Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 5, 607-614
2025
Publisher: Learning Gate
DOI: 10.55214/25768484.v9i5.6956

Testing real-time system algorithms performance on synthetic Data: Analytical study of hard and soft system tasks

Noor K. Younis[1], Marwa Riyadh Ahmed[1], Azhar W. Talab[1], Rabei Raad Ali[1]*
[1]Technical Engineering College for Computer and AI, Northern Technical University, 41000, Mosul, Iraq
rabei@ntu.edu.iq (R.R.A).

Abstract: This study compares RMS, EDF, and LLF on synthetic datasets for hard and soft real-time systems to assess their feasibility and effectiveness in supporting real-time systems for various utilization levels. It has been designed to give each algorithm's various strengths, limits, and applicability in real-time application scenarios through total utilization computation and schedule-up-to-do ability analysis. It has been concluded that RMS is not schedulable due to its overutilization, while EDF is infeasible at Total > 1.0 for hard and soft real-time. LLF has limitations due to overutilization and frequent preemptions, making it suitable for soft real-time systems, unlike hard systems, because of the limitations of overutilization and frequent preemptions. RMS and EDF cannot meet deadlines under hard and soft real-time conditions. Future work should focus on hybrid algorithms or load balancing to overcome these limitations and process data in real time without tasks going beyond the time available to them in the CPU.

Keywords: *Early deadline first, Feasibility, Utilization, Least Laxity First, Rate Monotonic Scheduling, Real time system, Synthetic datasets.*

1. Introduction

Growing demand for them, which has led to the quick development of various techniques for scheduling simple multiprocessor tasks [1]. However, these systems have trouble keeping in synchronization when accessing shared resources. Real-time systems consist of two types. Missing a deadline can have serious consequences due to real-time systems' extremely tight timing constraints, including equipment damage or human injury. For crucial jobs, these systems usually need to respond in a certain amount of time and exhibit deterministic and predictable behavior. Soft Real-Time Systems prioritize timely processing, allowing for late processing without causing system failure or loss of life or property [2].

In this study, several scheduling techniques and multi-processor environments are systematically reviewed to evaluate current synchronization protocols for shared resources in real time [3]. A comparison of several scheduling methods and algorithms is also included in the publication, along with a multi-metric-based analysis of resource scheduling on a synthetic job dataset. Rate Monotonic Scheduling (RMS), Early Deadline First (EDF), and Least Laxity First (LLF) are the three suggested scheduling methods [4]. Multi-processor platforms and various systems, including hard and soft real-time systems, are required for the implementation of real-time systems. The two most significant issues in multi-processor environments are scheduling and synchronization, as well as system usage and practicality [5]. The nature of processors in embedded real-time systems is also evolving regularly.

The popularity of multi-core real-time systems in recent years has led to the quick development of various scheduling strategies for core multi-processor tasks; however, these systems have limitations when it comes to synchronization for shared resource access [6].

History: Received: 17 February 2025; Revised: 21 April 2025; Accepted: 24 April 2025; Published: 7 May 2025
* Correspondence: rabei@ntu.edu.iq

This study compares RMS, EDF, and LLF on synthetic datasets for hard and soft real-time systems. Also, LLF has limitations due to overutilization and frequent preemptions, making it suitable for soft real-time systems on synthetic datasets.

The study's remaining sections are structured as follows: Section 2 examines earlier research related to this study. The work strategy, standards, algorithms, and methodology are explained in Section 3. Experimental results, processing algorithm results, suggested criteria, and standard values were all provided and reviewed in Section 4. Section 5: Talk about the results and how to evaluate them. Information on the conclusions drawn is provided in Section 6.

2. Related Work

In Ismail and Jawawi [7] Ismail and Jawawi proposed a weakly hard scheduling approach for real-time systems, focusing on multiprocessor environments. The approach uses partitioned multiprocessor scheduling techniques, hyper period analysis, and deadline models to guarantee task timing requirements. The study uses a MATLAB simulation tool to validate the results, and the results show that the proposed approach outperforms existing approaches in terms of task deadline satisfaction. This approach is particularly useful for multiprocessor environments where task allocation is even harder than in uniprocessor cases.

In Deming, et al. [8] and Salih and Younis [9] the authors explored real-time task scheduling for optimizing energy use in renewable energy integration into smart grids. It analyses fluctuations in renewable energy, evaluates optimization methods, and identifies obstacles. Findings show renewable energy fluctuations impact power system stability, requiring advanced prediction methods and energy storage. Policy implications include supporting advanced technologies, encouraging real-time scheduling, and enhancing grid infrastructure for a reliable smart grid and sustainable future.

In Abolhassani Khajeh, et al. [10] and Alsammak and Mohammed [11] the Internet of Things (IoT) is a rapidly evolving telecommunication network that has significantly impacted various aspects of life. It includes hardware, telephony, communications, storage, secure platforms, software, and data processing platforms. This article reviews real-time scheduling in IoT from 2018 to 2022, focusing on practical applications such as healthcare, infrastructure, industrial applications, smart cities, commercial applications, environmental protection, and general IoT applications. The study analysed 162 articles published between 2018 and 2020, with 35 focusing on real-time IoT scheduling with the RMS algorithm. The study found that industrial energy efficiency had the highest percentage of implemented approaches, followed by healthcare, commercial sales systems, smart buildings, urban, traffic monitoring, smart homes, and smart farms. The cost had the highest rate of real-time scheduling in IoT at 19%, and this result motivated us to work more on RTS scheduling and search for better techniques.

In Wang and Li [12] discovered that, multiple tasks with certain timing requirements make up a real-time system (RTS). A system whose response time is a key factor in determining proper operation is referred to as an RTS. without and with deadlines, Periodic or aperiodic, all the immediate tasks in an RTS can be scheduled using either fixed or dynamic priorities, including least laxity first (LLF) and earliest deadline first (EDF), using pre-emptive or non-pre-emptive schemes. This chapter examines these traditional scheduling systems using an RTS example. For real-time systems, real-time reconfiguration is crucial because it can be used to identify potential safe execution sequences if the system cannot be scheduled. To save the entire system in the event of sporadic disruptions, a reconfiguration scenario could involve adding, removing, or updating tasks at runtime. One popular online reconfiguration approach that primarily reconfigures the periods of real-time jobs is the elastic period model. The elastic period model mostly influences the RTS reconfigurations throughout the remainder of this work.

3. Methodology

The study compares three real-time scheduling algorithms: Earliest Deadline First (EDF), Rate Monotonic Scheduling (RMS), and Laxity First (LF) on a synthetic dataset. The focus is on their

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 5: 607-614, 2025
DOI: 10.55214/25768484.v9i5.6956

utilization feasibility and ability to meet task deadlines in hard and soft real-time systems. Key metrics analysed include processor utilization, feasibility checks, and missed deadlines [13]. The comparative analysis provides insights into algorithm suitability for different real-time scenarios, highlighting trade-offs between predictability, flexibility, optimization [14] and computational complexity as in Figure 1.

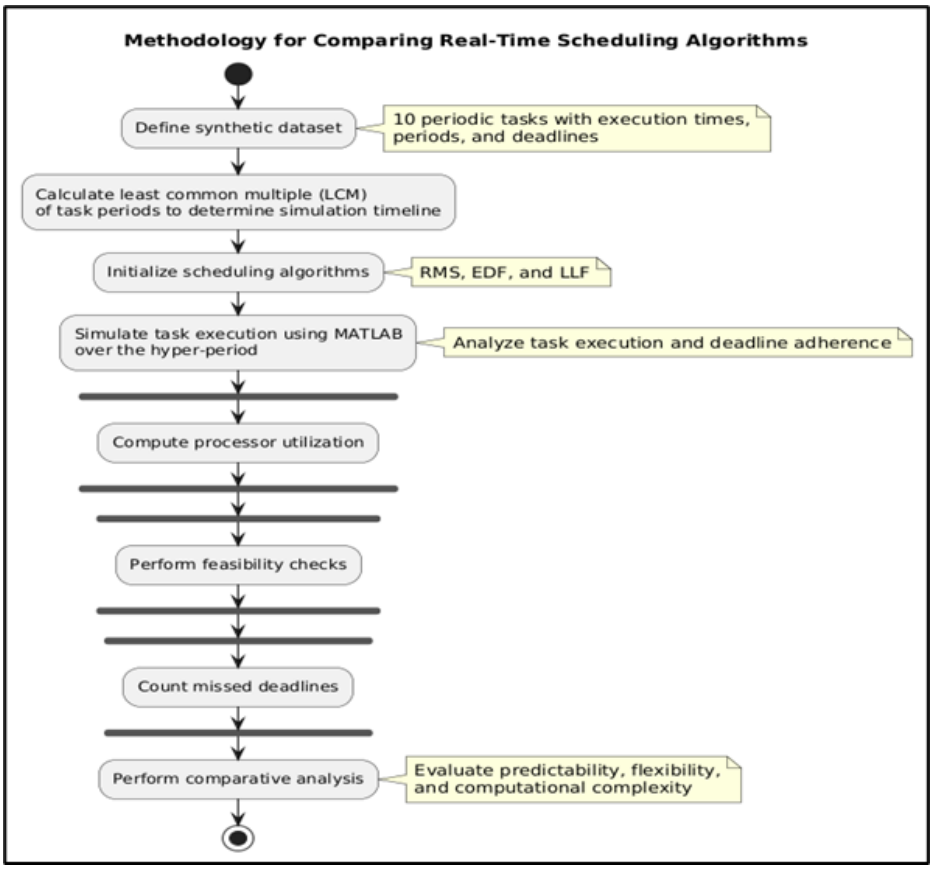

Figure 1.
Comparing real-time system scheduling algorithms.

A synthetic dataset of 10 periodic tasks was designed with attributes such as execution times, periods, and deadlines. The dataset ensures varied levels of utilization and system load to test algorithm efficiency. The least common multiple of the task periods was calculated to determine the simulation timeline. MATLAB applies the algorithms to simulate task execution over the hyper period.

3.1. Rate Monotonic Scheduling (RMS)

One priority approach that falls within Real Time Operating Systems' static priority scheduling category is rate monotonic scheduling. The nature of it is pre-emptive. The cycle time of the involved processes is used to determine the priority. The process has the highest priority if its task duration is short [15]. As a result, the process with the highest priority will start execution before the other processes. A process's priority is inversely related to how long it will take. Only when a group of processes meets the following criteria can they be scheduled [16]:

$$U=\sum_{i=1}^{n}\frac{Ci}{Ti} <= n(2^{1/n}-1) \quad (1)$$

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 5: 607-614, 2025
DOI: 10.55214/25768484.v9i5.6956

where U is the processor utilization, Ti is the time for the process to execute, Ci is the calculation time of the process, and n is the number of processes in the process set.

3.2. Earliest Deadline First (EDF)

It is a dynamic priority scheduling algorithm used in real-time systems for static and dynamic scheduling. assigns tasks based on absolute deadlines, with the closest deadline receiving the highest priority. EDF is efficient, allowing CPU utilization to be around 100% while maintaining task deadlines. It also includes kernel overload, ensuring all tasks meet deadlines [17]. EDF does not require periodic tasks or processes and allows preemption if a previous periodic instance with an earlier deadline becomes active. If the system is overloaded (total utilization > 100%), EDF might not guarantee that deadlines are met [18].

3.3. Least Laxity First (LLF)

It is a dynamic, prioritized algorithm for real-time scheduling. All the system's duties in LST are given priority based on their free time. The highest priority task is the one with the least amount of free time, and vice versa [19]. The tasks' priorities are changed on the flight. The following formula can be used to determine slack time [20]:

$$\text{Slack\_ time} = (D - t - e') \quad (2)$$

Where D: Deadline of the task, t: Real time when the cycle starts, and e: Remaining Execution Time of the task. The Least Slack Time scheduling algorithm is different from Earliest Deadline First due to its need for task execution times, which can be impractical in real-time systems. It may underutilize the CPU, decreasing efficiency and throughput. Tasks with similar slack time are dispatched first [19].

3.4. Synthetic Dataset

Information that is created artificially; rather, the term "synthetic data" refers to data that is derived from genuine events. It's generated using an algorithm. serves as a substitute for test sets of operational or production data to train machine learning (ML) models and validate mathematical models for real-time scheduling. While gathering high-quality data from the real world is difficult, expensive, and time-consuming, synthetic data technology enables users to quickly, simply, and digitally synthesize the data in any quantity they choose, customized to meet their own needs [21].

4. Experimental Result

Synthetic data was used to generate 10 tasks because of real events with synthetic data to test operational or production data and validate the mathematical models for real-time scheduling for each of the RMS, EDF, and LLF algorithms. Synthetic data technology allows data to be synthesized quickly, easily, and digitally. The algorithmic results for all tasks are tested and compared within the CPU to see if this data is valid for the real-time instance on the three algorithms, as well as their implementation potential for real-time systems, whether the system is hard or soft. A strong framework for methodically assessing and contrasting real-time scheduling algorithms is provided by synthetic data, which also offers insights into the algorithms' advantages and disadvantages in a controlled and repeatable way [22]. Using the MATLAB language, each algorithm is represented by its sequential steps and mathematical model to implement the proposed tasks and compare them in soft and hard real time.

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 5: 607-614, 2025
DOI: 10.55214/25768484.v9i5.6956

Table 1.
Synthetic dataset with 10 tasks.

Task ID	Execution Time (C)	Deadline (D)	Period (T)
T1	1	3	5
T2	2	4	7
T3	1	5	10
T4	3	8	15
T5	2	9	12
T6	4	10	20
T7	1	12	15
T8	2	14	18
T9	1	16	20
T10	3	20	25

Depending on Table 1, the dataset includes a mix of shorter, longer deadlines, and periods. Now we can start to work on this program to determine whether deadlines are fulfilled within the hyper period (the LCM of periods) by applying RMS, EDF, and LLF, then analysing utilization in a repeatable way.

4.1. RMS Task Set

Table 2 shows the Gantt Chart (Visualization Over a 20-Unit Period) under RMS and how operations are performed inside the CPU.

Table 2.
Gantt chart RMS.

Time	Task Executing
0-1	T1
1-3	T2
3-4	T1
4-5	T5
5-6	T3
6-7	T2
7-9	T4
9-10	T3
10-11	T2
11-12	T5
12-14	T4
14-15	T7
15-16	T2
16-17	T8
17-18	T6
18-19	T9
19-20	Idle

4.2. EDF Task Set

The Gantt Chart (Visualization Over a 20-Unit Period) under EDF and the way operations are carried out inside the CPU are displayed in Table 3.

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 5: 607-614, 2025
DOI: 10.55214/25768484.v9i5.6956

Table 2.
Gantt chart EDF.

Time	Task Executing
0-1	T1
1-3	T2
3-4	T1
4-5	T5
5-6	T3
6-7	T2
7-9	T4
9-10	T3
10-11	T2
11-12	T5
12-14	T4
14-15	T7
15-16	T2
16-17	T8
17-18	T6
18-19	T9
19-20	Idle

4.3. LLF Task Set

Figure 2 shows the Gantt Chart (Visualization Over a 20-unit Period) under LLF and how operations are performed inside the CPU. The results of the 3 algorithms bring out some very marked characteristics of the algorithms. RMS favors fixed periods, which brings simplicity at the probable infeasibility at high loads. EDF adjusts dynamically depending on the deadlines to ensure maximum schedule ability but requires more computation. LLF focuses on urgency through laxity, thereby minimizing missed deadlines, but risks frequent preemptions with possible consequences of affecting system stability under load.

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 5: 607-614, 2025
DOI: 10.55214/25768484.v9i5.6956

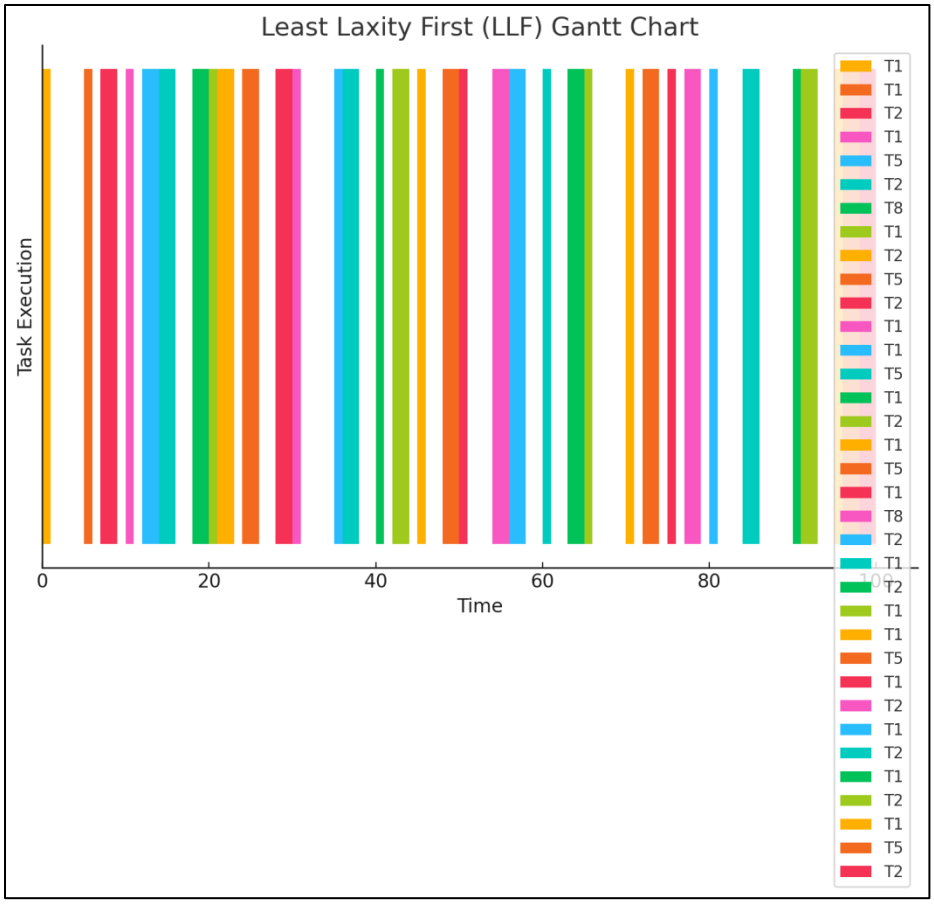

Figure 2.
Least laxity first Gantt chart.

6. Conclusion

The study compares RMS, EDF, and LLF on synthetic datasets for hard and soft real-time systems. RMS is not schedulable due to its overutilization for hard and soft RTS, while EDF is infeasible at total > 1.0. This failure points out the well-recognized limit of RMS and EDF in handling overloads, where deadlines can't be guaranteed by hard real-time systems. LLF has limitations due to overutilization and frequent preemption, making it suitable for soft real-time systems on synthetic datasets. LLF can handle overload conditions for soft real-time systems somewhat better because it adapts, whereas hard real-time systems cannot handle such a high level of overload. Future work should focus on hybrid algorithms or load balancing to overcome these limitations and optimal handling of real-time synthetic datasets.

Transparency:

The authors confirm that the manuscript is an honest, accurate, and transparent account of the study; that no vital features of the study have been omitted; and that any discrepancies from the study as planned have been explained. This study followed all ethical practices during writing.

Copyright:

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 5: 607-614, 2025
DOI: 10.55214/25768484.v9i5.6956

References

[1] F. Al-Dolaimy *et al.*, "Hybrid optimization with enhanced QoS-based path selection in VANETs," *International Journal of Intelligent Engineering & Systems*, vol. 16, no. 4, pp. 69-80, 2023. https://doi.org/10.22266/ijies2023.0831.06

[2] A. Q. Al-Dujaili, A. H. Shallal, A. H. Sabry, Y. M. Alkubaisi, and A. J. Humaidi, "Maximizing solar energy utilization and controlling electrical consumption in domestic water heaters by integrating with aluminum reflector," *Measurement*, vol. 230, p. 114558, 2024.

[3] N. Tantalaki, S. Souravlas, and M. Roumeliotis, "A review on big data real-time stream processing and its scheduling techniques," *International Journal of Parallel, Emergent and Distributed Systems*, vol. 35, no. 5, pp. 571-601, 2020.

[4] A. Kumar and S. K. Gupta, "A systematic survey of multiprocessor real-time scheduling and synchronization protocol," *International Journal of Sensors Wireless Communications and Control*, vol. 12, no. 3, pp. 212-229, 2022. https://doi.org/10.2174/2210327912666220105141851

[5] M. Cinque, "Virtualizing real-time processing units in multi-processor systems-on-chip," presented at the 2021 IEEE 6th International Forum on Research and Technology for Society and Industry (RTSI), IEEE, 2021.

[6] G. Xie, X. Xiao, H. Peng, R. Li, and K. Li, "A survey of low-energy parallel scheduling algorithms," *IEEE Transactions on Sustainable Computing*, vol. 7, no. 1, pp. 27-46, 2021. https://doi.org/10.1109/TSUSC.2020.2981234

[7] H. Ismail and D. N. Jawawi, "A weakly hard scheduling approach of partitioned scheduling on multiprocessor systems," *Jurnal Teknologi*, vol. 77, no. 9, pp. 1-12, 2015.

[8] C. Deming, P. Pasam, A. R. Allam, R. Mohammed, S. G. N. Venkata, and K. R. V. Kothapalli, "Real-time scheduling for energy optimization: Smart grid integration with renewable energy," *Asia Pacific Journal of Energy and Environment*, vol. 8, no. 2, pp. 77-88, 2021.

[9] T. A. Salih and N. K. Younis, "Designing an intelligent real-time public transportation monitoring system based on iot," *Open Access Library Journal*, vol. 8, no. 10, pp. 1-14, 2021. https://doi.org/10.4236/oalib.1107985

[10] S. Abolhassani Khajeh, M. Saberikamarposhti, and A. M. Rahmani, "Real-time scheduling in IoT applications: a systematic review," *Sensors*, vol. 23, no. 1, p. 232, 2022. https://doi.org/10.3390/s23010232

[11] H. N. Alsammak and Z. S. Mohammed, "Internet of things (IoT) work and communication technologies in smart farm irrigation management: A survey," *NTU Journal of Engineering and Technology*, vol. 1, no. 3, pp. 49–65, 2022. https://doi.org/10.56286/ntujet.v1i3.182

[12] X. Wang and Z. Li, "Real-time scheduling and reconfiguration," *Scheduling and Reconfiguration of Real-Time Systems: A Supervisory Control Approach*, pp. 55-70, 2023.

[13] C. A. Sari, M. H. Dzaki, E. H. Rachmawanto, R. R. Ali, and M. Doheir, "High PSNR using fibonacci sequences in classical cryptography and steganography using LSB," *International Journal of Intelligent Engineering & Systems*, vol. 16, no. 4, pp. 1-13, 2023.

[14] R. R. Ali, W. S. Al-Dayyeni, S. S. Gunasekaran, S. A. Mostafa, A. H. Abdulkader, and E. H. Rachmawanto, "Content-based feature extraction and extreme learning machine for optimizing file cluster types identification. In Future of Information and Communication Conference," presented at the Future of Information and Communication Conference, Cham: Springer International Publishing, 2022, 314-325.

[15] Z. Aliyah, R. M. R. Pambudhi, A. H. Ahnafi, H. H. Nuha, and S. Prabowo, "Comparison of earliest deadline first and rate monotonic scheduling in polling server," presented at the 2020 8th International Conference on Information and Communication Technology (ICoICT), IEEE, 2020.

[16] J. Cui, "Schedulability analysis of rate-monotonic algorithm on concurrent execution of digraph real-time tasks," International Workshop on Structured Object-Oriented Formal Language and Method, Springer, 2022.

[17] M. Chetto and R. El Osta, "Earliest deadline first scheduling for real-time computing in sustainable sensors," *Sustainability*, vol. 15, no. 5, p. 3972, 2023. https://doi.org/10.3390/su15053972

[18] G. Von Der Brüggen, "Efficiently approximating the worst-case deadline failure probability under EDF," presented at the 2021 IEEE Real-Time Systems Symposium (RTSS) IEEE, 2021.

[19] N. Chen, C. Kurniawan, Y. Nakahira, L. Chen, and S. H. Low, "Smoothed least-laxity-first algorithm for electric vehicle charging: Online decision and performance analysis with resource augmentation," *IEEE Transactions on Smart Grid*, vol. 13, no. 3, pp. 2209-2217, 2021.

[20] A. Finzi, S. S. Craciunas, and M. Boyer, "Integrating sporadic events in time-triggered systems via affine envelope approximations," presented at the 2024 IEEE 30th Real-Time and Embedded Technology and Applications Symposium (RTAS), IEEE, 2024.

[21] E. Wood, "Fake it till you make it: face analysis in the wild using synthetic data alone," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.

[22] M. Roberts, "Hypersim: A photorealistic synthetic dataset for holistic indoor scene understanding," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 5: 607-614, 2025
DOI: 10.55214/25768484.v9i5.6956