted to achieve cutting-edge security for IoT and OT systems. The primary objective of this document is, thus, to describe and to structure the current state-of-the-art as considered by the project partners.

The Structure of this Document

This deliverable is structured as follows: Section 2 gives an overview of the CISSAN project and its initial architectural plans and considerations; we also structure the related research and innovation areas. Section 3 presents in detail the state-of-the-art contributed by the project partners. Section 4 summarizes the document and gives an outlook for further research needs and plans.

2 Overview of the CISSAN Project

The CISSAN project aims at enhancing the cybersecurity, cyber resilience, and automation of Internet of Things (IoT) and Operational Technology (OT) ecosystems that utilize device, edge, and cloud computing capabilities (thus, including IT elements). A key project objective is to apply collaborative and intelligent mechanisms to enhance and aggregate local security awareness, which is especially important when local views are not sufficient for detecting and countering attacks. Hence, collaborative approaches, architectures, and algorithms for gathering and analysing locally available security information are required, leading to collective intelligence.

CISSAN's key method for researching and developing new cybersecurity solutions is based on an agile and DevOps inspired process that combines state-of-the-art technologies with the requirements and experiences from the use cases.

State-of-the-art techniques will be analysed whether they fit in the use cases and the CISSAN-powered network architectures. Based on this analysis, methods and techniques will be enhanced, refined, and combined. Each of the use cases can implement multiple iterations of this selection, analysis, refinement, and integration cycle.

The CISSAN project involves partners from Finland, Sweden, Spain, and Austria and focuses on three main use cases:

- smart public transportation (UC1; mainly located in Finland),
- smart energy grids (UC2; mainly located in Sweden),
- mining and tunnelling (UC3; mainly located in Austria).

The project leverages multiple paradigms and approaches to address security challenges and threats that IoT and OT systems face, such as collective intelligence, artificial intelligence (AI), and Blockchain. CISSAN will identify and analyse security challenges and threats in the three project use cases and more generally in the IoT and OT domains and will propose a conceptual CISSAN-powered platform and framework that integrate security solutions and technologies to prevent and mitigate cyberattacks.

CISSAN results will enable diverse applications for stakeholders, largely dependent on specific use cases. The utility of these results is influenced by factors such as the maturity of the target IoT or OT system/network, the application domain, prioritization of threats and risks, and technology choices. Consequently, the initial architecture is inherently flexible and structured around layered frameworks, core processes, functions, and key elements.

2.1 The Initial CISSAN Architecture

The initial CISSAN architecture is presented in deliverable D2.1, cf. [1]. In Figure 1, we see the main IoT / OT environment layers:

1. Perception / physical layer
2. Data layer
3. Network layer
4. Transport layer
5. Processing layer
6. Application layer

and security functions, components, and protocols that CISSAN envisions for improving cyber resilience of IoT / OT environments.

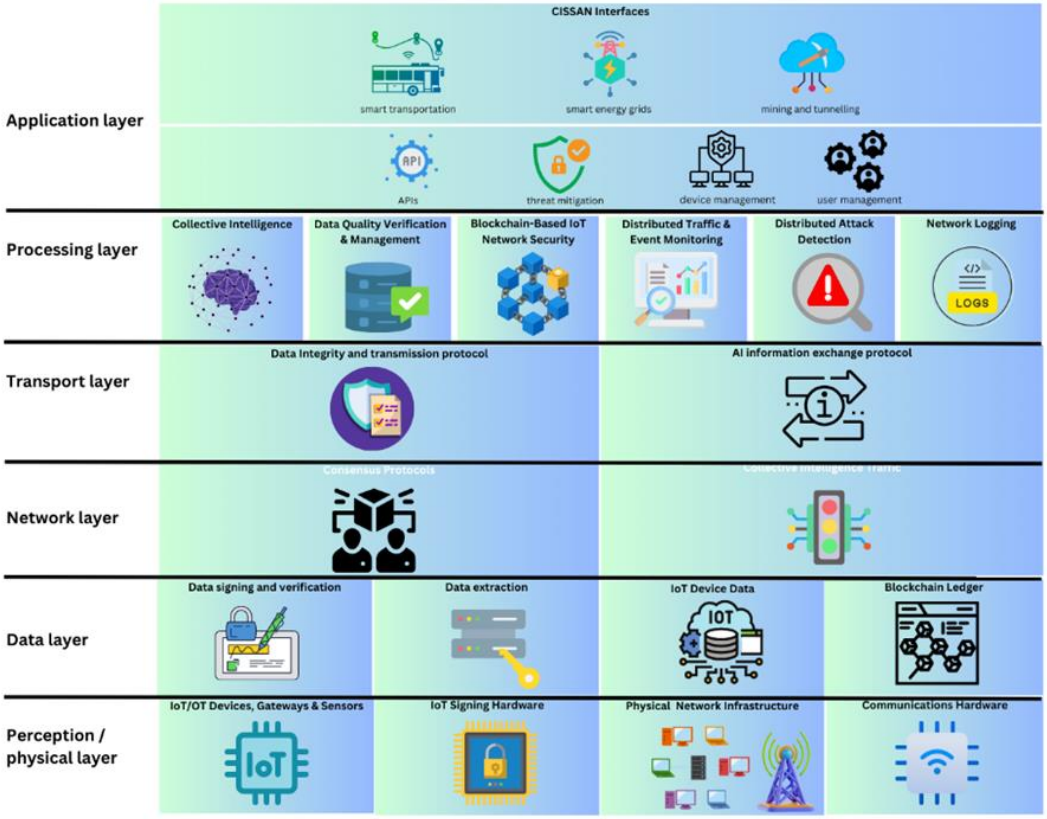

Figure 1. Layered structure of CISSAN framework

To guide the design and implementation of security solutions, the CISSAN framework should help elaborate a security model for IoT / OT environments, with four main cybersecurity functions: Detection, Response, Protection, and Intelligence. Each function is composed of several sub-functions that comprise specific security activities and objectives.

- Detection: The function of detecting cyberattacks in IoT and OT environments, using various methods and techniques, such as anomaly detection and signature-based (or rule-based) detection through network traffic analysis, device behaviour analysis, and user behaviour analysis. The sub-functions of detection are:
    - Meta Data: The process of extracting relevant information, e.g., from network traffic.
    - Labelling: The process of assigning labels to observations, e.g., network traffic, such as normal, suspicious, malicious, or unknown (based on the analysis of relevant metadata).
    - Source / Impact: The process of identifying the source and the impact of observations, e.g., network traffic, such as the device, the service, the vulnerability, etc.
    - Settings Management: The process of managing the settings and parameters of a detection function, such as thresholds, rules, policies, and alerts.
    - Reporting: The process of reporting the results and findings of a detection function, such as the metadata or labels.
- Response: The function of responding to cyberattacks and anomalies in IoT / OT environments, such as automated actions, manual actions, or collective actions. The response sub-functions are:

  - Self / Collective Awareness: The process of keeping aware of the current state of an IoT / OT environment and sharing this information with other systems and stakeholders, such as cloud backends, the protection function, or the intelligence function.
  - Automated Response: The process of executing appropriate actions to mitigate or prevent cyberattacks.
  - Reducing Attack Surface: The process of reducing the exposure and the risk of an IoT / OT environment, such as disabling or removing unnecessary or unused devices, services, etc.
  - Deny / Restrict: The process of denying or restricting the access or the communication of devices or services.
  - Configuration: The process of configuring and tuning settings and parameters for the response function.
- Protection: The function of protecting an IoT / OT environment from cyberattacks, using various methods and techniques, such as device security, network security, or cloud security. The sub-functions of protection are:
  - Identify posture improvement: The process of identifying and assessing the current security posture of an IoT / OT environment and suggesting improvements and recommendations to enhance the security level and performance.
  - Initiate change: The process of initiating and implementing changes and improvements to an IoT / OT environment, such as installing or upgrading devices or services, or applying patches or updates.
  - Implement protection: The process of implementing and enforcing protection measures and mechanisms for an IoT / OT environment.
  - Enterprise Posture Management: The process of managing and monitoring the security posture of an IoT / OT environment and reporting the status and the results to relevant stakeholders and systems, such as users, cloud backends, or the intelligence function.
- Intelligence: The function of providing and consuming intelligence (information and insights) for an IoT / OT environment, using various methods and techniques, such as threat intelligence, vulnerability intelligence, or collective intelligence. The intelligence sub-functions are:
  - Internal / External Threat Intelligence: The process of collecting, analyzing, and sharing threat information and indicators from internal or external sources.
  - Vulnerability Management: The process of identifying, assessing, and addressing security weaknesses in systems and software, collecting and analyzing vulnerability data from internal and external sources.
  - Collective Intelligence: The process of collecting, analyzing, and sharing intelligence (information and insights) from multiple sources and domains, including IoT, OT, IT, and cloud environments.
  - Protection Engineering: The process of applying intelligence (information and insights) to the protection function.

Figure *2* shows a typical mapping between security functions and core elements of IoT / OT systems and environments, which forms core elements of the initial CISSAN architecture.

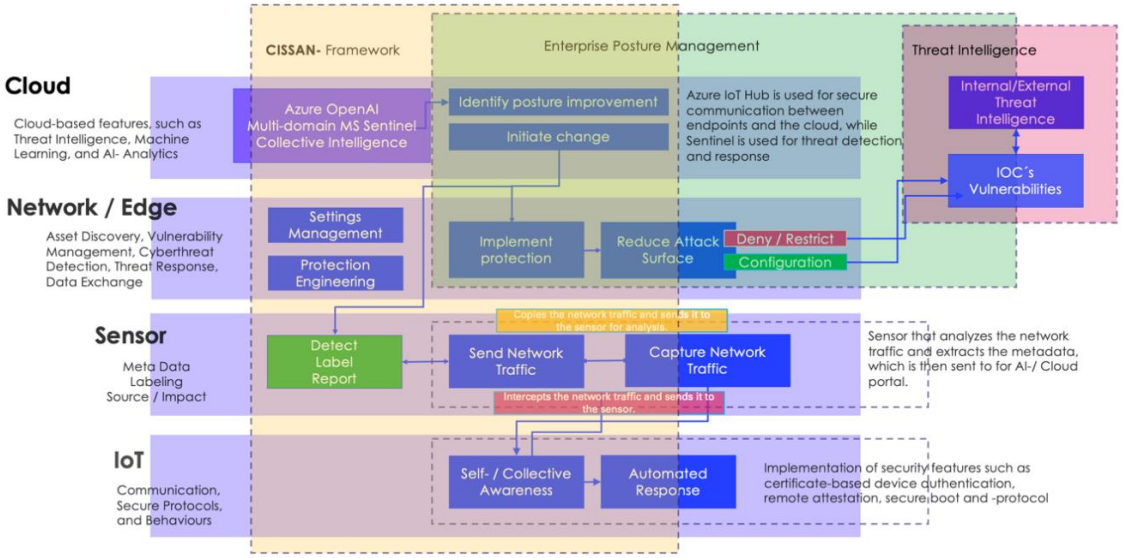

Figure 2. Key cybersecurity control points and functions at device, sensor, network/edge, and cloud levels.

References

[1] Niko Candelin, Ilgin Safak (eds.): Definition of the initial CISSAN architecture and distributed system elements and interfaces. CISSAN deliverable D2.1.

2.2 Related Research and Technological Innovation Areas

CISSAN is a highly collaborative endeavour that incorporates diverse use cases. This approach ensures that CISSAN technical solutions are reusable and broadly applicable while also drawing on multiple research concepts and approaches from the project partners. It is important to identify main research and innovation areas currently under consideration by the partners, aligning them with state-of-the-art solutions which can be enhanced and adapted to achieve cutting-edge security for IoT and OT systems. Furthermore, the use case-driven approach facilitates rapid progress towards relevant innovations, resulting in tangible outcomes. These results should include specific security techniques that can be quickly turned or integrated into marketable products and services.

Next, we attempt to outline the related (to the project developments) research and innovation areas and the foreseen innovative contributions by the CISSAN project.

Related Research and Innovation Areas

The research and innovation areas related to the CISSAN project are structured along the six layers presented above: perception, data, network, transport, processing and analysis, and application. As the project is work-in-progress, the related areas list will likely keep changing. Therefore, we focus in D1.1 on the current research areas and contributions from the consortium (and an updated list will be presented in D1.2):

- Data
  - Blockchains for data integrity protection (see Section 3.3)
  - Log file management (see Section 3.4)
- Network
  - Blockchain consensus mechanisms (see Section 3.3)
  - CI traffic mechanisms (see Section 3.6)

- Transport
  - CI exchange protocols (see Section 3.6)
- Processing and analysis
  - Collaborative / CI analysis and decision-making (see Section 3.1)
  - Generative Adversarial Networks (see Section 3.2)
  - AI for attack and potentially harmful actions detection (see Sections 3.1.1 and 3.5)
- Application
  - Security for the Use Cases (see Section 3.6)

In addition to these areas, D1.1 includes a section on the use of AI by cyber attackers (Section 3.7). It is in the CISSAN plan to follow the threat landscape and see whether project solutions should address threats and challenges specific to AI-powered attacks.

Furthermore, materials presenting state-of-the-art related to CI in IoT networks are included in other CISSAN deliverables: key architectural components for enabling CI in IoT networks can be found in Section 3 of D2.1 and architectural issues and threats in CI-enabled IoT networks and potential solutions to those can be found in D2.2.

Foreseen Innovative Contributions

Detailing innovative contributions by the CISSAN project is difficult at this stage. However, the foreseen innovations can be derived from the requirements of the use cases and classified by them:

- Smart public transportation (UC1; mainly located in Finland)
  - Innovation: Improved cybersecurity for traffic control systems
  - Innovation: GPS abuse detection in traffic control systems
- Smart energy grids (UC2; mainly located in Sweden)
  - Innovation: Local AI-based perception and anomaly detection for security aware smart grid elements, aggregation of local observations into global views
  - Innovation: Higher autonomy (through CI) for security in smart grids
- Mining and tunnelling (UC3; mainly located in Austria)
  - Innovation: Data quality verification for IoT-based construction monitoring systems
  - Innovation: Data integrity protection for IoT-based construction monitoring systems

3 Selected Related Research and Challenge Areas

3.1 CI Methods for IoT Network Security

This section describes the CI methods employed for IoT network security, with AI, crowdsourcing, multi-agent systems (MAS), and self-organizing systems (SOS) discussed.

3.1.1 Artificial Intelligence

CI can utilize AI to process extensive data produced by aggregated inputs, which provide insight into threats, to make or support informed decisions using predictive analytics, anomaly detection, pattern recognition, clustering, natural language processing, and other approaches. Mohamudally [1] provides a comparison of mathematical models for CI and a discussion of their suitability for implementation on mobile devices. He also proposes a framework for modeling CI systems using graph theory and artificial neural networks.

Hierarchical ML is a sophisticated methodology that arranges data and learning processes into stratified structures, mirroring the intrinsic hierarchical characteristics of numerous real-world issues. This methodology employs various degrees of abstraction, with each layer analyzing data at distinct granularity, hence improving the model's capacity to identify intricate patterns and relationships within the data. Hierarchical models frequently integrate unsupervised and supervised learning methodologies, facilitating enhanced accuracy and interpretability of outcomes. This method is very efficient in extensive data contexts, such as cloud computing, where it adeptly manages substantial data volumes while minimizing computational expenses and enhancing scalability. Moreover, hierarchical ML is significantly pertinent to CI, since it reflects the operational dynamics of collective systems through the utilization of several levels of abstraction and collaboration. Organizing data processing and learning activities into layered frameworks enhances cooperation and information sharing among agents, hence fostering CI and improved problem-solving abilities.

Verkerken et al. [2] proposes a scalable intrusion detection system (IDS) aimed at tackling the issues arising from the growing digitization and the expansion of linked devices. The proposed multi-tiered, hierarchical Intrusion Detection System (IDS) is validated utilizing public benchmark datasets and exhibits strong zero-day attack detection capabilities with superior classification performance. The system is adaptive without the need for retraining, minimizes bandwidth and computing demands, and upholds privacy regulations. Wu et al. [3] presents an innovative intrusion detection technique utilizing distributed computing to effectively manage extensive network data. The methodology integrates unsupervised and supervised learning approaches to improve detection accuracy and explainability. The system is engineered for cloud computing environments, providing scalability and diminished computation time. Experimental findings illustrate the method's efficacy in recording network traffic patterns and identifying many types of network intrusions with superior performance.

Federated learning (FL) is a methodology that enables the training of a machine learning (ML) model across several devices and/or servers, hence eliminating the need for data centralization. FL includes the following steps (see Figure 3):

1. A global model is established and sent to participant nodes in the network.
2. Each node autonomously trains and updates the model with its local dataset.
3. Nodes only transmit changes to the model, such as weights or gradients, to an aggregator, rather than sending their local data.
4. The aggregator enhances the global model by consolidating updates from all participant nodes by using various aggregation methods to enhance the learning process.
5. The revised global model is sent to participant nodes for further training or deployment.

This process continues across several iterations. The model is incrementally improved by using a broader data set with each iteration.

Figure 3. Federated Learning (source: *Sony AI*)

The FL paradigm harnesses the CI of distributed devices to facilitate collaborative model training. It leverages decentralized computation to improve network resilience against evolving threats [4], [5]. The authors in [4] have evaluated FL for intrusion detection in IoT networks using a shallow artificial neural network (ANN) as a shared model. The authors in [6] present an intrusion detection framework for 5G IoT based on federated transfer learning. federated transfer learning is a type of FL that entails using pre-trained models and refining them on decentralized data for certain purposes. This method is especially advantageous when local data is scarce or when the data distribution across devices is significantly different. It allows data aggregation with FL, the development of customized detection models by transfer learning, and information sharing between all IoT. The authors in [7] propose a collaborative Distributed Denial of Service (DDoS) detection and classification approach using FL for distributed multi-tenant IoT environments. The authors in [8] propose an instance-based transfer learning at the local level for training local models with non-independent and identically distributed (non-IID) data and introduce a rank aggregation algorithm with a weighted voting approach. A FL-based intrusion detection methodology proposed in [9] leverages multiple views of IoT network data in a decentralized format and uses multi-view ensemble learning for the detection, classification, and defense against attacks. Hei et al. [10] present a federated learning framework with a trusted feature aggregator to enhance the detection of distributed malicious attacks in IoT environments, leveraging CI by enabling multiple devices to collaboratively improve the detection model while maintaining data privacy.

FL can be categorized into three main types [11]:

1. Centralized FL: A central server orchestrates the training process in this manner. Local devices (clients) train models using their data and transmit model changes (e.g., weights and gradients) to the central server. The server consolidates these updates to create a global model, which is subsequently transmitted back to the clients for additional training. This approach improves privacy as raw data remains on local devices. Centralized FL can be categorized as follows:
    - Horizontal FL (HFL): In HFL, data is segmented by samples, indicating that several clients possess datasets with identical feature spaces but distinct sample spaces. This is beneficial when various organizations or devices gather analogous data kinds from distinct people. For instance, hospitals possess patient data, with each institution maintaining records for distinct patients yet utilizing same sorts of medical documentation.
    - Vertical FL (VFL) entails data segmented by features, wherein many clients possess datasets with an identical sample space but distinct feature spaces. This is relevant when various organizations possess complementary information regarding the same group of users. For example, a bank and an insurance business may partner, with the bank possessing financial data and the insurance company holding health data on the same individuals.

- Federated Transfer Learning (FTL) integrates FL with transfer learning to address situations where clients possess distinct feature and sample spaces. This approach is especially beneficial when there is minimal data overlap among clients, facilitating knowledge transfer across domains to enhance model performance.

2. Decentralized FL: In contrast to the centralized technique, decentralized FL operates without a central server. Clients engage in direct communication to exchange model updates. This peer-to-peer methodology can enhance resilience and mitigate the risk of a singular point of failure. Nevertheless, it may provide issues regarding the synchronization and consistency of the global model. A type of decentralized FL is *peer-to-peer FL*, in which clients interact directly among themselves, eliminating the dependence on a central server. Every client disseminates model updates to its counterparts, and the updates are consolidated in a decentralized fashion. This approach improves resilience and mitigates the risk of a single point of failure; yet, it may pose difficulties in preserving synchronization and consistency throughout the network.
3. Heterogeneous FL (HeteroFL): HeteroFL tackles the challenge of heterogeneity in FL settings, when clients exhibit varying compute capacities, data distributions, and network conditions. This approach facilitates the training of models capable of adapting to varied settings, so ensuring that all clients can effectively contribute to the global model. HeteroFL can enhance the overall efficacy and equity of the FL system and can be categorized based on device or data heterogeneity:
    - Device Heterogeneity: Examines the variations in computational capability and resources among clients. It guarantees that clients with differing capacities for processing power, memory, and battery life can nevertheless engage effectively in the federated learning process. Methods like model compression and adaptive training can be employed to address these disparities.
    - Data Heterogeneity: Data heterogeneity pertains to the discrepancies in data distributions and types among various clients. This is prevalent in real-world situations when data gathered by various devices or organizations may not exhibit identical distribution. HeteroFL methodologies seek to develop resilient models capable of effective generalization despite variances, frequently employing strategies such as personalized models or domain adaptation.

The CS4E (CyberSec4Europe) project [12] focused on improving cybersecurity throughout the EU, addresses critical sectors such as IoT, finance, and healthcare. This project includes an FL-based implicit collective threat intelligence (CTI) sharing Open Banking architecture for real-time fraud detection. Using a permissioned blockchain, data related to the CTI sharing event is securely and traceably stored. The incident response team sends real-time training data to multiple FL clients. They use a distributed identity management system for user authentication and authorization. Cyber Sandbox Creator was also developed as an open-source tool for cybersecurity testing in virtual environments and a governance model for facilitating stakeholder reporting of cybersecurity issues to improve collective defense mechanisms [13].

However, extensive use of AI for CI presents considerable risks, such as model poisoning and model evasion in FL. Model poisoning occurs when a malevolent individual intentionally introduces erroneous data into the training process of a ML model, leading to lower performance, incorrect learning by the model and predictions yielding erroneous or unfavorable outcomes [14]. Model evasion refers to the intentional alteration of input data by a malicious actor to deceive an ML model, resulting in misclassification by the model. This strategy is often used to obscure cyberthreats that would typically be identified by the system. Implementing multi-faceted solutions, including data cleaning, regular model retraining, continuous performance monitoring, differential privacy measures, and utilizing reliable data sources, may effectively prevent or minimize model evasion and model poisoning attacks. These models' resilience against such attacks may also be enhanced by training them employing challenging examples and ensemble techniques [15].

3.1.2 Crowdsourcing

Crowdsourcing, which can be used as a CI method, refers to combining the concepts of "crowd" and "outsourcing" to solicit frequent help from multiple sources through online platforms, to gather information, feedback, or labor [16]. Crowdsourcing creates synergy by gathering information and ideas using the contributions from large, geographically dispersed sources. Crowdsourcing systems typically have two main stakeholders:

1. *Crowdsourcing providers* optimize the quality of work while minimizing latency, often within a limited budget.
2. *Employees,* crowdsourcing institution's staff, enter data into the system using several platforms based on some given quality control criteria. Employees, who exhibit variability in their productivity, availability, and skill level, try to optimize the incentives they receive for completing assignments.

The efficacy of crowdsourcing may be enhanced by AI-driven threat detection that systems can identify threats more rapidly and accurately due to their capacity for extensive data processing. Protection against threats may be improved by augmenting the rate and sensitivity of threat detection. Crowdsourcing is efficient for identifying and rectifying cybersecurity vulnerabilities, bridging the detection gap, and reducing the dependence on a single cybersecurity specialist [17]. The procedures used for this objective are as follows:

- *Data Collection and Analysis*: Crowdsourcing and information acquired from a substantial cohort of users or specialists are crucial for comprehending the variety and intricacy of cyber threats. AI can be used to analyze data and develop threat models accordingly.
- *Threat Detection and Identification*: Crowdsourcing data may help AI-based systems discover threat trends and abnormalities faster.
- *Rapid Feedback and Update*: Crowdsourcing finds cybersecurity flaws quickly. This data may be used to enhance AI systems and threat response techniques.
- *Diversity of Technical Experience*: Crowdsourcing may assist discovering gaps in cybersecurity professionals' knowledge by pooling their viewpoints and experience.
- *Continuous Learning and Improvement*: Crowdsourcing data trains and strengthens AI models, hence improving AI-based threat detection systems.

Vlachos et al. [18] introduce a crowdsourcing platform that leverages CI by aggregating data from multiple users to evaluate real-time privacy threats, storing the results anonymously for further processing and visualization. Sun et al. [19] present a methodology for enhancing cybersecurity intelligence by collecting and analyzing data from both authoritative cybersecurity databases and social networks to monitor vulnerabilities, threats, and security trends, ultimately providing informed security recommendations. Christoforidis et al. [20] introduce a collaborative client application that uses crowdsourcing to enhance malware detection and mitigation, leveraging CI for real-time threat response and community involvement in cybersecurity. Moradi and Li [21] identify four typologies of adversarial crowdsourced attacks, namely malicious task design, deceptive task execution, collusion, and Sybil attacks, and propose strategies to counteract them. Jollès et al. [22] identify three CI dynamics, namely aggregation of indicators of compromise, collaborative analysis, and a feedback loop, which leverage crowdsourcing to enhance the detection, accuracy, and currency of cybersecurity information for critical infrastructure protection.

3.1.3 Multi-Agent Systems

Multi-Agent Systems (MAS) are made up of independent multiple agents that communicate with each other and cooperate to perform specific tasks (see Figure 4), whereas crowdsourcing utilizes a substantial collective of individuals to jointly contribute to a job or project. MAS systems offer great potential in solving complex problems, efficient decision-making processes, and adapting to dynamic environments [23]. These systems can detect threats, perform data analysis, and develop rapid response mechanisms in IoT networks [24]. Inter-agent interaction and coordination enable CI systems to work proactively against threats and produce safer and more effective solutions [25].

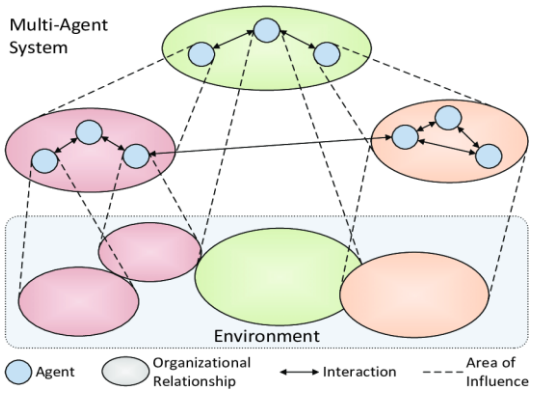

Figure 4. Multi-agent systems (source: [29])

The primary characteristics of MAS encompass [26]:

- *Decentralization*: MAS systems function autonomously, allocating work across several agents to improve robustness and scalability.
- *Autonomy*: Each agent is capable of independently and autonomously making choices using local knowledge and established procedures, without requiring an overarching system control.
- *Collaboration*: Devices (agents) exchange information and negotiate with one another to attain shared objectives, enhancing overall system efficacy and resilience. MAS can be used for CI in IoT networks, where distinct devices of various technologies collaborate to accomplish certain tasks (e.g., load balancing, and distributed decision-making).
- *Adaptability*: MAS systems can flexibly adjust to changes in environmental or system circumstances, making them appropriate for complex and evolving attack scenarios.
- *Heterogeneity*: Agents may possess varying capacities and fulfill distinct tasks within the system.

However, the MAS architecture also has security problems, including susceptibility to cyber-attacks, the need for secure communication protocols, and the significance of fault-tolerant technologies [26].

Example MAS usage in IoT networks [27]:

- Intelligent transportation systems, whereby vehicles (agents) cooperatively optimize routes.
- Distributed sensor networks, in which sensors (agents) exchange data and collaboratively choose environmental monitoring strategies.
- In a smart home IoT ecosystem, diverse products, such as thermostats, lighting systems, and security cameras, may independently collaborate to enhance energy efficiency according to customer preferences.

Reia et al. [28] examine the application of agent-based models in comprehending and simulating CI by analyzing the behaviors and interactions of individual agents as they collaboratively address complex problems, emphasizing the dynamics of emergent behaviors and the conditions that facilitate the emergence of CI. Kendrick et al. [29] present a novel decentralized MAS that enhances IoT security with enhanced real-time threat detection and response by distributing the processing of security events, thereby offloading computational costs from IoT devices. Stout [30] proposes a decentralized MAS architecture that improves cybersecurity of IoT networks by facilitating real-time threat detection and response through distributed intelligence, thus increasing the system's overall resilience and adaptability. Funchal et al. [31] introduce a MAS architecture that enhances the security of cyber-physical conveyor systems by integrating ML-based intrusion detection for improved real-time threat detection and system resilience. Choi et al. [32] introduces a cybersecurity-enhanced distribution automation system using a MAS that detects and mitigates cyber-attacks by monitoring communication protocols, protection schemes, and restoration applications for anomalies. This enables real-time, decentralized decision-making by enabling agents to coordinate and respond to threats effectively.

3.1.4 Self-Organizing Systems

Self-Organizing Systems (SOS) provide decentralized, autonomous decision-making and allow devices in a network to autonomously organize and coordinate without centralized control to solve complex problems and improve network efficiency. SOS depends on localized interactions among devices to attain overarching system functionality [33]. MAS and SOS both address intricate, distributed environments; yet, they diverge in their architecture and operating concepts. MAS comprises numerous interacting agents, each possessing distinct responsibilities and capacities, collaborating to attain a shared objective. These systems frequently depend on established protocols and centralized or decentralized control methods to synchronize agent operations. Conversely, SOS function without a central authority, depending on local interactions and straightforward principles to autonomously create structured structures and behaviors. This self-organization enables the system to react dynamically to environmental changes, enhancing resilience and adaptability. Although MAS can integrate self-organizing principles to improve flexibility, the primary difference is in the degree of autonomy and dependence on emergent behavior in SOS.

Essential attributes of SOS associated with CI include [33]:

- *Decentralization*: Devices inside the SOS framework function independently, rendering local judgments informed by their surroundings and adjacent devices. This decentralization diminishes the need for a central controller, which can be a bottleneck in large IoT networks.
- *Emergence*: SOS facilitates the development of sophisticated, intelligent behavior at the system level, beyond the aggregate capability of individual devices via localized interactions.
- *Adaptation and Scalability*: SOS can dynamically adjust to changes in the environment, network circumstances, or resource availability, making it extremely scalable. This versatility is crucial for overseeing IoT networks, where device status (e.g., failures, mobility) and network circumstances may fluctuate quickly.

CI mechanisms facilitated by SOS [33]:

- *Decentralized Decision-Making*: Enabling individual components to make decisions based on local information and interactions.
- *Modularity*: Breaking down complex systems into smaller, manageable modules that can operate independently and collaboratively.
- *Flexibility*: Adapting to changes and uncertainties in the environment without centralized control.
- *Robustness*: Maintaining functionality despite failures or disruptions within the system.
- *Reconfigurability*: Dynamically adjusting the system’s configuration to optimize performance under varying conditions.
- *Synergistic Collaboration*: Leveraging the collective capabilities of multiple agents to achieve goals that are beyond the reach of individual agents.

Casadei et al. [34] introduce a "pulverization" approach for cyber-physical systems, which separates self-organizing logic from deployment by breaking down global system behavior into smaller computational pieces executed across cloud, edge, and Long-Range Wide Area Network (LoRaWAN) infrastructures. This approach provides a flexible, deployment-independent application logic that improves system adaptability and resilience. Barletta et al. [35] introduce an intrusion detection method for in-vehicle communication networks using an unsupervised Kohonen self-organizing map network. This system detects attack messages on the controller area network bus using pattern and anomaly analysis, improving the network's capacity for autonomous cyber threat detection and response. Anwar et al. [36] present a data analytics model and self-organizing architecture for IoT networks, aimed at improving smart environmental monitoring systems via the management of massive data and the provision of strong protection against Denial of Service (DoS) and DDoS attacks. Sokolov et al. [37] introduce a methodology for validating the stability of self-organized wireless networks using block encryption, emphasizing the selection of unoccupied wireless channels via spectrum analysis and the evaluation of system throughput. This method mitigates the cybersecurity risk of

unauthorized access and data manipulation by guaranteeing reliable, secure, and efficient communication in self-organizing networks.

As a type of SOS and AI method, *swarm intelligence (SI),* derived from the collective behavior of social insects such as ants and bees, serves as an effective approach for facilitating CI in IoT networks. This method entails decentralized, self-organizing networks in which simple agents (IoT devices) adhere to fundamental laws and engage locally to produce intricate, emergent behaviors. In IoT networks, SI enables devices to collectively identify and address security threats in real-time, increasing resource efficiency and improving resilience. Utilizing distributed decision-making, each device can autonomously decide based on local input and collective knowledge within the swarm, enhancing adaptability and resilience. This approach diminishes dependence on centralized control, hence eliminating single points of failure and enhancing the network's capacity to react dynamically to fluctuating conditions and unforeseen occurrences. SI enables a scalable and efficient approach to managing and securing IoT networks, fostering transformational applications across diverse areas [38].

SI (AI) algorithms that can be utilized to address IoT architectural issues include the following [39]:

- Ant Colony Optimization (ACO): ACO algorithms, derived on the foraging behavior of ants, are employed for routing and resource allocation in IoT networks.
- Particle Swarm Optimization (PSO): Inspired by the social nature of avian flocks and aquatic schools, PSO is utilized to enhance work scheduling and load distribution in IoT systems.
- Bee Colony Optimization (BCO): Derived from the foraging behaviors of honeybees, BCO algorithms are employed for data clustering and network administration in IoT settings.
- Firefly Algorithm (FA): Derived from the luminescent behavior of fireflies, FA is employed to address optimization challenges associated with IoT security and energy efficiency.
- Bat Algorithm (BA): Inspired by the echolocation characteristic of bats, BA is utilized to improve the performance of IoT networks via effective resource management and anomaly detection.

Pham et al. [40] examine the utilization of SI methodologies in enhancing next-generation wireless networks (NGNs), encompassing 5G and subsequent technologies. The authors present a thorough examination of SI approaches, encompassing fundamental principles and prominent optimizers, while evaluating their applications in tackling new challenges in NGNs. Principal domains of application encompass spectrum management, resource allocation, wireless caching, edge computing, and network security. The paper emphasizes the capability of SI to manage the complexity and diversity of NGNs, providing solutions that improve efficiency and performance. The authors also address existing problems and prospective research avenues in the integration of SI with NGNs. Zhou et al. [41] introduce an innovative method for enhancing the security of IoT devices via effective work scheduling with SI. The authors present a hybrid approach that integrates PSO with Mixed-Integer Linear Programming to improve job scheduling, therefore improving security and energy efficiency. The method utilizes SI to dynamically assign jobs, guaranteeing strong security protocols while reducing energy usage. The study illustrates the efficacy of this approach via comprehensive simulations, revealing substantial enhancements in security and resource efficiency for IoT networks. Manal Abdullah Alohali et al. [42] investigate the application of SI methodologies to improve the security of IoT networks in fog-enabled cyber-physical systems. The authors advocate for a hybrid methodology that integrates PSO with additional ML techniques to efficiently identify and counteract cyber-attacks. This approach utilizes the decentralized and adaptable characteristics of SI to enhance the precision and efficacy of attack detection. The study illustrates the efficacy of this method via comprehensive simulations, revealing substantial enhancements in the detection of diverse attack types while preserving little computing overhead and energy usage.

3.1.5 References

[1] N. Mohamudally, "Paving the way towards collective intelligence at the IoT edge," Procedia Computer Science, vol. 203, pp. 8-15, 2022.

[2] M. Verkerken, L. D'hooge, D. Sudyana, Y.-D. Lin, T. Wauters, B. Volckaert and F. De Turck, "A Novel Multi-Stage Approach for Hierarchical Intrusion Detection," IEEE Transactions on Network and Service Management, vol. 20, no. 3, pp. 3915-3929, 2023.

[3] J. Wu, S. Nguyen, T. Kempitiya and D. A. Alahakoon, "Hierarchical Machine Learning Method for Detection and Visualization of Network Intrusions from Big Data," Technologies, vol. 12, no. 204, 2024.

[4] R. Lazzarini, H. Tianfield and V. Charissis, "Federated Learning for IoT Intrusion Detection," AI, vol. 4, p. 509–530, 2023.

[5] M. A. Ferrag, O. Friha, L. Maglaras, H. Janicke and L. Shu, "Federated Deep Learning for Cyber Security in the Internet of Things: Concepts, Applications, and Experimental Analysis," IEEE Access, vol. 9, pp. 138509-138542, 2021.

[6] Y. L. M. Z. H. C. a. Y. Z. Y. Fan, "IoTDefender: A Federated Transfer Learning Intrusion Detection Framework for 5G IoT," in 2020 IEEE 14th International Conference on Big Data Science and Engineering (BigDataSE), 2020 .

[7] S. D. a. A. A. G. E. C. P. Neto, "Collaborative DDoS Detection in Distributed Multi-Tenant IoT using Federated Learning," in 2022 19th Annual International Conference on Privacy, Security & Trust (PST), 2022 .

[8] C. L. M. C. a. G. M. J. Zhang, "Federated Learning for Distributed IIoT Intrusion Detection Using Transfer Approaches," IEEE Transactions on Industrial Informatics, vol. 19, pp. 8159-8169, 2023.

[9] D. C. Attota, V. Mothukuri, R. M. Parizi and S. Pouriyeh, "An Ensemble Multi-View Federated Learning Intrusion Detection for IoT," IEEE Access, vol. 9, pp. 117734-117745, 2021.

[10] X. Hei, X. Yin, Y. Wang, J. Ren and L. Zhu, "A trusted feature aggregator federated learning for distributed malicious attack detection," Computers and Security, vol. 99, 2020.

[11] R. Chaudhary, R. Kumar and N. Saxena, "A systematic review on federated learning system: a new paradigm to machine learning," Knowledge and Information Systems, 2024.

[12] "Cyber Security for Europe," [Online]. Available: https://cybersec4europe.eu/.

[13] A. Sforzin, "D5.6–Validation of Demonstration Case Phase 2," Cyber Security for Europe (CyberSec4Europe), 2022.

[14] G. Xia, J. Chen, C. Yu and J. Ma, "Poisoning Attacks in Federated Learning: A Survey," IEEE Access, vol. 11, pp. 10708-10722, 2023.

[15] L. Lyu, H. Yu, J. Zhao and Q. Yang, "Threats to Federated Learning," Lecture Notes in Computer Science (LNCS), vol. 12500, 2020.

[16] W. Li, W.-j. Wu, H.-m. Wang, X.-q. Cheng, H.-j. Chen, Z.-h. Zhou and R. Ding, "Crowd intelligence in AI 2.0 era," Frontiers of Information Technology & Electronic Engineering, vol. 18, p. 15–43, 2017.

[17] C. Sauerwein, M. Gander, M. Felderer and R. Breu, "A Systematic Literature Review of Crowdsourcing-Based Research in Information Security," in 2016 IEEE Symposium on Service-Oriented System Engineering (SOSE), Oxford, UK, 2016.

[18] V. Vlachos, S. Y.C. and S. Nikoletseas, "The Privacy Flag Observatory: A Crowdsourcing Tool for Real Time Privacy Threats Evaluation.," Journal of Cybersecurity and Privacy, vol. 3, pp. 26-43, 2023.

[19] N. Sun, J. Zhang, S. Gao, L. Zhang, S. Camtepe and Y. Xiang, "Data Analytics of Crowdsourced Resources for Cybersecurity Intelligence," in International Conference on Network and System Security, 2020.

[20] C. Christoforidis, V. Vlachos and I. Androulidakis, " A crowdsourcing approach to protect against novel malware threats," in 2014 22nd Telecommunications Forum Telfor (TELFOR), Belgrade, Serbia, 2014.

[21] M. Moradi and Q. Li, "Rogue people: on adversarial crowdsourcing in the context of cyber security," Journal of Information, Communication and Ethics in Society, vol. 1, pp. 87-103, 2020.

[22] E. Jollès, S. Gillard, D. Percia David, M. Strohmeier and A. Mermoud, "Building Collaborative Cybersecurity for Critical Infrastructure Protection: Empirical Evidence of Collective Intelligence Information Sharing Dynamics on ThreatFox," in Critical Information Infrastructures Security (CRITIS 2022), 2023.

[23] W. Lepuschitz, Self-Reconfigurable Manufacturing Control based on Ontology-Driven Automation Agents, Vienna, Austria: Technische Universität Wien, 2018.

[24] R. Coulter and L. Pan, "Intelligent agents defending for an IoT world: A review," Computers & Security, vol. 73, pp. 439-458, 2018.

[25] P. Hoen and S. M. Bohte, "COllective INtelligence with Sequences of Actions," in Machine Learning: ECML 2003. Lecture Notes in Computer Science (LNCS), Berlin, Heidelberg, 2003.

[26] R. Owoputi and S. Ray, "Security of Multi-Agent Cyber-Physical Systems: A Survey," IEEE Access, vol. 10, pp. 121465-121479, 2022.

[27] M. Gheysari and M. Tehrani, "The Role of Multi-Agent Systems in IoT," in Multi Agent Systems: Technologies and Applications towards Human-Centered, S. Gupta, I. Banerjee and S. Bhattacharyya, Eds., Singapore, Springer, 2022, p. 87–114.

[28] S. M. Reia, A. C. Amado and J. F. Fontanari, "Agent-based models of collective intelligence," Physics of Life Reviews, vol. 31, pp. 320-331, 2019.

[29] P. Kendrick, A. Hussain, N. Criado and M. Randles, "Multi-agent systems for scalable internet of things security," in Proceedings of the Second International Conference on Internet of things, Data and Cloud Computing (ICC '17), 2017.

[30] W. M. S. Stout, "Toward a Multi-Agent System Architecture for Insight & Cybersecurity in Cyber-Physical Networks," in 2018 International Carnahan Conference on Security Technology (ICCST), Montreal, QC, Canada, 2018.

[31] G. Funchal, T. Pedrosa, M. Vallim and P. Leitao, "Security for a Multi-Agent Cyber-Physical Conveyor System using Machine Learning," in "2020 IEEE 18th International Conference on Industrial Informatics (INDIN), Warwick, United Kingdom, 2020.

[32] I. -S. Choi, J. Hong and T. -W. Kim, "Multi-Agent Based Cyber Attack Detection and Mitigation for Distribution Automation System," EEE Access , vol. 8, pp. 183495-183504, 2020.

[33] P. Leitão, J. Queiroz and L. Sakurada, "Collective Intelligence in Self-Organized Industrial Cyber-Physical Systems," Electronics , vol. 11, no. 3213, 2022 .

[34] R. Casadei, D. Pianini, A. Placuzzi, M. Viroli and D. Weyns, "Pulverization in Cyber-Physical Systems: Engineering the Self-Organizing Logic Separated from Deployment," Future Internet, vol. 12, no. 203, 2020.

[35] V. Barletta, D. Caivano, A. Nannavecchia and M. Scalera, "Intrusion Detection for in-Vehicle Communication Networks: An Unsupervised Kohonen SOM Approach," Future Internet, vol. 12, no. 119, 2020.

[36] R. Anwar, K. Qureshi, W. Nagmeldin, A. Abdelmaboud, K. Ghafoor, I. Javed and N. Crespi, "Data Analytics, Self-Organization, and Security Provisioning for Smart Monitoring Systems," Sensors, vol. 22, no. 7201, 2022.

[37] V. Sokolov, P. Skladannyi and H. Hulak, "Stability verification of self-organized wireless networks with block encryption," Computer Modeling and Intelligent Systems, vol. 3137, pp. 227-237, 2022..

[38] W. Sun, M. Tang, L. Zhang, Z. Huo and L. Shu, "A Survey of Using Swarm Intelligence Algorithms in IoT," Sensors , vol. 20, 2020.

[39] L. Abualigah, D. Falcone and A. Forestiero, "Swarm Intelligence to Face IoT Challenges," Computational Intelligence and Neuroscience, vol. 4254194, p. 12, 2023.

[40] Q.-V. Pham, D. C. Nguyen, S. Mirjalili, D. T. Hoang, D. N. Nguyen, P. N. Pathirana and W.-J. Hwang, "Swarm intelligence for next-generation networks: Recent advances and applications," Journal of Network and Computer Applications, vol. 191, 2021.

[41] J. Zhou, Y. Shen, L. Li, C. Zhuo and M. Chen, "Swarm Intelligence-Based Task Scheduling for Enhancing Security for IoT Devices," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 42, no. 6, pp. 1756-1769, 2023.

[42] M. E. Manal Abdullah Alohali, F. N. Al-Wesabi, M. A. Duhayyim, A. M. Hilal and A. Motwakel, "Swarm intelligence for IoT attack detection in fog-enabled cyber-physical system," Computers and Electrical Engineering, vol. 108, 2023.

3.2 Generative Adversarial Networks (GANs)

3.2.1 Introduction

In the sphere of cybersecurity, the landscape is characterized by a constantly shifting and intricate network of threats. Organizations grapple with an unprecedented influx of cyber threats, each demonstrating a heightened level of sophistication [26]. Since the early 2000s, cyber-attacks targeting critical infrastructures have presented persistent challenges, leading to substantial disruptions in operational activities and service provisions [56]. Additionally, large-scale data breaches, exemplified by incidents like the Capital One breach, have compromised the personal information of millions, thereby fuelling the expansion of the cybercrime domain [45]. Notably, past successful attacks on critical infrastructure, such as Stuxnet and the Ukrainian Blackout, indicate the potential for further escalation and refinement of future cyber assaults [29, 47].

3.2.1.1 Motivation

One big challenge in cybersecurity is the fast-changing threat landscape. As technology moves forward, so do cyber attackers' tactics, techniques, and procedures (TTPs). This ongoing evolution means cybersecurity strategies must be just as flexible and responsive. Artificial intelligence (AI); machine learning (ML), deep learning (DL) and generative AI, are increasingly being used to create advanced defence systems that can adapt to new threats as quickly as they appear [43].

In response to the evolving threat landscape, there is a pressing need to enhance our capacity to defend against cyber-attacks. Considering the overwhelming volume of traffic, organizations must monitor, detect, and protect against, leveraging the latest AI technology to automate these tasks is critical. This approach is essential for adapting to new types of attacks and keeping pace with changing system patterns [29].

The rationale behind incorporating AI into cybersecurity measures lies in the necessity for dependable data solutions. The integrity and precision of data are fundamental to the success of AI-driven approaches, necessitating that the data remains impervious to unauthorized alterations [33]. Whether detecting anomalies or preventing security breaches, the efficacy of these systems across various domains is contingent on access to a substantial volume of high-quality data. Despite the advancements in the significant data era, there persists a challenge in acquiring data that is not only voluminous but also maintains a high level of quality and balance [17].

In advancing enhanced security protocols, deploying an Intrusion Detection System (IDS) is paramount. Within cybersecurity, IDS emerges as a critical component, identifying unauthorized user activities, thereby safeguarding against manoeuvres aimed at undermining or causing breaches in the system or network's confidentiality, integrity, and availability (CIA) stipulations. Its design is purposed to meticulously distinguish between malicious and legitimate traffic or actions. The accuracy of IDS in identifying and classifying security threats is heavily dependent on the quality of the underlying data. Absent high-calibre data, the efficacy of IDS in differentiating normal operations from malicious intents may be compromised, potentially culminating in false positives or negatives [21].

In pursuing advancing automated security measures, particularly the development of Intrusion Detection Systems (IDS), high-quality data becomes paramount. Current trends suggest that leveraging new advancements in AI is crucial to empower robust and intelligent IDS capable of safeguarding

systems effectively. Specifically, the advent of generative AI technologies, such as Generative Adversarial Networks (GANs), offers promising avenues for generating synthetic data. This section focuses on the application of GANs within the realm of synthetic data generation and evaluates their potential to enhance the efficiency of IDS in the cybersecurity landscape.

3.2.1.2 Application Under CISSAN Scenarios

The CISSAN project aims to enhance the development, testing, and validation of secure Internet of Things (IoT) applications and solutions. By leveraging collective intelligence and security-aware nodes, CISSAN contributes to a more secure energy supply. Under CISSAN scenarios, applying GANs or synthetic data generation is particularly valuable. GANs can generate realistic synthetic data to simulate various cyber-attack scenarios, providing a robust testing ground for Intrusion Detection Systems (IDS). This capability is crucial for developing and validating secure IoT applications within the CISSAN framework. By generating high-quality, diverse, and balanced synthetic data, GANs support the training of IDS to recognize and respond to both known and novel threats effectively. The adaptability of GANs ensures continuous data generation that mirrors the evolving nature of cyber threats, allowing IDS to remain responsive and accurate in real-time threat detection and mitigation. Integrating GAN-generated data within the CISSAN platform enhances the development of advanced AI-driven defence mechanisms. These mechanisms leverage synthetic data to improve their learning processes, boosting the overall resilience of network infrastructures against sophisticated cyber-attacks. This proactive approach ensures that CISSAN can maintain a high level of security, preparing for and mitigating potential threats with greater efficiency and precision.

3.2.2 Synthetic Data

Synthetic data is generated through computer algorithms to mimic the attributes of real-world data [23]. There are several key reasons why synthetic data is needed:

Privacy Protection: Synthetic data can be utilized to anonymize sensitive information, which is particularly crucial in sectors such as cybersecurity, where privacy protection is of utmost importance. By substituting actual data with artificial yet realistic data, researchers and organizations can develop and test algorithms without breaching confidentiality [22].

Data Scarcity: In scenarios where authentic data is scarce or challenging to acquire, synthetic data emerges as a significant substitute. It enables the artificial enlargement of data collections, essential for training proficient machine learning models. This advantage is especially pronounced when gathering actual data is costly, labour-intensive, or poses potential hazards [60, 61].

Model Development and Testing: Synthetic data enables the generation of simulated scenarios for training and testing purposes. This is particularly advantageous in cybersecurity, where it aids in developing and evaluating AI systems for threat detection and response in contexts where real-world experimentation might be unfeasible or hazardous [23].

In essence, synthetic data offers a way to overcome real-world data collection and utilization limitations. It safeguards privacy, addresses data scarcity, and facilitates the development of robust AI models in various fields.

It may appear that the significance of synthetic data has emerged predominantly in the recent era, coinciding with the advancements in generative AI. However, the utilization of synthetic data is not a novel concept; it has been integral to the field of statistics and data science since its early days in the 1960s [51]. Over the years, many methods for creating synthetic data have been developed.

The focus here is on tabular synthetic data, hence the following techniques discussed herein are particularly relevant and tailored to fit within the scope of this project focus.

Table 1 provides a high-level overview of the prominent techniques that have been used for the generation of tabular synthetic data, as discussed in the recent comprehensive survey on synthetic data by Alvaro Figueira and Bruno Vaz [23].

Table 1: Methods for the Generation of Synthetic Data

Classification	Methods

Standard (non-AI)	Random Oversampling [5]
	Synthetic Minority Oversampling Technique: SMOTE [9], Borderline-SMOTE [30], Safe-level-SMOTE [8]
	Adaptive Synthetic Sampling [31]
	Cluster based oversampling [38]: K-Means SMOTE [20]
	Gaussian Mixture Model [13]
AI-based Methods	Bayesian Neural Networks [59]
	GenerativeMTD [58]
	Autoencoders [24]: Variational Autoencoders (VAEs [34], TVAEs [68])
	Generative Adversarial Networks [27]: CTGAN [60], TGAN [61], Copula-GAN [7], SMOTE-GAN [36], CTAB-GAN [68], DC [40], GANBLR [66], MTS-TGAN [62]
	Diffusion Models: TABDDMP [46]

3.2.3 Generative Adversarial Networks

From a broader view, there are two types of AI. One is discriminative AI, and the other is generative AI [39]. Today, everyone is familiar with the power of AI, let alone generative AI. There are two ways to do generative modelling. One way is to find a density function describing the probability distribution. This is known as density estimation and, more precisely, **explicit density estimation**. The other way is **implicit density estimation**, which does not focus on defining an explicit density function. Instead, it learns a function to generate new samples that resemble the training data. The model can generate new data points but cannot tell you the exact probability of a given data point. Broadly, there are five main explicit generative models: variational autoencoder, energy-based models, diffusion models, autoregressive models, and normalization flow models, and one implicit generative model named Generative Adversarial network (GAN) (Figure 5) [24].

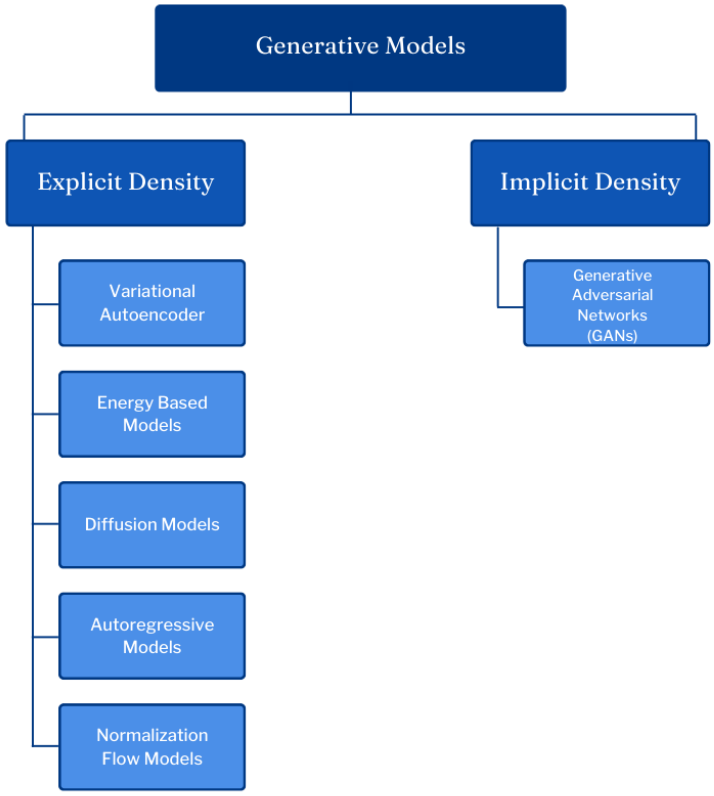

Figure 5: Taxonomy of Generative Models

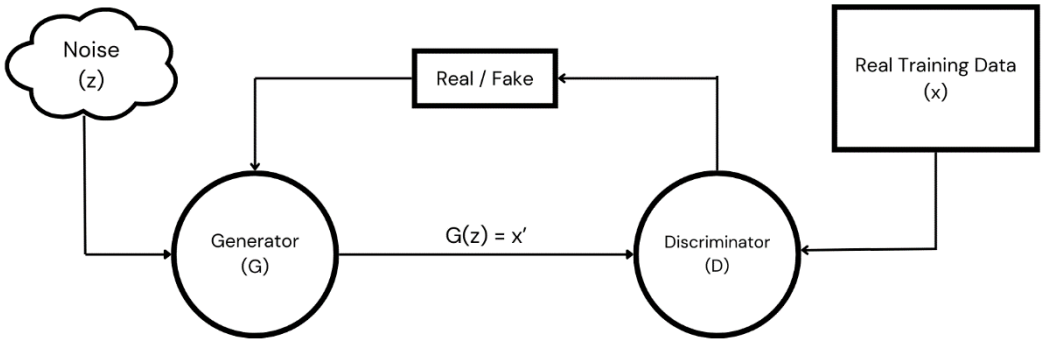

Figure 6: Original GAN structure by Goodfellow

GAN is a pivotal milestone in unsupervised machine learning, first introduced by Goodfellow et al. [27]. The GAN model comprises two main components, the Generator, and the Discriminator, and follows the framework depicted in Figure 6.

Generator: The Generator assumes the crucial role of crafting novel samples, be it images or data points, deriving them from random noise or a designated seed. In the training phase, initiation occurs with a random seed, promptly generating samples with the objective of crafting outputs that mirror reality. Post-training, the generator can generate synthetic samples virtually indistinguishable from authentic ones.

Discriminator: Conversely, the Discriminator assesses samples, categorizing them as authentic or generated. Through the training regimen, it hones its ability to differentiate between samples

originating from the real dataset and those produced by the Generator. Notably, the Discriminator is typically discarded after the successful training of the Generator.

Training Process: Simultaneous training of both the Generator and Discriminator unfolds in a competitive interplay. The Generator endeavours to produce convincingly realistic samples to deceive the Discriminator, while the latter strives for heightened accuracy in distinguishing between real and generated samples. This training loop persists until the Generator achieves the capability to generate samples indistinguishable from real ones, rendering the Discriminator only capable of making random guesses.

3.2.3.1 Different GANs

Originally, GAN was invented for the production of synthetic image data, and the base variant of GAN by Ian Goodfellow is known as vanilla GAN [27]. So far, there are numerous GAN variants that it is almost near impossible to count and mention each of the variants here. The reason is that the application level of GAN is limited to images and many other data formats like video, audio, text, and tabular data. However, in Table 2, some prominent GAN variants are covered by a recent comprehensive review paper by Aeryn Dunmore et al. [21].

Table 2: Prominent GAN Variants

GAN Variants	**Purpose**
CGAN (Conditional GAN) [49]	Provides control over the output of a GAN model by focusing on specific aspects, such as class labelling or certain features.
DCGAN (Deep Convolutional GAN) [54]	Modelled architecture based on original Convolutional Neural Networks for effective image generation and classification tasks.
Wasserstein GAN (WGAN) [4]	Measures distance and divergences between two distributions using Total Variation (TV), Kullback-Leibler (KL), and Jensen-Shannon (JS) divergences.
BiGAN (Bi-directional GAN) [19]	Introduces inverse mapping for feedback to the network and offers supervision for learning with different focuses.
GAN for Malicious Android Apps (GANG-MAM) [55]	Creates actual API calls to compromise infected devices, augment datasets, and increase the robustness of Android antivirus software.
CycleGAN [67]	Translates images from one domain to another.
AC-GAN (Auxiliary Classifier GAN) [52]	Increases structural requirements of latent space, characterizes the structure of natural images.
PassGAN [32]	Learns likely password distributions from real lists and creates password guesses.
IS GAN (Identity Sensitive GAN) [63]	Generates sketches based on photographs for detail extraction.
BEGAN (Boundary Equilibrium GAN) [6]	Combines WGAN and GANs using trained autoencoders, introduces equilibrium factor for balance.
ProGAN (Proximity GAN) [25]	Preserves important semantic data during down sampling.

MSG-GAN (Multi-Scale Gradients GAN) [42]	Addresses domain transferability issues with multiple scales of gradients.
SAGAN (Self-Attention GAN) [65]	Adds a self-attention module for long-range image tasks.
For greater functionality, IW-GAN (Inferential Wasserstein GAN) [12]	Melds Autoencoders and GANs guard against mode collapse.
InfoGAN (Information Maximising GAN) [10]	Untangles images of handwritten characters, adds negligible complexity to vanilla GAN.
SeqGAN (Sequence GAN) [64]	Generates data sequences using gradient policy.
TranGAN [11]	Undertakes social tie prediction in transfer learning.

From Table 2 and recent research, it is clear that GAN is more than generating synthetic images [2,41]. GANs have been successful in generating extensive data, including images, audio, and text, and have many potential applications such as generating synthetic training data, improving data augmentation, and creating new content in fields such as art and music [44]. GANs have been applied in various sectors, including computer vision, science-related activities such as cybersecurity, protein engineering, astronomical data processing, remote sensing image dehazing, and crystal structure synthesis, as well as in finance, marketing, fashion design, sports, and music [15].

3.2.3.2 Tabular GANs

Now, different data type formats have been produced by GAN [37]. Each one of them has its importance, but tabular data is one of the foremost important datatypes, which is versatile in almost every domain. Tabular data is a type of dataset that is structured in terms of rows and columns. Therefore, GAN has shown promise in addressing the challenges of generating synthetic tabular data [53]. Many GAN versions are specifically designed to generate tabular data. Still, some models gain more hype than others due to their performance, such as TGAN, CTGAN, TableGAN CasTGAN, Copula-GAN, SMOTE-GAN, and CTAB-GAN [3,23,36,68].

3.2.4 Challenges of Cybersecurity

In cybersecurity, several critical challenges persist that hinder the development and implementation of effective defence mechanisms. These challenges underscore the need for innovative solutions like Generative Adversarial Networks (GANs) to enhance cybersecurity measures.

Evolving Threat Landscape: The cybersecurity landscape is marked by its constantly evolving nature, with cyber threats becoming increasingly sophisticated and frequent. Organizations face a relentless influx of new threats that leverage advanced techniques to exploit vulnerabilities. This dynamic environment necessitates continuous adaptation and improvement of cybersecurity strategies to keep pace with emerging threats. Traditional methods often fail to respond quickly and effectively to these evolving threats, highlighting the need for more adaptive and intelligent solutions [29].

Attack Data Imbalance: One of the most significant challenges in cybersecurity is dealing with imbalanced datasets. Cybersecurity datasets typically contain a disproportionate ratio of normal network behaviour instances to cyber attack instances. This imbalance can severely skew the performance of machine learning (ML) and deep learning (DL) models, biasing them towards the majority class—usually normal behaviour. As a result, these models may exhibit reduced sensitivity to actual attacks, leading to higher false negative rates where malicious activities are misclassified as benign. This compromise in detection capabilities can have severe implications for the robustness of cybersecurity measures [35]. The issue of data imbalance also exacerbates the risk of overfitting in ML and DL models. Overfitting occurs when a model learns the noise or random fluctuations in the training data rather than the underlying patterns. Consequently, the model performs exceptionally well on the training data but fails to generalize to new, unseen data. In the context of cybersecurity, overfitting is particularly problematic because the ability to predict new and unseen attacks is crucial accurately.

Addressing overfitting requires methods to enhance the model's generalization capabilities, making it more effective in real-world scenarios [1].

Anomaly Detection and Threat Prediction: Anomaly detection is a critical aspect of cybersecurity, involving identifying patterns that deviate from normal behaviour. Traditional anomaly detection methods often struggle to accurately identify sophisticated and nuanced attack vectors, especially in the face of evolving threats. The need for more advanced techniques to detect anomalies effectively is paramount, as undetected anomalies can lead to significant security breaches [21].

3.2.5 GAN for Cybersecurity

The problem from where we started is cybersecurity and the need for quality data to keep pace with the evolving nature of the threat landscape. GANs offer promising solutions to many of the cybersecurity challenges. By generating high-quality synthetic data, GANs can help balance datasets, improve the diversity and volume of training data, and enhance the overall effectiveness of ML and DL models in cybersecurity applications. Specifically, GANs can be used to:

Data balancing: In the domain of ML and DL cybersecurity applications, the frequently encountered challenge is imbalanced datasets, particularly when distinguishing between normal network behaviour and cybersecurity attacks. This imbalance is not merely a statistical inconvenience but a substantial hindrance that can significantly skew the performance of the models. Specifically, the disproportionate ratio of normal instances to attack instances tends to bias the model towards the majority class—usually the normal behaviour—thereby reducing its sensitivity to detect actual attacks. Such bias can inadvertently lead to a higher rate of false negatives, where attacks are misclassified as normal behaviour, compromising the robustness of cybersecurity measures [35].

Moreover, this skewed data distribution exacerbates the model's susceptibility to overfitting. Overfitting occurs when a model learns the noise or random fluctuations in the training data to an extent where it performs exceptionally well on the training data but fails to generalize to unseen data. This situation is particularly dire in cybersecurity, where the ability to generalize and accurately predict new, unseen attacks is paramount [1]. Various techniques have been explored to mitigate these issues. Among these, GANs have gained noteworthy attention for their ability to generate synthetic data statistically similar to the original dataset. In the specific context of balancing datasets for cybersecurity, GANs have been deployed to augment underrepresented attack classes, thereby creating more balanced training datasets. This approach has been exemplified in work done on prominent cybersecurity datasets such as NSL KDD and UNSW-NB15 [18]. By synthesizing realistic attack data, GANs help enhance the diversity and volume of attack instances available for training, thus enabling models to learn the characteristics of various attacks better. Consequently, this improves attack detection capabilities as models become adept at identifying a broader spectrum of cybersecurity threats more accurately.

Synthetic Adversarial samples: GANs extend beyond merely addressing imbalances in datasets, especially within cybersecurity, where they significantly contribute by enhancing minority sample representation. Their capability encompasses the generation of varied and realistically simulated attack scenarios, which plays a crucial role in amplifying the richness of cybersecurity datasets and advanced development of threat detection methodologies. In this regard, a notable application of GANs is their use in benchmark datasets of KDD Cup and CICIDS2017 for generating synthetic adversarial samples. The methodology integrates GANs with attention mechanisms— methodology enhances the model's ability to focus on relevant features, essential for detecting subtle and complex attack patterns. Such a focused approach is indispensable for identifying sophisticated and nuanced attack vectors that might otherwise evade detection. Thus, this approach set a new benchmark for advanced cyber-defence strategies [57].

Intrusion Detection: GANs are more than just data generation, owing to their distinctive architectural framework. Two intricately linked neural networks undergo simultaneous training at the core of GANs. The first network generates data, encapsulating the ability to produce novel, synthetic data instances that mimic the real datasets it learns from. The second neural network is known as the discriminator. Its primary function is to evaluate the authenticity of the data generated by the first network, determining whether it is real or synthetic [27]. This unique tandem operation allows for a highly dynamic and adaptive model training process. Expanding upon this foundational understanding, the discriminator's function within a GAN configuration presents an intriguing opportunity for

divergent applications, particularly anomaly detection. By adjusting the discriminator network within a GAN, researchers have begun to harness its potential for detecting such anomalies, effectively repurposing the model for cybersecurity needs. A substantial literature has emerged, offering a comprehensive review of GAN applications aimed explicitly at intrusion detection within cybersecurity frameworks [21].

Several GAN models are in the spotlight for anomaly detection. A noteworthy example includes the adaptation of a Deep Convolutional GAN (DCGAN), which has been specifically modified to detect Android malware. This adaptation demonstrates GAN architectures' flexibility and potential in addressing cybersecurity challenges. In comparative analyses with traditional ML models, this modified DCGAN model showcases a commendable performance in identifying Android malware across the same datasets. However, its false positive rate of only 0.2% shows significant potential for GAN-based models in future anomaly detection development \cite{jan2018deep}.

3.2.6 Future research

Recent developments in GANs, synthetic tabular data, and cybersecurity IDS point to substantial progress in these domains. Nevertheless, specific areas still need to be explored, signaling the need for more focused research. A particular point of contention arises in standardizing evaluation methodologies for synthetic tabular data. Presently, both AI and non-AI methodologies are employed, yet without a universally accepted evaluative framework. The introduction of TabSynDex, a proposed universal metric for the robust evaluation of synthetic tabular data, partially addressed this gap [14]. However, the rationale behind selecting its five underlying formulas and metrics must be included.

Moreover, existing research predominantly emphasizes machine learning utility, gauging accuracy through various permutations of training and testing on natural and synthetic data [16,28,50]. This approach, while informative, needs to provide a comprehensive justification for the evaluation methods used on synthetic data. Therefore, there's a compelling need for a logical and standardized framework for evaluating synthetic tabular data.

Since 2014, GANs have evolved, finding applications across diverse fields. Despite these advancements, GANs grapple with the mode collapse dilemma, drastically limiting the generated datasets' diversity and thereby restricting the resulting data's novelty. Additionally, GANs occasionally need help generating varied data types and maintaining the correlation between the features of real tabular data. To overcome these limitations, it is imperative to direct future research toward enhancing GANs' capabilities. The goal should be to produce novel, diverse data that offers interpretability in feature representation and mirrors the correlation and distribution of real data [48].

3.2.6.1 Next steps – CISSAN

From the CISSAN perspective, GAN usage can be further researched to develop efficient synthetic attack data focusing on power grid electricity network flow. For this, recent state-of-the-art GAN models in the cyber security domain must undergo a testing phase, comparative analysis, and evaluation. Then, based on the results, the definition of the mathematical models (generator and discriminator) can be tweaked to achieve the desired output. There is also room for hybrid GAN testing, e.g., Federated GAN for the decentralized synthetic traffic generation.

3.2.7 References

[1] Sanad Aburass. Quantifying overfitting: Introducing the overfitting index. arXiv preprint arXiv:2308.08682, 2023.

[2] Renuka Agrawal, Kanhaiya Sharma, Sudhanshu Gonge, Rahul Joshi, and Dinesh Kumar Singh. Towards the applications of generative adversarial networks beyond images. In 2023 International Conference on Computational Intelligence and Sus tainable Engineering Solutions (CISES), pages 1010–1017. IEEE, 2023.

[3] Abdallah Alshantti, Damiano Varagnolo, Adil Rasheed, Aria Rahmati, and Frank Westad. Castgan: Cascaded generative adversarial network for realistic tabular data synthesis. IEEE Access, 2024.

[4] Martin Arjovsky, Soumith Chintala, and L´eon Bottou. Wasserstein generative adversarial networks. In International conference on machine learning, pages 214–223. PMLR, 2017.

[5] Rukshan Batuwita and Vasile Palade. Efficient resampling methods for training support vector machines with imbalanced datasets. In The 2010 International Joint Conference on Neural Networks (IJCNN), pages 1–8. IEEE, 2010.

[6] David Berthelot, Thomas Schumm, and Luke Metz. Began: Boundary equilibrium generative adversarial networks. arXiv preprint arXiv:1703.10717, 2017.

[7] Stavroula Bourou, Andreas El Saer, Terpsichori-Helen Velivassaki, Artemis Voulkidis, and Theodore Zahariadis. A review of tabular data synthesis using gans on an ids dataset. Information, 12(09):375, 2021.

[8] Chumphol Bunkhumpornpat, Krung Sinapiromsaran, and Chidchanok Lursinsap. Safe-level-smote: Safe-level-synthetic minority over-sampling technique for handling the class imbalanced problem. In Advances in Knowledge Discovery and Data Mining: 13th Pacific-Asia Conference, PAKDD 2009 Bangkok, Thailand, April 27-30, 2009 Proceedings 13, pages 475–482. Springer, 2009.

[9] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. Journal of artificial intelligence research, 16:321–357, 2002.

[10] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. Advances in neural information processing systems, 29, 2016.

[11] Yanjiao Chen, Yuxuan Xiong, Bulou Liu, and Xiaoyan Yin. Trangan: Generative adversarial network based transfer learning for social tie prediction. In ICC 2019 IEEE International Conference on Communications (ICC), pages 1–6. IEEE, 2019.

[12] Yao Chen, Qingyi Gao, and Xiao Wang. Inferential wasserstein generative adversarial networks. Journal of the Royal Statistical Society Series B: Statistical Methodology, 84(1):83–113, 2022.

[13] Chanachok Chokwitthaya, Yimin Zhu, Supratik Mukhopadhyay, and Amirhosein Jafari. Applying the gaussian mixture model to generate large synthetic data from a small data set. In Construction Research Congress 2020, pages 1251–1260. American Society of Civil Engineers Reston, VA, 2020.

[14] Vikram S Chundawat, Ayush K Tarun, Murari Mandal, Mukund Lahoti, and Pratik Narang. Tabsyndex: a universal metric for robust evaluation of synthetic tabular data. arXiv preprint arXiv:2207.05295, 2022.

[15] Ankan Dash, Junyi Ye, and Guiling Wang. A review of generative adversarial networks (gans) and its applications in a wide variety of disciplines: From medical to remote sensing. IEEE Access, 2023.

[16] Hongwei Ding, Leiyang Chen, Liang Dong, Zhongwang Fu, and Xiaohui Cui. Imbalanced data classification: A knn and generative adversarial networks-based hybrid approach for intrusion detection. Future Generation Computer Systems, 131:240–254, 2022.

[17] Hongwei Ding and Xiaohui Cui. A clustering and generative adversarial network based hybrid approach for imbalanced data classification. Journal of Ambient Intelligence and Humanized Computing, 14(6):8003–8018, 2023.

[18] Gcinizwe Dlamini and Muhammad Fahim. Dgm: a data generative model to improve minority class presence in anomaly detection domain. Neural Computing and Applications, 33:13635–13646, 2021.

[19] Jeff Donahue, Philipp Kr¨ahenb¨uhl, and Trevor Darrell. Adversarial feature learning. arXiv preprint arXiv:1605.09782, 2016.

[20] Georgios Douzas, Fernando Bacao, and Felix Last. Improving imbalanced learning through a heuristic oversampling method based on k-means and smote. Information sciences, 465:1–20, 2018.

[21] Aeryn Dunmore, Julian Jang-Jaccard, Fariza Sabrina, and Jin Kwak. A comprehensive survey of generative adversarial networks (gans) in cybersecurity intrusion detection. IEEE Access, 11:76071–76094, 2023.

[22] Khaled El Emam, Lucy Mosquera, and Richard Hoptroff. Practical synthetic data generation: balancing privacy and the broad availability of data. O'Reilly Media, 2020.

[23] Alvaro Figueira and Bruno Vaz. Survey on synthetic data generation, evaluation methods and gans. Mathematics, 10(15):2733, 2022.

[24] David Foster. Generative deep learning. " O'Reilly Media, Inc.", 2022.

[25] Hongchang Gao, Jian Pei, and Heng Huang. Progan: Network embedding via proximity generative adversarial network. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pages 1308–1316, 2019.

[26] Saeid Ghasemshirazi, Ghazaleh Shirvani, and Mohammad Ali Alipour. Zero trust: Applications, challenges, and opportunities. arXiv preprint arXiv:2309.03582, 2023.

[27] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. Communications of the ACM, 63(11):139–144, 2020.

[28] Omar Habibi, Mohammed Chemmakha, and Mohamed Lazaar. Imbalanced tabular data modelization using ctgan and machine learning to improve iot botnet attacks detection. Engineering Applications of Artificial Intelligence, 118:105669, 2023.

[29] James Halvorsen and Dr Assefaw Gebremedhin. Generative machine learning for cyber security. Military Cyber Affairs, 7(1):4, 2024.

[30] Hui Han, Wen-Yuan Wang, and Bing-Huan Mao. Borderline-smote: a new over sampling method in imbalanced data sets learning. In International conference on intelligent computing, pages 878–887. Springer, 2005.

[31] Haibo He, Yang Bai, Edwardo A Garcia, and Shutao Li. Adasyn: Adaptive synthetic sampling approach for imbalanced learning. In 2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence), pages 1322–1328. Ieee, 2008.

[32] Briland Hitaj, Paolo Gasti, Giuseppe Ateniese, and Fernando Perez-Cruz. Passgan: A deep learning approach for password guessing. In Applied Cryptography and Network Security: 17th International Conference, ACNS 2019, Bogota, Colombia, June 5–7, 2019, Proceedings 17, pages 217–237. Springer, 2019.

[33] Cesar Humberto Ortiz Huam´an, Nilcer Fernandez Fuster, Ademir Cuadros Luyo, and Jimmy Armas-Aguirre. Critical data security model: Gap security identification and risk analysis in financial sector. In 2022 17th Iberian Conference on Information Systems and Technologies (CISTI), pages 1–6. IEEE, 2022.

[34] Zubayer Islam, Mohamed Abdel-Aty, Qing Cai, and Jinghui Yuan. Crash data augmentation using variational autoencoder. Accident Analysis & Prevention,151:105950, 2021.

[35] Benjamin N Jacobsen. Machine learning and the politics of synthetic data. Big Data & Society, 10(1):20539517221145372, 2023.

[36] Elaheh Jafarigol and Theodore B Trafalis. A distributed approach to meteorological predictions: addressing data imbalance in precipitation prediction models through federated learning and gans. Computational Management Science, 21(1):22, 2024.

[37] Sushma Jaiswal and Priyanka Gupta. Glstm: A novel approach for prediction of real & synthetic pid diabetes data using gans and lstm classification model. International Journal of Experimental Research and ReviewOpen Access, 30:32–45, 2023.

[38] Taeho Jo and Nathalie Japkowicz. Class imbalances versus small disjuncts. ACM Sigkdd Explorations Newsletter, 6(1):40–49, 2004.

[39] Sanaa Kaddoura. A Primer on Generative Adversarial Networks. Springer Nature, 2023.

[40] Ha Ye Jin Kang, Erdenebileg Batbaatar, Dong-Woo Choi, Kui Son Choi, Minsam Ko, and Kwang Sun Ryu. Synthetic tabular data based on generative adversarial networks in health care: Generation and validation using the divide-and-conquer strategy. JMIR Medical Informatics, 11:e47859, 2023.

[41] Sanghoon Kang, Donghyeon Han, Juhyoung Lee, Dongseok Im, Sangyeob Kim, Soyeon Kim, and Hoi-Jun Yoo. 7.4 ganpu: A 135tflops/w multi-dnn training processor for gans with speculative dual-sparsity exploitation. In 2020 IEEE International Solid-State Circuits Conference-(ISSCC), pages 140–142. IEEE, 2020.

[42] Animesh Karnewar and Oliver Wang. Msg-gan: Multi-scale gradients for generative adversarial networks. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 7799–7808, 2020.

[43] Ramanpreet Kaur, Duˇsan Gabrijelˇciˇc, and Tomaˇz Klobuˇcar. Artificial intelligence for cybersecurity: Literature review and future research directions. Information Fusion, page 101804, 2023.

[44] Jyoti Kesarwani and Himanshu Rai. Generative adversarial networks (gans): Introduction and vista. In Artificial Intelligence, Blockchain, Computing and Security Volume 2, pages 27–34. CRC Press, 2024.

[45] Shaharyar Khan, Ilya Kabanov, Yunke Hua, and Stuart Madnick. A systematic analysis of the capital one data breach: Critical lessons learned. ACM Transactions on Privacy and Security, 26(1):1–29, 2022.

[46] Akim Kotelnikov, Dmitry Baranchuk, Ivan Rubachev, and Artem Babenko. Tabddpm: Modelling tabular data with diffusion models. In International Conference on Machine Learning, pages 17564–17579. PMLR, 2023.

[47] Gaoqi Liang, Steven R Weller, Junhua Zhao, Fengji Luo, and Zhao Yang Dong. The 2015 ukraine blackout: Implications for false data injection attacks. IEEE transactions on power systems, 32(4):3317–3318, 2016.

[48] Tongyu Liu, Ju Fan, Guoliang Li, Nan Tang, and Xiaoyong Du. Tabular data synthesis with generative adversarial networks: design space and optimizations. The VLDB Journal, 33(2):255–280, 2024.

[49] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784, 2014.

[50] Samuel Ndichu, Tao Ban, Takeshi Takahashi, and Daisuke Inoue. Security-alert screening with oversampling based on conditional generative adversarial networks. In 2022 17th Asia Joint Conference on Information Security (AsiaJCIS), pages 1–7. IEEE, 2022.

[51] Sergey I Nikolenko. The early days of synthetic data. In Synthetic Data for Deep Learning, pages 139–159. Springer, 2021.

[52] Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier gans. In International conference on machine learning, pages 2642–2651. PMLR, 2017.

[53] Eugenia Papadaki, Aristidis G Vrahatis, and Sotiris Kotsiantis. Exploring innovative approaches to synthetic tabular data generation. Electronics, 13(10):1965, 2024.

[54] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434, 2015.

[55] G Renjith, Sonia Laudanna, S Aji, Corrado Aaron Visaggio, and P Vinod. Gangmam: Gan based engine for modifying android malware. SoftwareX, 18:100977, 2022.

[56] Hugo Riggs, Shahid Tufail, Imtiaz Parvez, Mohd Tariq, Mohammed Aquib Khan, Asham Amir, Kedari Vineetha Vuda, and Arif I Sarwat. Impact, vulnerabilities, and mitigation strategies for cyber-secure critical infrastructure. Sensors, 23(8):4060, 2023.

[57] Mohammed Abo Sen. Attention-gan for anomaly detection: A cutting-edge approach to cybersecurity threat management. arXiv preprint arXiv:2402.15945, 2024.

[58] Jayanth Sivakumar, Karthik Ramamurthy, Menaka Radhakrishnan, and Daehan Won. Generativemtd: A deep synthetic data generation framework for small datasets. Knowledge-Based Systems, 280:110956, 2023.

[59] Andrew G Wilson and Pavel Izmailov. Bayesian deep learning and a probabilistic perspective of generalization. Advances in neural information processing systems, 33:4697–4708, 2020.

[60] Lei Xu, Maria Skoularidou, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. Modeling tabular data using conditional gan. Advances in neural information processing systems, 32, 2019.

[61] Lei Xu and Kalyan Veeramachaneni. Synthesizing tabular data using generative adversarial networks. arXiv preprint arXiv:1811.11264, 2018.

[62] Parul Yadav, Manish Gaur, Nishat Fatima, and Saqib Sarwar. Qualitative and quantitative evaluation of multivariate time-series synthetic data generated using mts-tgan: A novel approach. Applied Sciences, 13(7):4136, 2023.

[63] Lan Yan, Wenbo Zheng, Chao Gou, and Fei-Yue Wang. Isgan: Identity-sensitive generative adversarial network for face photo-sketch synthesis. Pattern Recognition, 119:108077, 2021.

[64] Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. Seqgan: Sequence generative adversarial nets with policy gradient. In Proceedings of the AAAI conference on Artificial intelligence, volume 31, 2017.

[65] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. In International conference on machine learning, pages 7354–7363. PMLR, 2019.

[66] Yishuo Zhang, Nayyar Zaidi, Jiahui Zhou, and Gang Li. Interpretable tabular data generation. Knowledge and Information Systems, 65(7):2935–2963, 2023.

[67] Miaomiao Zhu, Shengrong Gong, Zhenjiang Qian, and Lifeng Zhang. A brief review on cycle generative adversarial networks. In The 7th IIAE international conference on intelligent systems and image processing (ICISIP), pages 235–242, 2019.

[68] Yujin Zhu, Zilong Zhao, Robert Birke, and Lydia Y Chen. Permutation-invariant tabular data synthesis. In 2022 IEEE International Conference on Big Data (Big Data), pages 5855–5864. IEEE, 2022.

3.3 Blockchains

Blockchain technology goes far beyond an application in cryptocurrencies. In fact, it could become a tool for many companies and sectors (fashion industry, banking and finance, pharmaceuticals are some examples). Digitalization has changed our way of interacting with the environment, devices have also changed the way they interact.

3.3.1 Why use blockchain in this process?

These are some of the reasons why Councilbox has decided to make blockchain technology the basis of its developments: Why use blockchain in this process?

3.3.2 Basic introduction to Blockchain

Since Satoshi Nakamoto (pseudonym) introduced blockchain technology and its applicability to P2P commerce to the world, its evolution has not stopped. Blockchain can not only be used for commercial transactions but has the potential to be applied in any type of agreement between parties.

Basically, blockchain is an account book, a huge database, in which all types of transactions are recorded. Everything works by consensus of the parties, and the past cannot be erased or modified, nor operate outside the rules that the network itself has given itself.

So, a simple definition of blockchain could be: “A large immutable database in which records are securely recorded thanks to cryptography”. Blockchain is a specific typology of the more general concept of Distributed Databases (DLT -Distributed Ledger Technology).

3.3.2.1 How does a blockchain transaction work?

1. Some person/entity/node requests a transaction. The transaction could involve cryptocurrency, contracts, records or other information.
2. The requested transaction is transmitted to a P2P network with the help of different nodes.
3. The network of nodes validates the transaction and the user's status with the help of known algorithms.
4. Once the transaction is complete, the new block is added to the existing blockchain. In such a way that it is permanent and unalterable

Blockchain is not Bitcoin, but it is the technology behind the Bitcoin cryptocurrency. Bitcoin is a digital token and blockchain is the “ledger” for keeping track of who owns the digital tokens.

3.3.2.2 Basic blockchain architecture

A blockchain is a chain of blocks that contain information. For example, a Bitcoin block contains information about the sender, receiver, and the number of bitcoins to be transferred.

The first block in the chain is called the “Genesis” block. Each new block in the chain is linked to the previous block.

A block also has a hash. It is like a fingerprint that is unique to each block. It identifies a block and all its contents, and is always unique, like a fingerprint. So once a block is created, any changes within the block will cause the hash to change.

All blocks contain hashes from previous blocks. This is the technique that makes a blockchain so secure.

Hashes are an excellent mechanism to prevent tampering, but today's computers are becoming more powerful and can calculate hundreds of thousands of hashes per second. In a matter of minutes, an attacker can manipulate a block and then recalculate all the hashes of other blocks to make the blockchain valid again.

To avoid the problem, blockchains use the concept of “proof of work.”

A proof of work is a computational problem that requires some effort to solve. This type of mechanism makes it quite difficult to manipulate blocks, so if even a single block is manipulated, you will need to recompute the proof of work for all subsequent blocks. Therefore, the joint hashing and proof-of-work mechanism makes a blockchain secure.

However, there is one more method that blockchains use to protect themselves, and that is through distribution. Instead of using a central entity to manage the chain, blockchain uses a distributed P2P network. When someone joins this network, they will get the complete copy of the blockchain. Each computer is called a node.

There are 3 types of blockchain: public, private, consortium or permissioned.

3.3.2.3 Public blockchain networks

Public blockchain networks have four basic characteristics:

1. Anyone can download the code and manage a public node on their computer, validating transactions on the network and participating in the consensus process. This means that any user can be a miner within a blockchain network.
2. Anyone can make transactions on the chain. Any valid transactions will be added.
3. Transparency. Anyone can access and view the transactions using a block explorer, however, these transactions, although public, are anonymous so as not to give details of the participating parties.
4. Decisions within the network are made through decentralized consensus algorithms. Consensus can be by the PoW (Proof of Work) algorithm or the PoS (Proof of Stake) algorithm.

In public networks two models can be differentiated:

1) Public networks that operate without permissions

Public networks that operate with permits operate under the Proof of Work algorithm. In this type of networks, anyone can participate without needing permission or authorization. This implies that none of the participating parties know each other, but they have full trust in each other thanks to a series of rules already previously defined within the network protocol (Bitcoin).

2) Public networks that operate with permits

These types of networks operate with the Proof of Stake algorithm. Here participation is also open, but in order to participate in the network validation process, the user must meet certain requirements previously established in the network protocol (Ethereum).

As an advantage, public blockchain networks allow any actor to participate and use them to create new business models, and can even reduce, such as DApps (decentralized applications that use 'blockchain' so that users relate directly to each other and close agreements without there being a central entity that manages the service): e-chat, Zooz, CryptoCribs….

3.3.2.4 Private blockchain networks

On private networks, write permissions are restricted to one organization. On the other hand, reading permissions can be open to the public or restricted to the extent desired.

They are mainly used to maintain completely secure databases to which only the owners or authorized administrators have access (Multichain or Apla).

3.3.2.5 Consortium, or permitted, or hybrid, or federated networks

A consortium network borrows certain characteristics from public and private networks.

In consortium networks, “power” does not reside in a singular entity but is directed by the leadership of a group. That is, it is like a private network for a group of companies or entities.

Unlike public blockchains, consortium networks do not allow anyone with internet to participate in the transaction verification process.

They have the advantage that they are faster (since it eliminates the excessive data redundancy that usually occurs in public networks) and have greater scalability and transaction privacy.

The consensus mechanism is maintained by a series of nodes that have been preselected and trusted in advance. The difference with private ones is that these preselected nodes are not part of a single company (R3 is applying this type of networks for banking institutions and EWF for the electrical system, BigchainDB, Evernym).

3.3.3 Relevance identity models

3.3.3.1 AlastriaID

Alastria is a non-profit organization that aims to promote the digital economy through the development of decentralized technologies such as Blockchain. AlastriaId is a digital identity project carried out by the identity commission of this association, based on its blockchain platform, and with the objective of providing a framework for the development of projects of this nature.

3.3.3.2 EBSI /ESSIF

Permitted blockchain ecosystem based on open standards, with a transparent governance system, and that aims to align with current European regulations such as the RGPD, eIDAS, and others. Its architecture is divided into three different layers: central services made up of a set of APIs that allow applications to be developed in accordance with the principles established by the EBP (European Blockchain Partnership), the blockchain itself and a distributed storage system, and the infrastructure that provides connectivity with the blockchain network. Among its use cases, the European Self-Sovereign Identity Framework (ESSIF) stands out, which defines a governance model for the entities of the system, managed through a series of records that are stored in EBSI.

3.3.3.3 Hyperledger Indy

It is probably the most mature identity model today. It is a technology based on blockchain in which we find a distributed registry created with the aim of managing decentralized identities. This registry is used along with elements such as verifiable credentials based on ZKP (Zero-Knowledge Proofs), and DIDs among others to provide a development environment in which to control identity securely.

3.3.3.4 Microsoft Entra

Microsoft has established an active collaboration with members of the Decentralized Identity Foundation (DIF), the W3C Credentials Community Group, and the broader identity community. The product developed on Azure, Microsoft Entra, consists of a service called Verified ID that can issue verifiable credentials by retrieving claims of an id token, generated by the identity provider compatible with the organization's OpenID instance.

Among the relevant identity models, and their corresponding blockchain networks, EBSI is clearly the one with the greatest impact at the European level. Its key lies in the fact that the nodes that make up the network are distributed in the different EU member states, and it is specially designed for public administrations, with a governance system that easily adapts to the different institutions that have the capacity to issue attributes. of identity within the context of a European country. Another of the main reasons for choosing EBSI/ESSIF is its alignment with current European regulations in terms of identity, such as eIDAS and GDPR, and possible amendments to both.

3.3.4 The trilemma of blockchain technology

The trilemma of blockchain technology (Vitalik Buterin) is based on three fundamental pillars: security, decentralization and scalability. This trilemma, like others, refers to the fact that any network implementation has to choose two of those three characteristics and leave one of them in the background. Therefore, by definition, public blockchain networks must be primarily secure and decentralized, thus scalability takes a backseat. On the contrary, in the case of private blockchains, they are secure and scalable, but they are not as decentralized as public ones.

3.3.4.1 Decentralization

When Bitcoin and therefore blockchain was introduced, the idea behind the creation of a cryptocurrency was to make the current highly centralized financial system more decentralized, more democratic.

Decentralization has to do with the number of nodes (computers) that run the blockchain. The decentralized nature of a blockchain infrastructure is of central importance here, as Bitcoin's breakthrough was solving the double-spending problem without a central authority; something that is considered a trivial problem in centralized environments.

3.3.4.2 Security

Security has to do with encryption and, above all, with consensus mechanisms (proof of work vs proof of stake). Consensus mechanisms relate to “how many” of the network nodes must acknowledge a transaction before it is final and how those nodes are rewarded.

3.3.4.3 Scalability

Finally, scalability refers to the ability of a blockchain to maintain desirable transaction speed performance in the face of an ever-growing network and increasing number of transactions per second.

This is where the problem arises. At the time of the inception of blockchain technology the trend was to compromise the ability to scale efficiently in favor of decentralization and security. However, achieving a scalable blockchain is the only way to compete with much faster centralized networks.

It is important to note that while the blockchain trilemma poses a challenge to widespread adoption of the technology, there is no real law that prevents all three aspects from being achieved at the same time.

3.3.5 Limitations of blockchain technology

1) High cost of transactions: Nodes seek higher rewards for completing transactions in a company that operates on the principle of supply and demand.
2) Slower transactions: Nodes prioritize transactions with higher rewards, transaction backlogs build up.
3) Smaller ledger: It is not possible to obtain a complete copy of the blockchain, which can potentially affect immutability, consensus, etc.
4) Inefficiency: Each node running the blockchain must maintain consensus across the entire blockchain. This means very low downtime and makes the data stored on the blockchain unalterable forever. However, all this is an inefficient process, because each node repeats a task to reach a consensus.
5) Not suitable for high number of transactions: The next surge in Internet-connected devices needs a platform for machine-to-machine payments and automated micropayment services.

3.3.6 Blockchain evidence, registration and traceability

Digital Evidence (also referred to as Electronic Evidence) allows third parties to prove the occurrence of certain circumstances. It is an essential resource for companies and organizations whose validity and proper management must be guaranteed. The proliferation and need for legal value of electronic evidence leads to the approval of Regulation (EU) 910/2014 (also known as EIDAS) regarding electronic identification and trust services for electronic transactions in the internal market, which establishes a legal framework for electronic signatures, electronic seals, electronic time stamps, electronic documents, certified electronic delivery services and certificate services for website authentication.

This regulation (EU) 910/2014 sets the guidelines to guarantee the security of electronic transactions between citizens, companies and the public administration, applicable to all member countries of the European Union. Its articles cover the security standards that transactions must comply with and includes the figure of Trusted Service Providers.

Thus, the eIDAS Regulation defines this concept as “a trust service provider that provides one or more qualified trust services and to which the qualification has been granted by the supervisory body.” To be recognized as such and to be able to issue qualified services with full legal validity, the organization must undergo an audit by an accredited certification body in each EU Member State.

As a result, the provider complies with a series of technical, organizational and legal measures that guarantee the security of its solutions. During recent years, these providers have based their products and solutions on traditional technologies such as digital certificates, temporary stamping and cryptography, applied to electronic signature, identification, certified electronic notification.

With the arrival of blockchain technology, a new paradigm has opened for its application in the creation of immutable, distributed and time-stamped electronic evidence. This core technology provides the two main attributes that define electronic evidence:

1) Authenticity: encompasses the authentication and integrity of the data, guaranteeing persistence over time by demonstrating that the information has not been altered or modified to ensure non-repudiation.

2) Durability: Electronic evidence must be able to be located, recovered and interpreted in the long term. Thus, the organization must demonstrate that security standards have been met throughout the entire life cycle of electronic evidence.

The eIDAS 2 regulation arises because today, the original eIDAS is presented as an insufficient framework with regard to the implementation of a European electronic identity system that is truly secure, useful and efficient in the territorial scope of the EU.

Thus, the objectives of this eIDAS 2 are to facilitate, for cross-border use, the following issues:

1) Access to highly secure and reliable electronic identity solutions.

2) That public and private services can have reliable and secure digital identity solutions.

3) That natural and legal persons have the ability to use digital identity solutions.

4) That these solutions are connected with a variety of attributes and allow the specific sharing of identity data that is required for each specific service.

5) The acceptance of qualified trust services in the European Union and on equal terms with regard to their provision.

All current platforms that offer blockchain evidence services use transactions to integrate a unique identifier, which is mathematically related to specific data. This operation is direct on certain platforms. In this way, the expense associated with generating evidence in the blockchain is equivalent to the cost of executing a transaction in the corresponding blockchain network. Solutions based on public blockchains, such as Bitcoin or Ethereum, face scalability challenges, which can result in high transaction costs and long confirmation times, affecting the accuracy of timestamping. On the other hand, private or permissioned blockchains offer advantages in terms of speed and costs. However, its less decentralized nature could compromise the long-term immutability of evidence and raise questions about its quality.

Various organizations that offer services have been reviewed, among the main ones are:

GuardTime:

- It is a company that offers solutions based on KSI (Keyless Signature Infrastructure) blockchain. This technology allows verification of data integrity without the need for keys.
- Limitations: Although KSI offers advantages in terms of speed and security, it is not a solution based on a decentralized public blockchain, which could lead to questions about long-term immutability and quality of evidence.

Telefónica TrustOS:

- It is a platform developed by Telefónica that facilitates the adoption of blockchain technologies in companies. TrustOS provides modules that allow organizations to take advantage of the benefits of blockchain without the need for deep technical knowledge.
- Limitations: Being supported by a central entity such as Telefónica, there could be concerns about the centralization of the solution. Additionally, if you use private blockchains, the quality of evidence may be lower due to less decentralization.

IBM Blockchain:

- It is a platform that offers blockchain solutions for companies based on Hyperledger Fabric, which is a permissioned blockchain.
- Limitations: Being a permissioned blockchain, decentralization is limited. This can lead to inferior immutability guarantees compared to public blockchains.

POEX.io:

- Platform specialized in creating blockchain evidence in the Bitcoin network, providing proof of existence and time stamping.
- Limitations: Subject to limitations of the Bitcoin blockchain, with costs varying depending on network congestion. There may be a margin of error in timestamping due to block generation time in Bitcoin.

Stampery:

- Allows users to certify and verify documents using blockchain. It uses both the Bitcoin and Ethereum blockchains to ensure immutability and verification of data existence.
- Limitations: Given its dependence on public blockchains, it may face scalability issues and costs associated with rising gas prices (on Ethereum) or transaction fees (on Bitcoin). Furthermore, the accuracy of timestamping may not be accurate depending on network congestion.

Blocknotary:

- Is a solution that allows data verification and authentication using blockchain.
- Limitations: Similar to Stampery, by relying on public blockchains, you could face scalability and accuracy issues in timestamping.

Chainpoint:

- It is an open source solution that allows users to anchor data in the Bitcoin blockchain, generating proof that verifies the integrity and timestamp of the data.
- Limitations: Like Stampery, it depends on the Bitcoin blockchain, which can lead to scalability and time-stamp accuracy issues.

OpenTimestamps:

- It is an open source solution that allows you to prove that certain data existed at a specific time by anchoring it in the Bitcoin blockchain.
- Limitations: Shares similar limitations to Stampery and Chainpoint due to depending on the Bitcoin blockchain.

OriginStamp:

- It is a platform that uses multiple blockchains to anchor data, thus providing proof of existence and time stamping for that data.
- Limitations: By relying on multiple blockchains, including some public ones, you may face scalability and cost issues. Time stamping may not be accurate due to variability in confirmation times between different blockchains.

Proof of Existence:

- One of the first services that allowed users to pin documents to the Bitcoin blockchain, thus confirming their existence at a given time.
- Limitations: Being based on the Bitcoin blockchain, it faces the same limitations of scalability and variability in transaction costs, as well as accuracy in time stamping.

Tierion:

- Platform that allows users to pin a large number of tests on the blockchain by creating a single "merkle root".
- Limitations: Although Tierion mitigates some of the scalability issues by pinning multiple documents with a single transaction, it still relies on the public blockchain, which can influence confirmation costs and times.

Factom:

- It is a separate blockchain designed specifically to maintain data records without the need to store the entire data. It allows those records to be anchored in more secure blockchains such as Bitcoin.
- Limitations: Although Factom improves scalability and reduces costs by not storing all data on major blockchains such as Bitcoin, the adoption and security of its own blockchain is a consideration. The accuracy of timestamping also depends on how often you pin to more secure blockchains.

The evolution of the eIDAS Regulation is eIDAS 2

eIDAS 2 regulation arises because today, the original eIDAS is presented as an insufficient framework with regard to the implementation of a European Electronic Identity System that is truly secure, useful and efficient in the territorial scope of the EU.

Thus, the objectives of eIDAS 2 are to facilitate, for cross-border use, the following issues:

- Access to highly secure and reliable electronic identity solutions.
- That public and private services can have reliable and secure digital identity solutions.
- That natural and legal persons have the ability to use digital identity solutions.
- That these solutions are connected with a variety of attributes and allow the specific sharing of identity data that is required for each specific service.
- The acceptance of qualified trust services in the European Union and under conditions of equality with regard to their provision.

3.3.7 Blockchain Platforms

Below are different blockchain networks:

3.3.7.1 Ethereum (Ether -ETH-)

- High transaction fees and processing is quite slow.
- Allows you to develop applications or solutions with a modular architecture using plug-and-play components that are intended for use in private companies.
- Private blockchain technology whose objective is to have a business application, because it allows private transactions to be carried out
- Zero transaction fees.
- High scalability options and their reliance on delegated proof-of-stake consensus mechanism to enhance blockchain security.
- High scalability and compatibility with EVM (Ethereum Virtual Machine).

3.3.7.2 Avalanche (AVAX)

- Creation of decentralized solutions at scale.
- The Avalanche network consists of multiple blockchains.
- Combine the benefits of the “Nakamoto consensus” (robustness, scale and decentralization) with those of the “classical consensus” (speed, rapid finality and energy efficiency).
- “Platform of platforms”, which consists of thousands of subnetworks to form a single interoperable network.

3.3.7.3 Cardano (ADA)

- Consensus: Proof of Stake (PoS)
- Block time: 20 seconds

- Speed: 1,000 TPS per Hydra (multiple chains)

Created by mathematician Charles Hoskinson, Bitcoin millionaire and former co-founder of Ethereum.

A blockchain designed especially for “changemakers, innovators and visionaries”. Cardano is a two-layer platform that seeks to improve the scalability issues of first-generation blockchains.

You could say that the heart of the Cardano platform is “Ouroboros”, an algorithm that uses the Proof of Stake consensus protocol.

The system allows miners to validate block transactions based on the amounts of tokens they hold. Cardano's native token, ADA, has managed to position itself in the No. 4 place according to its market capitalization volume.

It is becoming a platform for the development of multiple assets, decentralized applications (DApps), that can be interoperable as well as sustainable.

3.3.7.4 Cosmos (ATOM)

- Consensus: Proof of Stake (PoS)
- Block time: 1 second
- Speed: Thousands of TPS

Created by Tendermint inventor Jae Kwon alongside Ethan Buchman.

With the goal of becoming the “Internet of blockchains,” Cosmos seeks to create a network that allows all other blockchains to communicate with each other and do so as efficiently and quickly as possible. An essential component of this platform to achieve its mission is: Tendermint Core.

Simply put, Cosmos is a network made up of many independent blockchains using Byzantine Fault Tolerant (BFT) consensus mechanisms, including Tendermint BFT. Each individual Blockchain maintains control of its own governance but is interoperable with other blockchains on the network.

In this regard, Cosmos offers a new approach to solving the interoperability and scalability problems of previous generation blockchains. Although it is a multi-asset project, Cosmos has its own native token: ATOM, which plays an important role in the protection and security of the network.

3.3.7.5 EOS (EOS)

- Consensus: (DPOS)
- Block time: 0.5 seconds
- Speed: +4,000 TPS

Created by the founder of Bitshares and Steem, Daniel Larimer together with Brendan Blumer.

The EOS.IO/ EOS Blockchain platform, developed by the private company Block.one and launched in 2018, offers “a Blockchain architecture designed to enable vertical and horizontal scaling of decentralized applications,” according to its white paper.

To achieve this goal, EOS uses a Delegated Proof-of-Stake (DPoS) consensus mechanism and a role-based permissions concept. These allow the flexibility to make instant high-level decisions, such as rolling back, freezing, and debugging failed applications, through majority agreement among designated stakeholders.

EOS has been one of the fastest networks in the entire cryptocurrency market, in addition to not including payment of commissions to its users for making transactions or using the network. However, its development seems to have stalled and part of its community has lost confidence in the project. Especially after Daniel Larimer resigned from his position as CTO of Block.one.

3.3.7.6 Harmony (ONE)

- Consensus: Effective Proof of Stake

- Block time: 5 seconds
- Speed: +200,000 TPS

Created by Stephen TSE together with the group of engineers Rongijan Lan, Nick White and Sahil Diwan.

Harmony is a public Blockchain with sharding infrastructure that focuses on speed and security in decentralized applications. Fully interoperable with Ethereum, its mainnet was launched in 2019 and consists of four shards, each with 1,000 nodes that produce blocks in eight seconds.

The underlying structure of the Harmony network is based on a specific type of PoS called Effective Proof of Stake or EPoS, focused on decentralization and fair distribution of rewards. Its mechanism has two main parts: delegators and validators. For its part, the native token, called ONE, is an important element in the governance of this network.

Harmony is becoming a viable extension for Ethereum applications and asset.

3.3.7.7 Near (NEAR)

- Consensus: Thresholded Proof of Stake (TPoS)
- Speed: +1,000 TPS

Created by Alexander Skidanov (ex Microsoft) Illia Polosukhin (ex Google/TensorFlow) Other Googlers.

Aiming to be “the developer's fastest path to market,” Near Protocol launched its mainnet on October 13, 2020. Designed for high-performance DApps and their use among millions of users, Near is also a Blockchain interoperable with other networks such as Ethereum.

Its founders introduced the sharding algorithm called “Nightshade” for segmenting the transaction calculation load and maintaining complete decentralization. Additionally, Near uses a block generation mechanism called Doomslug to process over 100,000 transactions per second.

One of the most distinctive features of Near is its sharding mechanism: all code running on the blockchain is sharded into shards, with each shard running on a single node parallel to each other. This project is young and its evolution is just beginning.

3.3.7.8 Polkadot (DOT)

- Consensus: Proof of Stake (PoS)
- Block Time: 6 seconds

Created by Gavin Wood, inventor of Solidity, co-founder of Ethereum.

One of the most talked about projects, Polkadot was built to connect private and consortium chains, public and permissionless networks and “future technologies that are yet to be created.”

Simply put, Polkadot is a next-generation Blockchain project that connects multiple specialized blockchains into a unified network. It is secured by a staking consensus mechanism, called Nominated Proof-of-Stake (NPoS), which allows two types of network actors - validators and nominators - to secure the network.

Polkadot has its own token: DOT.

By allowing unrelated blockchains to share data efficiently without the need for a third party, Polkadot has been one of the most innovative platforms in the blockchain space. Capable of processing more than 1,000 transactions per second, it is also one of the fastest thanks to its parachain technology. A system that uses multiple parallel blockchains (parachains) to divert the processing load from the main chain, the growing number of parachains on the Polkadot network means it is only a matter of time before it reaches speeds of one million transactions per second.

Kusama is Polkadot's pre-production environment that allows developers to experiment and test new blockchains and/or DApps before they are officially released on the network.

Kusama can be seen as a sandbox for developers where they can test early versions of their Polkadot projects but with real cryptocurrencies that are traded on an open market.

3.3.7.9 Solana (SOL)

- Consensus: Proof of History and PoS
- Block time: < 1 second
- Speed: 50,000 TPS between 200 nodes (without sharding)

Created by Anatoly Yakovenko.

Created in 2017, Solana emerged with the goal of becoming a censorship-resistant Blockchain network that provides the open infrastructure necessary for global adoption.

This platform addresses one of the long-ignored problems of blockchain, but fundamental to decentralization: time. To do this, the project has designed a kind of “Blockchain clock.” It employs what is known as the Proof of History (PoH) consensus method, which acts as a complementary component to the Proof of Stake (PoS) consensus.

Just like its rivals on this list, this platform seeks to address the problems of scalability, speed, interoperability while promoting an ecosystem for building decentralized applications.

Applies timestamps to the approval of each transaction. These allow network nodes to find the correct sequence of events, playing a crucial role in the blockchain's cryptographic clock.

The Gulf Stream Mechanism also further increases the speed of the solana blockchain when deleting the mempool. In a normal blockchain, the mempool is the place where transactions are collected before a node selects them for validation and inserting them into a new block.

With Gulf Stream, the network sends new transactions to validators before all transactions in the current block are approved. Solana also makes use of a sea level system to develop smart contracts that can run in parallel and that can use the same protocols. Through this, thousands of smart contracts can run simultaneously without slowing down the Solana blockchain.

3.3.7.10 Zilliqa (ZIL)

- Consensus: PoW, PoR (Proof of Reputation)
- Block time: 45 seconds
- Speed: +2,000 TPS

Created by Max Kantelia.

Zilliqa (June 2017) launched its mainnet in January 2019 and ensures that its commitment is to offer a scalable and secure platform for developers and companies that want to create decentralized applications.

The Blockchain platform uses a process known as sharding, which divides the network nodes into groups of 600 each. Zilliqa has a native token known as ZIL, which acts as an incentive for miners, powers smart contracts, and allows transaction fees to be covered. But it also has gZIL, a governance token with which you can stake.

3.3.7.11 RSK (rootstock)

Smart contract platform for Bitcoin.

The security of the RSK system is set through a mixed system. First, a federation of trusted entities that helps move bitcoins between the two blockchains, and then, a mining power that receives transaction fees. RSK uses GHOST to achieve average confirmation times between blocks of 10 seconds. The maximum transaction capacity for simple payments is estimated at approximately 400 transactions per second.

It maintains a 1:1 parity with bitcoin thanks to the use of a “bridge” or a “two way peg” mechanism between both blockchains.

Notable proposals:

- RSKIP53 (or LTCP protocol): The LTCP protocol aims to reduce the size of transactions using a technique called delta compression, which basically takes an original version of the transactions as a “template” so as not to repeat the same content in the next one, but only saving the modified bytes. Likewise, with LTCP, transaction signatures will be removed and only transaction chains will be signed instead.
- RSKIP04 (parallel transactions): a new field would be added to the header of each block to tell miners how to partition transactions into sets and how full nodes should process them in parallel (simultaneously) securely. This would save space and gain speed.

3.3.8 BAAS Suppliers in the Market

Figure 7: BAAS vendors

3.3.8.1 Microsoft

They offer solutions through their Microsoft Azure Blockchain as a service platform.

They generally offer three types of service:

- Dev / Test Topology. You work with a virtual machine. It is only for developers.
- If it is a single organization: single member topology.
- If it is a consortium: multi-member topology.

3.3.8.2 IBM

- Blockchain as a service of the IBM platform on Hyperledger Fabric.
- Efficient development environment for your applications.
- Extensive development tools make it easy to make changes.
- High availability and also offer a disaster recovery option
- Particularly suitable for financial services, banking and supply chain management

3.3.8.3 Oracle

- It works similarly to public blockchains.

3.3.8.4 Amazon

- Blockchain as a Service AWS.
- They offer three types of blockchain platforms: Hyperledger Fabric, Corda, and Ethereum.

3.3.8.5 Alibaba

- Open ecosystem.
- Alibaba Cloud works with Quorum and Hyperledger Fabric.

3.3.8.6 SAP

- They offer Hyperledger Fabric, Quorum and MultiChain integration.

3.3.8.7 Baidu

- XuperChain blockchain platform.
- They offer three types of blockchain platforms: Hyperledger Fabric, XuperChain, and Ethereum. XuperChain is Baidu's original blockchain platform, and it is open source.

3.3.8.8 Huawei

- Also new to the business.
- Huawei Cloud Platform

3.3.8.9 Kaleido

- Kaleido is one of the Blockchain as a Service providers in the market in partnership with ConsenSys.
- They offer three types of blockchain platforms: Hyperledger Fabric, Corda, and Ethereum.

3.3.8.10 Hewlett-Packard Enterprise

- HPE is launching its relatively new Blockchain as a Service solution for mission-critical environments.

3.3.9 Risks

Scalability is understood as the ability of any system or process to adapt to increases in demand without affecting performance.

To deal with the large number of transactions/evidence generated by IoT networks, a scalability problem has been detected in the design of blockchain networks.

Scalability is related to the performance of the blockchain network in the face of high increases in the number of transactions. This performance is mainly related to the number of users on the network and the consensus algorithm it uses.

A large number of transactions implies that many nodes are required to validate and verify all these processes. Also, a greater number of users can slow down the network, since, when talking about a decentralized system (that is, one that does not depend on a central entity), more users imply a greater number of “copies” of the network. With all the data contained in it from the beginning. And, in addition, it must be constantly and permanently updated as more information is created and added.

An increase in the number of transactions in addition to implying a longer confirmation time, as well as an increase in commissions (networks like Bitcoin charge an increasing commission the higher the priority of incorporating the transaction into the registry).

Transactions per second (TPS) refers to the number of transactions the network can process in a second, followed by how quickly the network can confirm a trade or exchange. The average transaction speed is important because it indicates the current capacity of the network to process

transactions. If a cryptocurrency is experiencing an increase in its transaction volume, the average speed will decrease.

Transactions are the underlying unit of activity on any blockchain. The speed with which transactions are processed is critical in determining the usefulness of a given network.

Table 3: Transaction speed

Cryptocurrency	Transaction Per Second (TPS, ms)	Average Transaction Confirmation Time (Block)
Bitcoin	3-7	10 min
Ethereum	15-25	6 min
Solana	2.825	0.4 sec
Polkadot	1.000	4-5 sec
EOS	4.000	0.5 sec
Cosmos	10.000	2-3 min
Stellar	1.000	2-5 sec
Dogecoin	30	1 min
Litecoin	56	30 min
Avalanche	5.000	1-2 sec
Algorand	1.000	45 sec
Ripple (XRP)	1.500	4 sec
Bitcoin Cash	61	60 min
IOTA	1.500	1-5 min
Dash	10-28	15 min

To give us an idea, Bitcoin is capable of processing an approximate average of 7 transactions per second (TPS), while Ethereum reaches 20 TPS and a traditional and centralized system such as the issuance of Visa credit cards can reach 56,000 TPS.

The fastest and most scalable of all, by far, is believed to be Solana (SOL). According to its white paper, Solana should be able to reach 710,000 TPS. However, this is only theoretical. During testing, the project easily reached 65,000 TPS and the developers believe it could go up to 400,000 TPS. Additionally, the block completion time is 21-46 seconds. As already mentioned, Bitcoin requires a full hour, at minimum, to achieve the same.

Following its upgrade to Ethereum 2.0, the project increased its maximum TPS to 100,000. Given that its previous TPS was 12-15, this is quite an achievement. However, Ethereum is a very popular and widely used blockchain. As such, it definitely has the need for high TPS to handle the traffic and microtransactions of smart contract-based DApps.

Thus, transaction speeds are severely affected. Reaching points where networks can collapse and demand a large amount of time to process a transaction. Something that is not at all convenient for the long-term sustainability and adoption of this technology.

Although this methodology and work structure provides the blockchain with a high level of security and protection, it also causes the scalability of these networks to be limited. Making blockchain networks unable to process more information than each of its nodes can process individually.

On the other hand, the consensus algorithm will determine the difficulty and time necessary for a new transaction to be incorporated into the general registry.

Despite the use of equipment with great computational power, the scalability problem is still present.

Some drawbacks of the decentralized blockchain design are:

- Confirmation of transactions on the bitcoin blockchain takes up to an hour before they are irreversible.
- Micropayments, or smaller payments, are confirmed inconsistently.
- Current networks have a fixed rate per transaction, making micropayments unviable.

Regardless of the model, the question of their speed is ultimately a question of the need they fill.

3.3.10 Possible solutions to the blockchain technology trilemma

In researching and developing solutions for the trilemma, different approaches can be taken: direct modifications to the blockchain network (Layer 1 solutions), running a different network from the main blockchain (Layer 2 solutions). Ethereum, for example, introduced Proof-of-Stake as a Layer 1 solution, while Bitcoin introduced the Lightning Network as a Layer 2 solution. All of these potential solutions have been applied in the cryptocurrency financial domain, not in the of applications with legal validity.

- Layer 1 solutions (network configuration) or what we could also call in-chain.
    - Improved consensus mechanisms to make it faster when confirming operations: proof-of-work consensus is secure and decentralized, but it is slow; trending is a proof-of-stake consensus mechanism.
    - Greater block size.
    - Solutions based on network division. One option is Fragmentation (sharding). This solution involves breaking transactions into smaller “chunks” to speed up the transaction onboarding process. These are then processed simultaneously in parallel by the blockchain so that it can run on multiple transactions at the same time. Additionally, nodes do not have to contain a copy of each source block; instead, this information is divided and stored by different nodes.
- Layer 2 solutions or out-chain solutions
    - Solutions based on reducing transactions. All developments based on reducing the number of transactions sent to the main network would be included.
        - An alternative is the nested blockchain. In this type of system, the main blockchain sets the rules for the entire network and is not expected to participate in any operations unless a dispute needs to be resolved. There are several levels of blockchains, which are built on top of each other and connected using a parent-child chain connection. The delegates of the upper chain work below their children, carry out the actions and send the result to the main chain, reducing its workload and increasing scalability.
    - State Channels: Create two-way communication between a blockchain and off-chain transaction channels. State channels do not require node verification to validate transactions; Instead, this off-chain resource seals transactions with smart contracts. When transactions are completed on a state channel, the final state of the “channel” and all of its transactions are added to the underlying blockchain.
        - A possible implementation is the creation of so-called “payment channels”, which basically consists of grouping transactions by pairs of users (between whom there must be trust) to send only the final balances of the accounts to the blockchain. In this way, the number of final transactions that must be confirmed is much smaller.

In accordance with the previous trends, there are several blockchain technologies that can be used to improve its ability to scale. However, they are still far from traditional systems, and it is not easy to improve in this aspect without harming the security of the network or guaranteeing its decentralization.

3.3.11 Most significant technologies to investigate

- Permission-required blockchains

Permission-enabled blockchains are a necessity for Blockchain as a Service providers. Basically because of the need for privacy.

An example is the Ethereum platform, but by itself it cannot offer adequate applicability within companies or public administrations. Therefore, it is necessary to investigate alternatives to provide traditional blockchain networks with the required privacy.

- Monitoring tools: Block monitoring and exploration tools

Tools are required to maintain the overall health of the BAAS solution. The tools should make it possible to know how the nodes work or if there is any type of error in the ledger.

You can also use these tools to see if a node is trying to cause registry corruption and manipulate information.

Also, the use of performance verification tools will allow us to know the behavior of the service.

- Blockchain-First Services

Collection of tools that simplifies the development of blockchain-based applications. They are developer-oriented, considering the development cycle, from coding and testing to debugging and deployment.

It should allow developers to quickly and easily build and test their applications, reducing development time and increasing the reliability of the final product.

They should be platform independent and interoperable, meaning they can be connected to other tools quite easily.

Technologies like ENS, Metamask, Truffle, Swarm, BigChainDB, IPFS and many more offer some innovative solutions.

- Sidechain

In 2014, a group of well-known Bitcoin researchers and developers published an article in which they presented the idea of the sidechain. A Bitcoin sidechain would be another blockchain where the native currency is also bitcoin.

RSK is a sidechain of the Bitcoin blockchain, but with similarities to Ethereum. It is a “Turing complete” system and the Rootstock Virtual Machine (RVM) is almost identical to the Ethereum Virtual Machine (EVM). Therefore, any contract on Ethereum is portable to RSK and can be programmed in Solidity, the industry standard language for smart contracts and decentralized applications.

- Merged Mining

Basically, it is about mining two or more different cryptocurrencies at the same time, with the mining solutions for only one of them, the same equipment and almost the same software. In other words, a two for one (or three and more for one?) offer in mining.

Unlike the Liquid Network sidechain, for example, block creation and transaction processing is not done with a federation. RSK uses “merged mining”, a more decentralized solution.

In merged mining, it is the Bitcoin miners themselves who mine the secondary network, in this case RSK. Because miners send a proof of work every 30 seconds to prove their work to the mining centers, they can also send it to the RSK network (hence a block is produced every 30 seconds).

In this way a Bitcoin miner can additionally earn commissions from RSK transactions and, in fact, large part of miners does so.

- Federation model (Bridge between the main chain -L1- and the secondary chain -L2-)

The Bitcoin blockchain itself cannot operate a bridge. Therefore, currently there is no choice but to use a federation, that is, a group of people who keep your bitcoin on the main network while you use the sidechain and who promise to return it when you require it.

This is Liquid's model and it was also RSK's at first, but they decided to make a modification for greater security. The new bridge mechanism was called “powpeg”.

In the powpeg bridge, federation members use devices known as “hardware security modules” (HSM) to store the private keys that control the locked bitcoin. HSMs do not show keys to anyone, are tamper-proof, and sign bitcoin return transactions autonomously. In other words, the members of the federation cannot agree to steal the bitcoin, in the worst possible case they can only disconnect the HSMs, preventing the recovery of the bitcoin. Until this happens, 1 RBTC can be exchanged for 1 BTC using the bridge.

Although when we say “they cross” we are actually talking about an illusion, because what happens is that the side chain has its own token equivalent to the main chain cryptocurrency. Thus, when a user needs to access the sidechain functions, he transfers from his wallet on the main chain the amount of cryptocurrency he needs to a multi-signature address on it where it remains locked.

Once cryptocurrencies are locked on the main chain, the same number of equivalent tokens are unlocked on the secondary chain. And vice versa, if you want the balance of your cryptocurrencies back on the main blockchain to use them outside the sidechain.

For example, let's say you want to use the RSK sidechain, so you lock a BTC on the Bitcoin blockchain. That BTC is frozen out of your reach, but in exchange you now have 1 RBTC on the RSK sidechain and with that RBTC you can start paying the small amounts required for transactions to create smart contracts or make instant payments.

RSK Federation: a group of 15 members in a semi-brokered system, which are responsible for freezing and unfreezing BTC as necessary. Those 15 members are mostly prominent companies in the ecosystem with the technical capacity to maintain and manage their own node. In exchange for its services, the Federation receives 1% of the transaction fees generated in RSK.

- Fragmentation (Sharding)

Sharding was created with the purpose of allowing greater scalability in distributed and decentralized systems. But today, its application in blockchain technology could considerably improve the scalability problems faced by current networks.

Since transactions can be processed and validated more quickly, reducing the amount of time required for this process. And consequently, the network will have the capacity to process a greater number of transactions per second.

Simply put, sharding technology is a form of slicing to distribute computational and storage load across a P2P network. This way each node is not responsible for processing the entire transactional load. On the contrary, each node only considers the information related to its partition (shard).

With the implementation of sharding, it will no longer be necessary to store the entire blockchain on the same node, so the purchase of expensive equipment will not be required. This would allow many more people to have the possibility of participating in the network with their conventional equipment, guaranteeing its decentralization.

The implementation of sharding can represent a viable and feasible solution that eliminates scalability problems, and that allows the processing of a greater number of transactions in a shorter amount of time, safely and efficiently.

With sharding applied to the blockchain environment, it will no longer be necessary for all blockchain nodes to work linearly to validate all the data that is added to the chain. Rather, they will operate in parallel and will manage specific fragments where the information will be distributed and in these fragments they will be in charge of validating and processing only the data that corresponds to them. And when all the groups of nodes finish executing the assigned process, all the information will be added to the blockchain, keeping it complete and complete, with the difference that the nodes will not manage the information in its entirety as is the case until now.

It is necessary to analyze whether the application of the fragmentation method will allow transactions to be processed much more quickly, improving the scalability and efficiency of the network.

- Merkle tree

This design was created by Ralph Merkle in 1979, with the aim of streamlining the verification process of large amounts of data, relating them through various cryptographic and information management techniques.

Or in other words, it is a data structure divided into several layers that aims to relate each node to a unique root associated with them. To achieve this, each node must be identified with a unique identifier (hash). These initial nodes, called child nodes (leaves), are then associated with a parent node called the parent node (branch). The parent node will have a unique identifier resulting from the hash of its child nodes. This structure is repeated until reaching the root node or “Merkle Root”.

Merkle trees allow a large amount of data to be related to a single point. In this way, the verification and validation of this data can become very efficient, by only having to verify the “Merkle Root” instead of the entire structure.

Some of the most notable characteristics of Merkle trees are:

1. They are an efficient means of generating a distributed data structure.

2. They provide great security and resistance to data alterations.

3. They enable a high level of data transmission performance in distributed networks.

4. They are computationally inexpensive and efficient when creating, processing and verifying information.

5. They allow “dissection” to make verification searches faster. All this, without compromising the security and traceability of the transactions carried out.

6. Thanks to the “dissection” feature they are also able to save storage resources.

Although Merkle trees are a powerful tool for data verification, they alone do not guarantee data security. Parallel processes are required to verify that the data is secure.

- Payment Channels

It is a typology of techniques designed to allow users to make multiple Bitcoin transactions without compromising all transactions on the Bitcoin blockchain. In the standard model only two transactions are added to the blockchain, but an unlimited or almost unlimited number of payments can be made between participants.

Payment channels are transactions on the Bitcoin network that lock funds based on 2 of 2 multisignatures. Payment channels are a trust-less mechanism for exchanging bitcoin transactions between two parties outside of the bitcoin blockchain.

- Multiple signatures (Multisignature transaction)

Multi-signature technology refers to requiring multiple keys to authorize a Bitcoin transaction, rather than a single signature of one key. It has several applications.

Multi-signature has been used for thousands of years to protect the security of crypts containing the most precious relics of saints. The superior of a monastery would only give monks partial keys to access relics. Therefore, no monk could access and possibly steal the relics.

Public key encryption, also known as asymmetric cryptography, is a key technology of blockchains. Chain participants generate their own pairs of private keys and public addresses. They keep the private keys secret, but freely distribute the associated addresses. A blockchain transaction that performs an action for a particular address must be signed by the corresponding private key. All participants in the chain can verify these signatures, using only public addresses, without needing to see each other's private keys.

Standard transactions on the Bitcoin network could be called "single-signature transactions" because transfers require only one signature: from the owner of the private key associated with the Bitcoin address. However, the Bitcoin network supports much more complicated transactions that require the signature of multiple people before funds can be transferred. These are often called “m of n” transactions. Any m private keys out of “n” possible are required to move the money. For example, a “2 of 3” multisignature transaction may have its private keys distributed across a desktop computer,

a laptop, and a smartphone, two of which are needed to move the money, but the compromise of one key cannot end. in a robbery.

Multiple signature is a digital signature scheme that allows a group of users to sign a single document. Typically, a multisignature algorithm produces a joint signature that is more compact than a collection of distinct signatures from all users.

The idea is that Bitcoins are taxed by providing addresses from multiple parties, so the cooperation of those parties is required to do anything with them. These parties can be people, institutions or programmed scripts.

It allows increasing the security of transactions.

- Solidity

Solidity, a kind of mix between JavaScript, Python and also C++, specially designed to create smart contracts.

3.3.12 Terminology

- Transactions

Transactions are the leitmotiv and reason for being of Blockchain. All other parts are built to ensure that transactions are created correctly, propagated on the network (P2P), verified and added to the Blockchain. Transactions are data structures that store transfers of “value” between system participants. Each transaction is stored as an entry on the Blockchain.

- Digital evidence

In the broadest definition of the term, we understand by digital evidence all types of information in digital format that serves as testimony or evidence in a judicial process to relate a crime to its perpetrator and/or its victim. In short, any digital data that serves as evidence in a trial. From an email, an image, a video, a snapshot of a web page, to information stored on a hard drive, a mobile phone, a server, and many other variants.

- Smart contracts

Cryptologist Nick Szabo was the first to think of computer protocols that would allow electronic commerce between strangers and would replace legal paperwork. Today a smart contract refers to a contract that executes itself without third parties and is written as a computer program instead of using a printed document with legal language.

Smart contracts are one of the most important parts of any Blockchain as a Service solution. Mainly because they are a great way to enforce an agreement between the parties. Smart contracts are a special type of digital contract that directly controls the transfer of assets or documents between users under certain rules. The smart contract not only defines the rules like typical contracts, but also imposes penalties in case any rules are broken. However, the term smart contract is a bit of a stretch because they are neither legal nor smart contacts.

In 'smart contracts', computers play an active role. It is not only about electronically storing documentation or allowing electronic signatures, as has been done until now, but these programs perform analysis and execute some of the parts of their internal logic.

The program can define strict rules and consequences in the same way that a traditional legal document would, but unlike traditional contracts, it can also take information as input, process it according to the rules established in the contract, and take any action that is necessary. required as a result.

- Differences between a 'token' and a cryptocurrency

A digital 'token' is a unit of value based on cryptography and blockchain, issued by a private entity so that it has a specific functionality in the digital world, with the value that the entity establishes. A 'token' represents a utility or a digital asset that can have very different purposes.

A cryptocurrency is a 'token' whose main purpose is to serve as a means of decentralized payment for products and services in the virtual environment.

Another way to distinguish them is by the properties that usually characterize cryptocurrencies, which 'tokens' do not have to meet: being fungible, divisible and portable and having a limited number.

In general, it can be said that the 'token' is at the foundation of all digital transactions, from the simplest to the most complex, while the cryptocurrency is a 'token' oriented to a specific use (BBVA creative).

- DApp (Decentralized Application)

Platforms that allow any interaction between their members, from the web or through a mobile 'app', without the need for a central agent to manage this service or to keep track of each of the records and actions carried out.

Each of the users of the same DApp is a node of a decentralized network in which all members act jointly as a collective notary of any movement that is made on that platform.

No one must expressly give their consent, but everything works automatically and the system itself is responsible for corroborating the validity of each interaction through a smart contract.

3.4 (Secure) Log File Management for IoT Devices

3.4.1 The DistLog System

According to [1], modern IoT attacks can employ anti-forensic techniques to destroy or modify evidence, such as log files, which will complicate investigations that could help identify the attack source. In order to ensure the availability and integrity of log files, they proposed a distributed logging scheme. Using the Modified Information Dispersal Algorithm (MIDA), log files generated by IoT devices are aggregated, compressed and encrypted, followed by fragmentation, authentication and distribution over n storage nodes. Availability of the fragmented log files is guaranteed with a degree of $(n - t)$, n being the total number of fragments and t being the number of fragments needed for data reconstruction. The values of n and t set the levels of availability as well as security and can be changed to fit the number of available nodes. Increasing n increases both communication and storage overhead.

The network in the proposed solution consists of N interconnected devices where some of the devices act as storage or aggregation nodes. The MDIA solution is applied to logs collected during a session which can be configured to suit the target system's needs. According to the security analysis performed by the authors, the solution exhibits sufficient security against statistical attacks, brute-force attacks, plain/cipher-text attacks as well as linear and differential attacks, providing great overall security, more specifically to availability, integrity and confidentiality. This allows for more robust investigations after security incidents since log files often play a central role in digital investigations. Furthermore, some intrusion detection systems can use log files to determine if the system has been compromised, giving alerts to system administrators. Log files could also be used to detect anomalies in the system, one of the goals of the CISSAN project.

The computational difficulty for the solution is low, exhibiting linear computational complexity. The encryption scheme also requires only a few operations, and as the authors mentioned, look-up tables can be used to speed up the process. Furthermore, the MIDA process can be sped up with parallel computation, if available. Since log collection is done periodically, with the period being configurable, the time needed for execution should not be an obstacle for use.

Some open questions are how well the solution works in an ad-hoc network and how much computational resources the reverse MIDA process will take during system operation. In addition, if enough storage nodes are lost, the logs can no longer be retrieved. Finally, aggregation may cause information to be lost, though this depends on the chosen scheme. Nevertheless, the proposed solution exhibits traits that make it beneficial for CISSAN, more specifically to WP5 tasks 5.2 and 5.3. Since multiple nodes on the network can perform aggregation, the network won't have a single point of failure for securing logs. Distributing the compressed log files among the network nodes also allows for efficient use of limited resources. Of special note are how only a fraction of the distributed fragments are needed to reconstruct the log data, meaning that some storage nodes on the network can

be lost without affecting the availability of the data. Access to logs by malicious actors is also unlikely since they would need all the fragments for reconstruction.

3.4.2 Probabilistic Logging

Byzantine nodes can negatively affect the performance of an IoT network by acting outside their intended purpose, due to hardware/software issues or due to malicious actors. Having full logs on the events of the network despite this would allow better auditing and anomaly detection. To solve this problem, [2] presented a logger selection method and a weak probabilistic method for logging messages. The solution assumes that the network includes a sink node is needed, that the network is $k + 1$ connected, where k is the number of Byzantine nodes, with a maximum of $(n - 1)/3$ Byzantine nodes (n being the total number of nodes in the network). Since the solution relies on clustering the network, Byzantine nodes are also expected to be uniformly distributed across the network.

Logger selection is solved by using a clustering algorithm that assigns each node to a single logger. This ensures that there aren't too many loggers which would scatter witness data, and that messages don't have to travel too far, which would deplete energy from nodes. The following is a brief description of the probabilistic solution with weak validity for logging messages, using the receiver-oriented algorithm which exhibited the best performance among their explored solutions.

When an event is generated and a related data message is sent to the sink, the neighbor devices of the sender send "witness" messages to the logger, confirming the data message. These control messages don't interfere with other messages (e.g. application messages) and have the necessary information to identify the sender and the event the control message refers to. If the number of confirmations exceeds the number of Byzantine nodes in the cluster, the control message is logged. The solution guarantees the validity of the messages, and that the logger will log either a control message or a fault message with a probability of p.

The clustering used in the solution allows the devices to save energy since control messages do not need to travel far across the network, which will extend the nodes' operation time, something the CISSAN project is interested in. Since confirmation messages need to exceed the number of Byzantine nodes, incorrectly functioning devices will not be able to forge logs easily, which would affect anomaly detection, another point of interest for the CISSAN project. Accurate logs would also help during investigations following security incidents.

The solution is not particularly taxing when it comes to computation and unless enough nodes leave or join the network, the clusters will not need to be reconfigured. With witnesses and control messages confined to their own clusters, the logging of activities is distributed across the network limiting network overhead. While the solution was not able to achieve 100% logging rate during simulations, the authors pointed out that as the size of the network rises, so should the number of correctly logged messages. By developing the solution further, a network size where full logging rate could be achieved may be possible. Even if this turns out to be unachievable, a system that prevents individual nodes from creating false logs due to the requirement of collective agreement is still desirable for ad-hoc networks where nodes may join and leave a network freely.

3.4.3 The Transparent IoT (T-IoT) Framework

[3] pointed out that in IoT platforms, the audit and diagnosis of trigger-action-based event chains (e.g. a camera detects movement which leads to a light getting activated) are both untrustworthy and unreliable due to device-centric logging mechanisms as well as the heterogeneity and vulnerability of the IoT devices, which can cause a lack of causal dependencies between events. An example given by the authors where such a lack of event ordering may cause issues is a smart health platform, where identifying the root cause of a patient going critical is imperative. To solve this issue, they proposed the Transparent IoT (T-IoT) framework, where causal relationships between devices become transparent and tamper-proof via a Blockchain protocol and a vector clock system, both tailored for IoT devices with resource constraints. The network model for the framework consists of IoT devices connected to a gateway, a computationally powerful device that is connected to the Cloud, which acts as a data storage for applications of the users. The gateway and the Cloud are assumed to be secure.

In the framework, a distributed ledger of events is generated by the IoT nodes, acting as miners and adding events to the ledger as blocks of events. The gateway manages the IoT nodes, allowing action events to be registered in a block only if the triggering events of that action are already registered in the ledger. Each node saves only a portion of the ledger, replacing it over time using the proposed partial consistent cut replacement policy, while the gateway saves the entire ledger. Event ordering runs parallel to the registration process and is done by logically synchronizing the nodes by extending Lamport's logical clock system to vector clocks. The gateway generates one proof of work puzzle for each event so that each event becomes tamper-proof and so that no miner can dominate the proof of work protocol. Block validation follows the Nakamoto consensus.

The authors implemented the framework with an IoT gateway and 30 devices. Experiments demonstrated that the average storage size, latency and energy expenditure per node for the Blockchain protocol used make the solution viable for IoT applications, with smart homes and traffic monitoring mentioned as examples. However, the viability of the solution in systems with larger amounts of nodes is yet to be confirmed. Considering the subject matter of resource constrained IoT devices in the CISSAN project, the T-IoT shows promise in being a viable option for logging with a low energy requirement via blockchain. In addition, low energy consumption would extend the battery life of a node, leaving more resources for collective intelligence functions. Having tamper proof records of events that have been ordered should also serve anomaly detection well, since causal dependencies between events are clearly visible. Lastly, one of the goals of CISSAN is the development of Blockchain based security solutions for IoT networks and the solution presented in the paper could potentially be used in other aspects of the project than just secure logging.

3.4.4 The LogSafe System

Due to the resource constraints of IoT devices, collecting logs at the device level for storage may not be feasible. [4] noted that while cloud services can provide the necessary storage for large-scale data logging, they are still vulnerable to security breaches in addition to not having the precise architecture known making security evaluations difficult. In order to improve logging for IoT systems on untrusted cloud infrastructure, [4] proposed a logger utilizing the security guarantees of Intel Software Guard Extensions (SGX) to provide confidentiality, integrity and availability. According to the authors SGX was originally designed for executing software applications on remote computers owned by untrusted parties and is comprised of a new set of instructions and memory access changes that allow the creation of enclaves, isolated and encrypted execution environments.

The LogSafe architecture consists of IoT devices, Loggers, a Manager (a local trusted platform), Trackers and the database for long term storage. Consistent hashing is used to assign IoT devices to Loggers. Logger nodes in LogSafe are organized in a decentralized cluster that can be seen as an n-node ring (n being the maximum number of nodes at a time) where IoT devices are assigned to a specific Logger and each primary Logger replicates their state on backup nodes. In case of machine failures, responsibilities are transferred to a backup machine.

High level data flow is as follows. An IoT device establishes a secure TLS connection with the Logger. If the Logger has not done so, it verifies its identity with the Manager via SGX remote attestation and provisions for the IoT device's meta-data (e.g. root certificate, device specific encryption key). Next the IoT device sends the data to the Logger which goes through the encryption and backup process. A snapshot of the Logger's counter values, a monotonically increasing sequence number for logs, is created and saved on a Tracker, ensuring that logged data can be verified in case of system topology changes. Finally, the encrypted data is stored in the database cluster.

The solution was tested by simulating three different setups (1-3 Logger nodes) with 1 to 256 IoT devices with the 3 Logger setup outperforming the others. Though remote attestation and snapshot creation took time, the decrease in performance was considered negligible due to them occurring rarely. According to the authors of [4], assuming the IoT devices send data every 1 second, the three-node setup should be able to support up to 5000 devices.

Apart from the IoT devices and the Manager, all the rest of the devices are situated on the cloud. This provides a research opportunity for evaluating the solution in a scenario where the Loggers are moved to the edge of the network. Overall, the proposed solution shows promise in securing log files in IoT environments. Having multiple loggers managing the IoT devices should help relieve the network overhead that would occur with only one logger device, relating to task 5.2 of the project. The

solution also provides means of handling loggers leaving the network, something that can easily happen in ad-hoc networks that the CISSAN project is interested in. The authors also noted that the enclaves created by SGX should maintain confidentiality even when the device is physically controlled by an attacker. Since the writing of the paper, Intel has deprecated SGX from their 11th and 12th generation Intel Core processors onwards, though development of SGX continues on some lines of Intel Xeon for cloud and enterprise use, meaning that other means for creating enclaves may need to be found.

3.4.5 The DASLog System

[5] noted that in addition to the potential challenges of creating secure logs for IoT devices, there is also a need for security proofs so that external parties can trust the data retrieved is correct if there is a need for public auditing, for example in areas where the government needs to inspect whether regulations are being followed. To address this, the authors designed a secure logging scheme for use in aerial transport via drones providing public auditability of records where all logging system components are controlled by the operator of the system and presented two solutions, DASLog and Simple-DASLog, with Simple-DASLog being the more simple and less secure one. Noted that while the case presented is related to UAVs, the concept itself is applicable to any IoT system with a single controller for all the logging components.

DASLog and Simple-DASLog both consist of four types of entities, 1) the central logging system managed by the operator, 2) data sources, i.e. entities generating logging records 3) data consumers, i.e., entities requesting logged records from the central logging system and 4) the 3rd party register (BFT-based private blockchain) containing proofs for logging records on the central logging system.

The following is a description for the DASLog solution. Using secure TLS connections, logging records are encrypted and stored in the logging database with authentication needed to access them. Each individual record is signed digitally, and a set of proofs are generated from the records in the form of hash chains. Since logging records are signed with a private key, the data sources cannot deny signing the records. The generated proofs are used to create a Merkle tree with the proof at the root stored in the blockchain. The BFT- based private blockchain used ensures that the proofs cannot be deleted or altered. Proofs in the blockchain are used to verify that the records are correct and complete. Simple-DASLog is largely similar to DASLog, with a few security functions missing either altered or missing, e.g. no digital signing with private keys.

Evaluation of performance and demonstration of practicality was done by replicating an unmanned aerial vehicle (UAV) ecosystem on Amazon Elastic Compute Cloud (EC2) instances. DASLog was able to create proofs faster than Simple-DSALog. Verification of logs took approximately the same time for both solutions but was faster with Simple-DASLog if the logs had not been tampered with. Based on the security analysis in the paper, DASLog provides excellent security for logs. The immutability of proofs would guarantee that any changes in logging records could be easily detected allowing forensic investigations to be focused on accurate data improving incident response. The implementation of a private blockchain in an IoT environment is also directly related to task 4.3 of the project. Due to the resources employed in the demonstration (Amazon EC2), the solutions would have to be tested in a different environment to see how it fits general use. While the requirement of a single system operator and having a centralized database may make the solution unviable for systems where devices can join and leave the network at will and a network that may not always have access to the Internet, DASLog could fit the use cases of the CISSAN project. This would not only allow to test the solution in different environments but also with different device types, since UAVs have better computational capabilities than basic sensors, which would shed light on the generalizability of the solution.

3.4.6 Optimal Data Witnessing

According to [6], IoT sensors are generally not aware of attackers in the middle of communication which leaves them open to attacks where the goal is to manipulate data during transit. To address this problem, they developed a system architecture based on Ethereum blockchain for an on-demand auditing of wearable IoT device data in the realm of healthcare IoT by using witnesses. The primary

focus of the solution was on the verification of data by collecting witness statements that can be used to determine whether data has been tampered with during transit.

In the case presented in the paper, the system architecture consists of the healthcare service provider (HSP), wearable IoT sensors, witnesses, the blockchain network and smart contracts. The HSP provides patients with services based on the received data which is provided by the IoT sensors. Witnesses are wireless nodes located near the sensors sharing the same wireless broadcast domain. They are assumed to have sufficient computational resources to interact with the blockchain and to be able to overhear data packets sent by the sensors. Witness statements are collected in the blockchain. Smart contracts run as distributed applications on top of the blockchain providing the necessary functions for the witnessing system.

In the system, sensors transmit their data via their gateway to the server. Witnesses residing in the same local network can overhear the sent data packets. Once witnesses receive a witness request from the HSP, they check if they are in the vicinity of the target device. Witness nodes in the vicinity offer their services based on their resources to the HSP, which selects a set of witnesses (zero, one or more) based on selection criteria and announces them on the blockchain. After this, the witnesses generate witness statements, in the form of bloom filters with a fixed size, on the blockchain based on the overheard packets. The HSP can then check if the received packets are present in the bloom filter, acting as a guarantee that the data packets have not been altered during transmission.

While the solution presented by [6] is not a secure logging solution in and of itself, it is still related to secure logging. The encryption and secure storage of sent data means little if the data has been false from the get-go. Having witness statements that can be checked for data packet tampering can be used during forensic investigations to narrow the investigation area away from the network traffic. Witness statements could also have potential use in anomaly detection since they could rule out certain attacks and be used to detect compromised witness devices as well. If transmitted data is found out to be compromised yet the witness statements claim that nothing of the sort has happened, it would likely mean that the witness device itself is acting in an anomalous manner. Furthermore, the way the HSP requests and selects witness nodes based on their resources can be relevant to WP5 task 5.3 of optimally distributing security functions. Though the presented solution concerns itself with the healthcare environment, the architecture should be applicable in other areas as long as the requirements of a central server, a blockchain network and a wireless network with both IoT devices and witnesses are satisfied, making the solution an interesting option for the use cases of CISSAN.

Since the authors were focused on optimizing the monetary costs of using Ethereum blockchain for storing the statements and the evaluation of verification errors due to the inherent false positives of using bloom filters, less focus was gives to the computational requirements of the witness devices. Examples such as smart phones and tablets were mentioned which would imply that regular IoT devices may not be sufficient for witness statement generation. One of the first steps to be taken should therefore be determining the minimum computational requirements.

3.4.7 The LogStack System

The use of Named Data Networking (NDN) simplifies programming and use of Internet applications with the trade-off of comprehensive infrastructure of the traditional Internet (Hail et al., 2022). The increased use of IoT devices also places more demands on their reliability which is impacted by their resource constrained nature. To enable the use of IoT devices in NDN, [7] proposed the Smart Logging Stack (SLS) for recording IoT device activities. To accomplish this, the authors extend the Data Plane of the IoT-NDN containing the caching and forwarding strategies, with the SLS table, an additional table which caches logged messages. All messages have four parameters, 1) log level, which assigns a priority value to the message, e.g. messages delivering general device information or messages detailing fatal errors, 2) scope, which defines the hop number, i.e. how far the log message should be forwarded within the network (e.g. cached in the SLS table, forwarded to all devices in the network), 3) age functioning as a counter that is incremented after every new log message, and 4) send status, showing whether the log message has been sent or not. Following the name structure of NDN, requesting log messages should be simple.

The development of the proposed solution is still in the early stages, but initial tests have been performed by the authors on an ESP32 developing board and an application for switching device LEDs.

While the use cases of CISSAN do not use NDN, the solution could be used for inspiration for a logging solution. Since IoT devices are resource constrained, having a logging system where a portion of the logs are saved locally, and the rest are sent for storage elsewhere based on the log message type (e.g. sensor recording vs fatal error) could be a viable solution for distributing logs across the network. With the correct setup, log messages could be immediately sent to those devices that need them the most, perhaps to be used in collective decision making. How viable this is, depends on the required computational capabilities needed to implement such a system. Unfortunately, the authors did not provide specific information of this, which means that such a system would need to be tested at a larger scale, something that would also provide an opportunity to experiment with other aspects of the CISSAN project in an NDN.

3.4.8 Integrity Verification

In IoT ecosystems the collection, processing and forwarding of data is often conducted on the edge of the network by edge devices [8]. However, deploying edge devices in outdoor environments will leave the devices vulnerable to not only electronical access by adversaries, but also physical access. This would allow attackers to extract information. modify programming of the devices or even replace the devices with nodes of their own. The proposed logging scheme by [8] deals with the problem of validating a streaming IoT application in a hostile environment. The solution assumes a standard edge-cloud setting with basic nodes sensing and generating observations while secure nodes verify computations, and a central auditor periodically verifies data consistency. Basic and secure nodes are connected to the Internet through routers with there being significantly less secure nodes than basic nodes. A setup for the distribution of long-term keys and for the registration of sensor identities is required.

The full scheme contains three mechanisms, Time-out Detection to check for lost or stolen basic nodes by checking if messages are received within pre-estimated latency, Code Validation to detect premature code execution and Integrity Protocol to deal with overwriting of post-computation output. Algorithms for Code Validation as well as the Merkle Hash Tree and auditing portions of the Integrity Protocol are provided in full in the paper. The phases of the Integrity Protocol are presented in more detail here, since it is the most relevant mechanism in relation to secure logging in the CISSAN project.

The Integrity Protocol is divided into four phases, 1) output generation, 2) output collection, 3) Merkle Tree building and 4) verification. In the first phase, basic nodes execute their computations based on inputs, producing outputs and creating hash-chains for them. During output collection the secure nodes collect the hash-chain ends from the basic nodes. After this the secure nodes create and manage output logs by forming a bottom-up Merkle Hash Tree making the verification overheads constant. Lastly, the auditor reconstructs the Merkle Hash Tree by repeating the previous steps locally and comparing the produced Merkle Hash Trees verifying the output validity.

The NS-3 simulator was used to test the solution in three different scenarios with cost of verification compared to a straightforward baseline where basic nodes push outputs directly to the secure node and results are compared without hash-chains and the Merkle Hash Tree. The topology contained three basic nodes and one secure node. The latency costs for verification dropped between 11% and 34% depending on the scenario.

Secure logging is often focused on methods that prevent malicious actors from accessing logs. The solution presented by [8] focuses more on verification instead by securely storing a Merkle Tree containing hashes that can be used to verify the actual logs. Based on the description in the paper, the solution has potential for providing the CISSAN project a way to detect log anomalies caused by tampering more efficiently. By using multiple secure nodes for forming the Merkle Hash Trees allows for distribution of computation across the network decreasing the burden placed on each node. However, IoT devices are often resource constrained and therefore may not be able to function as fully secure nodes. This could make the solution unviable for some scenarios. Testing the viability of the solution by replacing secure nodes with collective functions could help determine whether the solution can be applied to more general IoT networks.

3.4.9 References

[1] Noura, H. N., Salman, O., Chehab, A., & Couturier, R. (2020). DistLog: A distributed logging scheme for IoT forensics. *Ad Hoc Networks, 98.* https://doi.org/10.1016/j.adhoc.2019.102061

[2] Alhajaili, S., & Jhumka, A. (2021). Reliable Logging in Wireless IoT Networks in the Presence of Byzantine Faults. 2021 IEEE 20th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), 17-25. https://doi.org/10.1109/TrustCom53373.2021.00021

[3] Rahman, M., & Saifullah, A. (2023). Transparent and Tamper-Proof Event Ordering in the Internet of Things Platforms. IEEE Internet of Things Journal, 10(6), 5335-5348. https://doi.org/10.1109/JIOT.2022.3222450

[4] Nguyen, H., Ivanov, R., Phan, L. T. X., Sokolsky, O., Weimer, J., & Lee, I. (2018.) LogSafe: Secure and Scalable Data Logger for IoT Devices. IEEE/ACM Third International Conference on Internet-of-Things Design and Implementation (IoTDI), 141-152. https://doi.org/10.1109/IoTDI.2018.00023

[5] Sarenche, R., Aghili, F., Yoshizawa, T., & Singelée, D. (2023). DASLog: Decentralized Auditable Secure Logging for UAV Ecosystems. IEEE Internet of Things Journal, 10(23), 20264-20284. https://doi.org/10.1109/JIOT.2023.3281263

[6] Chinaei, M. H., Gharakheili, H. H., & Sivaraman, V. (2021). Optimal Witnessing of Healthcare IoT Data Using Blockchain Logging Contract. IEEE Internet of Things Journal, 8(12), 10117-10130. https://doi.org/10.1109/JIOT.2021.3051433

[7] Hail, M. A. M., Dietrich, L., & Fischer, S. (2022). LogStack: A Smart Logging Stack Approach for IoT Devices based NDN (IoT-NDN). International Conference on Software, Telecommunications and Computer Networks (SoftCOM). https://doi.org/10.23919/SoftCOM55329.2022.9911233

[8] Lou, S., Panwar, N., & Agrawal, G. (2021). Integrity Verification for Streaming IoT Applications with a Minimalist Logging Scheme. IEEE International Conference on Smart Computing (SMARTCOMP), 197-202. https://doi.org/10.1109/SMARTCOMP52413.2021.00047

3.5 Generative AI for Cybersecurity in IoT Networks

In this section, we review several recent papers discussing generative AI applications to intrusion detection and threat hunting in IoT networks and in other potentially relevant contexts and cybersecurity tasks. Since this is a truly vibrant research domain at the time of writing, our goal here is to briefly introduce the current capabilities of quickly evolving generative AI-based models and certain research directions that can influence CISSAN plans.

Various approaches and solutions – both non-AI and AI-based – proposed for addressing the problem of IoT intrusion detection are presented in a comprehensive survey by Arisdakessian et al. in [1]. They categorize intrusion detection approaches into four high-level classes and then further into subclasses. The classes, along with their advantages and disadvantages, are shown in Table 4. The authors mention that traditional IDS mechanisms are often unsuitable for IoT environments. The challenges usually relate to the heterogenous nature of IoT devices, their limited resources, and diverse communication protocols. We note that the IoT cybersecurity framework proposed in [1] has explainable artificial intelligence (XAI) as a key element, and the use of Large Language Models (LLMs) for explainability can be explored in CISSAN.

Table 4: Advantages and disadvantages of different intrusion detection approaches in IoT networks (Arisdakessian et al., 2023)

Category	Approach	Advantages	Disadvantages
Trust	Direct Trust	• Information can contain objective sources such as historical trust values which can assist in getting more accurate information about the environment • Objective trust feedback can be used in aggregation methods combined with subjective trust feedback to avoid biased results and combat slandering and promotion attacks	• The information acquired might be out of date since only subjective trust values are taken • Subjective trust is prone to collusion attacks such as slandering and promotion, where several devices collude to provide negative feedback about other devices for malicious purposes • In larger environments, the reputation of the entity needs to also be considered to avoid collusion attacks which increases the complexity of the solution
Trust	Indirect Trust	• Information acquired is non-biased because it takes the trust from direct neighbors as well as from neighbors of neighbors • Easy to acquire accurate information about neighbor using single hopping in the form of subjective-trust	• Excessive resource overhead (storage, computation) due to the fact that some devices (for example, a central server) has to keep historical data and run additional computations to calculate subjective scores and compare with past results or perform regression analysis
Math	Statistical	• Dynamic which is an advantage in IoT environments where devices enter and leave the network frequently	• Might need to gather additional information from the devices in the network such as resource availability, latency, network capacity, which can cause delays and overheads
Math	Game Theory	• Takes into account the strategies and preferences of both entities in the network, i.e., the attacker and the defender	• For the system to apply the game, additional information need to be gathered. • The game needs to be formulated and played which can add layers of complexity and cause overhead in the system
Math	Graph Based	• Provides a visual and communicative connection between the devices in the network which can assist in finding other devices and extracting information about them such as their reliability and trust	• In dynamic systems such as IoT based systems, every time a device leaves the network, or another one joins, the graph needs to be changed. Similarly, if there are calculations relying on the graph, they need to be recalculated after each change.
Math	MOO	• Works directly on any population without any assumptions • Can provide more than one-single solution for the underlying problem	• Can be time-consuming which might not be adequate for mission-critical systems • Constraints need to be collected pre-hand, thus the solution might not scale and be dynamic
Data	Centralized	• Full control over the model by the central authority, thus the model can be easily tuned and edited upon need	• Provides a single point of failure to the system. If the central authority is attacked, the entire system collapses
Data	Distributed	• Fixes the problem of single point of failure	• More prone to internal attacks, since devices in the distributed system can collude together and turn malicious • In case of updating the detection system, more than one entity would need to be provided with the new updates and models. This can be hard given the network and latency constraints for IoT based environments.
Data	Ensemble	• Uses a combination of several classifiers to form a robust attack detection mechanism	• Extra steps and additional algorithms are needed to be applied to work through several classifiers to come up with the ensemble classifier which is time-consuming and adds overhead to the system
Data	Federated	• Preserves the privacy of the data instead of sharing them into entities outside the control of the end-user or the system-owner	• The central server presents a single point of failure • Prone to poisoning attacks for the global model which decreases its accuracy • Challenge of non-IID intrusion data since different devices can have different sizes and classes
Blockchain	Smart Contracts	• Works autonomously based on predefined rules without any human intervention • Everything is transparent on the blockchain where everyone can access and see all the information	• If a weakness is detected in the contract, it would be highly abused and because information on the blockchain is immutable and cannot be altered • Would need the intervention of humans to edit the code of the contract and fix the error

The survey by Alwahedi et al. [2] fully focuses on the use of Machine Learning techniques for cyber threat detection in IoT environments. It includes a comparative analysis of state-of-the-art ML-based Intrusion Detection Systems (IDSs) for IoT and discusses unresolved issues and challenges. The authors express their belief that generative AI and LLMs promise “a more secure, intelligent, and adaptive approach to protecting the ever-expanding universe of IoT devices and networks”. They expect that LLMs will revolutionize threat detection in IoT due to their ability to analyze unstructured data from various sources including IoT device logs and network traffic. They envision other generative AI applications in IoT contexts, e.g., to smart contracts, access control, vulnerability detection, and penetration testing.

Ferrag et al. propose in [3] an LLM-based cyber threat detection architecture, called SecurityBERT (Bidirectional Encoder Representations from Transformers), for IoT networks. According to the authors, this model is suitable for real-world traffic analysis in IoT networks because it is efficient and lightweight: it has an inference time of less than 0.15 seconds and a model size of 16.7 MB. The model is based on a 15-layered BERT-based architecture with 11 million parameters and achieved 98.2% overall accuracy in identifying distinct attack types (real-world network traffic dataset called Edge-IIoTset was used for training and testing). One part of the architecture was a privacy-preserving encoding technique, Privacy-Preserving Fixed-Length Encoding (PPFLE). Apart from protecting sensitive network traffic data, the PPFLE converts unstructured traffic data to a format somewhat similar to the English language, and thus BERT models understand it better.

In [4], Zhang et al. introduce a framework for LLM-based intrusion detection in the field of wireless communication. They propose four main steps in their process: i) selecting the most relevant network features using LLM, ii) collecting and processing data to send to LLM as input, iii) building prompts for LLM, and iv) extracting decisions from LLM output. They experiment with different models, including GPT-3.5, GPT-4 (which showed the best performance with only ten in-context examples) and Llama-2-7b, and a real-world network intrusion detection dataset, and they compare their framework with a CNN-based network intrusion detection model. The authors designed and compared three in-context learning methods for OpenAI’s models: illustrative, heuristic and interactive in-context learning. These methods bring good performance with only a few learning examples while fine-tuning normally requires many more examples. The authors also identified challenges and risks with LLM-based approaches, such as adversarial prompting, hallucinations, and stochastic output.

Ali and Kostakos introduce in [5] a prototype, called HuntGPT, to analyze network traffic, focusing on explainable anomaly detection complemented by actionable recommendations. Their model has three layers and two systems. The layers, from lower to higher, are analytics engine (for anomaly detection), data storage, and user interface. The systems are an Anomaly Detection Application server and an Intrusion Detection System Dashboard (a screenshot of which can be seen in Figure 8), with the latter explaining detected anomalies (applying LIME and SHAP techniques) and suggesting appropriate courses of action (connecting to OpenAI models, e.g., GPT-3.5-turbo). While the work does not specifically address IoT security challenges, the focus on explainability and the use of LLMs for providing actionable recommendations is of high relevance to CISSAN.

Figure 8: The dashboard of HuntGPT (Ali & Kostakos, 2023)

In addition to intrusion detection and response, LLMs can be used, e.g., for code-level vulnerability detection (see, e.g., SecureFalcon LLM presented in [6]), planning and carrying out parts of penetration testing (see, e.g., examples in [7]), and other security-related tasks. A general review of LLM applications to cybersecurity can be found in [8]. Of course, one also has to remember about challenges which the use of LLMs, and more generally AI, can be connected to. Fui-Hoon Nah et al. discuss in [9] such examples as:

- Technological (hallucinations, quality of training data, explainability, authenticity, prompt engineering)
- Ethical (harmful or inappropriate content, bias, over-reliance, misuse, privacy, security, digital divide)
- Regulation and policy compliance (copyrights, AI governance)

Generative AI is not limited to LLMs, and we also see applications of GANs and other generative models to intrusion detection and threat hunting. Abdalgawad et al. propose in [10] three generative deep learning-based models to detect various cyberattacks in IoT networks. Using the labeled IoT-23 dataset, which contains network traffic obtained from IoT devices and includes 20 malware captures and 3 benign captures, the authors trained an Adversarial Autoencoder (AAE) model and two Bidirectional Generative Adversarial Networks (BiGAN). The AAE model achieved the F1-Score of 0.97 and one of the BiGAN models achieved the F1-Score of 0.974 in detecting attacks of known types, while the other BiGAN model detected unknown (Zero-Day) attacks with the F1-score between 0.85 and 1.

Ferrag et al. presented in [11] a GAN- and Transformer-based model for Cyber Threat Hunting in 6G-enabled IoT Networks. They utilized the Edge-IIoT dataset which comprises one class of normal traffic and 14 attack classes, with the data sourced from over ten types of IoT devices. The objective of the GAN in the proposed model was to increase the gap between the generated IoT data and the actual IoT data. The model achieved the accuracy of around 95% in the conducted tests. The authors

also discussed the resilience of GANs to adversarial examples and several challenges regarding the use of generative AI in IoT networks, including scalability, decentralized training, data quality, energy consumption, privacy preservation, and tokenization.

In [12], Xiong et al. propose a federated learning-based generative model as a solution to challenges posed by multiple heterogeneous data sources (such as devices in an IoT network) in intrusion detection. They introduce a three-layered hierarchical framework (device, edge server, cloud) for deploying a federated generative model and consider two scenarios: a feature-related scenario, where data from different communities have the same features but different labels, and a label-related scenario, where data from different communities have the same labels but different features. The model can learn a powerful generator for hierarchical IoT systems, which is potentially of relevance in CISSAN T4.2 and methodologically complements or extends the approaches considered in Section 3.2 above. Particularly, the generative model framework can solve the problem of distributed data generation on multi-source heterogeneous data. Additionally, the framework addresses privacy and communication cost concerns by eliminating the need to send data to a central server.

3.5.1 References

[1] Arisdakessian, S., Wahab, O. A., Mourad, A., Otrok, H. & Guizani, M. (2023) A Survey on IoT Intrusion Detection: Federated Learning, Game Theory, Social Psychology, and Explainable AI as Future Directions. IEEE internet of things journal, 10(5), pp. 4059-4092. doi:10.1109/JIOT.2022.3203249

[2] Alwahedi, F., Aldhaheri, A., Ferrag, M. A., Battah, A., & Tihanyi, N. (2024) Machine learning techniques for IoT security: Current research and future vision with generative AI and large language models. Internet of Things and Cyber-Physical Systems, 4, 167–185. https://doi.org/10.1016/j.iotcps.2023.12.003

[3] Ferrag, M. A., Ndhlovu, M., Tihanyi, N., Cordeiro, L. C., Debbah, M., Lestable, T. & Thandi, N. S. 2024. Revolutionizing Cyber Threat Detection with Large Language Models: A privacy-preserving BERT-based Lightweight Model for IoT/IIoT Devices. IEEE access, 12, p. 1. doi:10.1109/ACCESS.2024.3363469

[4] Zhang et al. (2024b) / Zhang, H., Sediq, A. B., Afana, A. & Erol-Kantarci, M. (2024b) Large Language Models in Wireless Application Design: In-Context Learning-enhanced Automatic Network Intrusion Detection. arXiv.org. doi:10.48550/arxiv.2405.11002

[5] Ali, T. & Kostakos, P. (2023) HuntGPT: Integrating Machine Learning-Based Anomaly Detection and Explainable AI with Large Language Models (LLMs). arXiv.org. doi:10.48550/arxiv.2309.16021

[6] Ferrag, M. A., Battah, A., Tihanyi, N., Debbah, M., Lestable, T., & Cordeiro, L. C. 2023. SecureFalcon: The Next Cyber Reasoning System for Cyber Security. ArXiv.Org. https://doi.org/10.48550/arxiv.2307.06616

[7] Happe, A., & Cito, J. 2023. Getting pwn'd by AI: Penetration Testing with Large Language Models. ArXiv.Org. https://doi.org/10.48550/arxiv.2308.00121

[8] Zhang, J., Bu, H., Wen, H., Chen, Y., Li, L., Zhu, H. (2024a) When LLMs meet Cybersecurity: A Systematic literature review. doi: arXiv:2405.03644v1

[9] Fui-Hoon Nah, F., Zheng, R., Cai, J., Siau, K. & Chen, L. (2023) Generative AI and ChatGPT: Applications, challenges, and AI-human collaboration. Journal of information technology cases and applications, 25(3), pp. 277-304. doi:10.1080/15228053.2023.2233814

[10] Abdalgawad, N., Sajun, A., Kaddoura, Y., Zualkernan, I. A. & Aloul, F. (2022) Generative Deep Learning to Detect Cyberattacks for the IoT-23 Dataset. IEEE access, 10, pp. 6430-6441. doi:10.1109/ACCESS.2021.3140015

[11] Ferrag, M. A., Debbah, M., Al-Hawawreh, M. (2023) Generative AI for Cyber Threat-Hunting in 6G-enabled IoT Networks. https://arxiv.org/pdf/2303.11751.pdf

[12] Xiong, Z., Li, W. & Cai, Z. (2023) Federated Generative Model on Multi-Source Heterogeneous Data in IoT. Proceedings of the AAAI Conference on Artificial Intelligence, 37(9), pp. 10537-10545. doi:10.1609/aaai.v37i9.26252

3.6 Specifics of Cybersecurity in IoT Networks

In this section, we present potentially relevant state-of-the-art on threat intelligence, communication, information sharing, trust management and authentication schemes.

3.6.1 Honeypots for Cyberattacks Targeting IoT Devices and Networks

Honeypots are successfully utilized in cyber defense. Usually, their aim is to engage the attacker, encouraging interaction with them instead of real devices, but they also enable defenders to detect and monitor attacks. Nowadays, traditional honeypots have a relatively high risk of being detected as honeypot (Ragsdale & Boppana, [1]). LLMs can potentially make honeypots more believable as they can mimic various devices and do not rely on manually defined responses. LLM-based honeypots are also safe since they do not execute commands ([1]). The costs (e.g., time and computational resources) of developing an LLM-based honeypot may be much lower than those of an equally effective traditional honeypot because there is no need to manually define all responses or use real devices.

The varying ability of LLMs to act as high/intelligent-interaction honeypots has already been demonstrated in several papers. However, their usage in IoT networks has received very little attention in the literature. In CISSAN, papers about LLM-based honeypots can provide ideas on how to develop a new LLM-based honeypot for an IoT network.

McKee and Noever simulated in [2] ten honeypot tasks using ChatGPT as a honeypot. They grouped the tasks into three categories: i) operating systems (command-line interactivity with Windows, Linux and Mac. For example, ChatGPT acted as a Linux terminal), ii) applications (ChatGPT acted as a jupyter notebook and as a Linux terminal when prompted to install Team Viewer), and iii) networks (ChatGPT acted as command-line tools such as nmap and ping). Usually, the model provided correct or expected responses to command-line inquiries, but sometimes it started its answer like “I’m sorry, but …” even if the model was prompted to act like a terminal. According to the authors, ChatGPT has the potential to be a valuable tool as a honeypot. The authors only simulated and demonstrated the usage of ChatGPT as a honeypot by prompting; they did not test it in real-world situations.

The paper by Sladić et al. [3] introduces a novel method to create a software honeypot based on LLMs. The authors utilized the GPT-3.5-turbo-16k model and prompted it to act as a Linux terminal. Each processed prompt consists of the original personality prompt (e.g., to act as a Linux terminal), all previous interactions in the current session, and the new user command. The honeypot was evaluated by 12 security experts, who were specifically asked to consider which outputs helped them discern whether they were interacting with a honeypot. According to the authors, the honeypot was dynamic, realistic, and difficult to distinguish from a real system by humans. Additionally, the authors analyzed that the cost of actively using their model is roughly $0.8 per hour.

Ragsdale and Boppana identified in [1] two main threats to honeypots in their paper: an attacker’s ability to detect them and to break out of the scope of actions. Their proposed model is designed to address these issues by being easy to update and maintain. The model customizes both the input and the output of the LLM: the input to ensure that the LLM gets all necessary information (e.g., conversation history and context) and the output to sanitize the response (e.g., if the LLM refuses to answer questions). The authors simulated five tasks, including system reconnaissance, data obfuscation, lateral propagation, persistence, and exfiltration. They utilized GPT-3.5-turbo as an LLM and evaluated the model against Cowrie, which is as low-risk honeypot as the proposed model. The evaluation showed that a honeypot using an LLM as its backend can better emulate outputs than traditional honeypots of a similar level of risk. The authors mentioned that using generative model-based honeypots is as safe as low-interaction honeypots since no commands are executed. They also discussed the limitations of using LLMs as honeypots, such as responsiveness, non-deterministic output, non-verifiable output, costs, memory limit and training bias.

Guan et al. propose in [4] an adaptive high-interaction honeypot for IoT devices, called HoneyIoT. First, the authors built an attack trace collection system based on real devices. Their goal was to learn how attackers interact with IoT devices by modeling their behavior through a Markov decision progress. The authors also used reinforcement learning to learn the best responses to engage attackers. The honeypot consists of two components: i) a frontend virtual machine running on AWS that opens ports similar to a real IoT device and forwards attributes of attackers' request packets to the reinforcement learning agent, and ii) a reinforcement agent that selects a proper response based on the attacker's request. According to the authors, HoneyIoT performed well and mislead attackers into uploading malware better than their baseline models. HoneyIot also evaded honeypot detection tools effectively.

Mfogo et al. propose in [5] an intelligent-interaction honeypot for IoT devices. The model has three components: a honey-chatbot, a req/res database, and a request evaluator. The honey-chatbot is responsible for replying to the attacker's requests. The chatbot is first trained offline based on the transformer architecture (BERT), which records the response candidates. After that, the most probable response is chosen using a reinforcement learning and Markov decision process (MDP) model. The reinforcement learning models the future direction of the conversation. The req/res database is a database of possible requests and corresponding responses. The training dataset (including HTTP protocol data) serves as its baseline, and all new trusted requests and corresponding responses are added to the database. The request evaluator evaluates whether the request is malicious before requests and responses are collected in the database. The model was evaluated against IoTCandyJar and FirmPot honeypots, as well as against a model where the response was chosen randomly among all responses. The evaluation, which took 20 days, indicated that the proposed model improved the length of interaction with the attacker. However, 54% of interactions still only had one request and one response before the communication was terminated. The model captured about 30% of DoS attacks and 60% of R2L attacks. Additionally, the model was running online and was updated in real-time.

References

[1] Ragsdale, J. & Boppana, R. V. (2023) On Designing Low-Risk Honeypots Using Generative Pre-Trained Transformer Models With Curated Inputs. IEEE access, 11, pp. 117528-117545. doi:10.1109/ACCESS.2023.3326104

[2] McKee, F. & Noever, D. (2023) Chatbots in a Honeypot World. arXiv.org doi: arXiv:2301.03771

[3] Sladić, M., Valeros, V., Catania, C. & Garcia, S. (2024) LLM in the Shell: Generative Honeypots. arXiv.org. doi:10.48550/arxiv.2309.00155

[4] Guan, C., Liu, H., Cao, G., Zhu, S. & Porta, T. L. (2023) HoneyIoT: Adaptive High-Interaction Honeypot for IoT Devices Through Reinforcement Learning. arXiv.org. doi:10.48550/arxiv.2305.06430

[5] Mfogo, V. S., Zemkoho, A., Njilla, L., Nkenlifack, M. & Kamhoua, C. (2023) AIIPot: Adaptive Intelligent-Interaction Honeypot for IoT Devices. doi:10.1109/PIMRC56721.2023.10293827.

3.6.2 Sharing Security Information between Resource Constrained Network Nodes

Challenges in resource constrained IoT networks are driven by the miniaturisation of purpose-specific hardware and the imperative for cost-efficiency. These characteristics make traditional security solutions, which focus on securing individual devices, unrealistic. Instead, decentralised solutions that leverage the collective resources of a network must be considered. For these solutions, key enablers include self-configuration, scalability, resilience to extreme dynamics, and the overall sustainability and reliability of the network, suggesting the ad hoc paradigm an attractive foundation for development. Therefore, the objective of this master's thesis is to examine the feasibility of ad hoc networking in providing a reliable communication framework for sharing security information within CISSAN networks.

The research consists of the examination of major challenges in resource constrained IoT systems and exploration of promising solutions from academic literature. It details various categories of ad hoc networking, key standards, and routing protocols. Additionally, a simulation artifact was

developed to assess the feasibility of using ad hoc networking to distribute security information among resource constrained devices across different network sizes. The findings indicate that ad hoc networking is an efficient and scalable framework for sharing security information in terms of throughput, latency, and the overall reliability of the introduced security functionality. The key factors affecting these values could be identified as the physical topology of the network and the availability of existing routes between nodes. Furthermore, the alignment of ad hoc networking with academically proposed solutions was affirmed, highlighting benefits such as enhanced local processing capabilities, overcoming the limitations of centralized architectures, and improved management efficiency.

The results from this thesis can provide a reference base for further steps in validating ad hoc networking as a suitable communication framework upon which more complex, decentralised security functionalities can be implemented.

References

[1] Markkanen, V. (2024). Sharing Security Information between Resource Constrained Network Nodes. University of Jyväskylä. http://urn.fi/URN:NBN:fi:jyu-202405304109

3.6.3 ReLI: Real-Time Lightweight Byzantine Consensus in Low-Power IoT-Systems

In large decentralised systems like CISSAN there is a chance of malfunctioning due to one or more components of the system getting compromised. Byzantine fault tolerance support is essential in combating the presence of such malicious forces affecting these networks. However, according to the authors of ReLI: Real-Time Lightweight Byzantine Consensus in Low-Power IoT-Systems (2022) existing solutions for consensus or data aggregation in IoT/WSN systems either assume non-Byzantine node failures or use only simulation/theoretical models to address the existence of Byzantine nodes or are designed in a way that is not suitable for low-power IoT-systems. Theoretically, a decentralised system can effectively tolerate Byzantine characteristics of up to a certain fraction of the nodes. However, to achieve even that, the nodes need to interact extensively and share data with each other which makes the practicality of such solutions a major challenge, especially in resource-constrained IoT systems. The current work is one such solution directed for achieving Byzantine consensus in low-power IoT systems.

In their study, the researchers adapted and optimised the Practical Byzantine Fault Tolerant (PBFT) consensus strategy for IoT and Wireless Sensor Networks. The key methodological proposal involved the use of Synchronous-Transmission based mechanisms to handle the extensive data sharing required for reaching consensus, which typically challenges resource-constrained devices due to high data collision rates and energy demands. In PBFT, consensus is achieved if a certain number of devices can agree on the state of the system or the validity of data/actions, even in the presence of potentially disruptive elements. Three specific protocols: Glossy, MiniCast and Chaos were implemented to facilitate this process. Glossy was used for network-wide data sharing initiated by a primary node during the PBFT's PRE-PREPARE phase, while MiniCast enabled efficient many-to-many data sharing during the PREPARE phase. Chaos was used to streamline the COMMIT phase via sharing single bit flags to indicate whether the node could reach the quorum in the previous phase. To experimentally validate their approach, the researchers implemented the framework within the Contiki operating system for TelosB devices and tested it in Cooja simulator and publicly available IoT/WSN testbeds DCube and FlockLab. The comparison was made against a baseline PBFT method, adapted to IoT/WSN context with no optimisation.

The ReLI framework was observed performing significantly better than the baseline method, even when there are many traitors in the system. ReLI achieved greater optimisation particularly in environments with wider areas. The results observed involved up to 80% decrease in latency and up to 78% decreased radio-on time compared to the baseline PBFT method. The results show promise in developing sustainable Byzantine fault tolerant services, required for maintaining trust and integrity in decentralised systems. These directions may be leveraged in guiding CISSAN R&D efforts.

References

Goyal, H., H, M. K., & Saha, S. (2022). ReLI: Real-time lightweight Byzantine consensus in low-power IoT-systems. *2022 18th International Conference on Network and Service Management (CNSM)*, Thessaloniki, Greece (pp. 275-281). https://doi.org/10.23919/CNSM55787.2022.9965123

3.6.4 Survey of Secure Routing Protocols for Wireless Ad Hoc Networks

The paper: Survey of Secure Routing Protocols for Wireless Ad Hoc Networks, by Boulaiche (2020), is a comprehensive review of the state-of-the-art secure routing protocols proposed in the literature, that address routing security issue. The paper classifies secure routing protocols according to the secured protocol and discuss the proposed solutions. This paper has been reviewed because routing security is a prerequisite for ensuring reliable communication between network nodes, which in turn constitutes the foundation for collective intelligence. Key points involve potential routing attacks against CISSAN networks, security properties for communication, and potential routing solutions addressing both.

3.6.4.1 Various routing attacks against ad hoc networks

The research paper systematically categorises and details various security threats specifically targeting the routing protocols of wireless ad hoc networks. In this work, attacks are organised into categories based on their characteristics and objectives:

External vs. Internal Attacks:

1. External Attacks: Launched by nodes not belonging to the network, these can often be mitigated with firewalls and authentication measures.
2. Internal Attacks: More pernicious, these are carried out by compromised nodes within the network, making them harder to detect and counter.

Passive vs. Active Attacks:

3. Passive Attacks: These involve eavesdropping on the network to gather sensitive information without affecting network operations. Strong encryption can mitigate their impact.
4. Active Attacks: These involve overt actions like modifying data, injecting false information, or disrupting network operations to degrade the network's functionality and integrity.

The purpose of the attack: An adversary may target data exchanged between nodes in the network or targets the topology of the network created by the routing protocol in a way that allows it to act maliciously.

3.6.4.2 Security requirements

The research paper provides a detailed analysis of the essential security properties required to ensure safe and reliable communication in wireless ad hoc networks. The key security services identified include availability, authentication, integrity, confidentiality, and non-repudiation. To guarantee these security services, many cryptographic primitives are used. These are classified into three categories: symmetric cryptography, asymmetric cryptography, and digital signature and hash function.

3.6.4.3 Secure routing in ad hoc networks

After surveying secure routing protocols in wireless ad hoc networks a general understanding suggests, that most of the proposed protocols are extensions to already existing protocols to strengthen their security efficiency against some specific attacks. The major concern identified in topology-based category is to protect route construction phase of some specific routing protocols such as AODV and DSR. Some other solutions implement mechanisms to protect packet forwarding phase against tampering and dropping attacks. In position-based routing protocols, verifying node positions is identified as significant importance to minimise the impairments of routing. Other solutions aim to protect packet forwarding phase against tampering and dropping attacks.

Generally, existing solutions enhance security in wireless ad hoc networks primarily through cryptographic means to ensure the authentication, integrity, and confidentiality of control messages, while also attempting to maintain availability through node isolation mechanisms. Despite these efforts, comprehensive security must encompass both route establishment and data transmission phases, and further work is required to effectively isolate attackers and secure the neighbourhood discovery phase. The complexity of fully securing routing in resource-constrained environments, where heavy cryptographic solutions may be impractical, highlights the need for innovative approaches that

balance security with network performance. Additionally, some of the most severe attacks have yet to be fully addressed, underscoring the ongoing challenges in this field.

3.6.4.4 Conclusion

In conclusion, there are various security protocols designed to counter threats stemming from these characteristics, but completely securing such networks remains a challenging, unresolved issue. Designing novel solutions requires careful analysis of the adversary's scenarios and its objectives. These elements are classified in this paper providing helpful instruments for understanding secure routing protocols in wireless ad hoc networks.

References

Boulaiche, M. (2020). Survey of secure routing protocols for wireless ad hoc networks. *Wireless Personal Communications, 114*(1), 483–517. https://doi.org/10.1007/s11277-020-07376-1

3.6.5 Peer-to-Peer Trust Management in Intelligent Transportation System: An Aumann's Agreement Theorem Based Approach Cyberattacks Targeting IoT Devices and Networks

Operating in Peer-to-Peer manner necessitates trust among each node in the network. However, trust management among participants is a challenging task. This work presents truth of consensus, to identify the malicious nodes with no further delay, by considering the decision of each device on finding the trust values. This work has been recognised beneficial for CISSAN. The envisioned peer-to-peer capabilities in CISSAN networks depend on advanced trust management schemes, rendering recent work such as Ramesh et al. (2022), who detail their approach and measure their performance useful.

The approach detailed in this work assigns trust values to each node based on past interactions and behaviour. Nodes within the network use a voting mechanism to categorise each node as either compromised or un-compromised. This categorisation is based on direct observations and previously established trust values. Initially a node becomes a target of voting when its behaviour needs to be evaluated for trustworthiness. The process involves selecting voting participants through a hash function. Aumann's agreement theorem is applied to the voting: it assumes that rational nodes with common knowledge of each other's beliefs should converge towards a consensus about the trustworthiness of a node. The voting results are divided into two groups, those who voted for un-compromised and those who voted for compromised. The group with majority vote determines the trust status of the node. Based on the voting outcome, a decision parameter is updated, which influences the trust values of the voting nodes. If a node is determined to be compromised the trust values of the nodes who voted so, is increased. On the other hand, if a node is voted un-compromised, the nodes that voted so will improve their trust value. The target node is assessed based on the majority opinion of the voting nodes, and its status is updated to reflect whether it is considered trustworthy or not. In essence, this dual weight trust system aims to improve the accuracy of malicious node detection by dynamically updating trust values based on collective node decisions, thereby reinforcing the network's resilience to attacks.

The system's effectiveness is measured through network simulations that demonstrate superior detection rates of malicious nodes and better performance in terms of packet delivery ratio, energy consumption, and latency compared to other existing systems like Firecol. The research suggests that this dual weighted trust mechanism, reinforced by consensus-based decision-making and the strategic application of Aumann's agreement theorem, significantly enhances the security and reliability of VANETs. Based on these results, similar trust management schemes could be examined for CISSAN scenarios. VANET considerations involve resilience to extreme dynamics, which is a highly attractive attribute for any IoT networks. However, VANETS are also commonly characterised by relatively high resources. Therefore, the first step would be to examine its applicability in a more static and resource constrained environment.

References

Ramesh, T. R., Vijayaragavan, M., Poongodi, M., Hamdi, M., Wang, H., & Bourouis, S. (2022). Peer-to-peer trust management in intelligent transportation system: An Aumann's agreement theorem based approach. *ICT Express*, *8*(3), 340-346. https://doi.org/10.1016/j.icte.2022.02.004

3.6.6 A Comprehensive Review of Authentication Schemes in Vehicular Ad-Hoc Network

Among all security requirements, authentication is of prime importance. It is the first line of defence that guarantees that the message has been received from an authentic sender and hence checks masquerading attacks. There must be a way to assure the legitimacy of all parties as well as the message through which they communicate. This work presents a taxonomy of authentication schemes in vehicular ad hoc networks (VANET) and was chosen for review due to extreme importance of sophisticated authentication schemes in CISSAN networks, which utilise ad hoc networking principles. Furthermore, with public transport as a use case, schemes focusing on ensuring security services in mobile conditions are imperative.

Taxonomy of authentication schemes

Authentication schemes based on cryptography: These schemes primarily leverage asymmetric, symmetric, and ID-based cryptographic techniques to authenticate the identity of entities involved in communication. Cryptographic authentication mechanisms ensure that the information exchanged is accessible only to intended and verified users. These methods often employ public key infrastructures (PKI) where keys are distributed and managed through a trusted central authority, or decentralised approaches like blockchain where the authentication process is distributed among multiple nodes. Significant work is also put in preserving privacy during authentication process. Recent improvements in cryptography based authentication schemes focus on anonymity, increasing efficiency and decreasing delays during authentication.

Authentication schemes based on signature: Signature-based authentication involves verifying the digital signature attached to a message to confirm the sender's identity. The sender uses a private key to create a signature on the message, which can then be verified by others using the sender's public key. This type of scheme is pivotal in scenarios where non-repudiation is crucial, as it not only authenticates the sender but also ensures that the sender cannot deny the authenticity of the sent message later. In particular, Group signature (shared public key) is highlighted as promising approach on alleviating the burden of key management, frequent change of key pair and computation/communication overhead, typical for cryptography schemes.

Authentication schemes based on verification: Generally there is a time limit for verifying messages. Under high volume and strict requirements the verification process can become a bottleneck. Two main types of authentication schemes are introduced: batch verification and cooperative verification. Batch verification groups messages for simultaneous verification, reducing delays and computational load. Identified solutions include sharing verification responsibilities between different units. Cooperative verification involves vehicles within the network assisting each other to authenticate messages, which reduces redundancy and the computational burden on individual vehicles. Promising work in this direction involves ensuring each devices contribution and avoiding free riding problems.

Recent advancements in VANET authentication

Recent advancements in VANET authentication are significantly enhanced by integrating 5G technology and software-defined networking, improving data rates, latency, and overcoming the limitations of previous technologies like IEEE 802.11p and LTE, which faced issues with scalability and interference. Innovations such as millimeter waves, edge computing, and advanced cryptographic methods have improved security in vehicular communications, supporting a higher density of connected vehicles and ensuring privacy and cyber-attack resistance. However, traditional PKI-based systems, which rely on certificate authorities and management, face challenges with cumbersome certificate handling and key escrow problems in ID-based schemes, complicating scalability and key management, especially as Certificate Revocation Lists (CRLs) grow. In parallel, the advent of vehicular social networks (VSNs) and the integration of electric vehicles (EVs), which frequently require charging, introduce new privacy concerns, necessitating robust solutions for managing voluminous user data securely. To address these multifaceted challenges, blockchain technology, characterised

by decentralisation, tamper-proof nature, trustworthiness, and anonymity, is being increasingly applied in VANETs to streamline key management, improve security, and preserve user privacy, thereby enhancing overall network management and reliability in complex vehicular environments Understanding these elements significantly benefits the CISSAN project by promoting potential research directions and fostering innovations tailored to meet the evolving security demands of decentralized networks.

References

Azam, F., Yadov, S. K., Priyadarshi, N., Padmanaban, S., & Bansal, R. C. (2021). A comprehensive review of authentication schemes in vehicular ad-hoc network. *IEEE Access, 9*, 31309-31321. https://doi.org/10.1109/ACCESS.2021.3060046

3.7 Malicious Use of AI

In this section, we present an initial analysis of the use of AI in cyberattacks. Since a key objective of this analysis is to find out whether the malicious use of AI should be reflected in the CISSAN plans and efforts (and in what ways), we look carefully not only at research experiments and hypothetical AI applications but also at the real-world situation. After all, CISSAN's research and development scope is limited, and it is important for the partners to prioritize their efforts based on the current threat landscape and expectations for the coming years.

In cybersecurity, Artificial Intelligence (AI) – as nearly any other technology – can be used for both good and malicious purposes. The use of AI for carrying out or facilitating cyberattacks has been actively studied for over ten years, starting with AI applications to generating domain names for command and control (C&C or C2) connections which bypass detectors (e.g., [1]), solving CAPTCHAs (e.g., [2]), generating malware (e.g., [3]), and producing social engineering content (e.g., [4]). It is in the CISSAN plan to keep track of the malicious use of AI systems and techniques to identify, analyze, and, if feasible, propose countermeasures to AI-assisted attacks relevant to the project objectives and use cases. In addition to a brief overview of malicious AI applications, we consider in D1.1 the current state of knowledge about real-world AI-assisted cyberattacks and look at certain misconceptions and confusion in the discussions around such attacks. We note right away that the focus of this section is mainly on Machine Learning (ML) systems and techniques, which enable machines to learn from data to solve tasks without being explicitly programmed to do so. ML is arguably the most successful and vibrant subfield of AI as of today. We also note that while ML can be used for both cyber-dependent and cyber-enabled crime and other malicious activities[1], the former is much more relevant for CISSAN and will receive our full attention[2].

The framework proposed in [5] categorizes the AI / ML techniques used for offense in accordance with the six stages of the (slightly modified) cyber kill chain: reconnaissance, access and penetration, delivery, exploitation, C2, and action on objectives. For the review, the authors selected 46 papers published between 2014 and 2021 in the following databases: ACM, arXiv, Blackhat, MDPI, Scopus, Springer, and IEEE Xplore. The following types of AI-assisted malicious operations were found in the selected papers:

- Reconnaissance stage: spear phishing; vulnerability identification; target person or organization selection, classification and profiling (e.g., based on their social media activity and public social media profiles); target digital estate identification (e.g., for carrying out malicious operations only in targeted systems and networks); normal target behavior learning; attack outcome prediction.
- Access and penetration stage: password guessing; CAPTCHA solving; keyboard data stealing (and other applications of data analysis to obtaining confidential information, e.g., based on side-channel attacks); intelligent abnormal behavioral generation; ML model manipulation (attacking ML models used in cyber defense via poisoning and evasion techniques).

[1] See, e.g., https://nationalcrimeagency.gov.uk/cyber-choices

[2] We, however, mention quickly certain cases where ML is used for cyber-enabled crime, e.g., the use of deepfakes for financial fraud.

- Delivery stage: automated generation of phishing URLs and malware (to bypass ML-based and other detection systems); malicious payload hiding.
- Exploitation stage: vulnerability exploitation, target's security weaknesses learning; personalized disinformation generation.
- C2 stage: automated generation of domain names for C2 communication (Domain Generation Algorithms), including data exfiltration; malware learning for more autonomous operations (e.g., lateral movement) to eliminate or minimize the need for C2 communication.
- Action on objectives stage: Distributed Denial of Service (DDoS) attack orchestration.

An earlier survey [6] discusses several other types of ML applications in cyberattacks and considers some of the ML-based malicious techniques in greater detail:

- Anti-reversing techniques complicating malware analysis: detection of sandboxed environments (to hide malicious payload and behavior), malicious code obfuscation. (DeepLocker [7] serves as a good example in this category.)
- Analysis of the data from earlier malicious campaigns and security tools testing for selecting and adjusting attack operations (primarily to evade detection).
- ML-supported decision-making for building more autonomous malware: selecting content for exfiltration; selecting among vulnerability exploitation, credentials brute-forcing, and installing a key-logger to capture credentials; adapting malicious functionality to the compromised environment (e.g., ransomware and exfiltration at a computer of a company executive vs. crypto-mining on a server).
- Learning patterns of normal user communication behavior, e.g., in email and chat traffic, for further social engineering activities.
- Use of ML for model stealing of ML models used in cyber defense, based on blackbox probing (an early example can be found in [8]). Replicated defensive models can be used then for poisoning and evasion, e.g., for building undetectable malware or phishing messages.

In addition to ML techniques, [6] also discusses a few other malicious AI applications, such as those based on bio-inspired algorithms and swarm intelligence:

- The use of genetic algorithms and mutation for generating more effective or evasive malware (e.g., [9], [10]).
- The use of swarm intelligence-based algorithms to avoid the need for centralized control over malicious code in multiple devices, such as C2 servers in botnets (e.g., [11]).

In recent publications, we also find several other ML applications in cybercrime and cyberattacks:

- The use of deepfakes for impersonating existing people (e.g., [12], [13]). This is an example of cyber-enabled crime.
- Building and operating fake online personas for establishing contact with targeted victims, for disinformation activities and other purposes (mentioned, e.g., in [13]).
- Generating data that resembles a learned distribution (e.g., in network traffic or OS events) for hiding malicious operations (e.g., [14]).
- Collection and mining of Open Source Intelligence (OSINT) data, including data leaks, primarily for reconnaissance (e.g., [15]).
- Confidential data extraction from ML models (e.g., [16], [17]).

Unsurprisingly, the recent advances in generative AI[3] have led to new studies of the use of ML models in the cybersecurity domain. Generative Adversarial Networks (GANs), which learn to generate new data resembling their training set, have found multiple applications in cyber defence, for instance, for augmenting training sets of attack detection ML models. However, GAN is a dual-use

[3] See, e.g., https://en.wikipedia.org/wiki/Generative_artificial_intelligence

approach, which can also be used for generating realistic data and objects that can fool attack detectors and other security systems. A survey [18] discusses both GAN applications that support cyber defenders and those that can assist in cyberattacks. We mention here a few examples in the latter category:

- Assuming that an attacker knows what features a malware detection algorithm uses but does not know anything else about the detection ML model, the authors of [19] proposed MalGAN – a GAN that transforms malware samples to evade detection while preserving their malicious functionality (with the GAN discriminator trained to fit the black-box malware detection model). Multiple enhancements and variations of this approach were proposed (e.g., in [20] and [21]).
- PassGAN proposed in [22] is a GAN for learning, in an unsupervised way, the distribution of real passwords from actual password leaks. PassGAN is a clear improvement over the most popular rule-based password guessing tools (such as HashCat [23] and John the Ripper [24]) and is not limited by the number of rules and by the password dataset size.
- Several papers applied GAN techniques for attacking ML models. In [17], for example, GAN is used for recovering confidential data from trained models. AdvGAN in [25] is used for generating adversarial examples, and PoisonGAN in [26] is used for poisoning Federated Learning in Edge Computing Systems.

Malicious applications of models based on Generative Pre-trained Transformers (GPTs) and other large foundation models came into focus only very recently. In CISSAN, we will follow these developments and consider them in D1.2 (and possibly other deliverables, as appropriate), and here we only mention a few examples.

- Fang et al. [27] demonstrated how Large Language Models (LLMs), interacting with appropriate tools and APIs, can be used to autonomously hack websites without human feedback and without knowing site vulnerabilities beforehand. They also showed that a GPT-4[4] based agent was capable of identifying vulnerabilities in real-world websites.
- Gupta et al. [28] demonstrated several ways to bypass restrictions and use ChatGPT[5] for malicious purposes. After bypassing the system safeguards, it was possible to use ChatGPT to generate malicious content of various types, including highly convincing and personalized phishing emails, attack payload (e.g., code for SQL injections and Web Application Firewall bypass payloads), ransomware and malware.
- Similarly to [28], Alawida et al. [29] produced various malicious code snippets using the ChatGPT API, overcoming the safeguards by dividing (prohibited) prompts into sub-questions and collecting the model responses. They demonstrated that ChatGPT can create constantly evolving polymorphic malware, a Java program that starts Notepad in the background without showing its window to the user (which can be used to run malicious PowerShell commands, e.g., for downloading and installing malware on a victim's computer), phishing emails, Living off the Land Binaries (LOLBIN), and other malicious content.
- Malicious applications of ChatGPT were reviewed by Al-Hawawreh et al. in [30], including malware development, phishing and social engineering, vulnerability discovery and exploitation, disinformation and misinformation. The authors also demonstrated how to use ChatGPT for simulating an Industrial Control System device to mount a data injection attack against critical infrastructure.
- The LLMorpher project [31] shows how to encode a self-replicating virus code entirely in the natural language, as a list of well-defined sentences in English used as an input to one of the OpenAI's GPT models.
- Pentesting (penetration testing) is used extensively by cybersecurity professionals for identifying security weaknesses, but it can also be used for preparing cyberattacks. Pentesting

[4] https://openai.com/index/gpt-4/
[5] https://openai.com/chatgpt/

usually requires significant manual efforts, so an LLM-based pentesting automation tool proposed in [32] and called PentestGPT can be an asset for both defenders and attackers.

Microsoft and OpenAI recently published their analysis of the current use of LLM technology by threat actors [33]. The study presents identified activities associated with several known threat actors (most of which are nation-state-affiliated), including prompt-injections, attempted misuse of LLMs, and fraud. The report notes that "Microsoft and OpenAI have not yet observed particularly novel or unique AI-enabled attack or abuse techniques resulting from threat actors' usage of AI."

In some cases, opinions may vary whether a specific malicious activity or technique belongs with the use of ML for attacking. Several of such "borderline" cases come from the GPT domain. For example, [34] presents an indirect prompt injection technique demonstrated with Microsoft's Bing Chat LLM Chatbot: if a user gives Bing Chat a permission to view and access currently open websites in the chat session, an attacker can plant an injection in a website the user is visiting and make Bing Chat exfiltrate personal information of the user and send it to the attacker. Another example is "AI package hallucination" explained in [35]: if an attacker finds a ChatGPT's recommendation for an unpublished SW package, they can publish their own malicious package in its place, and then a user prompting ChatGPT about software with a similar functionality will be directed to the malicious package. Furthermore, prompt engineering can be used for extracting training data from LLM-based chatbots [36] and for altering the behaviour of LLM-based agents in ways selected by the attacker [37]. In all these examples, the attacker does not employ ML techniques but rather abuses ML-based systems implemented by other parties and used by potential attack victims (or trained on their data). In some cases, this can be easier to prevent or defend against and should be taken into account in threat analysis efforts.

With the rich existing research and numerous proposed malicious applications of ML, it is important to understand how little we currently know about the use of ML in real-world attacks. It is confirmed that deepfakes were used in financial fraud (so, cyber-enabled crime). For instance, one case where deepfaked voice of a company's director was used to demand a bank manager in Hong Kong to authorize a $35M transfer can be found in [38], and a similar example was reported by NCSC-FI in [12][6]. We also have strong evidence that LLMs have been widely used for producing phishing and spam messages, including the 1,265% increase in phishing emails since the launch of ChatGPT reported by SlashNext in [39] and the analysis of Abnormal in [40]. Apart from these examples of the use of ML for social engineering, however, it is not easy to find adequate evidence of other applications of ML in real-world cyberattacks. There are only a handful of cases that go from one article or report to another:

- Emotet malware is claimed in [41] to use ML for selecting its victims and, possibly, hiding its malicious functionality "in environments used by cybersecurity researchers and investigators." Also [42] calls Emotet "a real trojan malware AI tool that cyber-criminal use" but provides no AI-related details.
- There are claims, e.g., in [43], [44], [45], and [46], that the botnet used in the DDoS attack against TaskRabbit in 2019 was AI-controlled or AI-enabled.
- A botnet attack on WordPress is presented as AI-assisted in [43] and [45], with no AI-related details.
- The authors of [43] and [45] also claim the use of AI in attacks on Instagram. While [43] mentions that "many have speculated that hackers are using AI systems to scan Instagram user data for potential vulnerabilities" (which is not easy to interpret), [45] provides no AI-related details.
- In a very similar manner, [44] and [46] present the ransomware attack on Yum! Brands in January 2023 as AI-assisted. They claim that "this attack used AI technology to determine the most valuable data based on how much damage it could cause to the target company."

[6] We note, however, that the number of the publicly reported deepfake-based fraud cases is still modest.

- Another case similarly presented in [44] and [46] is a data breach at T-Mobile. The two articles say that the threat actor used an API equipped with AI capabilities to secure unauthorized access to sensitive data.

Unfortunately, in all these cases, we see no technical analysis providing evidence of the use of ML / AI. The claims in [41] – [46] either have no references to technical reports or refer to the analysis where ML and AI are not mentioned at all[7]. In the case of Emotet, for instance, the malware's ability to avoid honeypots and distinguish between real and sandboxed environments can be explained by a good set of rules instead of the use of ML, and we do not find even assumptions of the latter in such detailed technical reports as [47] and [48]. Furthermore, some of the claims are likely a result of misunderstanding and may confuse the reader. For example, the use of an API equipped with AI capabilities to secure unauthorized access to sensitive data in the T-Mobile data breach case (claimed in [44]) likely incorrectly refers to the abuse of the T-Mobile's API that provides access to the data used for ML models training (see, e.g., the analysis in [49]), which is not related to employing ML / AI in cyberattacks.

It is also regrettable that despite the lack of evidence, we see claims about real-world malicious applications of ML / AI expressed in the no-doubt manner, such as "Many ransomware attacks that took place after the creation of AI tools leveraged AI technology to automate decisions on which data to take ..." in [44] and "The AI botnets often look for vulnerability in devices, after which they exploit the system to discover chief elements and then attack the target, …" in [45]. A good example of careful phrasing is [40]: while their method of attributing phishing messages to LLMs appears sound, the report is titled "5 Real-World Email Attacks *Likely* Generated by AI in 2023".

What we, however, do have evidence for is interest of cybercriminals in ML-powered tools. The study of underground forums presented in [50] found discussion threads, in an exploratory tone, on such tools (mainly open-source ones) for penetration testing (DeepHack and DeepExploit), CAPTCHA breaking (XEvil), finding social media accounts associated with a specific profile (Eagle Eyes), and password guessing (likely similar to PassGAN mentioned above).

One case where an "AI used in attacks" claim is backed by analysis is presented in a recent Threat Insights Report by HP Wolf Security [51], in "Generative AI assisting malware developers in the wild" section. The report says: "Based on the scripts' structure, consistent comments for each function and the choice of function names and variables, we think it's highly likely that the attacker used GenAI to develop these scripts (T1588.007[8]). The activity shows how GenAI is accelerating attacks and lowering the bar for cybercriminals to infect endpoints."

Several reasons can be behind the lack of evidence of the use of AI in real-world attacks, including:

- Attack victims do not want to talk about their AI-assisted breaches and incidents
- "AI techniques serve only to improve existing attack techniques, leaving little trace that can help differentiate AI-enabled cyberattacks from their conventional counterparts." [13]
- Low attacker's motivation – conventional attack techniques work well, and attackers fail to gain significant new benefits with AI
- Attackers lack data
- Attackers lack skills

We still have to see how the evolution of generative AI and other ML techniques will change the threat landscape in the coming months and years, but the greatest concerns in the short to medium term are perhaps the use of AI in social engineering and reconnaissance. While countering social engineering is not part of CISSAN's agenda, we will analyse whether AI-assisted reconnaissance should be prioritized by the project. We will also consider two other potentially interesting research directions: (i) attackers targeting our ML models (exploiting AI techniques) and (ii) attackers building and operating AI-powered IoT botnets.

[7] Instead, the articles describe how devastating the presented attacks were.

[8] https://attack.mitre.org/techniques/T1588/007/

3.7.1 References

[1] Anderson, H. S., J. Woodbridge, and B. Filar. 2016. Deepdga: Adversarially-tuned domain generation and detection. In Proceedings of the ACM Workshop on Artificial Intelligence and Security, Vienna, Austria, 13–2409. (https://dl.acm.org/doi/10.1145/2996758.2996767)

[2] Bursztein, E., J. Aigrain, A. Moscicki, and J. C. Mitchell. 2014. The end is nigh: generic solving of text-based CAPTCHAs. 8th Usenix workshop on Offensive Technologies WOOT '14, San Diego, CA, USA. (https://www.usenix.org/conference/woot14/workshop-program/presentation/bursztein)

[3] Cani, A., M. Gaudesi, E. Sanchez, G. Squillero, and A. Tonda (2014). Towards automated malware creation. Proceedings of The 29Th Annual ACM Symposium On Applied Computing, Gyeongju Republic of Korea, 157–60. doi: 10.1145/2554850.2555157

[4] Seymour, J., and P. Tully. 2016. Weaponizing data science for social engineering: Automated E2E spear phishing on Twitter. https://www.blackhat.com/docs/us-16/materials/us-16-Seymour-Tully-Weaponizing-Data-Science-For-Social-Engineering-Automated-E2E-Spear-Phishing-On-Twitter-wp.pdf

[5] The Emerging Threat of Ai-driven Cyber Attacks: A Review (https://www.tandfonline.com/doi/full/10.1080/08839514.2022.2037254)

[6] Thanh, C., and I. Zelinka. 2019. A survey on artificial intelligence in malware as next-generation threats. MENDEL 25 (2):27–34. doi:10.13164/mendel.2019.2.027

[7] https://i.blackhat.com/us-18/Thu-August-9/us-18-Kirat-DeepLocker-Concealing-Targeted-Attacks-with-AI-Locksmithing.pdf (IBM Research work)

[8] Tramer, F., Zhang, F., Juels, A., Reiter, M. K., and Ristenpart, T. 2016. Stealing machine learning models via prediction apis. In 25th USENIX Security Symposium (USENIX Security 16), pp. 601–618.

[9] Ferrie, P. and Shannon, H. 2005. Virus analysis 2 – It's zell(d)ome the one you expect. Virus Bulletin, pp. 7–11.

[10] Xu, W., Qi, Y., and Evans, D. 2016. Automatically evading classifiers. In Proceedings of the 2016 network and distributed systems symposium, pp. 21–24.

[11] Zelinka, I., Das, S., Sikora, L., and Senkerik, R. 2018. Swarm virus-next-generation virus and antivirus paradigm? Swarm and Evolutionary Computation 43, 207–224.

[12] Deep fakes are used as part of cybercrime (in Finnish). https://www.kyberturvallisuuskeskus.fi/fi/ajankohtaista/kyberturvallisuuskeskuksen-viikkokatsaus-032024

[13] The security threat of AI-enabled cyberattacks (https://www.traficom.fi/sites/default/files/media/publication/TRAFICOM_The_security_threat_of_AI-enabled_cyberattacks%202022-12-12_en_web.pdf)

[14] M. Rigaki and S. Garcia, "Bringing a GAN to a Knife-Fight: Adapting Malware Communication to Avoid Detection," 2018 IEEE Security and Privacy Workshops (SPW), San Francisco, CA, USA, 2018, pp. 70-75, doi: 10.1109/SPW.2018.00019.

[15] Open-Source Intelligence (OSINT), https://www.imperva.com/learn/application-security/open-source-intelligence-osint/

[16] Song, Congzheng and Ananth Raghunathan. "Information Leakage in Embedding Models." Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security (2020). Available at: https://arxiv.org/abs/2004.00053.

[17] Zhang, Yuheng, R. Jia, Hengzhi Pei, Wenxiao Wang, Bo Li and Dawn Xiaodong Song. "The Secret Revealer: Generative Model-Inversion Attacks Against Deep Neural Networks." 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2019): 250-258. Available at: https://arxiv.org/abs/1911.07135.

[18] Yinka-Banjo, C., Ugot, OA. A review of generative adversarial networks and its application in cybersecurity. Artif Intell Rev 53, 1721–1736 (2020). https://doi.org/10.1007/s10462-019-09717-4

[19] Hu, Weiwei & Tan, Ying. (2017). Generating Adversarial Malware Examples for Black-Box Attacks Based on GAN. Available at: https://arxiv.org/pdf/1702.05983.pdf.

[20] Enmin Zhu, Jianjie Zhang, Jijie Yan, Kongyang Chen, Chongzhi Gao. N-gram MalGAN: Evading machine learning detection via feature n-gram. Digital Communications and Networks, Volume 8, Issue 4, 2022, Pages 485-491, https://doi.org/10.1016/j.dcan.2021.11.007.

[21] Trehan, H., Di Troia, F. (2022). Fake Malware Generation Using HMM and GAN. In: Chang, SY., Bathen, L., Di Troia, F., Austin, T.H., Nelson, A.J. (eds) Silicon Valley Cybersecurity Conference. SVCC 2021. Communications in Computer and Information Science, vol 1536. Springer, Cham. https://doi.org/10.1007/978-3-030-96057-5_1

[22] Hitaj, Briland & Gasti, Paolo & Ateniese, Giuseppe & Perez-Cruz, Fernando. (2017). PassGAN: A Deep Learning Approach for Password Guessing. Available at: https://arxiv.org/pdf/1709.00440.pdf.

[23] Hashcat: https://hashcat.net/hashcat/

[24] John the Ripper: https://www.openwall.com/john/

[25] Chaowei Xiao, Bo Li, Jun-Yan Zhu, Warren He, Mingyan Liu, and Dawn Song. 2018. Generating adversarial examples with adversarial networks. In Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI'18). AAAI Press, 3905–3911. DOI: 10.5555/3304222.3304312.

[26] J. Zhang, B. Chen, X. Cheng, H. T. T. Binh and S. Yu, "PoisonGAN: Generative Poisoning Attacks Against Federated Learning in Edge Computing Systems," in IEEE Internet of Things Journal, vol. 8, no. 5, pp. 3310-3322, 1 March1, 2021, DOI: 10.1109/JIOT.2020.3023126.

[27] Fang, R., Bindu, R., Gupta, A., Zhan, Q. & Kang, D. (2024) LLM Agents can Autonomously Hack Websites. arXiv.org. doi:10.48550/arxiv.2402.06664

[28] Gupta, M., Akiri, C., Aryal, K., Parker, E., & Praharaj, L. (2023) From ChatGPT to ThreatGPT: Impact of Generative AI in Cybersecurity and Privacy. IEEE Access, 11, 80218–80245. https://doi.org/10.1109/ACCESS.2023.3300381

[29] Alawida, M., Abu Shawar, B., Abiodun, O. I., Mehmood, A., Omolara, A. E., & al Hwaitat, A. K. (2024) Unveiling the Dark Side of ChatGPT: Exploring Cyberattacks and Enhancing User Awareness. Information (Basel), 15(1), 27. https://doi.org/10.3390/info15010027

[30] Al-Hawawreh, M., Aljuhani, A., & Jararweh, Y. (2023) Chatgpt for cybersecurity: practical applications, challenges, and future directions. Cluster Computing, 26(6), 3421–3436. https://doi.org/10.1007/s10586-023-04124-5

[31] The LLMorpher project: https://github.com/SPTHvx/SPTH/blob/master/articles/files/LLMorpher.txt

[32] Deng, Gelei & Liu, Yi & Mayoral-Vilches, Víctor & Liu, Peng & Li, Yuekang & Xu, Yuan & Zhang, Tianwei & Liu, Yang & Pinzger, Martin & Rass, Stefan. (2023). PentestGPT: An LLM-empowered Automatic Penetration Testing Tool. Available at: https://arxiv.org/abs/2308.06782.

[33] Staying ahead of threat actors in the age of AI. (By Microsoft Threat Intelligence, 2024) https://www.microsoft.com/en-us/security/blog/2024/02/14/staying-ahead-of-threat-actors-in-the-age-of-ai/

[34] Greshake, Kai, Sahar Abdelnabi, Shailesh Mishra, Christoph Endres, Thorsten Holz and Mario Fritz. “Not What You've Signed Up For: Compromising Real-World LLM-Integrated Applications with Indirect Prompt Injection.” Proceedings of the 16th ACM Workshop on Artificial Intelligence and Security (2023). Available at: https://arxiv.org/abs/2302.12173.

[35] Bar Lanyado, Can you trust ChatGPT's package recommendations? Available at: https://vulcan.io/blog/ai-hallucinations-package-risk.

[36] Nasr, Milad, Nicholas Carlini, Jonathan Hayase, Matthew Jagielski, A. Feder Cooper, Daphne Ippolito, Christopher A. Choquette-Choo, Eric Wallace, Florian Tramèr and Katherine Lee. "Scalable Extraction of Training Data from (Production) Language Models." ArXiv abs/2311.17035 (2023).

[37] Donato Capitella. "Synthetic Recollections: A Case Study in Prompt Injection for ReAct LLM Agents." Available at: https://labs.withsecure.com/publications/llm-agent-prompt-injection.

[38] Hong Kong Bank Manager Swindled by Fraudsters Using Deepfaked Voice of Company Director (https://incidentdatabase.ai/cite/147/)

[39] The State of Phishing 2023 (https://slashnext.com/state-of-phishing-2023/)

[40] AI Unleashed: 5 Real-World Email Attacks Likely Generated by AI in 2023 (https://cdn2.assets-servd.host/gifted-zorilla/production/files/AI-Unleashed_5-Real-World-Attacks-Likely-Generated-by-AI-in-2023.pdf)

[41] Machine learning: How does it help IT admins (and hackers too)? https://digitalsecurityguide.eset.com/apac/machine-learning-how-does-it-help-it-admins-and-hackers-too

[42] The Shifting Cyber Security Threats and the Role of AI. https://clairelogic.net/the-shifting-cyber-security-threats-and-the-role-of-ai/

[43] Has an AI cyberattack happened yet? https://www.infoq.com/articles/ai-cyber-attacks/

[44] Real-Life Examples of How AI Was Used to Breach Businesses, by Kelle White, https://oxen.tech/blog/real-life-examples-of-how-ai-was-used-to-breach-businesses-omaha-ne/

[45] 5 Artificial Intelligence-Based Attacks That Shocked The World In 2018, https://analyticsindiamag.com/ai-origins-evolution/5-artificial-intelligence-based-attacks-that-shocked-the-world-in-2018/

[46] AI Was Used to Breach These Major Businesses, https://mwdata.net/ai-was-used-to-breach-these-major-businesses-rockport-mo/

[47] Emotet Malware: The Enduring and Persistent Threat to the Health Sector. https://www.hhs.gov/sites/default/files/emotet-the-enduring-and-persistent-threat-to-the-hph-tlpclear.pdf

[48] Emotet Malware Analysis. Druan. https://medium.com/@dryan5346/emotet-malware-analysis-25a31d325731

[49] Is T-Mobile's AI training model the reason it keeps getting hacked? Mike Dano. https://www.lightreading.com/ai-machine-learning/is-t-mobile-s-ai-training-model-the-reason-it-keeps-getting-hacked-

[50] Malicious Uses and Abuses of Artificial Intelligence. Vincenzo Ciancaglini, Craig Gibson, David Sancho, Odhran McCarthy, Philipp Amann, Aglika Klayn, Maria Eira. https://unicri.it/sites/default/files/2020-11/AI%20MLC.pdf

[51] Threat Insights Report by HP Wolf Security, September 2024. Available at: https://threatresearch.ext.hp.com/wp-content/uploads/2024/09/HP_Wolf_Security_Threat_Insights_Report_September_2024.pdf

4 Summary and Outlook

This document, D1.1, reviews, summarizes, and organizes approaches and technologies relevant for the project, mapping “essential research findings” to the project needs and supporting the project plan relevance by reflecting and adapting to such findings dynamically. The primary objective of the document is to describe and structure the current state-of-the-art as considered by the project partners. The focus areas of the document include AI-based methods, blockchain-based technologies, and collective decision-making, which correspond to the key research and innovation directions in CISSAN.

D1.1 is produced in the first stage of the project, and D1.2 will present an update, collected, analysed, and taken into use throughout the project timeframe. Enabling techniques, methods, and algorithms for collective intelligence in IoT networks will be considered in greater detail, such as collective intelligence communication protocols, security task delegation (run-time) and security functionality distribution (design-time) algorithms, node trust management, aggregation of AI decisions. Other areas of attention will likely include explanations of detected anomalies, support of response actions, and data quality verification methods.