OPEN ACCESS
International Journal of Innovations in Science & Technology
RESEARCH & INNOVATION DIVISION
IJIST

Building Robust Context Aware IoT Applications: Methods and Strategies for Detecting and Resolving Context Inconsistencies

Asia Khatoon Soomro[1], Yasir Arfat Malkani[1], Lachhman Das Dhomeja[2], Iqra Kanwal Lakho[3]
[1]Institute of Mathematics and Computer Science, University of Sindh, Jamshoro, Pakistan
[2] Department of IT, Faculty of Engineering and Technology, University of Sindh, Jamshoro, Pakistan
[3] Department of IT, Government College University, Hyderabad, Pakistan
*** Correspondence:** Asia Khatoon Soomro, asia.soomro@usindh.edu.pk

Citation | Soomro. A. K, Malkani. Y. A, Dhomeja. L. D, Lakho. I. K, "Building Robust Context Aware IoT Applications: Methods and Strategies for Detecting and Resolving Context Inconsistencies", IJIST, Vol. 07 Special Issue. pp 12-22, May 2025
Received | April 07, 2025 **Revised |** May 06, 2025 **Accepted |** May 07, 2025 **Published |** May 07, 2025.

The Internet of Things (IoT) has revolutionized connectivity, creating a vast network of interconnected devices that seamlessly exchange and analyze data. Within this dynamic IoT ecosystem, context-aware applications have emerged, enabling autonomous responses to events triggered by contextual information, thereby enhancing user experiences and facilitating intelligent decision-making. However, the utilization of context data in IoT applications has introduced a key challenge to context inconsistency. Context inconsistency is defined as the condition in which context data collected from multiple sources is inaccurate, incomplete, or conflicting, leading to incorrect processing may disrupt the behavior of context-aware applications. Context inconsistencies arise from various factors, including sensor noise, communication errors, and contradictory data sources (e.g.- two motion detection sensors located in the same area may report different readings, where one sensor detects one person, and another sensor detects three people). These inconsistencies can significantly impact the reliability and precision of IoT applications, potentially resulting in erroneous decisions and degraded user experiences. To address this critical concern, this research paper undertakes a comprehensive review of contemporary methodologies developed for detecting and resolving context inconsistencies in IoT environments. This study explores various strategies, discusses their features in detail and contributes by classifying them into different categories for better understanding. Through a detailed examination of the effectiveness, strengths, and limitations of each classified method, the paper aims to offer valuable insights into managing context inconsistencies in IoT applications. More precisely, this paper serves as a valuable resource for researchers, practitioners, and industry professionals in the IoT domain, providing them with a comprehensive understanding of context inconsistency detection and resolution methods.

Keywords: Context Awareness, Context Inconsistency Detection, Context Inconsistency Resolution, Internet of Things.

Introduction:

In 1999, Kevin Ashton coined the term Internet of Things (IoT), which refers to a network of physical devices embedded with sensors, software, and other technologies to connect and exchange data with other devices and systems over the internet [1]. The main concept is that IoT devices are designed to automatically collect, process, and share data without the need for human intervention. Nowadays, IoT technology is used in a variety of applications, including smart homes, smart cities, industrial automation, healthcare, transportation, etc. IoT has rapidly grown in the number of context-aware applications that provide personalized services to users. In the dynamic IoT environment, context-aware applications allow systems to adapt themselves based on the context [2]. This enhances user experience, provides real-time insights, and supports intelligent decision-making. However, the increasing use of contextual information in IoT-based context-aware applications has led to a growing need for effective solutions to address the challenges posed by context inconsistencies. Context inconsistencies arise from various factors, including sensor noise, communication errors, or conflicting data sources. It is a situation where the collected context may be inaccurate, incomplete or inconsistent [3], [4]. For example, two motion detection sensors located in the same area may provide completely different readings, where one sensor detects one person, and the other sensor detects three people. An inaccurate, incomplete, or inconsistent context may cause context-aware applications to operate improperly and deviate from their original functionality. For instance, a security management system may fail to correctly identify the number of people present due to inconsistent values reported by different motion detection sensors monitoring the area. Consequently, detecting and resolving context inconsistencies is critical to ensure that context-aware applications operate accurately and reliably, and provide a seamless user experience.

Aim and Objectives of the Study:

The main aim of this research is to provide valuable insights for researchers, practitioners, and industry professionals in the IoT domain, as it provides them with a comprehensive understanding of the methods used for detecting and resolving context inconsistency in context-aware IoT applications. To achieve the aim the following objectives are set:

- With thorough and comprehensive literature review on the contemporary methodologies of detecting and resolving context inconsistencies in IoT environments, we have proposed the classification of these methods and strategies for better understanding of this domain for the research community.
- To compare the strengths and limitations of context inconsistency detection and resolution methods as per our proposed classification.
- To emphasize key aspects such as scalability, accuracy, efficiency etc. that are crucial for evaluating these methods.
- To provide a comparison based on key aspects, emphasizing that selecting an approach for context inconsistency detection and resolution requires careful consideration in terms of scalability, accuracy, efficiency etc. to effectively address the needs of IoT applications.

Material and Methods:

The research methodology for this study involves a comprehensive review of existing literature on the detection and resolution of context inconsistencies. After a thorough and comprehensive literature review on context inconsistency detection and resolution, we have proposed a classification of these methods and strategies for a better understanding of this domain. This study also compares the characteristics and weaknesses of each method. It also examines key aspects such as scalability, accuracy, efficiency etc. that are crucial for evaluating these methods.

As per our proposed classification, context inconsistency detection and resolution approaches can be categorized as shown in Figure 1. In subsequent sub-sections each of these categories is discussed in detail. For the sake of understanding and clarity, some prevalent context inconsistency detection and resolution approaches placed in each of these categories are also discussed.

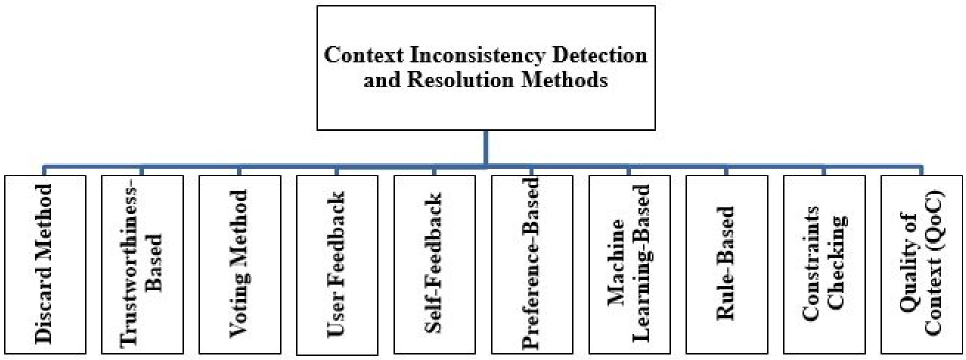

Figure 1. Proposed Classification of Context Inconsistency Detection and Resolution Approaches

Discard Method:

The discard method [4], [5], [6] for the detection of context inconsistency involves discarding the context based on predefined rules. This method includes discarding all contexts, discarding the latest context, and discarding bad context, which is perceived as incorrect. Dropping all refers to rejecting all inconsistent contexts, dropping the latest focuses on the most current context, and dropping bad removes context that has been identified as inconsistent. Although this method is used to maintain context consistency and system accuracy by removing information that deviates from expected norms, one major drawback of this method is the loss of significant amounts of information. Moreover, it might not be suitable for dynamic environments.

Xu C. et al. [4] proposed a drop-bad context inconsistency resolution method for pervasive computing systems. The study emphasizes the need to enhance heuristics-based inconsistency resolution by estimating the impact on applications and adjusting resolution actions accordingly. Bu et al. [5] suggested a drop-all method for context inconsistency. In this method all contexts leading to an inconsistency except the latest one are simply discarded. Chomicki et al. [6] proposed an approach based on dropping the latest context assuming that the most recent context is more likely to be inconsistent and thus should be removed to resolve the inconsistency.

Trustworthiness-based:

The trustworthiness-based method [7]. On the evaluation of quality of context. In European conference on smart sensing and context detects context inconsistency in IoT by analyzing the accuracy of data from different IoT devices. The accuracy of data is assessed using criteria such as correctness, credibility, and consistency. This approach enhances overall system reliability and appears suitable for accurate decision-making in dynamic IoT environments. However, due to limited trustworthiness information it is challenging to obtain reliable assessments, especially in rapidly changing IoT scenarios. In [7], the authors present a trustworthiness-based algorithm that selects context information with the highest sensor perception precision.

Voting Method:

In this method [8], multiple devices work together by voting to determine a situation to detect context inconsistency. Each device shares its view, and the opinion that is most frequently accepted is considered the accurate one. Device collaboration can boost enhance accuracy in the dynamic nature of the IoT environment. However, this method can lead to inaccurate context assessments if a device generates inaccurate incorrect data due to damage.

In [8], the voting algorithm is applied to select context information by plurality. This research contributes to the understanding of reliability in fault-tolerant software systems by proposing new voting strategies and analyzing their effectiveness in scenarios with limited output options.

User Feedback:

This method [9], [10], [11] relies on user feedback for context inconsistency detection in context-aware IoT applications. Users provide feedback about their current context, which is then used to evaluate the perception precision of each sensor, helping to identify inconsistencies and enhance accuracy. However, excessive user feedback may disturb users and violate the principle of context awareness in IoT which emphasizes minimizing user involvement. Handling a large volume of user input can also be challenging. Moreover, this method becomes less applicable when users do not provide feedback or have limited understanding of the IoT systems.

Work in [9], [10], [11] is based on user feedback. Lee and Kim [9] utilized user feedback data to obtain context information that can be used to select the adaptive context awareness method. Xu H. et al. [10] addressed the challenge of managing context inconsistency in dynamic environments and presented a framework for parallel processing of context information based on user feedback and adjusted basic reliability distribution. The work in [11] proposed a user feedback approach to address inconsistencies in context information within context aware systems. Through user feedback, the precision of each sensor's perception is assessed while employing modified evidence theory to adjust the impact of context on decision-making.

Self-Feedback:

In the Self-feedback method [12], devices use their own internal input as feedback to evaluate correctness and adapt performance. By using this method, the system becomes adaptable to the dynamic environment of IoT. However, due to the lack of external validation the system is more vulnerable to attacks. Furthermore, creating robust self-feedback in resource-constrained IoT systems is challenging and may lead to delays or inaccuracies. Ji M et al. [12] proposed an approach based on limited self-feedback for measuring the Probability of Correctness (PoC) of context in context-aware systems. The system uses its own reliable output to evaluate the correctness of input context information and calculate the PoC for each context provider. The PoC of each source plays a vital role in the decision-making process.

Preference-based:

In this method [13], context inconsistencies are detected by analyzing user preferences. The system may discover inconsistencies in information gathered from different IoT devices by understanding user preferences, thus improving the reliability of IoT applications. However, this method is challenging due to the dependability of preferences, dynamic nature of user behavior, and potential conflicts among users. Zhang D. [13] proposed a preference-based approach for Decentralized Checking of Context Inconsistency (DCCI) in resource-constrained and decentralized pervasive computing environments.

Machine Learning-based:

This method [14] is based on training models or algorithms. In this approach, models are trained using historical data, allowing them to detect inconsistencies in the data gathered from various IoT devices. The performance of these models can be affected by insufficient training data. Moreover, to adapt to the dynamic environments of IoT, these models must be continuously updated with new data, which requires significant resources and increases operational costs.

The work done in [14] is based on machine learning methods for detecting and resolving context inconsistencies. The authors proposed a model based on the random forest algorithm to predict the most effective detection method using prior knowledge, thereby increasing resolution accuracy.

Rule-based:

This method [15] in IoT context-aware systems utilizes heuristics or predefined rules to identify inconsistencies in context data gathered from IoT devices. It is based on thresholds for evaluating inconsistencies. Moreover, this method adapts to the changing contexts of the environment. To enhance its effectiveness and applicability across various IoT applications, it can be integrated with other methods such as machine-based approaches.

The work done in [15] presented the rule-based context elimination scheme for context inconsistency management. Context elimination rules specify the conditions for contexts, allowing the removal of inconsistent contexts from the repository.

Consistency Constraints:

The constraint checking method [16], [17] for detecting context inconsistency in IoT context-aware systems operates on the assumption that well-defined constraints govern system behavior, enabling the identification of inconsistencies. Its characteristics lie in the explicit validation of system states against predefined constraints, ensuring coherence, and the ability to handle interdependent constraints. These sets of constraints or logical rules that context data must satisfy can be temporal (time-related), spatial (location-related), or value constraints (acceptable range of sensor readings). However, some of its limitations include the difficulty of accurately defining constraints for complex IoT systems, scalability challenges as systems expand, and a reliance on the quality of context data for precise inconsistency detection, which impacts its reliability in dynamic and evolving IoT ecosystems.

Wang H et al [16] proposed an approach named Generic Adaptive Scheduling (GEAS), which detects context inconsistencies through consistency constraints to prevent abnormal behavior or failure in applications. GEAS is a scheduling strategy that groups multiple consecutive context changes into one batch and checks them together to reduce the number of scheduled constraint check. Research in [17] is based on consistency constraints for detecting context inconsistency. The proposed approach called Partial Constraint Checking (PCC), identifies reusable parts of previous checking results to expedite the detection of context inconsistencies.

Quality of Context (QoC):

Quality of Context (QoC) is defined as "any information that describes the quality of information used as context information" [18]. QoC plays a crucial role in ensuring the quality of context information in context-aware systems. By evaluating various QoC parameters or attributes (such as precision, trustworthiness, up-to-dateness etc.), this method effectively detects and resolves context inconsistencies. Researchers have developed numerous QoC parameters to address context inconsistency problems, enabling systems to enhance the reliability and usefulness of context information.

Manzoor et al. [19] proposed conflict resolving policies based on QoC parameters. These parameters, including up-to-dateness, trustworthiness, completeness, and significance, are used to evaluate the quality of context information. Fan SD et al. [20] developed an approach in which they introduced a new (QoC) parameter relevance and used it alongside other multiple QoC parameters to address the inconsistency between sensed and non-sensed contexts. Chen M et al. [21] presented a context inconsistency elimination algorithm and introduced a new overall quality indicator (OQoC) parameter for improving context information quality. OQoC effectively combines the parameters of reliability, up-to-dateness, and modified correctness.

The following theories are also integrated with the aforementioned methods to improve the detection and resolution of context inconsistencies.

Dempster-Shafer Theory (DST):

Dempster-Shafer Theory (DST) also known as Evidence Theory or Belief Function Theory, was developed by Arthur P. Dempster in the 1960s and later expanded by Glenn Shafer in the 1970s [22]. It is a mathematical framework designed for reasoning under uncertainty, particularly when dealing with incomplete contexts or uncertain information. DST is used to detect inconsistencies by applying belief functions to assign degrees of belief to different hypotheses or propositions based on the available evidence. The core concept of DST is belief function, which reflects the degree of belief in a particular proposition. This theory has been widely applied in fields, such as artificial intelligence, decision support systems, and information fusion. In the context of context-aware systems and IoT environments, DST is used to build mass belief functions and resolve context inconsistencies, enhancing the accuracy of context inference. It provides a logical approach for handling uncertainty and conflicting evidence in decision-making processes. However, DST can be complex and face challenges in detecting inconsistencies, particularly in large-scale scenarios. Additionally, the theory's reliance on evidence source independence and difficulties in handling continuous data make it less adaptable and less effective in dynamic IoT environments. Research in [9], [11], [23] has incorporated Dempster-Shafer theory and presented combined approaches to address context inconsistencies.

Bayesian Networks:

Bayesian networks, also known as Bayes nets, belief networks, or probabilistic graphical models [24], are a method of managing uncertainty by identifying system variables, establishing probabilistic connections based on data (or domain expertise), and updating probabilistic estimates through Bayesian inference. This allows the assessment of context consistency or inconsistency. Bayesian networks utilize decision thresholds to identify inconsistencies in IoT environments, providing a systematic and probabilistic approach to uncertainty management. Their adaptability makes them applicable across various fields. However, when ~~is~~ used for context fusion, Bayesian networks require mutual exclusivity for computing hypotheses and are unable to fully account for general uncertainty. In [25], the authors applied Bayesian networks to model uncertain contexts.

Fuzzy Logic:

Fuzzy logic, introduced by Lotfi A. Zadeh in 1965, is a mathematical framework that initiated by allowing for intermediate values between true/false, yes/no, and high/low evaluations [26], [27]. It handles uncertainty by permitting degrees of truth between completely true and false values, making it an effective tool for managing context inconsistency. Fuzzy logic operates by defining certain variables, creating membership functions to quantify the degrees of membership, and establishing fuzzy rules to describe relationships between variables. It then converts observed data into fuzzy sets. and uses fuzzy inference to produce fuzzy output sets that indicate the degree of inconsistency. A threshold is set to determine when the degree of inconsistency requires action. The flexibility of fuzzy logic allows for continuous improvement by adjusting membership functions and rules in response to emerging evidence or system changes. This approach is particularly useful in situations where precise data is hard to obtain, making fuzzy logic highly applicable in fields such as control systems, decision-making, and context inconsistency detection. In [28], the authors developed an algorithm for a security system that uses fuzzy logic as its core framework to detect inconsistencies.

Results and Discussion:

This research provided comprehensive review of the current methodologies for detecting and resolving context inconsistencies in IoT environments. It explored various strategies, thoroughly discussed their features classifying them into distinct categories for better understanding. By examining the effectiveness, strengths, and limitations of each method, the paper aimed to offer valuable insights into managing context inconsistencies in IoT applications. More specifically, this work served as a valuable resource for researchers, practitioners, and industry professionals in the IoT domain, equipping them with a thorough understanding of context inconsistency detection and resolution techniques.

Features Analysis of Prevalent Methods:

As discussed in the previous section, various approaches exist for detecting and resolving context inconsistencies in context-aware systems within the Internet of Things (IoT) domain. However, there remains a need to improve these methods without sacrificing accuracy or reliability. In Table 1, we present the key characteristics and limitations of the context inconsistency detection and resolution methods based on the proposed classification. While some techniques are user-friendly, they may not scale effectively, whereas others excel at handling large amounts of data but require significant resources. Methods that prioritize accuracy tend to be resource-intensive and challenging to maintain, while some methods may offer lower accuracy. We suggest that selecting the appropriate approach involves finding a balance between accuracy, scalability, and complexity to effectively address the needs of IoT applications.

Table 1. Classification of Context Inconsistency and Detection Methods

Method	Characteristics	Limitation
Discard Method – Drop-Bad [4] – Drop-All [5] – Drop-Latest [6]	– Removes inconsistent context (latest, all, or erroneous). – Quick responses. – Easy to implement. – Efficient in handling simple inconsistencies.	– Significant loss of information. – Not suitable for dynamic environment.
Trustworthiness-Based Method [7]	– Focuses on the reliability of data sources. – Trust values determine data acceptance or rejection.	– Requires accurate trust measurement. – Possibility of eliminating valid data.
Voting Method [8]	– Aggregates data from multiple sources. – Majority decision determines context validity.	– Requires numerous data sources. – Delays caused by unreliable sources.
User feedback [9-11]	– Involves user input to confirm context accuracy.	– Excessive user feedback hinders the goal of context-awareness. – Relies on user availability. – User bias may affect the results.
Self- feedback [12]	– System monitors and refines by own context accuracy. – Reduces inconsistency autonomously.	– Difficult to implement in complex systems. – Limited by the system's self-awareness.
Preference-based [13]	– Eliminates inconsistencies based on user-defined preferences.	– Depends on personal user preferences. – Difficult to generalize.
Machine Learning [14]	– Uses algorithms to learn and adapt to inconsistencies	– Requires a large dataset for training. – Complex to understand and maintain.
Rule-based [15]	– Removes inconsistency based on predefined rules.	– Requires frequent rule updates. – Inflexible to new scenarios.
Constraint Checking [16-17]	– Ensures data adheres to predefined constraints.	– Computationally expensive.
Quality of Context (QoC) [19-21]	– Eliminates low-quality data based on QoC parameters.	– QoC parameters are Complex to define and measure.
Dempster-Shafer Theory [11,19,21,23]	– Combines evidence from multiple sources.	– Computationally complex. – Difficult to interpret.
Bayesian Network [25]	– Uses probabilistic models to reason under uncertainty.	– Requires accurate probabilistic data. – Complex and dependent on assumptions.
Fuzzy Set Theory [28]	– Manages imprecise and uncertain data effectively.	– Difficult to interpret and implement.

Comparative Analysis and Discussion:

In this section, we discuss key aspects that are essential for evaluating context inconsistency detection and resolution approaches for IoT.

Scalability:

Scalability refers to the ability of a system to expand and manage an increasing volume of data from IoT devices over time. Detection and resolution methods must be scalable to handle the growing demands of IoT systems without sacrificing performance as data volume increases.

Flexibility:

Flexibility is crucial for identifying and addressing context inconsistencies in dynamic IoT environments. It allows the system to adapt seamlessly to changing scenarios, incorporate new devices and meet evolving needs without extensive reconfiguration, ensuring the system's continued functionality.

Generalizability:

Generalizability ensures that a context inconsistency detection and resolution approach can be applied across various application scenarios. This reduces the need for creating distinct solutions for each specific situation, making the system adaptable and more efficient in diverse use cases.

Efficiency:

Efficiency ensures that context inconsistency detection and resolution methods achieve their goals quickly and with minimal resource consumption. Given that IoT systems often operate in environments with limited processing power and energy resources efficiency is crucial for real-time applications and extending the lifespan of battery-powered devices.

Accuracy:

Accuracy is essential for reliably detecting and resolving inconsistencies. High accuracy in context consistency detection is vital for ensuring the system's reliability particularly in safety-sensitive IoT applications, where errors can have significant consequences.

Dependency:

Dependency refers to the system's reliance on external data sources or frequent human intervention. To enhance the reliability and robustness of IoT applications, context inconsistency detection and resolution methods should operate autonomously, minimizing external dependencies and reducing the need for manual intervention.

Complexity:

The complexity of context inconsistency detection and resolution methods must be minimized to simplify implementation, maintenance, and deployment of IoT systems. A simplified approach ensures that the system remains responsive and adaptable real-time monitoring and control scenarios.

Heterogeneity:

Managing heterogeneity is critical for seamless integration and interoperability of various system components. By handling multiple devices and data sources efficiently, it enhances context awareness and promotes consistent system behavior, contributing to more reliable and coherent IoT applications.

Table 2. Comparative Analysis

Method	Scalability	Flexibility	Generalizability	Efficiency	Accuracy	Dependability	Complexity	Heterogeneity
Discard Methods	High	Low	Low	High	Low	Low	Low	Low
Trustworthy-Based	Medium	Medium	Medium	Medium	High	High	Medium	Medium
Voting Method	Medium	Medium	Medium	Medium	Medium	Medium	Medium	Medium
User feedback	Low	High	Medium	Low	High	High	Medium	High
Self -Feedback	Medium	Medium	Medium	Medium	Medium	Low	High	Medium
Preference based	Medium	High	High	Medium	High	Medium	Medium	High
Machine Learning	High	High	High	High	High	High	High	High
Rule Based	Medium	Low	Low	High	High	Low	Medium	Low
Consistency Constraints	Medium	Medium	Medium	High	High	Medium	Medium	Medium
Quality of Context	Medium	High	High	Medium	High	High	High	High
DST	Low	Medium	Medium	Low	Medium	High	High	Medium
Bayesian	Medium	High	High	Medium	High	High	High	High
Fuzzy Logic	Medium	High	High	Medium	High	Medium	High	High

Table 2. provides a comparative analysis of the different methods and strategies for context inconsistency detection and resolution. Some methods are scalable, resource-efficient, adaptable, and. accurate across a wide range of scenarios. Approaches that rely on user input tend to be more accurate and flexible but may vary in terms of efficiency. While some techniques excel in efficiency and scalability, they may sacrifice adaptability, and others are more accurate and adaptable, but require greater computational resources. Therefore, selecting an approach for context inconsistency detection and resolution requires careful consideration of the parameters outlined in Table 2 to effectively meet the needs of robust, reliable, and efficient IoT applications.

Conclusion:

The Internet of Things (IoT) enables context-aware applications that enhance user experiences and facilitate intelligent decision-making, However, the dynamic nature of IoT environments leads to the challenge of context inconsistencies, which can undermine the reliability and accuracy of these systems. Context inconsistency detection and resolution are essential for ensuring that IoT applications operate effectively and provide a seamless user experience. This paper categorizes various methods, including discard-based, trustworthiness-based, voting, user feedback, and machine learning-based approaches, and provides a comprehensive review of current techniques for detecting and resolving context inconsistencies in IoT environments. Each method presents a different set of trade-offs concerning accuracy, efficiency, scalability, and complexity. There is no one-size-fits-all solution; the selection of an appropriate method depends on the specific requirements of the IoT application.

Future research should focus on developing adaptable approaches that can function effectively across the diverse and evolving scenarios inherent in IoT environments. This will ensure that context-aware IoT applications remain resilient, scalable, and capable of delivering consistent performance despite the challenges posed by dynamic and heterogeneous contexts.

Acknowledgement: I am grateful to my supervisor, Prof. Dr. Yasir Arfat Malkani, for his kind, generous help and guidance in the writing of this research paper.

Author's Contribution: This research is a result of teamwork in which all the authors have equally contributed.

Conflict of interest: There is no conflict of interest of any type between authors about publishing this manuscript.

Project details. N/A

References:

[1] Y. A. Malkani, P. Bai, L. Das Dhomeja, and M. K. Abbasi, "Policy-Based Contextual Access Control Mechanism for Smart IoT Devices," *Asian Bull. Big Data Manag.* , vol. 4, no. 02, p. Science 4(2)-334, Jun. 2024, doi: 10.62019/ABBDM.V4I02.178.

[2] M. Aljawarneh, L. Das Dhomeja, and Y. A. Malkani, "Context-Aware service composition of heterogeneous services in pervasive computing environments: A review," *Proc. 2016 19th Int. Multi-Topic Conf. INMIC 2016*, Feb. 2017, doi: 10.1109/INMIC.2016.7840119.

[3] C. Chen, C. Ye, and H. A. Jacobsen, "Hybrid context inconsistency resolution for context-aware services," *2011 IEEE Int. Conf. Pervasive Comput. Commun. PerCom 2011*, pp. 10–19, 2011, doi: 10.1109/PERCOM.2011.5767574.

[4] C. Xu, S. C. Cheung, W. K. Chan, and C. Ye, "Heuristics-based strategies for resolving context inconsistencies in pervasive computing applications," *Proc. - 28th Int. Conf. Distrib. Comput. Syst. ICDCS 2008*, pp. 713–721, 2008, doi: 10.1109/ICDCS.2008.46.

[5] Y. Bu, T. Gu, X. Tao, J. Li, S. Chen, and J. Lu, "Managing quality of context in pervasive computing," *Proc. - Int. Conf. Qual. Softw.*, pp. 193–200, 2006, doi: 10.1109/QSIC.2006.38.

[6] J. Chomicki, J. Lobo, and S. Naqvi, "Conflict resolution using logic programming," *IEEE Trans. Knowl. Data Eng.*, vol. 15, no. 1, pp. 244–249, 2003, doi: 10.1109/TKDE.2003.1161596.
[7] A. Manzoor, H. L. Truong, and S. Dustdar, "On the Evaluation of Quality of Context," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 5279 LNCS, pp. 140–153, 2008, doi: 10.1007/978-3-540-88793-5_11.
[8] D. F. McAllister, C. E. Sun, and M. A. Vouk, "Reliability of Voting in Fault-Tolerant Software Systems for Small Output-Spaces," *IEEE Trans. Reliab.*, vol. 39, no. 5, pp. 524–534, 1990, doi: 10.1109/24.61308.
[9] B. H. Lee and D. H. Kim, "Efficient context-aware selection based on user feedback," *IEEE Trans. Consum. Electron.*, vol. 58, no. 3, pp. 978–984, 2012, doi: 10.1109/TCE.2012.6311345.
[10] H. Xu, L. Wang, H. Xiong, Z. Du, and Z. Xie, "Effective context inconsistency elimination algorithm based on feedback and reliability distribution for IoV," *China Commun.*, vol. 11, no. 10, pp. 16–28, Oct. 2014, doi: 10.1109/CC.2014.6969790.
[11] L. Wang, H. Xu, G. Sun, Z. Du, Z. Xie, and L. Zheng, "Context Inconsistency Elimination Based on User Feedback and Modified Evidence Theory," *Int. J. Mach. Learn. Comput.*, vol. 3, no. 6, pp. 473–478, Dec. 2013, doi: 10.7763/IJMLC.2013.V3.363.
[12] M. Ji, H. Xu, L. Wang, J. Dang, Z. Xu, and H. Fang, "Approach of measuring PoC of context using limited self-feedback in context-aware systems," *IET Wirel. Sens. Syst.*, vol. 6, no. 5, pp. 158–165, Oct. 2016, doi: 10.1049/IET-WSS.2015.0132.
[13] D. Zhang, M. Chen, H. Huang, and M. Guo, "Decentralized checking of context inconsistency in pervasive computing environments," *J. Supercomput.*, vol. 64, no. 2, pp. 256–273, May 2013, doi: 10.1007/S11227-011-0661-X/METRICS.
[14] M. Hamed, H. Abdelkader, and A. Abdelatey, "Resolving Context Inconsistency Approach Based on Random Forest Tree," *Lect. Notes Data Eng. Commun. Technol.*, vol. 113, pp. 192–199, 2022, doi: 10.1007/978-3-031-03918-8_18.
[15] Y. J. Lee, J. Lim, S. J. Hyun, and D. Lee, "Rule-based approach for context inconsistency management scheme in ubiquitous computing," *Proc. - IEEE/IFIP Int. Conf. Embed. Ubiquitous Comput. EUC 2010*, pp. 120–126, 2010, doi: 10.1109/EUC.2010.27.
[16] H. Wang, C. Xu, B. Guo, X. Ma, and J. Lu, "Generic Adaptive Scheduling for Efficient Context Inconsistency Detection," *IEEE Trans. Softw. Eng.*, vol. 47, no. 3, pp. 464–497, Mar. 2021, doi: 10.1109/TSE.2019.2898976.
[17] C. Xu, S. C. Cheung, W. K. Chan, and C. Ye, "Partial constraint checking for context consistency in pervasive computing," *ACM Trans. Softw. Eng. Methodol.*, vol. 19, no. 3, Jan. 2010, doi: 10.1145/1656250.1656253/SUPPL_FILE/A9-XU-APNDX.PDF.
[18] "(PDF) Quality of Context: What it is and why we need it." Accessed: Apr. 21, 2025. [Online]. Available: https://www.researchgate.net/publication/303963518_Quality_of_Context_What_it_is_and_why_we_need_it
[19] A. Manzoor, H. L. Truong, and S. Dustdar, "Using Quality of Context to Resolve Conflicts in Context-Aware Systems," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 5786 LNCS, pp. 144–155, 2009, doi: 10.1007/978-3-642-04559-2_13.
[20] S. Fan *et al.*, "A new QoC parameter and corresponding context inconsistency elimination algorithms for sensed contexts and non-sensed contexts," *Appl. Intell.*, vol. 52, no. 1, pp. 681–698, Jan. 2022, doi: 10.1007/S10489-021-02226-4/METRICS.
[21] M. Chen, H. Xu, H. Xiong, L. Pan, B. Du, and F. Li, "A new overall quality indicator OQoC and the corresponding context inconsistency elimination algorithm based on

OQoC and Dempster–Shafer theory," *Soft Comput.*, vol. 24, no. 14, pp. 10829–10841, Jul. 2020, doi: 10.1007/S00500-019-04585-0/METRICS.
[22] A. P. Dempster, "Upper and Lower Probabilities Induced by a Multivalued Mapping," *Stud. Fuzziness Soft Comput.*, vol. 219, pp. 57–72, 2008, doi: 10.1007/978-3-540-44792-4_3.
[23] Q. Liu *et al.*, "A novel context inconsistency elimination algorithm based on the optimized Dempster-Shafer evidence theory for context-awareness systems," *Appl. Intell.*, vol. 53, no. 12, pp. 15261–15277, Jun. 2023, doi: 10.1007/S10489-022-04223-7/METRICS.
[24] I. Ben-Gal, "Bayesian Networks," *Encycl. Stat. Qual. Reliab.*, Dec. 2007, doi: 10.1002/9780470061572.EQR089.
[25] K. E. Ko and K. B. Sim, "Development of context aware system based on Bayesian network driven context reasoning method and ontology context modeling," *2008 Int. Conf. Control. Autom. Syst. ICCAS 2008*, pp. 2309–2313, 2008, doi: 10.1109/ICCAS.2008.4694191.
[26] S. Dutta, "Fuzzy Logic Applications: Technological and Strategic Issues," *IEEE Trans. Eng. Manag.*, vol. 40, no. 3, pp. 237–254, 1993, doi: 10.1109/17.233185.
[27] M. Hellmann, "Fuzzy Logic Introduction".
[28] M. Pérez-Gaspar, J. Gomez, E. Bárcenas, and F. Garcia, "A fuzzy description logic based IoT framework: Formal verification and end user programming," *PLoS One*, vol. 19, no. 3, Mar. 2024, doi: 10.1371/JOURNAL.PONE.0296655.