OPEN ACCESS
International Journal of Innovations in Science & Technology

Challenges and Practices Identification via Systematic Literature Review in the Design of Green/Energy-Efficient Embedded Real-Time Systems

Muhammad Salman[1], Siffat Ullah Khan[2], Nasir Rashid[3]
[1,2,3]Department of Computer Science and IT, University of Malakand, Pakistan
* Correspondence: Nasir Rashid nasir@uom.edu.pk

Citation | Salman. M, Khan. S. U, Rashid. N, "Challenges and Practices Identification via Systematic Literature Review in the Design of Green/Energy-Efficient Embedded Real-Time Systems", IJIST, Vol. 07 Issue. 01 pp 190-209, Jan 2025
DOI | https://doi.org/10.33411/ijist/202571190209
Received| Dec 15, 2024 **Revised|** Jan 16, 2025 **Accepted|** Jan 20, 2025 **Published|** Jan 22, 2025.

As most embedded devices are portable, that is they are operated by batteries, early battery exhaustion is likely to cause the failure of the embedded real-time systems (ERTS). Therefore, developers and users enjoy the services of the ERTS but face green and energy consumption challenges. Studies show that attempting to design green ERTS may lead to some serious issues or deteriorate some of the quality characteristics of the embedded systems. Energy conservation in ERTS has continued to be an area of interest in the past years. Energy efficiency or certain quality features are considered while designing ERTS, but these two factors are not often considered together because they have direct impact on each other in ERTS. The purpose of this research is to identify the challenges in the design of green ERTS and the solutions that can be employed to address those challenges. A review of the relevant literature was conducted to define the problems and practices under consideration. Based on a comprehensive Systematic Literature Review (SLR), we have found 8 challenges and 34 practices from 65 papers in the green ERTS context. The results of our SLR will help us develop a framework for creating green ERTS in the future.

Keywords: Energy-Efficient Embedded Real-Time Systems; Green Embedded Real-Time Systems; Challenges; Practices; Systematic Literature Review

Introduction:

ERTS consist of two main components: software and hardware, both of which operate within a real-world environment [1, 2]. The software is integrated into the hardware system, enabling interaction with external systems. A key requirement of ERTSs is meeting specific response times while performing tasks. Involving execution time, system often fail to produce the required result within real-time constraint, a chronic issue of the embedded system. Thus, an ERTS is a system that performs a specific task in a specified period [3]. Moreover, an ERTS is a computer application that can execute a job and retaliate to the exterior environment within a specified time [4]. ERTS are computer systems that examine, monitor, retaliate to, or manage exterior events. These events are linked to the computer system via input-output linkages such as sensors and actuators; it can be comprised in any form, either biological or physical object's structure. Typically, humans are part of the external environment, but a wide range of artificial and natural entities, including animals, can also be involved. ERTSs must adhere to the timing constraints imposed by the external environment they interact with, which is why they are referred to as real-time systems. Another term commonly used for such systems is "reactive systems," as noted in various publications. This is because one of their primary objectives is to respond to signals from their surroundings. An ERTS can also function as a component within a larger system [5].

Designing ERTS is a complex task, and advancements in digital system design point toward future directions for improvement [6]. During the design of ERTS, some designers may overlook important factors such as performance, reliability, and safety. This can lead to additional challenges, as the system may face issues stemming from insufficient attention to these aspects during the design phase [7]. In case of ERTS, the security is not a design goal for the designers; rather, physical security is employed to secure access [8]. Although the use of ERTSs in sensitive applications, such as medical devices, is growing exponentially, security concerns have become increasingly critical during the design of these systems [9]. The study of general threats affect security as SCA [10,11], LMFAs [12], SLCAs, CIAs [13,14] and others. Another approach to addressing system security is by implementing security services, which encompass a security-aware model and framework, security-driven hardware design, and encryption algorithms. These services are intended to be more power intensive to create. Since many ERTS are dependent on low or unbalanced cell/battery or renewable energy, the power supply capability is considerably restricted. Consequently, energy efficiency becomes an important factor which must be addressed during the design of security services.

However, there are two kinds of problems that still hinder ERTSs from being applied extensively [30, 31]. The first one is that in most cases, an ERTS's energy sources, e.g., power harvesters and cell/battery are energy limited [32]. The second issue is that transient faults in ERTSs may occur due to external disturbances or imbalanced energy supply [33]. A model-driven approach has been proposed to design secure ERTS concerning their energy consumption [34]. Some of the embedded systems are charged with battery or cell and the majority of it has critical power control. While some ERTSs can survive on a single cell for weeks, months, or years, many can garner battery power daily. An attacker may attempt to disable ERTSs by, for example, draining power from the device's battery, even without penetrating the system. This vulnerability is particularly critical for battery-powered devices [35]. Attempts to evaluate power are important because power considerations are crucial in ERTSs, impacting their performance and reliability. However, there are several approaches proposed for modeling secure ERTSs, which are [36] and [37]. The drawback is that the research efforts mentioned above have typically been developed independently, even though these aspects are closely interconnected within an ERTS [34]. Contemporary approaches for task scheduling rarely take into consideration both security management and energy efficiency; they do so either or. software programs that provide services in security [38].

From the above discussion, we conclude that in developing ERTS, either security management or energy efficiency is considered during the design phase, and these two factors are not often considered together. The same way each of them is met by interacting with the other; for example, meeting the security requirement impacts other system requirements such as the performance, energy, and maintainability requirements. Similarly, achieving energy efficiency goals during the development of ERTSs may require compromising on other system quality features, such as performance, security, and timing, among others. Thus, the purpose of this research is to analyze the development of ERTS to identify the list of the design challenges for green/energy efficient ERTS as well as the solutions/practice to address them.

Novelty Statement:

Literature reveals that no such research efforts have been reported using SLR on the subject domain till date. Thus, we have planned to use SLR, followed by an industrial survey, to complement the research work by investigating and validating the identified challenges and its solutions in Green ERTS.

To attain the mentioned goals, we have designed the below Research Questions (RQs):

RQ1: What are the challenges and its solutions/practices, as identified in the literature, in the design of Green/Energy-efficient embedded real-time systems?

RQ1.1. Do these challenges in the design of Green/Energy-efficient embedded real-time systems may vary with time?

RQ1.2. How are these challenges related to the study strategies used in the literature?

RQ1.3. Do these challenges in the design of Green/Energy-efficient embedded real-time systems may vary concerning publication venue?

RQ2. As identified in the literature, what are the practices/solutions for addressing the identified challenges in the design of Green/Energy-efficient embedded real-time systems?

Objectives of the Study:

The primary objective behind this research involves identifying and addressing fundamental challenges in developing ERTS. The research establishes a complete understanding of energy limitations together with time constraints along with maintainability requirements and performance trade-offs and transient fault challenges. The research presents 34 practical guides including scheduling algorithms and reliability methods that help researchers and practitioners improve green ERTS development.

The research examines temporal shifts and contextual variations of these challenges by analyzing how they have evolved while also comparing methods used across different publication venues. This research aims to develop next-generation frameworks that optimize both energy efficiency and embedded system performance. The research demonstrates how Artificial Intelligence (AI) and Machine Learning (ML) integration can optimize energy efficiency and deliver predictive maintenance while adapting system configurations to address present and future challenges in green ERTS development.

Significance of the Study:

This research presents a unique approach to addressing the challenges in the design of Green ERTSs by employing SLR methodology. Moreover, the study stands out by being the first of its kind to employ a systematic review in this domain. Previous efforts have typically explored green ERTS through case studies or experimental methods, often addressing narrow or specific aspects of the problem. In contrast, this research adopts a comprehensive approach, aiming to establish a foundational understanding of the challenges and solutions in green

ERTS design. The novelty of the work lies in its dual focus: not only identifying the issues but also documenting actionable practices to tackle them, thereby paving the way for future advancements in the field.

Background:

ERTSs are categorized into three major types. One of these is "Hard Real-Time Embedded Systems," where meeting deadlines is absolutely critical. In these systems, missing a deadline is intolerable under any circumstances. These are time-sensitive devices, and their outputs must be accurate and delivered on time. Failing to meet deadlines in such systems can result in complete system failure [39]. Second category "Firm real-rime systems" is like hard real-time systems. However, in these systems, slight delays beyond the time limit are somewhat tolerable, though they may compromise the quality of the system's services [39]. "Soft Real-Time Systems" differ from hard real-time systems in that while hard real-time systems guarantee meeting time limits at all costs, soft real-time systems aim to meet certain subsets of time limits in order to enhance the overall performance of the ERTS services [39].

The complexity of ERTS is increasing and to improve the advancement, sustainability, and maintainability of embedded systems, advanced development approaches are essential. NFRs in ERTS design must be managed properly prevent a disintegrated system mess and systemic failure [40, 41]. In the ERTS designing stage, the first stage is the requirements stage that has been improved using the FRIDA technique which is the refined aspect in the ERTS area. During the FRIDA stage, the FRIDA technique should be applied to capture both functional and non-functional requirements effectively [42]. When designing ERTSs, two primary challenges arise. The first is predictability: designers often struggle to create a system with predictable behavior while simultaneously meeting Non-Functional Requirements (NFRs), such as resource usage. The second challenge is developing a system capable of operating effectively in potentially hazardous environments [1]. Both the software and hardware efficiency aspects of the embedded system must be considered when designing the ERTS, which consists of two main components: software and hardware [43].

In ERTS development, two primary gaps are identified: the software design gap and the hardware design gap. According to [44], bridging these two gaps through synthesis results in the creation of a comprehensive system design. The primary issue in designing ERTSs lies in "Hardware-Dependent Software" (HDS), since it is dependent on the software section of the ERTS with a direct correlation to the fundamental hardware [45]. Basit et al., [45] argued that before the overall system is developed and manufactured, ERTS warrants appropriate support and authentication. Otherwise, the defects discovered at the last moment, not only raises the cost at most but also the probability of the system failure. Such products should be safe, sustainable and should not contain any deadlock. ERTSs are highly critical and complex systems. Moreover, addressing issues systematically during the early stages significantly reduces the error rate in later stages. In the context of the identified user requirements, they can be easily validated and verified before the detailed design and development. Energy saving management in ERTS is experiencing extra consciousness attention, mostly in mobile or hand computing devices which are using battery/cell (even the non-interchangeable battery) as their energy. With the constantly growing demand for mobile device applications, high-speed processors are expected to deliver optimal quality of service. However, due to energy limitations, minimizing power consumption in processors has become a critical priority. Currently, the central challenge for embedded real-time system is how to assure the requirements of ERTSs and minimize power utilization [46].

Methodology:

A Systematic Literature Review (SLR) was employed in this study to collect data from the existing body of literature. SLR offers highly stable results, unlike Ordinary Literature Reviews (OLRs). Moreover, SLR is a preplanned approach, which is based on certain protocols [47, 48]. It consists of three main stages: planning, implementation, and evaluation. These steps are intended to extract, organize, and distill the available papers based on the strength of their evidence. One of the advantages of SLR is that it is transparent when it comes to data collection and synthesis [49]. Step by step process of the research study conducted is presented in Figure 1.

Search Strategy:

A research was conducted using a set of keywords related to the research. These keywords were subsequently combined to create "search string 1," which was used to address the research questions. ScienceDirect and ACM digital libraries do not allow for long search strings; thus, "search string 1" was split by excluding some synonyms to make a "search string 2." The derived search string was then used in those specific libraries as explained below.

Search String:

We reviewed Kitchenham [47] guidelines and the improvements proposed by Mendes et al. [50] for designing the SLR protocols. Kitchenham's guidelines created a structured methodology for defining specific search strings through key concepts along with synonyms and Boolean operators. While Mendes et al.'s refined procedures enhanced the search process through their recommendation of multiple approaches which guarantee extensive literature coverage including alternative vocabulary and domain-specific terms as well as search string testing cycles. The combined principles of these methods delivered exhaustive search strings which maintained alignment with research objectives. PICO criteria was used to find keywords, where P stands for population, I – intervention, C – comparison, and O – outcome. We then compiled and combined the search strings based on the research questions as described above. This approach has also been adopted by other researchers [51, 52].

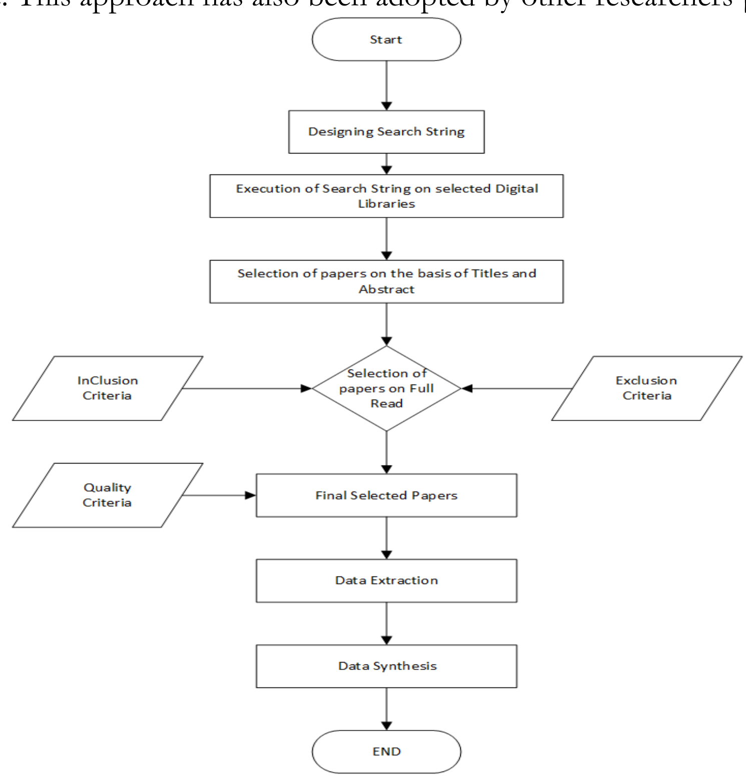

Figure 1. SLR-Studies Selection Flow Diagram

Population: Green/Energy-efficient Embedded Real-Time Systems
Intervention: challenges and practices
Comparison: No comparisons take place for the current study.
Outcomes of relevance: Energy-efficient design, Green relationship.
For constructing the search string, "Boolean connectors" and "OR" are used to join PICO.
Search string 1.
("Embedded real-time system" OR "Embedded real time system") AND ("Energy-efficient" OR Green OR "Green ERTS" OR "Energy-efficient ERTS") AND ("challenges" OR "barriers" OR "risks" OR "Design issues" OR "practices" OR "solutions")
Search string 2.
("Embedded real-time system" OR "Embedded real time system") AND ("Energy-efficient" OR Green OR "Green ERTS" OR "Energy aware") AND ("challenges" OR "practices" OR "solutions")

Literature Resources:

Applying our search strings, we used several digital libraries including IEEE Xplore, Science Direct, ACM, Springer Link, Google Scholar for searching the relevant papers. Table 1 summarizes the findings of the libraries that were used in our search strings.

Criteria for Study Selection:

In addition to the reviewers' (members of Software Engineering Research Group, University of Malakand, Pakistan SERG-UOM-PK) the researchers also stimulate the strings according to the guidelines [53]. First, we added the string to the library's metadata. The same level of caution was applied to ensure compliance with the restrictions on the title, abstract, and keywords. The first reviewer (member of SERG-UOM-PK) pays close attention to details of every paper, retrieved, and recorded it in a systematic way. According to the recommendations, the following inclusion and exclusion criteria have been developed [47, 50].

Inclusion Criteria and Exclusion Criteria:

The research examines papers which focus on the design challenges affecting green and energy-efficient embedded real-time systems. This review includes research articles that present solutions to design issues or recommend best practices to improve green embedded real-time systems. This research includes papers from peer-reviewed journals alongside repu Table conferences that are written in English and have accessible review versions. The research excludes studies about unrelated research questions or challenges in green embedded real-time systems together with non-refereed works such as books, tutorials, editorials and workshop papers and inaccessible full texts and non-peer-reviewed or duplicate sources. Table 1 illustrates the encapsulated outcomes for the search strings. Based on inclusion criteria, we selected 122 research papers from 1,129 total studies. Based on our exclusion criteria, we initially excluded 44 papers from a total of 122. Subsequently, by applying snowballing techniques, we added 9 more papers, bringing the final selection to a total of 65 papers. All steps of the data extraction procedure were accomplished by the primary reviewer (member of SERG-UOM-PK) and latter then it was consciously analyzed by secondary reviewer (member of SERG-UOM-PK).

Table 1. Search String Outcomes Per Database

Search Strings	**Digital-Libraries**	**Total Results**	**Initial Selection**	**Final Selection**
Search string 1 ("Embedded real-time system"	IEEE Xplore	332	43	19

OR "Embedded real time system") AND ("Energy-efficient" OR Green OR "Green ERTS" OR "Energy-efficient ERTS") AND ("challenges" OR "barriers" OR "risks" OR "Design issues" OR "practices" OR "solutions") Search string 2 ("Embedded real-time system" OR "Embedded real time system") AND ("Energy-efficient" OR Green OR "Green ERTS" OR "Energy aware") AND ("challenges" OR "practices" OR "solutions")	Science Direct	116	24	08
	ACM	5	0	0
	Springer Link	347	10	11
	Google Scholar	329	45	15
Total		1129	122	44
Snowballing				09
Grand total				65

Quality Assessment:

ID	Quality-Assessment-Criteria	Answer
1	Does the paper provide well-defined aims and objectives?	Yes/no/partially
2	Does the article have a clear context, e.g., industry or laboratory setting?	Yes/no/partially
3	Does the paper explicitly discuss the limitations?	Yes/no/partially
4	Does the paper add challenges for energy-efficient ERTS?	Yes/no/partially
5	Does the paper add practices for energy-efficient ERTS?	Yes/no/partially

On the basis of findings to our research question, we evaluated the primary studies. From a few selected findings we designed our quality assessment criteria. The queries of the quality criteria are extended in Table 2.

Table 2. Quality assessment criteria

We utilized a three-tier scale for the purpose of evaluating all questions of the findings for quality assessment criteria, i.e., Yes, Partially, or No. To yield quantifiable outcomes, we have specified the values for Yes → 2, partially → 1, and No → 0.

We included papers only if they met an average count of 0.5. Throughout this process, the primary reviewer (member of SERG-UOM-PK) was responsible for applying the quality assessment benchmark to the studies, while the additional reviewers (members of SERG-UOM-PK) verified and validated these assessments until a small set of randomly selected findings was extended. Based on our quality assessment benchmark, the secondary reviewer (member of SERG-UOM-PK) rejected some papers. Any distinctions were decided via contemplation. The central objective of the quality criteria was to keep out literature having low quality and assign the rectitude of the results.

Extraction and Synthesis of Data:

From the entire 65 selected studies, we extracted the following information from all:

In the data extraction process, data was collected systematically from the papers regarding the research questions formulated in the study while applying the inclusion/exclusion and quality assessment criteria described above. Finally, 65 papers were included in this review. The extracted data matched a set of pre-established criteria, namely Paper ID, the title of the study, year of publication, country of origin, the digital library source, type of publication (journal, conference, and grey literature), the methodology used in the paper, such as interviews, case studies, reports, surveys, meta-literature reviews, systematic literature reviews, company size, and a focus on design of ERTS design challenges, as described. After carrying out consultations with the members of the Software Engineering Research Group at the University of Malakand (SERG-UOM), we merged this into 8 aggregated groups. The study is distinctive as it systematically analyzes and synthesizes findings from 65 selected research papers to identify 8 critical challenges and 34 best practices for designing green ERTSs. Unlike traditional approaches that focus on either energy efficiency or quality attributes in isolation, this research highlights the interconnectedness between energy conservation and other critical aspects, such as system performance, security, and maintainability.

Result and discussion:

This research work had the main focus on finding challenges in the design of green/energy efficient ERTS and on imparting the practices/solutions to allay those challenges. Our research finding details are stated as follows:

Challenges in the design of green energy efficient ERTSs

The study has recorded eight critical challenges in the design of green/energy-efficient ERTSs.

CC #1 Security-Critical Real-Time Systems (SCRTS):

Energy efficiency is one of the most crucial characteristics of Security-Critical Real-Time Systems (SCRTS). The provision of security-guaranteed services in ERTSs will undoubtedly consume more power, and early battery discharge may result in system collapse. Nevertheless, usual security features in ERTSs are expected to demand an incredibly significant amount of energy. As mentioned before energy efficiency should therefore be properly addressed when designing secure ERTSs. Implementing security to ERTSs would increase the times taken to process them, and in the end, lead to reduced energy consumption. Thus, one of the major issues for the design of embedded SCRTSs is the energy usage limitation [54].

CC#2 Time Constraints:

ERTSs are not convenient to use when there is a need to meet the time-sensitive objectives since the applications of ERTSs are time-sensitive. In ERTSs there is no possibility to tolerate the violation of timing constraints. Although we may save a substantial amount of energy if we lower the processor speed and run ERTSs on low power, this will result in another major problem: failure or disaster. Also, the execution time will increase and it will take a long time before we produce results. Due to this ERTSs require a compromise between energy efficiency and speed of result generation. Therefore, the design of ERTSs under time and energy constraints is virtually impossible [55-56].

CC#3 Transient fault rate increases:

Earlier, ERTSs used methods named "dynamic voltage frequency scaling" (DVFS) for minimizing energy consumption. But these methods do not consider the potential of increasing the transient fault rate for a processor working with low power. As has been illustrated before, transient fault rates can be guaranteed services in ERTSs will, without doubt, dissipate more power; early battery discharge leads to system collapse. Nevertheless, usual security features in ERTSs are expected to demand an incredibly significant amount of energy.

As mentioned before energy efficiency should therefore be properly addressed when designing secure ERTSs. Implementing security to ERTSs would increase the times taken to process them, and in the end, lead to reduced energy consumption. Thus, one of the major issues for the design of embedded SCRTSs is the energy usage limitation [54].

CC#4 Cyber-physical systems (Adding cyber security increase energy):

As ERTSs become more integrated and have a tight coupling with the physical world, they are becoming cyber-physical systems. Communication devices, cloud computing, and sensors are requirements for the integration of the real world with the connected virtual world. These devices represent a large portion of the total energy consumption of ERTS and energy consumption cannot be ignored. Energy minimization across the system is therefore an impressive challenge [59].

CC#5 Increases transistors counts and system on a chip-design:

Conserving power is emerging as a key design issue for ERTSs because they are more creative and efficient complex systems. When ERTS power is low heat is generated in the system this can lead to more transistors and perhaps system on chip design. However, this has an adverse impact on the availability of the system. Consequently, an algorithm to minimize power consumption at both system and run-time operating system levels is required [60].

CC#6 Lack of Maintainability & Performance:

Almost all ERTSs consist of various hardware elements. I think it is not very hard to predict that reducing the power consumption of one device will lead to an increase in the power consumption of others; either the amount of power saved will decrease or the total amount of power consumed will rise. Despite the fact that performance of ERTS will be significantly influenced, the power consumption will be reduced by lowing the supply power of the processor. ERTSs' problem is that, to address energy concerns, one needs an algorithmic solution that can support ERTS performance [61–63].

CC#7 Memory Consumption:

Consequently, simple data structures are employed in ERTSs to provide small memory requirements and usage but power requirement increase. On the same note, the complex data structures in the evaluation yield low power utilization, but at the cost of high memory utilization and consumption. Thus, a technique for designing the data structure in ERTSs that can be potentially memory efficient and energy efficient is needed [64].

CC#8 Reliability:

Two primary methods to reduce the power consumption in ERTSs are dynamic power management (DPM) and dynamic voltage frequency scaling (DVFS, where in fact the frequency of the processor changes during the runtime. However, the occurrence of a system failure will be higher whenever the frequency is low, and this will reduce the dependability of the system. One of the main problem areas is how to achieve dependable ERTSs, while at the same time minimizing power consumption. [65-66].

Analysis of the identified challenges in green/energy-efficient ERTS based on the publication venue:

Out of the eight challenges, six were highlighted in studies published in journals, with the publication venues categorized as journals and conferences. The challenge "maintainability and performance-62%" is the most frequently mentioned challenge in >=50% of the studies. Among the eight challenges that were highlighted in a study presented in the conference, two of them, namely 'time constraints' noted 56% and 'maintainable & performance' noted 59%, have not been reported in less than 50% in the cross-sectional studies. For searching for the difficulties in the disparate study publications, the Linear-by-Linear Chi-Square test was used. The available information is as follows Table 3.

Table 3. Analysis of the challenges based on publication venue

Challenges	Occurrence in SLR (N=65)				Chi-square Test (Linear-by-Linear Association) a = .05	
	Journal (N=27)		Conference (N=35)		X2	P
	Freq	%	Freq	%		
Security-Critical Real-Time Systems (SCRTS)	2	10	9	28	2.619	.106
Time Constraints	3	14	18	56	.164	.686
Transient fault rate increases	4	19	5	16	.103	.748
Cyber-physical systems (Adding cyber security increase energy)	0	0	3	9	2.048	.152
Increased transistors counts and system on a chip-design	0	0	2	6	1.338	.247
Maintainability & Performance	13	62	19	59	5.847	.016
Memory consumption	1	5	4	13	.872	.350
Reliability	8	38	9	28	.568	.451

Comparison of the challenges based on the methodology:

The first search techniques were developed by the main reviewer (member of SERG-UOM-PK) at the time of data extraction, and the second reviewer (member of SERG-UOM-PK) confirmed these approaches. In the course of the assessment of the final selected papers, different difficulties were revealed in four types of study approaches. Based on our findings we were able to identify eight specific issues that were reported in the experiment, survey, OLR, and other studies. Out of these eight challenges, seven originated from experimental approaches, three from surveys, two from OLR, and two from other methods. To compare the differences found in the various challenges across these study methodologies, we utilized the Linear-by-Linear Chi-Square test. Additional information about this work is presented in Table 4.

Table 4. Analysis of the challenges based on publication venue

Challenges	Occurrence in SLR (N=65)								Chi-square Test (Linear-by-Linear Association) a = .05	
	Experiment (N=49)		Survey (N=4)		OLR (N=7)		Others (N=5)		X2	P
	Freq	%	Freq	%	Freq	%	Freq	%		
Security-Critical Real-Time Systems (SCRTS)	0	0	0	0	1	50	0	0	.041	.840
Time Constraints	8	2	1	100	2	100	0	0	.025	.875

Transient fault rate increases	9	18	0	0	0	0	0	0	.765	.382
Cyber-physical systems (Adding cyber security increase energy)	3	6	0	0	0	0	0	0	.224	.636
Increased transistors counts and system on a chip-design	2	4	0	0	0	0	0	0	.147	.702
Maintainability & Performance	20	41	1	100	0	0	1	100	.111	.739
Memory consumption	5	10	0	0	0	0	0	0	.390	.533
Reliability	15	31	1	100	0	0	1	100	.551	.458

Comparison of the challenges based on year wise:

In the SLR, there were no geographical restrictions; however, the selected studies were published between 2001 and 2024. Based on the literature from this period, our study identified six issues that were consistently highlighted in the reviewed papers. Among these challenges, two were reported in at least 50% of the studies: The highest importance is given to 'time constraints' at 55% and the second highest to 'Maintainable & Performance' at 59%. It is noteworthy that during the period 2001-2010, the factor 'Maintainability & Performance' was mentioned most often (59%), which suggested that it presented challenges for energy minimization in Energy-Related Technology Systems (ERTS). Although, this trend changes with time and in the second period, we observed that only 20% of the respondents have this challenge. In the earlier timeframe, seven challenges were observed across the studies, but none of the challenges it has achieved more than 50% prevalence level. For identifying significant differences of these challenges, Linear-by-Linear Chi-Square test was used. Further particulars of this research study are presented in Table 5.

Table 5. Analysis of the challenges based on year wise

Challenges	Occurrence in SLR(N=65)				Chi-square Test (Linear-by-Linear Association) a = .05	
	From 2001-2010 (N=20)		From 2011-2024 (N=45)		X2	P
	Freq	%	Freq	%		
Security-Critical Real-Time Systems (SCRTS)	2	9	12	27	2.116	.137
Time Constraints	12	55	26	58	.030	.863
Transient fault rate increases	3	14	8	18	.088	.767
Cyber-physical systems (Adding cyber security increase energy)	0	0	4	9	1.891	.169
Increased transistors counts and system on a chip-design	2	9	0	0	3.365	.067
Maintainability & Performance	13	59	12	27	7.163	.007
Memory consumption	0	0	5	11	3.283	.070

Reliability	6	27	11	24	.062	.803

Comparison of the challenges based on continents:

This study shows the number of studies done in the different continents in our analysis. We focused on comparing the challenges identified in three continents: Countries like Asia, Europe, and North America, and many others. The intended objective was to find out whether or not these challenges vary from one continent to the other or not. Our rationale for studying these challenges is that knowing how often they occur in various regions can help provide insights into energy-efficient Energy-Related Technology Systems (ERTSs). It consists of the same type of challenges as well as the different ones when the difficulties in the specified continents are compared. The identified challenges for these regions are 'time constraints' (65%, 54%, and 57%), 'maintainability and performance' (39%, 38%, and 57%), 'reliability' (32%, 38%, and 14%), and 'security critical real time systems (SCRTS)' (26%, 15%, and 14%).

Further, 'transient fault rate increases' and 'cyber-physical systems which enhance energy due to cybersecurity' are the problems clearly mentioned in North America and Asia. Table 6 also shows that 'memory consumption' is a common issue for organizations in Asia and Europe, and the 'increased transistor counts and system-on-chip design' is only mentioned as a problem in North America. In addition to the above mentioned three continents, the issues in 'transient fault rate increase,' 'maintainability & performance' and 'reliability' has also been reported in other regions of the world. The need for this study is to determine the various difficulties experienced in the different continents of the world.

Table 6. Analysis of the challenges based on continent

Challenges	Occurrence in SLR (N=65)								Chi-square Test (Linear-by-Linear Association) a = .05	
	Asia (N=31)		North America (N=13)		EUROPE (N=19)		Others (N=2)		X2	P
	Freq	%	Freq	%	Freq	%	Freq	%		
Security-Critical Real-Time Systems (SCRTS)	8	26	2	15	3	14	0	0	1.259	.262
Time Constraints	20	65	7	54	11	57	0	0	1.945	.163
Transient fault rate increases	7	23	1	8	0	0	1	50	.466	.495
Cyber-physical systems (Adding cyber security increase energy)	1	3	2	15	0	0	0	0	.008	.927
Increased transistors counts and system on a chip-design	0	0	2	15	0	0	0	0	.400	.527

Maintainability & Performance	12	39	5	38	11	57	1	50	.557	.456
Memory consumption	4	13	0	0	3	14	0	0	.370	.543
Reliability	10	32	5	38	3	14	1	50	.040	.841

Practices/Solutions to address Challenges in designing of Green/energy-efficient ERTS:

We have identified 34 practices in total to address the eight found challenges during this study. These practices are listed in the following Tables, started from Table 7 to 14.

Table 7. Practices for addressing security critical real-time system

CC #1 Security critical real-time system		
S.NO	**Practices to address Security critical real-time system**	**Frequency**
CCP#1.1	The Degree of Security Gap of the Application (DSGA) approach should be put into use to measure the security level of the system in question with regards to the compliance with the time and energy constraints.	2
CCP#1.2	Design an approximation algorithm using the dynamic programming approach, which meets some security requirements, consumes less energy and has low memory complexity.	2

Table 8. Practices for addressing time constraints

CC #2 Time Constraints		
S.NO	**Practices to address time constraints**	**Frequency**
CCP#2.1	Adopt the FITS technique to make a 16 bit Instruction set to minimize energy, size of code and time constraints with a limited system interference	4
CCP#2.2	Use DFR-EDF for energy-efficient scheduling and DPM and DVS for timing constraint.	6
CCP#2.3	Optimise the use of energy and complete tasks on time by using Adaptive Processor and Voltage Assignment with Probability APVAP.	5
CCP#2.4	Choose non-preemptive scheduling and Energy-efficient load scheduling algorithm" (EELSA) to decrease energy consumption and to achieve the intended time-frame of the tasks.	11
CCP#2.5	The quality of ERTS can be improved under energy-efficient conditions and within the time limit using the REW-Pack and REW-Unpack algorithms.	1
CCP#2.6	Use the non-preemptive scheduling for energy efficient outcome on variable speed processors.	1
CCP#2.7	The best tools for managing processor speeds with regard to energy consumption and time constraints are Modified energy aware Dynamic voltage and speed algorithm MEADVSA and energy aware Dynamic voltage and frequency algorithm EA_DVFA.	2
CCP#2.8	Use a Power-State Transition Graph to minimize energy consumption to achieve timing constraints in ERTSs.	1

CCP#2.9	Design a schedule and power control mechanism using a dynamic programming approach subject to the scheduling and security constraints.	2
CCP#2.10	To guarantee the strength of the system security together with energy and deadline requirements, one can use Degree of Security Gap of the Application DSGA metric	2
CCP#2.11	Integrated Frequency locking of Priority Ceiling Protocol FL-PCP and Frequency Assignment of an Independent Fixed-Priority Task Set ”FA-ITS algorithms to minimize task energy consumption and enhance timely task accomplishment.	1
CCP#1.12	Use POET, an open-source C library and runtime system because it is efficient in terms of time and energy for ERTSs.	1

Table 9. Practices for addressing Memory Consumption

CC #3 Memory Consumption		
S.NO	**Practices to address Memory Consumption**	**Frequency**
CCP#3.1	Use a Pareto distribution to determine a data structure that maximizes energy efficiency while using minimum memory.	1
CCP#3.2	Design an approximation algorithm using dynamic programming to solve this problem and propose a solution to reduce energy consumption with little memory overhead.	2

Table 10. Practices for addressing Maintainability and Performance

CC #4 Maintainability and performance		
S.NO	**Practices to address Maintainability and performance**	**Frequency**
CCP#4.1	Propose an “efficient mode assignment algorithm” to minimize power consumption in the wireless ERTSs while keeping the performance of the system intact besides the radio sleep schedule.	2
CCP#4.2	Improve the quality of services offered in ERTSs by minimizing energy consumption through the use of reliability maximization with energy constraint RMEC algorithm.	2
CCP#4.3	Employ the DVFS-energy-efficient scheduling algorithm and Non-DVFS energy-efficient scheduling algorithms (NDERG and DERG) for high power reduction while meeting the services goal in ERTSs.	2
CCP#4.4	Implement the Constrained Power Management (CPM) for an overall energy saving approach on a system wide basis while maintaining quality of service in ERTS.	4
CCP#4.5	Use the “Markov Reward Model” and number of visits, collected state, and impulse rewards to assess system power usage in terms of component idle states.	2
CCP#4.6	Employ the writer reader aware speed assignment (WRAS) heuristic algorithm to conserve energy in a processing element by partitioning tasks and utilizing EDF policy for the best performance.	4

CCP#4.7	Develop a "Dynamic adaption algorithm" that uses three approaches namely Migration, Degradation, and Removal, in the scenario of unpredicTable conditions in ERTSs.	3
CCP#4.8	Apply the Adaptive Processor and Voltage Assignment with Probability (APVAP) algorithm in ERTSs in order to minimize power consumption, time and probability.	5

Table 11. Practices for addressing Transient fault tolerance

CC #5 Transient fault tolerance		
S.NO	**Practices to address Transient fault tolerance**	**Frequency**
CCP#5.1	Adopt the "Criticality-Aware Standby-Sparing" (CASS) framework, integrating criticality theory with the standby-sparing (SS) technique to ensure fault tolerance and enhance power efficiency.	3
CCP#5.2	Apply the "Greedy Energy-Efficient" (GEE) algorithm to optimize energy efficiency in ERTSs by reducing execution frequency. It allows task re-execution at maximum frequency to maintain fault tolerance under transient faults.	3
CCP#5.3	Utilize the "frequency assignment method" with "dynamic priority assignment" to significantly reduce power consumption while handling transient faults in ERTSs.	1

Table 12. Practices for addressing Reliability

CC #6 Reliability		
S.NO	**Practices to address the Reliability**	**Frequency**
CCP#6.1	Extend the "Greedy Energy-Efficient" (GEE) algorithm to increase energy efficiency in Energy-Reliable Time-Sensitive Systems (ERTS) under reliability bounds.	2
CCP#6.2	An Efficient Resource Trading System (ERTS) should incorporate a Directed Acyclic Graph (DAG) based parallel application coupled with a DVFS-energy-efficient scheduling algorithm to implement power control and enhance the reliability of the system.	2
CCP#6.3	Use the DERG and NDERG algorithms that are Non-DVFS energy-efficient scheduling techniques to reduce power consumption necessary for the reliable operation of ERTSs.	2
CCP#6.4	During the architectural phase, fulfill both quality attributes – reliability and power consumption – by applying the Markov Reward Model and the Reliability Evaluation Model, which improve Kubat's model.	1
CCP#6.5	Introduce the "Dynamic Low Power Scheduling Algorithm" (DLPSR) for enhancing dynamic slack time and minimizing energy consumption in ERTSs with reliability constraint.	1

Table 13. Practices for addressing Increase transistor counts and system on a chip design

CC #7 Increase transistor counts and system on a chip design		
S.NO	**Practices to address the Increase transistor counts and system on a chip design**	**Frequency**

CCP#7.1	Schedule the "Low-Energy Earliest-Deadline-First" (LEDF) algorithm and the improved version, the "Extended-LEDF" (E-LEDF) that includes extra experimental scenarios. Also, use what is known as Mixed-Integer Linear Programming as a technique for solving moderate-sized problem well.	1

Table 14. Practices for addressing Cyber physical system

CC #8 Cyber physical system		
S.NO	**Practices to address the Cyber physical system**	**Frequency**
CCP#8.1	Adopt Genetic Algorithm Frequency Scaling GAFS and Differential Evolution Frequency Scaling DEFS in order to minimize the total energy usage.	1

Limitations and Threats to Validity:

Our research method was a systematic literature review (SLR); we aimed at being systematic in our research. However, one may argue that our approach can miss some high-quality data points. According to the recommendations of both the primary and secondary reviewers (members of SERG-UOM-PK) and after a thorough consideration the following keywords were used in this study, to make the study valid and include as many related studies as possible. In order to minimize risks associated with the conduct of searches in digital libraries, we only searched the best known and most relevant libraries in the computing disciplines, as discussed in Section 2 of this paper. However, this search was done with some constraints in the number of libraries that were to be searched, this makes it more likely that important data may be left out. Each phase of the SLR was checked and confirmed with the help of a reviewer (member of SERG-UOM-PK) in a strict procedure. Also, we excluded articles only published in English to minimize the risk of not including the studies available in other languages, which could have added to the sources used. One more limitation of the present study is that the search results included only articles up to the year 2019. We intend to build upon these results in subsequent sections.

Conclusion and Future Work:

In conclusion, the design of energy-efficient Green Embedded Real-Time Systems (ERTSs) is getting more important to handle the issues raised by system failures in the present world dominated by complex technologies. The present systematic literature review has found that there are several critical concerns that developers and users experience when developing these systems and best practices to address these concerns. But we also know that there is still much that needs to be done in this field. For the future works, our near-term strategy is to use a questionnaire survey to replicate the results of the SLR in the industry. This will be useful in helping us get other issues and practices that we may not have seen when we were reviewing the literature. Since the present work covers literature up to 2019, we plan to extend our research in future work towards constructing a framework for the design of Green ERTSs. Specifically, the integration of AI and ML will be crucial in our future research activities. Thus, the problem of using the potential of AI-based optimization in improving energy efficiency in real-time scheduling is relevant. Furthermore, the incorporation of machine learning methods contributes to the ability of preemptive maintenance, which enables the systems adaptively reconfigure its provided services according to the data and usage rate. With the help of these technologies, we are to develop new effective Green ERTSs with greater reliability and more impact on environmental issues.

References:

[1] T. A. Henzinger, “Two challenges in embedded systems design: predictability and robustness,” Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, vol. 366, no. 1881, pp. 3727-3736, 2008.

[2] T. Soyata, L. Copeland, and W. Heinzelman, “RF energy harvesting for embedded systems: A survey of tradeoffs and methodology,” IEEE Circuits and Systems Magazine, vol. 16, no. 1, pp. 22-57, 2016.

[3] H. Chai, G. Zhang, J. Zhou, J. Sun, L. Huang, and T. Wang, “A short review of security-aware techniques in real-time embedded systems,” Journal of Circuits, Systems and Computers, vol. 28, no. 02, pp. 1930002, 2019.

[4] Y. Jiang, H. Zhang, Z. Li, Y. Deng, X. Song, M. Gu, and J. Sun, “Design and optimization of multiclocked embedded systems using formal techniques,” IEEE transactions on industrial electronics, vol. 62, no. 2, pp. 1270-1278, 2014.

[5] R. M. Hierons, "Real-Time Systems and Software, Alan C. Shaw, John Wiley & Sons, 2001, ISBN: 0-471-35490-2, 215 pages, Soft cover," Wiley Online Library, 2001.

[6] P. Bertels, M. D'Haene, T. Degryse, and D. Stroobandt, “Teaching skills and concepts for embedded systems design,” ACM SIGBED Review, vol. 6, no. 1, pp. 4, 2009.

[7] I. Ibrahim, R. Ali, M. Zulkefli, and N. Elfadil, “Embedded systems pedagogical issue: Teaching approaches, students readiness, and design challenges,” American Journal of Embedded Systems and Applications, vol. 3, no. 1, pp. 1-10, 2015.

[8] S. Gürgens, C. Rudolph, A. Maña, and S. Nadjm-Tehrani, "Security engineering for embedded systems: the secfutur vision." p. 7, 2010.

[9] S. Ravi, S. Ravi, A. Raghunathan, P. Kocher, and S. Hattangady, “Security in embedded systems: Design challenges,” ACM Transactions on Embedded Computing Systems (TECS), vol. 3, no. 3, pp. 461-491, 2004.

[10] C.-Y. Chen, A. Ghassami, S. Mohan, N. Kiyavash, R. B. Bobba, R. Pellizzoni, and M.-K. Yoon, “A reconnaissance attack mechanism for fixed-priority real-time systems,” arXiv preprint arXiv:1705.02561, 2017.

[11] C. Bao, and A. Srivastava, "A secure algorithm for task scheduling against side-channel attacks." pp. 3-12, 2014.

[12] N. Druml, M. Menghin, D. Kroisleitner, C. Steger, R. Weiss, A. Krieg, H. Bock, and J. Haid, "Emulation-Based Fault Effect Analysis for Resource Constrained, Secure, and Dependable Systems." pp. 337-344., 2013.

[13] K. Patel, S. Parameswaran, and S. L. Shee, "Ensuring secure program execution in multiprocessor embedded systems: a case study." pp. 57-62, 2007.

[14] K. Patel, and S. Parameswaran, "SHIELD: a software hardware design methodology for security and reliability of MPSoCs." pp. 858-861, 2008.

[15] R. Pellizzoni, N. Paryab, M.-K. Yoon, S. Bak, S. Mohan, and R. B. Bobba, "A generalized model for preventing information leakage in hard real-time systems." pp. 271-282, 2015.

[16] B. Tan, M. Biglari-Abhari, and Z. Salcic, “Towards decentralized system-level security for MPSoC-based embedded applications,” Journal of Systems Architecture, vol. 80, pp. 41-55, 2017.

[17] M. Saadatmand, A. Cicchetti, M. Sjödin, and T. Leveque, “Managing timing implications of security aspects in model-driven development of real-time embedded systems,” Int. J. Adv. Secur, vol. 5, pp. 68-80, 2012.

[18] X. Zhang, J. Zhan, W. Jiang, and Y. Ma, "A vulnerability optimization method for security-critical real-time systems." pp. 215-221, 2013.

[19] S. Mohan, M.-K. Yoon, R. Pellizzoni, and R. B. Bobba, “Integrating security constraints into fixed priority real-time schedulers,” Real-Time Systems, vol. 52, no. 5, pp. 644-674, 2016.

[20] W. Jiang, X. Zhang, J. Zhan, Y. Ma, and K. Jiang, “Design optimization of secure message communication for energy-constrained distributed real-time systems,” Journal of Parallel and Distributed Computing, vol. 100, pp. 1-15, 2017.

[21] N. Botezatu, V. Manta, and A. Stan, "Self-adapTable Security Architecture for Power-aware Embedded Systems." pp. 98-103, 2010.

[22] F. A. T. Abad, R. Mancuso, S. Bak, O. Dantsker, and M. Caccamo, "Reset-based recovery for real-time cyber-physical systems with temporal safety constraints." pp. 1-8, 2016.

[23] A. Easwaran, A. Chattopadhyay, and S. Bhasin, "A systematic security analysis of real-time cyber-physical systems." pp. 206-213.

[24] C.-W. Lin, B. Zheng, Q. Zhu, and A. Sangiovanni-Vincentelli, “Security-aware design methodology and optimization for automotive systems,” ACM Transactions on Design Automation of Electronic Systems (TODAES), vol. 21, no. 1, pp. 18, 2015.

[25] L. Apvrille, and Y. Roudier, “SysML-Sec: A SysML environment for the design and development of secure embedded systems,” APCOSEC, Asia-Pacific Council on Systems Engineering, pp. 8-11, 2013.

[26] T. Xie, X. Qin, and A. Sung, "SAREC: A security-aware scheduling strategy for real-time applications on clusters." pp. 5-12, 2005.

[27] S. Tripathi, R. S. Yadav, and R. P. Ojha, "A utilization based approach for secured real time applications on clusters." pp. 433-438, 2009.

[28] T. Xie, and X. Qin, "Enhancing security of real-time applications on grids through dynamic scheduling." pp. 219-237, 2005.

[29] S. Singh, S. Tripathi, and S. Batabyal, "Utilization Based Secured Dynamic Scheduling Algorithm for Real-Time Applications on Grid (U-SDSA)." pp. 606-613, 2017.

[30] E. A. Lee, "Cyber physical systems: Design challenges." pp. 363-369, 2008.

[31] I. Lee, O. Sokolsky, S. Chen, J. Hatcliff, E. Jee, B. Kim, A. King, M. Mullen-Fortino, S. Park, and A. Roederer, “Challenges and research directions in medical cyber–physical systems,” Proceedings of the IEEE, vol. 100, no. 1, pp. 75-90, 2011.

[32] S. Liu, J. Lu, Q. Wu, and Q. Qiu, “Harvesting-aware power management for real-time systems with renewable energy,” IEEE transactions on very large scale integration (VLSI) systems, vol. 20, no. 8, pp. 1473-1486, 2011.

[33] P. Derler, E. A. Lee, and A. S. Vincentelli, “Modeling cyber–physical systems,” Proceedings of the IEEE, vol. 100, no. 1, pp. 13-28, 2011.

[34] M. Saadatmand, A. Cicchetti, and M. Sjödin, "A methodology for designing energy-aware secure embedded systems." pp. 87-90, 2011.

[35] P. Koopman, “Embedded system security,” Computer, vol. 37, no. 7, pp. 95-97, 2004.

[36] T. Lodderstedt, D. Basin, and J. Doser, "SecureUML: A UML-based modeling language for model-driven security." pp. 426-441, 2002.

[37] J. Jürjens, "UMLsec: Extending UML for secure systems development." pp. 412-425, 2002.

[38] H. Chai, G. Zhang, J. Zhou, J. Sun, L. Huang, and T. Wang, “A Short Review of Security-Aware Techniques in Real-Time Embedded Systems,” Journal of Circuits, Systems and Computers, vol. 28, no. 02, pp. 1930002, 2018.

[39] H. Alhussian, N. Zakaria, F. A. Hussin, and H. T. Bahbouh, "A review of the current status of the Java programming on embedded real-time systems." pp. 836-842, 2012.

[40] L. Chung, and B. A. Nixon, "Dealing with non-functional requirements: three experimental studies of a process-oriented approach." pp. 25-25, 1995.

[41] S. L. Tsang, S. Clarke, and E. Baniassad, "An evaluation of aspect-oriented programming for java-based real-time systems development." pp. 291-300, 2004.
[42] M. A. Wehrmeister, E. P. Freitas, C. E. Pereira, and F. R. Wagner, "An aspect-oriented approach for dealing with non-functional requirements in a model-driven development of distributed embedded real-time systems." pp. 428-432, 2007.
[43] F. Tian, S. Wang, and Y. Yu, "Design of Smart Home System based on ARM and ZigBee Technology," International Journal of Smart Home, vol. 10, no. 2, pp. 235-244, 2016.
[44] H. Ahlendorf, and L. Gopfert, "Hardware/software design challenges of low-power sensor nodes for condition monitoring." pp. 659-659, 2010.
[45] M. A. Basit-Ur-Rahim, F. Arif, and J. Ahmad, "Modeling of real-time embedded systems using SysML and its verification using UPPAAL and DiVinE." pp. 132-136, 2014.
[46] J. Niu, C. Liu, Y. Gao, and M. Qiu, "Energy efficient task assignment with guaranteed probability satisfying timing constraints for embedded systems," IEEE Transactions on Parallel and Distributed Systems, vol. 25, no. 8, pp. 2043-2052, 2013.
[47] B. Kitchenham, and S. Charters, "Guidelines for performing systematic literature reviews in software engineering," 2007.
[48] T. C. Lacerda, and C. G. von Wangenheim, "Systematic literature review of usability capability/maturity models," Computer Standards & Interfaces, vol. 55, pp. 95-105, 2018.
[49] N. Dissanayake, A. Jayatilaka, M. Zahedi, and M. A. Babar, "Software Security Patch Management--A Systematic Literature Review of Challenges, Approaches, Tools and Practices," arXiv preprint arXiv:2012.00544, 2020.
[50] E. Mendes, C. Wohlin, K. Felizardo, and M. Kalinowski, "When to update systematic literature reviews in software engineering," Journal of Systems and Software, vol. 167, pp. 110607, 2020.
[51] R. A. Khan, and S. U. Khan, "Empirical exploration of communication and coordination practices in offshore software development outsourcing," Proceedings of the Pakistan Academy of Sciences: A. Physical and Computational Sciences, vol. 54, no. 1, pp. 41, 2017.
[52] A. Ahmad, S. U. Khan, H. U. Khan, G. M. Khan, and M. Ilyas, "Challenges and Practices Identification via a Systematic Literature Review in the Adoption of Green Cloud Computing: Client's Side Approach," IEEE Access, 2021.
[53] R. A. Khan, S. U. Khan, H. U. Khan, and M. Ilyas, "Systematic Mapping Study on Security Approaches in Secure Software Engineering," IEEE Access, vol. 9, pp. 19139-19160, 2021.
[54] W. Jiang, K. Jiang, X. Zhang, and Y. Ma, "Energy aware real-time scheduling policy with guaranteed security protection." pp. 317-322, 2014.
[55] R. Nath, A. K. Shukla, P. K. Muhuri, and Q. D. Lohani, "NSGA-II based energy efficient scheduling in real-time embedded systems for tasks with deadlines and execution times as type-2 fuzzy numbers." pp. 1-8, 2013.
[56] W. Jiang, Z. Guo, Y. Ma, and N. Sang, "Measurement-based research on cryptographic algorithms for embedded real-time systems," Journal of Systems Architecture, vol. 59, no. 10, pp. 1394-1404, 2013.
[57] J. Zhou, and T. Wei, "Stochastic thermal-aware real-time task scheduling with considerations of soft errors," Journal of Systems and Software, vol. 102, pp. 123-133, 2015.
[58] A. Kumar, and B. Alam, "Real time scheduling algorithm for fault tolerant and energy minimization." pp. 356-360, 2014.

[59] A. Suyyagh, J. G. Tong, and Z. Zilic, "Analysis of meta-heuristics performance in energy aware scheduling of real-time embedded systems." pp. 1-6, 2015.
[60] V. Swaminathan, and K. Chakrabarty, "Real-time task scheduling for energy-aware embedded systems," Journal of the Franklin Institute, vol. 338, no. 6, pp. 729-750, 2001.
[61] O. Sokolsky, A. Philippou, I. Lee, and K. Christou, "Modeling and analysis of power-aware systems." pp. 409-424, 2003.
[62] B. Mochocki, D. Rajan, X. S. Hu, C. Poellabauer, K. Otten, and T. Chantem, "Network-aware dynamic voltage and frequency scaling." pp. 215-224, 2007.
[63] Z. Yuan, and G. Wang, "Power management for real-time tasks in wireless networked embedded systems." pp. 118-121, 2008.
[64] L. Jiang, C. Peng, H. Cai, Y. Chen, and X. Chen, "Low power consumption data structure design for embedded applications in real-time systems." pp. 13-18, 2012.
[65] H. Xu, R. Li, C. Pan, and K. Li, "Minimizing energy consumption with reliability goal on heterogeneous embedded systems," Journal of Parallel and Distributed Computing, vol. 127, pp. 44-57, 2019.
[66] J. Mei, K. Li, J. Hu, S. Yin, and E. H.-M. Sha, "Energy-aware preemptive scheduling algorithm for sporadic tasks on DVS platform," Microprocessors and Microsystems, vol. 37, no. 1, pp. 99-112, 2013.