CREATING AN ADAPTIVE DATA COLLECTING SYSTEM BASED ON IoT DEVICES

George CALIANU[1], Elena CARCADEA[2], Ciprian LUPU[3], Alina PETRESCU-NIȚĂ[4]

This paper presents an innovative Power-to-X system that integrates IoT devices and a novel caching mechanism to improve real-time data collection and distribution for industrial applications. The system enables concurrent and secure access to data across multiple installations, enhancing platform scalability and responsiveness. Compared to traditional SCADA systems, the proposed caching model reduces latency and ensures greater adaptability for managing large volumes of data. This design supports efficient monitoring, reliable operation, and energy optimization, addressing the challenges of modern industrial environments while providing a flexible and robust solution for diverse applications.

Keywords: IoT, SCADA. Renewable Energy, Cache

1. Introduction

The acquisition of data from IoT (Internet of Things) devices in industry represents a dynamic and evolving field, reflecting advances in technology, standards, and practical applications. To reduce network latency, various temporary data storage (cache) technologies are used. The literature reveals a multitude of such techniques and approaches. In traditional cloud computing, communication delays between IoT devices and the cloud make it unsuitable for latency-sensitive applications. A solution could be a cooperative cache system at the application level [1]. Thus, the most accessed data is either stored in a cloudlet[5] or distributed as replicas across multiple cloudlets. IIoT[6] generates a large volume of traffic, leading to issues such as high energy consumption and longer access times. A solution is

[1] Eng., National Research and Development Institute for Cryogenics and Isotopic Technologies – ICSI, Râmnicu Vâlcea, Romania, e-mail: george.calianu@icisi.ro

[2] Mat., National Research and Development Institute for Cryogenics and Isotopic Technologies – ICSI, Râmnicu Vâlcea, Romania, e-mail: elena.carcadea@icsi.ro

[3] Prof., Department of Automatic Control and Systems Engineering, National University of Science and Technology POLITEHNICA Bucharest, Romania; ciprian.lupu@acse.pub.ro

[4] Associated Prof., Dept. of Mathematical Methods and Models, National University of Science and Technology POLITEHNICA Bucharest, Romania, alina.petrescu@upb.ro

[5] A cloudlet is a small-scale data center or a cluster of computers that provides services similar to those of the cloud.

[6] IIoT (Industrial Internet of Things) represents the industrial application of the IoT concept.

storing IIoT content in network nodes called trusted cache nodes (TCN[7]), which can help mitigate these issues [2]. Unlike approaches that only focus on content placement, there are proposals that optimize both content placement and the node distribution model at different levels, considering cache size limits and energy consumption costs [3]. Fog computing can address these technical problems and could be one of the promising solutions for managing the big data produced by IoT, which are often critical for security and time-sensitive [4]. Security in IIoT applications must be analyzed at each layer of the typical IoT architecture: hardware, network, and application. The measures proposed in the literature include the use of firewalls, secure communication channels (IPSec, SSL/TLS), user-based authentication, and encryption. However, current solutions mainly address industrial systems as a whole, without specifically responding to the variability of IIoT applications, where each layer requires dedicated security measures [5]. For compensating communication delays in automatic control systems, a solution would be to use predictive control, which estimates the process response and adjusts commands in advance. This method helps maintain system stability and performance even in the presence of significant network delays [6]. TCP can introduce additional delays and performance bottlenecks due to its inherent congestion control mechanisms. To address these challenges, several performance measurement techniques have been proposed to assess and optimize communication channels. One such approach is ImTCP (TCP with Inline Measurement), which enhances TCP by embedding measurement capabilities directly into the protocol, allowing real-time estimation of available bandwidth without requiring additional hardware [7]. Geographically distributed systems, such as power plants, can successfully use IoT devices to achieve goals like real-time parameter monitoring, remote measurements, reducing labor costs, and increasing economic viability [8]. A large-scale adoption of IoT devices in a geographically dispersed structure must consider the potential vulnerabilities specific to these devices. Such a large-scale vulnerability can create serious security issues, as malicious actors can exploit vulnerabilities of low-resource devices in traditional networks [9]. To prevent malicious activities, some research introduces the concept of a trusted cache node (TCN) for IIoT [10]. TCNs cache frequently used content and ensure the security of connected links. Edge computing has been considered a primary paradigm to meet the low-latency demand for some computing- or data-intensive applications, particularly for IIoT applications. Considering this, some research proposes a deep learning-based cache memory optimization method called DLECO, to reduce costs during the cache memory planning process [11]. Information-Centric Networking (ICN) [12] is recognized as an important technology for rapid content retrieval in content-based IoT applications. Proactive caching in cooperative cache systems [13], in heterogeneous IoT systems, is an

[7] TCN – Trusted Caching Node

interoperable and flexible solution for meeting latency constraints in these environments, allowing efficient data processing and storage while reducing redundant computing and costs. As systems become increasingly complex and traditional maintenance and repair methods (such as preventive maintenance based on historical data) are no longer sufficient, the concept of antifragility is introduced into smart manufacturing systems, utilizing emerging technologies such as Digital Twin (DT), Artificial Intelligence (AI), Big Data, and Cloud Computing. This solution enables the creation of a digital model of the factory, which monitors and analyzes the state of equipment and processes in real time. By applying the Failure Mode and Effects Analysis (FMEA) method, the system can anticipate and prevent failures, thereby optimizing production and transforming problems into opportunities for improvement [14].

2. The proposed solution

The proposed solution is based on a cache system that collects data from various devices and exposes it to stakeholders in a unique and consistent manner, while maintaining traceability of the time each event occurred.

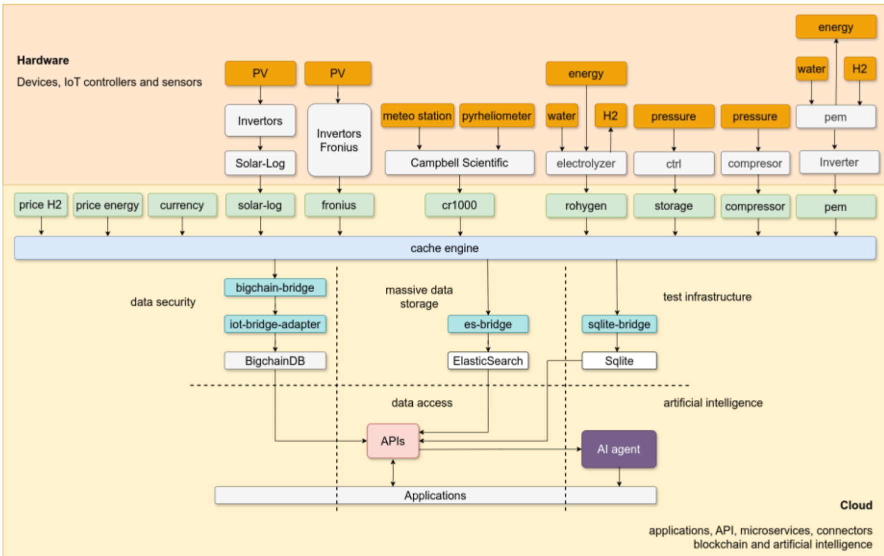

Fig. 1: General software architecture

This architecture integrates essential components for monitoring, data security, and process optimization in the hydrogen industry, utilizing IoT technologies, blockchain, distributed databases, and artificial intelligence. Unlike traditional systems where information is queried at various levels, here it propagates and, through successive aggregations, reaches its destination in real-time so that, on a large scale, the network is not overloaded with large amounts of simultaneous data requests. As illustrated in Fig. 1, the input level consists of connectors for primary data acquisition, structured according to equipment, devices, and installations. This establishes a direct connection with the hardware level and acts as a device driver. The input level plays an essential role in abstracting the existing hardware. The system monitors environmental parameters through weather stations and specialized sensors, data on solar energy production, inverter output, and energy consumption in the electrolysis process. Additionally, it tracks, in real-time, compression levels, stored hydrogen volume, and tank safety. Related data that can be correlated at higher decision-making levels to optimize production flows are also collected, such as utility prices, hydrogen, or energy costs. The intermediate cache storage level collects primary data and synchronizes it at time intervals. This level serves as a transit point for data, without maintaining a history, processing, or modifying the information in any way, acting as a buffer for building data flows destined for subsequent levels. At the output level of the cache, connectors perform data processing and aggregation. These connect to databases, forward data to other cache-type systems and beyond, or aggregate data using customized algorithms as is shown în Fig. 2.

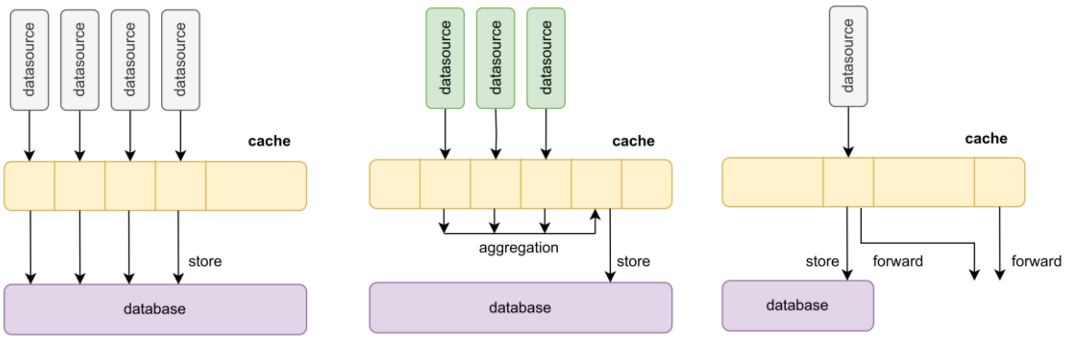

Fig. 2: Ways to use the cache engine

From a security perspective, vulnerable IoT devices are isolated in sub-networks that communicate exclusively with the cache system, thereby eliminating direct access to these devices. Additionally, a blockchain-based subsystem is used to ensure the integrity and protection of sensitive information. Elasticsearch is used for storing and analyzing large volumes of data, while SQLite serves as a testing infrastructure. Access to these data is facilitated through an API layer, allowing the system to interconnect with other applications and cloud services. Artificial intelligence agents use these data to optimize decisions related to energy and hydrogen production, storage, and sales, anticipating production fluctuations or suggesting business decisions based on input conditions from the process as well as market trends. The software architecture is agnostic, distributed, and microservices-oriented. This architectural choice was made to ensure the highest degree of horizontal scalability for the infrastructure. A key aspect of this data acquisition mechanism, with a cache-based system at its core, is its flexibility, as it can be developed at any scale. Regardless of the size or complexity of the application, this system can maintain optimal performance and efficiently manage resources. Its performance remains constant even in the face of large volumes of data or large-scale applications.

The mathematical model

To mathematically model a system that acquires data from IoT devices in a cache-based system, several aspects are considered, such as data flow, acquisition frequency, and cache capacity. The basic data storage unit in the cache system is the cell.

Notations and definitions:

- D_i IoT device i;
- $C(t)$ cache capacity at time t;
- Δt the time interval for data acquisition;
- $\lambda i(t)$ data rate generated by device i at time t (for example, bytes/second);
- $\mu(t)$ the rate at which the cache processes and flushes data (eg bytes/second);
- $Q(t)$ the amount of data stored in the cache at time t;
- t_{sample} the sampling interval of the application that collects data from the cache;
- t_{avail} the moment when the data becomes available in the final application;
- t_{gen} the original moment of event generation by the IoT device;
- t_{send} the time when the device sends the data to the cache;
- t_{cache} the time when the data is written in the cache (the timestamp added by the cache);
- Δt_{net} network and processing latency between IoT device and cache;
- Δt_{proc} the processing time in the IoT device before sending the data to the cache;
- Δt_{total} the total time after which the value collected by the IoT becomes available in the final application;

Each device Di sends data to the cache at a rate $\lambda i(t)$. The total amount of data received by the cache at time t is:

$$\Lambda(t) = \sum_{1}^{n} \lambda_i\,(t) \tag{1}$$

where n is the total number of IoT devices. The cache stores the data received at time t according to the following relation:

$$Q(t + \Delta t) = Q(t) + \Lambda(t)\Delta t - \mu(t)\Delta t \tag{2}$$

This equation shows that the data in the cache at time $t+\Delta t$ is the existing data $Q(t)$, plus the new data received $\Lambda(t)\Delta t$, minus the data processed $\mu(t)\Delta t$. The cache has a limited capacity $C(t)$. If at any moment t, $Q(t)$ exceeds $C(t)$, the additional data is lost.

$$Q(t) \leq C(t) \tag{3}$$

Assuming that t_{gen} is the moment when the event is generated by the IoT device. If the IoT device has a certain processing time Δt_{proc} before sending the data, then:

$$t_{send} = t_{gen} + \Delta t_{proc} \tag{4}$$

Therefore, t_{cache} becomes:

$$t_{cache} = t_{gen} + \Delta t_{proc} + \Delta t_{net} \tag{5}$$

To calculate the original time t_{gen} of the event, this equation becomes:

$$t_{gen} = t_{cache} - \left(\Delta t_{proc} + \Delta t_{net}\right) \tag{6}$$

To obtain the original event time t_{gen} , we need to know or estimate the latencies Δt_{proc} and Δt_{net}. These can be determined through experimental measurements or by monitoring the system in real-time. For Δt_{net} , either the network latency between the IoT device and the cache can be measured using test packets or timestamps of known events, or it can be estimated based on network conditions and communication infrastructure. In the application, a fixed-capacity cache $C(t)=1$ was used. In this case, the cache can store only one event at a time, meaning that any new incoming event will overwrite the previous event.

$$Q(t + \Delta t) = min(1, \Lambda(t)\Delta t) \tag{7}$$

Applications collect data from the cache at regular intervals of t_{sample}. The moment t_{avail} at which the data becomes available in the application can be calculated as the first sampling moment after t_{cache}:

$$t_{avail} = t_{cache} + t_{sample} - \left(t_{cache} mod t_{sample}\right) \tag{8}$$

The total time after which the value collected by the IoT device becomes available in the final application is given by the following relation:

$$\boldsymbol{\Delta t_{total} = t_{avail} - t_{gen}} \tag{9}$$

To optimize the sampling period t_{sample} for the application that collects data from the cache, several factors must be considered, including network latencies, data processing time, and cache capacity. The goal of the optimization is to minimize the total data availability time Δt_{total} and to maximize the efficiency of the data collection process. The objective is to optimize the sampling period t_{sample} to minimize Δt_{total} while ensuring that data is collected efficiently without exceeding the cache capacity. In this regard, we can consider the following formulas and criteria:

1. Minimizing total data availability time. Since Δt_{total} depends on t_{sample}, one approach would be to minimize Δt_{total} as a function of t_{sample} :

$$\min_{t_{sample}} \Delta t_{total} = \left(t_{cache} + t_{sample} - \left(t_{cache} mod t_{sample}\right) - t_{gen}\right) \tag{10}$$

This can be solved by searching for an optimal value for t_{sample} that minimizes the wait time for data collection after it has been stored in the cache.

2. Maximizing the sampling frequency. If t_{sample} is too large, the application may miss important data or collect older data than would be ideal. Therefore, we want to maximize the sampling frequency.

$$\max_{t_{sample}} \frac{1}{t_{sample}} \tag{11}$$

under the constraint that Δt_{total} remains acceptable.

3. Breaking down the optimization into intervals. We can optimize t_{sample} based on network and processing latencies. If Δt_{net} and Δt_{proc} are constant or estimated, we can determine t_{sample} so that it aligns well with these latencies.

$$t_{sample} \approx \Delta t_{net} + \Delta t_{proc} \tag{12}$$

4. Discrete optimization. If t_{sample} needs to be a multiple of a discrete time (for example, one second), we can perform a discrete optimization.

(13)

$$t_{sample} = k \times t_{unit}$$

where k is a positive integer, and t_{unit} is the time unit.

The optimal value for k can be chosen by evaluating different values and selecting the one that minimizes Δt_{total}.

5. Cache capacity. It is important to ensure that the sampling period t_{sample} is not too large, so that the cache can be cleared before new data becomes available.

$$\frac{Q(t)}{t_{sample}} \leq C \tag{14}$$

where Q(t) is the amount of data in the cache at time t, and C is the capacity of the cache.

3. Case Study: Data acquisition from a group of photovoltaic parks

The overall objective of the study was to develop an integrated Power-to-X system for the supply of hydrogen, electricity, and thermal energy whose architecture is shown in Fig. 3.

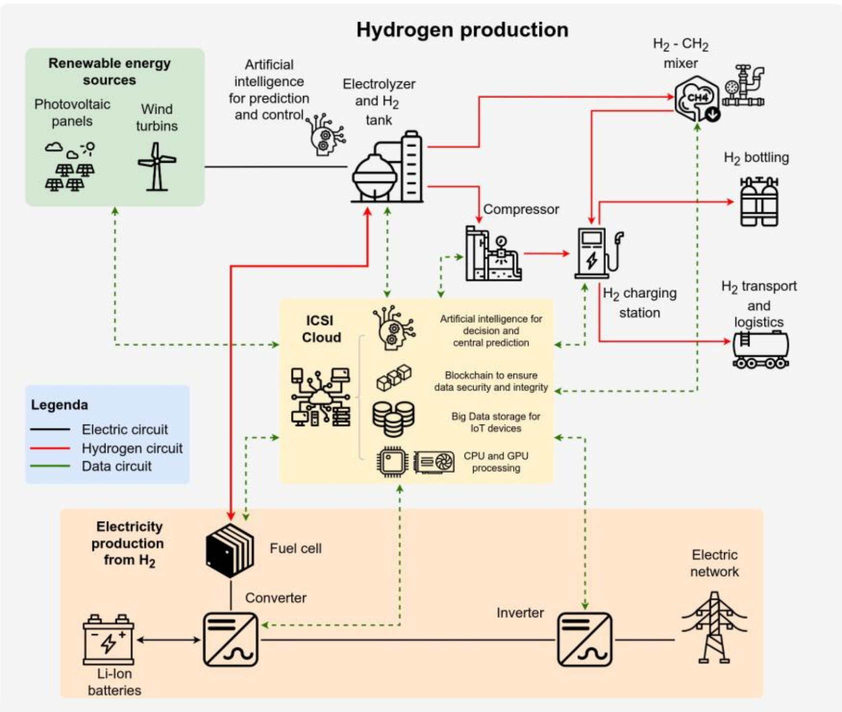

Fig. 3: The Power-to-X System of ICSI [15]

The aim is to facilitate the complete integration and interconnectivity of all components of the integrated HET (Hydrogen-Electricity-Thermal) unit with a synergistic hardware configuration of energy use/conversion/storage technologies (fuel cells, hydrogen storage tanks, battery energy storage systems, H2-CH4 mixing unit, electrochemical cell for hydrogen separation from a gas stream), using artificial intelligence algorithms and multivariate analyses to achieve intelligent energy management [15]. In the Power-to-X system, renewable energy production is ensured by a group of three photovoltaic parks, as shown in Table 1. Inverter manufacturers offer their own cloud solution application for monitoring energy production, but also offer an API for integration that was used to connect to the cache system. The aim of this case study is to acquire, in real-time, the production values from the photovoltaic parks, as well as additional elements useful for machine learning predictions, such as solar radiation and environmental parameters.

Table 1

Solar park	Installed power	Inverter type
A	50Kw	2 x FRONIUS Eco 25.0-3-S
B	10kw	2 x SMA Sunny Tripower STP 5000TL-20
C	10kw	2 x ABB TRIO-5.8-TL-OUTD-S-400

A limitation we had from the beginning was that the software solutions proposed by inverter manufacturers are not interconnectable. Thus, the chosen solution is based on collecting data in a cache (Fig. 4), aggregating it, and saving a snapshot with a sampling period of one minute in a database. This granularity was chosen based on the consideration that reports are generated for time intervals such as date, hour, and minute. The input connector for Fronius inverters present in the photovoltaic panel park uses an API/JSON mechanism provided by the Fronius inverter and acquires electricity production information with a sampling period of 1 second. The input connector for inverters controlled by Solar-Log datalogger devices uses a similar API mechanism to acquire data corresponding to the attached inverters, with a reading sampling period of 15 seconds, limited by the equipment. The aggregation connector is a particular connector that retrieves data about green energy production from the cells allocated to each solar park and calculates the total energy produced in the destination cell in real-time, with a sampling period of 1 second. An output connector retrieves data from the cache system and stores it in the databases for historical purposes, following an agreed schema. The retrieval of this information is done with a sampling period of 1 minute. As a result of data aggregation, the output cell provides the amount of green energy produced at the institute level in real-time (Fig. 5). It is observed that, at the software level, the cell also serves as an endpoint[8] for applications that can directly consume this

[8] Specific URL used to access a resource or functionality

information. Also, this approach is efficient, integrates various inverter manufacturers and can be implemented at very low costs.

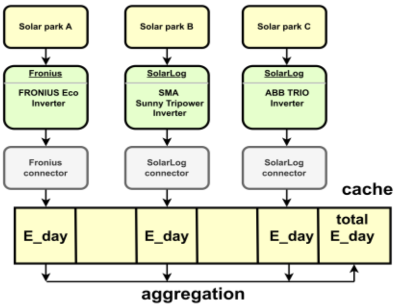

Fig. 4: Data acquisition from non-homogeneous photovoltaic parks.

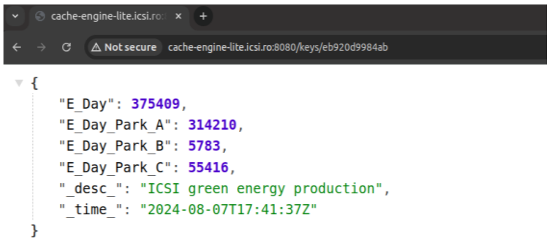

Fig. 5: Total green energy production

A summary of the analyzed metrics is presented in Table 2 below:

Table 2

KPI	Definition	Measured value	Impact
Data latency	Time delay between data generation at inverters and its availability in cache or database.	**Fronius Inverters:** 1.2 seconds on average (1-second sampling + processing overhead). **Solar-Log Devices:** 15.5 seconds (15-second sampling + processing overhead). **Aggregation Connector:** <0.3 seconds additional delay.	Low latency ensures accurate real-time monitoring and timely decision-making.
System uptime	Percentage of time the system is fully operational.	**Average Uptime:** 99.8%. **Downtime Causes:** API unavailability and scheduled maintenance. **Target:** Maintain >99.9% with API redundancy and enhanced monitoring.	High uptime ensures continuous energy monitoring and uninterrupted AI-driven optimization.
Data aggregation efficiency	Ability to process and aggregate data from heterogeneous sources within sampling intervals.	**Success Rate:** >99.5%. **Failures:** 0.5%, mainly due to network issues or API timeouts.	Reliable aggregation ensures a holistic and real-time view of energy production.

4. Conclusions

The use of a cache system in the IoT context offers multiple essential advantages for improving system performance and efficiency. Although the most common use of cache systems is to reduce latency by locally storing frequently accessed data, this study has explored their potential in managing real-time concurrent access to information. The proposed technique enables ultra-fast data access in large-scale IoT infrastructures, making it highly suitable for distributed real-time applications, where caching saves bandwidth, reduces network load, minimizes operational costs, and enhances communication stability. Unlike conventional caching solutions, which are primarily used to reduce latency, the proposed approach optimizes the scalability of communications in IoT infrastructures, allowing efficient management of a large number of devices without overloading the network. This is a key difference from traditional methods, which become inefficient in large-scale distributed environments. The proposed system reduces network congestion and improves concurrent data access, making it ideal for distributed real-time applications. In this context, caching is not only used to accelerate data access but also as an essential mechanism for balancing and optimizing communication flows in an extensive IoT network. To validate the efficiency of the proposed system, its behavior was analyzed in scenarios involving a large number of IoT devices, demonstrating more efficient management of concurrent access and a significant reduction in network overload compared to traditional methods. Future directions for this research could include integrating predictive analytics and machine learning to optimize real-time data management. These advancements could improve the system's adaptability, enabling it to anticipate data access patterns and dynamically enhance its performance. Beyond renewable energy, the industrial relevance of caching could extend to applications such as industrial production, supply chain optimization, and healthcare monitoring systems, where fast data access and reliability are equally critical. Integrating a cache system in this configuration represents a strategic solution for optimizing performance, reliability, and cost efficiency, paving the way for more robust and scalable IoT solutions in industrial and commercial domains.

Acknowledgements

This work was supported by a grant from the European Regional Development Fund, within the Competitiveness Operational Program, in the frame of the project "Increasing the research capacity of ICSI Ramnicu Vâlcea through the development of a CLOUD infrastructure connected to global information resources, 4C-ICSI", SMIS 125119, project no. 345/2021, and by a project supported by the Romanian Ministry of Research, Innovation, and Digitization under the National Plan for Research and Development, contract 20N/2023, project PN 23 15 01 01**.**

R E F E R E N C E S

[1] Kim CK, Kim T, Lee S, Lee S, Cho A, Kim MS. Delay-aware distributed program caching for IoT-edge networks. PLoS One. 2022 Jul 19;

[2] M. I. Aziz Zahed, I. Ahmad, D. Habibi and Q. V. Phung, "Content Caching in Industrial IoT: Security and Energy Considerations," in IEEE Internet of Things Journal, vol. 7, no. 1, pp. 491-504, Jan. 2020;

[3] J. Yang, C. Ma, B. Jiang, G. Ding, G. Zheng and H. Wang, "Joint Optimization in Cached-Enabled Heterogeneous Network for Efficient Industrial IoT," in IEEE Journal on Selected Areas in Communications, vol. 38, no. 5, pp. 831-844, May 2020;

[4] Bhatia, J.; Italiya, K.; Jadeja, K.; Kumhar, M.; Chauhan, U.; Tanwar, S.; Bhavsar, M.; Sharma, R.; Manea, D.L.; Verdes, M.; et al. An Overview of Fog Data Analytics for IoT Applications. Sensors 2023, 23, 199.

[5] Chenaru, O. (2021). Gateway for Secure IIoT Integration in Industrial Control Applications. U.P.B. Scientific Bulletin, Series C, 83(3), 117-128. https://www.scientificbulletin.upb.ro/rev_docs_arhiva/full273_572235.pdf

[6] Irimia, D-B., Lupu C. (2010). Process control over time delayed communication systems.https://www.researchgate.net/publication/229013110_Process_control_over_time_delayed_communication_systems

[7] Irimia, D.-B., Lupu, C., Junie, P. (2011). Control solution for processes with time delayed communication components, U.P.B. Scientific Bulletin, Series C, 73(2). ISSN 1454-234X.

[8] R.V. Ravi, S.B. Goyal, N.B. Constantin, M.S. Raboaca, C. Verma and V. Kustov, "Monitoring and Control of Electrical Sub Station Using IoT and Cloud," 2022 International Conference and Exposition on Electrical and Power Engineering (EPE), Iasi, Romania, 2022, pp. 654-657;

[9] Anand, Pooja & Selwal, Arvind & Singh, Pradeep & Raboaca, Maria & Felseghi, Raluca. (2020). IoVT: Internet of Vulnerable Things? Threat Architecture, Attack Surfaces, and Vulnerabilities in Internet of Things and Its Applications towards Smart Grids. Energies. 13. 4813. 10.3390/en13184813.

[10] Zahed, Mostafa & Ahmad, Iftekhar & Habibi, Daryoush & Phung, Quoc Viet. (2019). Content Caching in Industrial IoT: Security and Energy Considerations. IEEE Internet of Things Journal. PP. 1-1. 10.1109/JIOT.2019.2948147.

[11] Bowen Liu, Xutong Jiang, Xin He, Lianyong Qi, Xiaolong Xu, Xiaokang Wang, Wanchun Dou, A deep learning-based edge caching optimization method for cost-driven planning process over IIoT,Journal of Parallel and Distributed Computing, Volume 168,2022, Pages 80-89, ISSN0743-7315; https://www.sciencedirect.com/science/article/pii/S0743731522001435

[12] Hasan, K.; Jeong, S.-H. Efficient Caching for Data-Driven IoT Applications and Fast Content Delivery with Low Latency in ICN. Appl. Sci. 2019, 9, 4730;

[13] Ivan Zyrianoff, Lorenzo Gigli, Federico Montori, Luca Sciullo, Carlos Kamienski, Marco Di Felice, CACHE-IT: A distributed architecture for proactive edge caching in heterogeneous IoT scenarios, Ad Hoc Networks, Volume 156, 2024, 103413, ISSN 1570-8705; https://www.sciencedirect.com/science/article/pii/S1570870524000246

[14] Drăgănescu, C., Cristescu, G., Chenaru, O. (2021). Mechanism for Testing and Improving the Robustness of Smart Manufacturing Systems. U.P.B. Sci. Bull., Series C, Vol. 83, Iss. 2.

[15] PN 23 15 01 01, Development and demonstration of a synergistic resilient operating model of a hydrogen-based energy structure, using artificial intelligence – HyEnergy – HyEnergy, https://www.icsi.ro/pn-23-15-01-01