Adapting Event Processing in Dynamic IoT Applications, Meeting Evolving Requirements for Quality, Privacy, and Rule Autonomy

Majid Lotfian Delouee

rijksuniversiteit groningen / faculty of science and engineering / bernoulli institute

The work described in this thesis was performed at the *Distributed Systems* group, part of the Bernoulli Institute for Mathematics, Computer Science, and Artificial Intelligence at the University of Groningen, the Netherlands.

rijksuniversiteit
groningen

Adapting Event Processing in Dynamic IoT Applications, Meeting Evolving Requirements for Quality, Privacy, and Rule Autonomy

Proefschrift

ter verkrijging van de graad van doctor aan de
Rijksuniversiteit Groningen
op gezag van de
rector magnificus prof. dr. ir. J.M.A. Scherpen
en volgens besluit van het College voor Promoties.
De openbare verdediging zal plaatsvinden op
maandag 1 december 2025 om 9.00 uur

door

Majid Lotfian Delouee

geboren op 21 september 1989

Promotor
Prof. dr. B. Koldehofe

Copromotor
Dr. V. Degeler

Beoordelingscommissie
Prof. dr. D. Karastoyanova
Prof. dr. D. Eyers
Prof. dr. S. Bowmik

To Nafiseh and Elena,
my Family.

Summary

This thesis presents concepts and algorithms to realize a quality- and privacy-aware Complex Event Processing (CEP) system, supported by autonomous rule generation and generalizable to various IoT applications operating in dynamic and heterogeneous environments. In particular, four major contributions are introduced: (i) a mechanism for quality-aware selection and reconfiguration of sensor inputs for CEP systems, (ii) a graph-based mechanism for pattern-level privacy protection, (iii) an AI-based approach for autonomous rule generation and refinement, and (iv) an application of the privacy–utility trade-off (PUT) concept in vehicular networks for online object detection.

In Chapter 3, we introduce AQuA-CEP, in which consumer-defined quality policies guide complex event detection. These policies are evaluated to automatically select and configure appropriate data sources within the sensor infrastructure. We explored different ways of expressing quality policies and analyzed their influence on the quality monitoring process. Furthermore, several methods for evaluating and applying quality-related adaptations were investigated, and their impact on correlation efficiency was analyzed.

We evaluated AQuA-CEP in IoT scenarios by assessing performance based on defined quality policies and query adaptation informed by quality monitoring. The results demonstrate that AQuA-CEP enhances DCEP system performance in terms of result quality while satisfying consumer-defined quality requirements. Additionally, quality-based adaptation extended network lifetime by optimizing sensor energy consumption through efficient data source

selection.

In the second contribution, within the APP-CEP framework (Chapter 4), we address the challenge of integrating pattern-level privacy into event-based systems. APP-CEP employs selective obfuscation techniques to conceal private information. Unlike existing approaches, we enforce privacy without relying on actual events in the data streams. To achieve this, we use CEP-like patterns to express both user queries and privacy requirements.

Privacy is preserved through the generation of pattern-dependency graphs and the dynamic selection of obfuscation strategies that minimize interference with both sensitive and non-sensitive patterns—an essential aspect for maintaining acceptable Quality of Service (QoS). Moreover, we model potential attackers' background knowledge to evaluate their ability to compromise privacy and refine our obfuscation procedures accordingly. Evaluations using an online transaction dataset and a medical dataset demonstrate that APP-CEP effectively balances privacy and utility. Our approach to background knowledge modeling successfully prevents attackers from detecting adjustments in the input streams.

In Chapter 5, within the context of GPT-CEP, we employ large language models (LLMs) and prompt engineering to generate accurate and efficient CEP rules. To this end, we introduce the *AGES index*, a comprehensive metric evaluating model size, rule generation efficiency, and rule accuracy. The performance of GPT-CEP is assessed using this index and compared against baseline strategies. GPT-CEP consistently outperforms the baselines, with the AGES index playing a key role in selecting the optimal language model and prompt engineering techniques.

GPT-CEP also incorporates a cluster-based approach to refine the initial LLM-generated rules. Simulated annealing is applied to further optimize rule thresholds, leading to improved detection accuracy. Overall, GPT-CEP represents a significant advancement in autonomous rule generation for CEP systems. By combining large language models, prompt engineering, and rule refinement, it provides a promising solution for automating rule creation and enhancing the performance of CEP applications.

Finally, in Chapter 6, we explore the application of privacy–utility tradeoffs in stream processing systems, focusing on autonomous vehicular networks

within the AR-CFL framework. AR-CFL offers a compelling solution by enabling model training among distributed vehicles without sharing raw sensory data, thus addressing privacy concerns. In vehicular networks, continuous adaptation and real-time model updates are essential due to dynamic driving conditions and constant data generation. AR-CFL facilitates this adaptation through clustered federated learning, which reduces communication overhead and ensures efficient resource utilization by clustering participants based on available bandwidth, storage, and computational power.

AR-CFL thus provides a balanced trade-off between privacy and utility, essential for managing continuous data streams generated by vehicles while safeguarding sensitive information. The framework dynamically optimizes the number of clusters and selects participants based on available resources, ensuring efficient event processing. Evaluation results show that AR-CFL achieves robust detection performance for online perception model training, even with non-IID data across varying traffic densities. Moreover, training efficiency among participating nodes improved by up to 25%, while communication overhead was reduced by 33% compared to conventional federated learning methods, all while preserving privacy through local data analysis.

Contents

Acknowledgments **xv**

1 Introduction **1**
1.1 Research Challenges 4
1.2 Research Questions and Contributions 8
1.2.1 Mapping RQs and Contributions to the Pub-Sub Model 14
1.3 Structure of the Thesis 15
1.4 Publication Origins 15

2 Fundamentals and State-of-the-art **19**
2.1 Event-based Systems 19
2.1.1 Complex Event Processing 21
2.1.2 Quality Monitoring 32
2.1.3 Privacy Protection 38
2.2 Employing AI for Stream Processing 43
2.2.1 CEP Rule Generation 44
2.2.2 Federated Learning in CEP 48
2.3 Summary 50

3 Quality Monitoring in CEP **51**
3.1 Quality-Aware Event Source Selection Problem 54
3.2 The AQuA-CEP System Design 56
3.2.1 Quality Requirement Description 60

3.2.2 Quality Monitoring . . . 62
3.2.3 Sensing Deployment Adaptation . . . 64
3.3 Evaluation . . . 69
3.3.1 Simulation Setup . . . 69
3.3.2 Simulation Results . . . 76
3.4 Related Work . . . 83
3.4.1 Sensor Selection . . . 83
3.4.2 Quality Monitoring . . . 84
3.5 Summary . . . 84

4 Privacy Protection in CEP **87**
4.1 Pattern-level Privacy Protection Problem . . . 90
4.2 The APP-CEP System Design . . . 93
4.2.1 Pattern Dependency Graph . . . 99
4.2.2 Event Dependencies . . . 103
4.2.3 Stream Features . . . 105
4.2.4 Obfuscation Technique Assignment . . . 106
4.3 Evaluation . . . 109
4.3.1 Evaluation Results for Webshop Dataset . . . 110
4.3.2 Evaluation Results for Medical Dataset . . . 115
4.4 Related Work . . . 118
4.5 Summary . . . 119

5 CEP Rule Generation **121**
5.1 Autonomous Rule Generation Problem . . . 124
5.2 The GPT-CEP System Design . . . 126
5.3 Autonomous LLM-based Rule Generation . . . 130
5.4 Federated Rule Refinement . . . 133
5.5 Evaluation . . . 134
5.5.1 LLM Rule Generation . . . 134
5.5.2 Rule Refinement Evaluation . . . 146
5.6 Related Work . . . 151
5.7 Summary . . . 152

6 Federated Stream Processing: An application in Vehicular Networks **155**
6.1 Related Work . . . 157
6.1.1 FL-based Object Detection in Vehicular Context . . . 158
6.1.2 Clustered Federated Learning in Vehicular Context . . . 158
6.1.3 Research Gaps . . . 159
6.2 Object Detection Model Training in Vehicular Networks . . . 159
6.3 AR-CFL System Design . . . 161
6.3.1 RCFL Framework . . . 163
6.3.2 ICFL Framework . . . 165
6.3.3 Handling The Limited Storage Challenge . . . 165
6.4 Evaluation . . . 167
6.4.1 Evaluation Scenario and Experimental Setup . . . 168
6.4.2 Data Generation . . . 172
6.4.3 Results and Discussion . . . 173
6.4.4 Limitations of the Study . . . 179
6.5 Summary . . . 180

7 Conclusion and Future Work **181**
7.1 Contributions Revisited . . . 182
7.2 Key Results . . . 183
7.3 Future Work . . . 186

Bibliography **190**

Samenvatting **213**

Acknowledgments

As I look back on the years of my PhD journey, I find it difficult to believe how much has changed — in my work, in my life, and in myself. These years were not only about research, papers, and experiments, but also about growing as a person, learning new ways to think, and discovering what it truly means to persevere. None of this would have been possible without the guidance, patience, and kindness of many people to whom I owe my deepest thanks.

First and foremost, I would like to express my sincere gratitude to my promotor, Prof. dr. Boris Koldehofe. Working with Boris has been one of the most valuable experiences of my life. He was never just a supervisor to me, but a mentor and a friend who treated me with trust and respect. Our discussions were always inspiring, and his ability to look at problems from different angles helped me grow as a researcher. I learned from him not only how to conduct good research, but also how to be patient, open-minded, and creative. I will always remember our conversations, both about science and about life, and how they gave me the confidence to keep going, even in difficult moments.

I am equally grateful to my co-promotor, Dr. Victoria Degeler, whose guidance, warmth, and constant encouragement were invaluable to me. Working closely with Victoria, especially during my research stay at the University of Amsterdam, was one of the best parts of my PhD. She always listened carefully, offered honest feedback, and pushed me to think more clearly. I appreciate not only her scientific insight, but also her kindness and care for people.

I would also like to sincerely thank the members of my thesis committee: Prof. dr. Dimka Karastoyanova, Prof. dr. D. Eyers, and Prof. dr. S. Bowmik. Dimka, in particular, holds a special place in my journey. As my wife's supervisor and as a person, she has always been supportive and generous with her advice, not only about research but about life and parenting as well. My heartfelt thanks also go to Professors Eyers and Bowmik for their time and effort in evaluating my work and for their thoughtful feedback.

I am very proud to have been part of the Distributed Systems Group. It was more than just a workplace, it felt like a family. My deepest thanks go to Saad Saleh, with whom I shared countless hours in our office, room 596, at the very end of the building. We shared so many discussions, sometimes about complex technical problems, sometimes about life itself, and those moments shaped my PhD years in unforgettable ways.

I am also grateful to Mostafa Hadadian, my compatriot and friend, for his support, his honesty, and all our shared experiences as two Iranians far from home. To Andrés Tello and Bochra Boughzala, my fellow office mates, thank you for the good times, laughter, and support. My sincere appreciation also goes to Prof. Alexander Lazovik, the chair of our group, for leading by example and creating a friendly, collaborative environment. I am thankful as well to my collaborators and colleagues from other universities: the wonderful people at Ilmenau University, which I had the pleasure of visiting twice. I also had a great experience collaborating with the University of Oslo team on the Parrot project.

The journey of doing a PhD abroad comes with challenges beyond research. Moving to a new country, adapting to a new language, and adjusting to a new culture were not easy at first. But I am deeply grateful to the Dutch people for their openness, kindness, and sense of community. I found the Netherlands to be a welcoming and supportive place, and it eventually became a second home to me.

Among all the people who stood beside me, no one played a greater role than my wife, Nafiseh. She was my strength throughout every difficult day. We shared this journey together, two PhD students in the same building, Bernoulli, supporting each other through endless work, deadlines, and moments of exhaustion. I still remember the evenings when we both stayed late,

working silently in our offices, just a few doors apart. Her patience, her love, and her unshakable belief in me were what made it possible to reach this point. Moving to a new country together, facing uncertainty, and building a new life far from home, these experiences could have been overwhelming, but with her by my side, they became meaningful. She helped me through the hardest days and celebrated with me the small victories that made the difference. Nafiseh, thank you for being my partner in every sense, in life, in work, and in dreams. This achievement belongs to you as much as to me.

To our daughter, Elena, who arrived during this long journey, you brought new light and meaning into our lives. I will never forget the moment I first held you, it changed everything. Your laughter gave me strength when I was tired, and your presence reminded me of why all the effort was worth it. Watching you grow while I was finishing this thesis was the most beautiful reminder that life continues beyond research. You have been my greatest motivation, and I dedicate this work to you and your mother.

I would also like to thank my parents, who have always supported me with love and belief, even from far away. My father and mother taught me the value of hard work, patience, and kindness. Their encouragement gave me the courage to pursue this path, even when it meant living far from them. To my brother Amin and sister Elham, thank you for always being there, for cheering me on, and for reminding me of home when I missed it most.

These years have been full of challenges, discoveries, and moments I will never forget. I close this chapter with deep gratitude to all the people who made this journey possible, and to those who made it worthwhile. Whatever comes next, I know I carry with me everything I have learned, not only about science, but about life, friendship, and love.

Majid Lotfian Delouee
Groningen
November 17, 2025

Chapter 1

Introduction

> "Rest not a moment from learning's embrace, Let no doubt linger in pursuit of grace."
>
> - Ferdowsi (Revised by GPT4o)

Due to omnipresent advancement in sensing technology in the Internet of Things (IoT) environment, a continuous acceleration towards analyzing the generated data has emerged to react quickly and correctly to the environmental dynamics. As examples of rapidly evolving IoT-based scenarios, healthcare systems have attracted attention to facilitate health-related procedures ranging from hospitalizations to monitoring systems [1]. Besides healthcare, other industries, such as e-commerce [2], have gained many benefits from the data collected by sensors in real-time about customer preferences, behaviors, and usage patterns. This data can be leveraged to create personalized shopping experiences, such as product recommendations, customized promotions, and targeted advertisements. As per Statista's data [3], the count of connected IoT devices stood at around 15.14 billion by 2023, nearly double the global human population of eight billion. Finance Online predicts that this figure is set to grow annually, surpassing 25 billion in the next seven years, propelled by the adoption of 5G and other emerging technologies.

The value of IoT data tends to decrease over time, highlighting the importance of analyzing it as soon as it is generated, ideally in near real-time, since much of its usefulness depends on timely insights for monitoring and decision-making. Real-time data analytics mechanisms have proven effective for processing information in this context. Complex Event Processing (CEP) systems [4, 5] are widely employed to process data from IoT sources and extract insights for various applications (cf. Figure 1.1). CEP systems handle

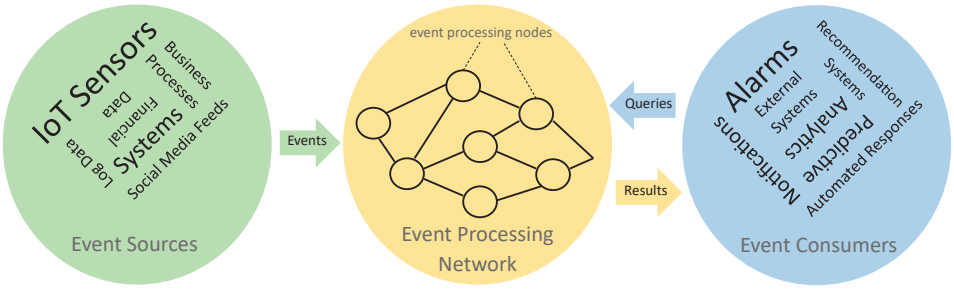


Figure 1.1: A comprehensive look at CEP systems. Simple events originating from event sources are fed into the Event Processing Network and undergo thorough analysis, aligning with the queries posed by event consumers and applications. Subsequently, the system generates results tailored to meet the demands of these queries.

large volumes of data in real-time, detecting situations of interest, such as environmental changes, through "events" (or simple events) like a temperature reading in a room. More complex scenarios, such as detecting a fire, are represented by "complex events", which are identified by applying continuous queries to data streams, for instance, recognizing patterns like "simultaneous high temperature and smoke". Additionally, CEP allows for the integration of user-defined requirements, making it adaptable to specific quality and privacy needs, such as customer requests for a particular level of accuracy in event detection.

Currently, many famous companies are utilizing event-based systems, including Heron (i.e., direct successor of Apache Storm), which is used by X (i.e., formerly Twitter) to process billions of events in real-time [6], Apache Kafka at JPMorgan Chase & Co (i.e., one the biggest financial institutions in the world) to create new products, respond to customers and make business decisions in real-time [7], and Apache Flink at Amazon Web Services to build and run real-time streaming applications [8].

Although CEP systems offer numerous benefits, they still face significant shortcomings in meeting evolving user expectations, particularly in dynamic IoT environments. These challenges stem from maintaining system perfor-

mance, data protection, and operational independence while balancing conflicting user requirements. Addressing non-functional demands such as quality, privacy, and autonomy is not always mandatory, but doing so significantly enhances users' trust in the system's reliability. For example, while a recommendation system can function without prioritizing privacy, neglecting it may lead to a loss of customers due to non-transparent privacy policies. CEP system developers have started to recognize the importance of these requirements, particularly for processing IoT data. However, current CEP systems still struggle with effectively supporting while balancing users' requirements.

Ensuring high-quality event processing in dynamic IoT environments remains a challenge. Most Complex Event Processing (CEP) systems optimize performance by strategically placing detection logic across computing resources to meet predefined Quality of Service (QoS) requirements [9–12]. However, these approaches assume static deployment conditions and fail to adapt to evolving IoT environments. While some studies have explored adaptive sensor deployments in response to environmental changes [13, 14], they focus on narrow system components such as query optimizations and lack mechanisms to dynamically balance competing quality requirements across multiple queries.

Existing CEP mechanisms primarily address attribute-level privacy using access control but overlook privacy risks stemming from event patterns. In many IoT applications, sensitive information can be inferred not just from individual attributes but from the sequence and correlation of events [15, 16]. This creates vulnerabilities in pattern-based event processing, where safeguarding user privacy requires strategies beyond traditional access control methods.

Another challenge in CEP systems is the lack of autonomy, particularly in the area of rule generation. While some systems attempt to generate CEP rules by extracting event relationships from historical data [17–20], they rely on clean and reliable datasets and struggle to generate meaningful rules tailored to users' specific needs. These methods are also ineffective in handling unforeseen events in dynamic environments, limiting their adaptability.

Balancing privacy and utility in CEP remains an open problem. Although previous studies have examined this trade-off in stream processing [21–27],

few address privacy risks associated with pattern-based processing [15]. The ability to dynamically adjust this balance at runtime and extend it to diverse IoT scenarios remains a significant challenge.

Despite advancements in CEP, existing systems remain rigid, relying on predefined rules optimized for fixed IoT configurations. This limits their ability to adapt to changing sensor availability, event patterns, and privacy constraints over time. This thesis addresses these limitations by making CEP systems more adaptive and resilient. The proposed approach enables CEP to dynamically adjust to evolving sensor deployments, ensuring effective event detection despite environmental changes. It also incorporates privacy-aware processing beyond traditional access control, protecting sensitive information at the pattern level. Finally, it introduces mechanisms for autonomous rule generation, allowing CEP systems to refine event processing rules without requiring manual intervention.

By tackling these challenges, this work enhances the ability of CEP systems to operate in real-world IoT environments, where system conditions and user needs continuously evolve. The proposed solutions introduce adaptive mechanisms that ensure CEP remains efficient, privacy-conscious, and capable of autonomously managing event-driven data streams.

1.1 Research Challenges

Building on the aforementioned motivations, this thesis delves into four core challenges within the context of CEP systems that we aim to address.

The Quality Problem.

IoT deployments generate a diverse range of sensor data streams, providing multiple sources for event detection. The availability and quality of these streams directly impact the accuracy and reliability of downstream systems that process them. For example, in location tracking applications, selecting data from different sensors, such as GPS, Wi-Fi access points, or Bluetooth beacons, affects how frequently an object's position is updated and how accurately movement is detected [28]. However, most Complex Event Processing (CEP) systems are designed for a fixed sensor deployment, meaning they pro-

cess data from predefined sources without dynamically adjusting to new or alternative sensor streams. This rigid approach limits their ability to adapt to changing environments, sensor failures, or user requirements.

In practice, this limitation becomes critical when sensors become unavailable, unreliable, or when multiple event sources provide conflicting data. Users expect accurate query results that align with predefined quality requirements, such as precision, recall, or update frequency. However, meeting these expectations at runtime is challenging if the system does not support adaptive stream selection. This thesis explores how to enable CEP systems to dynamically adjust input streams based on availability, reliability, and evolving user needs, ensuring that quality requirements are met without requiring manual intervention.

Without such adaptive mechanisms, applications themselves must handle sensor adaptation, which leads to additional complexity. Developers would often need to define new processing rules whenever sensor configurations change, leading to potential downtime, manual reconfiguration, and inconsistent event detection. Instead of placing this burden on individual applications, integrating dynamic stream adaptation into CEP systems allows for more seamless and reliable event processing in highly dynamic IoT environments.

CEP systems face significant challenges in adapting to quality dynamics in IoT environments. Quality, typically defined in terms of Quality of Service (QoS) and Quality of Results (QoR), is highly dependent on the characteristics of simple events generated by distributed sensors. These dynamics include factors such as stream availability, event rates, and evolving customer requirements. For instance, in a recommendation system, customer preferences may shift over time based on their experiences, requiring frequent updates to recommendations to remain relevant. However, streams that once met QoS and QoR standards may suddenly fail to do so due to changes in stream quality or user preferences.

The core problem lies in the inability of current CEP systems to adapt dynamically to such changes. While state-of-the-art approaches attempt to address this by decoupling detection procedures from adaptation strategies [10–12], they often rely on static information defined during the design phase, making them inadequate for real-time adjustments. This limitation

prevents CEP systems from meeting QoS and QoR requirements effectively as quality demands evolve. To address this, there is a need for mechanisms that enable CEP systems to continuously acquire and react to changing quality requirements in real-time, ensuring timely and reliable performance in dynamic IoT environments.

The Privacy Problem.

Privacy protection in IoT applications has traditionally focused on safeguarding sensitive information at the attribute level, as defined by data owners. However, this approach is inadequate when dealing with complex patterns formed by combining multiple individual events. Even if individual events are not sensitive on their own, their aggregation can unintentionally reveal sensitive information [29]. For example, while blood pressure and heart rate data may not seem sensitive in isolation, their combined analysis can indicate a medical condition. This gap in protection highlights a critical issue: existing privacy protection mechanisms (PPMs) are not designed to safeguard privacy at the pattern level, making it difficult to meet user privacy requirements in IoT scenarios.

A further challenge is that protecting privacy-sensitive patterns (referred to as private patterns) can interfere with the accurate detection of non-sensitive patterns (referred to as public patterns). CEP systems must balance both needs (i.e., supporting the detection of both types of patterns), yet most existing solutions fall short. While a few approaches offer privacy protection for patterns [15], they are typically limited to sequence-based patterns and rely heavily on input streams to apply obfuscation techniques (such as event dropping or reordering) [16]. These methods fail to address the obfuscation needs of a variety of private patterns, particularly in dynamic event streams, limiting their applicability in diverse IoT environments.

The core problem is that current privacy-enhancing techniques are unable to effectively protect against privacy-sensitive patterns in complex and dynamic IoT event streams, while also preserving the accuracy of non-sensitive pattern detection. This limits their ability to meet the dual requirements of privacy protection and reliable pattern detection in real-world applications.

The Rule Autonomy Problem.

CEP systems use operators such as sequence and conjunction to define patterns, known as CEP rules, that detect complex situations by identifying causal relationships between events and situations of interest [17, 30]. Accurate rule generation is critical for effective event detection, but traditionally, these rules are crafted manually by domain experts, introducing the risk of human error and bias [31–34]. As IoT environments become more dynamic, where sensor deployments change over time, manually defined rules become increasingly inadequate. This highlights the need for autonomous rule generation that can continuously adapt to evolving sensing configurations and event patterns, provided it does not propagate error or bias in the process.

Most existing CEP systems rely solely on historical data to generate rules. While this approach works for static deployments, it fails in real-time applications where new or unforeseen events occur. IoT systems introduce additional complexity because changes in sensor availability enable different ways to detect the same events. A rigid rule set based on past observations cannot accommodate such variations, limiting the system's ability to detect emerging patterns effectively. Although prior research has explored autonomous rule generation [18–20], existing methods are often domain-specific and lack the flexibility to generalize across different IoT scenarios. Furthermore, approaches that update rule bases [35] typically depend on predefined rule repositories, preventing them from dynamically generating new rules that reflect changing environments and evolving user requirements.

The core limitation is that current CEP systems lack the ability to autonomously and dynamically generate rules in real-time, particularly for new and unforeseen situations, raising questions about the appropriate level of rule abstraction that such systems can effectively manage. As IoT deployments evolve, event detection mechanisms must adapt to new sensor configurations and changing event sources, ensuring that rule generation models remain relevant over time. Moving beyond static, history-based methods, CEP systems need real-time, adaptive rule generation techniques that continuously refine rules based on dynamic sensing conditions and evolving event landscapes.

The Privacy-Utility Trade-Off (PUT) Problem.

In information processing systems, improving the quality of service (i.e., utility) typically requires collecting more data, but data owners are often reluctant to share large amounts of data due to privacy concerns. To balance these competing interests, CEP systems must enforce appropriate policies to establish a privacy-utility trade-off (PUT). Most existing CEP systems tend to focus on either privacy or utility, making it difficult to find a solution that satisfies both data owners and application customers. Any privacy-enhanced CEP system must consider the impact of obfuscation techniques on system utility, and in some cases, a trade-off may be necessary, where privacy is partially sacrificed to achieve the desired level of utility (or vice versa) [36]. Applying this trade-off in various IoT scenarios, such as autonomous driving in vehicular networks, underscores both the limitations and potential of this approach, which can lead to the development of a more generalizable PUT framework [37]. The concept of PUT has been widely explored in areas such as database analysis [38], smart grids [39], healthcare systems [40,41], and machine learning [42–44]. However, in the CEP domain, only a few studies have addressed the privacy-utility trade-off (PUT) [15, 45], and their approaches are tailored to specific applications. While PUT has been explored in other domains, its integration into CEP remains underdeveloped. A key challenge is balancing privacy protection with real-time event detection accuracy, particularly as sensor deployments and query requirements evolve. Existing methods lack adaptive mechanisms to dynamically adjust privacy constraints without compromising detection quality. This gap highlights the need for scalable solutions that enable CEP systems to refine privacy-aware event detection in changing environments, making this a promising direction for future research.

1.2 Research Questions and Contributions

The overall goal of this thesis is to develop models, methods, and algorithms that enable CEP systems to evolve dynamically in response to changing IoT environments and user requirements. Rather than a one-time customization, this work focuses on continuous adaptation, ensuring that CEP systems can

autonomously refine their event processing strategies as sensor deployments, event patterns, and privacy needs evolve. This approach addresses the key challenges of maintaining high-quality event detection, safeguarding user privacy, and autonomously generating CEP rules tailored to real-world scenarios where balancing privacy and utility is critical. The following sections outline the research questions and the corresponding contributions of this thesis.

Research Question 1: *How to define, measure, and monitor quality requirements adaptively to maintain QoS and QoR in the face of dynamics in the environment?*

To effectively leverage multiple sensor deployments in CEP systems, it is crucial to understand how the choice of a specific sensing configuration impacts the quality of detected events. Different sensor deployments influence both Quality of Service (QoS) and Quality of Results (QoR), requiring adaptive mechanisms to ensure consistent event processing. To address this, we propose AQuA-CEP, an adaptive quality-aware CEP method that dynamically switches between input streams to maintain predefined quality requirements. Seamlessly transitioning between sensing configurations while preserving continuous query output is a complex challenge, particularly when accounting for the specific requirements defined by consumers. A naive approach to this problem can lead to significant performance issues, including high switching overhead, incorrect query results, and interruptions in event delivery. These challenges can ultimately prevent the system from meeting QoS and QoR expectations, making adaptive stream selection a critical research problem in CEP. For more details, AQuA-CEP provides the following **contributions**.

1. This thesis introduces a policy-based approach to defining quality requirements for complex event processing, aiming to enhance data processing efficiency and optimize resource allocation.

2. This thesis presents a framework for integrating quality monitoring into CEP, enabling the adaptive selection of data sources while ensuring compliance with user-defined quality expectations.

3. This thesis explores the design of monitoring agents that detect policy

violations and initiate adaptive responses, assessing their influence on both service quality and result accuracy.

4. The effectiveness and constraints of the proposed approach are examined through evaluation on both real-world and synthetic datasets.

Research Question 2: *How can the emerging concept of pattern-level privacy improve data protection by enabling the selective assignment and adaptation of obfuscation techniques?*

In stream processing systems, privacy imposes a challenge in analyzing events. Achieving a feasible trade-off between preserving privacy and detecting useful information remains challenging. To this end, APP-CEP is introduced based on a dynamic assignment of obfuscation techniques (OT) to optimize the following constraints. First, complex event patterns exhibit causal dependencies, necessitating a solution based on multiple obfuscation techniques. Second, it is important to consider an adversary's background knowledge when selecting an obfuscation technique. For instance, if an event called e_1 appears in the result of one query, but not in another, the adversary may be able to deduce some parts of the original stream by combining the information obtained from both. Therefore, it is essential to choose an appropriate obfuscation technique to prevent the adversary from inferring sensitive information. Third, DCEP systems are dynamic, with changing queries and privacy requirements. Input sources can vary, and an adversary's knowledge increases over time. Therefore, we present APP-CEP as a solution to these challenges. Our **contributions** are as follows.

1. This thesis introduces a privacy-preserving mechanism that processes privacy requirements and event patterns as input, determining the most effective obfuscation strategy to protect sensitive patterns in query results.

2. This thesis presents a graph-based approach for selecting obfuscation techniques, modeling pattern dependencies, and analyzing input stream characteristics to predict optimal switching times, ensuring both privacy protection and scalability.

3. This thesis proposes a framework that allows data owners to dynamically model an adversary's potential background knowledge by leveraging event dependencies and statistical insights extracted from historical event streams.

4. This thesis evaluates the proposed approach using real-world data, demonstrating how APP-CEP enhances complex event processing systems in practical scenarios.

Research Question 3: *How and to what extent can large language models support the autonomous definition, validation, and refinement of CEP rules?*

The domain of natural language processing has witnessed significant improvements through the emergence of large language models, which have led to greater reliability and accelerated interpretation of human requests, e.g., AI-based e-commerce customer support [46]. The conversion of user queries and data owners' privacy requirements to CEP-like patterns can derive considerable benefits from such approaches. In doing so, the burden of errors posed by producers and consumers can be alleviated, thereby enhancing the overall efficiency and accuracy of the process. Another challenge is how prompt engineering can assist large language models (LLMs) in producing relevant and concise responses [47]. To this end, we propose GPT-CEP, a general platform that initiates the CEP-like representations of queries and privacy demands, powered by LLMs (e.g., ChatGPT [48], Google Gemini [49], etc.), followed by a federated rule validation and testing consuming local data in clients. Eventually, aggregating insight collected from clients' training processes produces and tests a generic version of each CEP rule. For more details, GPT-CEP provides the following **contributions**.

1. This thesis introduces an autonomous mechanism for generating and refining complex event processing (CEP) rules using the natural language processing capabilities of large language models (LLMs) and validates these rules over distributed data through a federated platform.

2. This thesis develops different strategies for prompt engineering and compares the effectiveness of multiple LLMs in generating CEP-like rules.

3. This thesis investigates methods for validating initial CEP rules using local client data, proposing techniques for generating rule updates and algorithms for aggregating them to produce a reliable, generalized rule set.

4. This thesis assesses the performance of prompt engineering techniques and the federated rule validation and refinement process using a real-world activity recognition datasets, highlighting the approach's strengths and limitations.

Research Question 4: *How to generate PUT-aware algorithms and investigate their limitations on specific IoT scenarios?*

The Internet of Things (IoT) enables devices to connect and share data, creating smarter systems that improve efficiency and safety in various domains. From healthcare to smart cities, IoT applications rely on real-time data processing to make informed decisions, e.g., Autonomous driving that promises to make vehicle movements more predictable and less reliant on the driver's decisions, increasing road safety and traffic efficiency. However, the limited onboard sensing of today's vehicles results in a narrow perception of the environment [50]. To overcome this limitation, exchanging collected data among vehicles and all road users can help achieve a better perception of the environment [51]. However, this can compromise drivers' privacy. As a solution to this, Federated Learning (FL) trains object detection models on distributed data while preserving privacy by keeping raw data local.

To provide PUT in autonomous driving, we propose a framework called AR-CFL that uses Adaptive Resource-aware Clustered Federated Learning, designed to optimize the factors impacting PUT requirements within vehicular environments. We have conducted a systematic comparison of the outcomes obtained with different design decisions and configuration options, and we discuss these results in detail. In summary, AR-CFL's **contributions** are:

1. This thesis presents a framework, AR-CFL, that enhances federated learning (FL) through adaptive clustering, enabling hierarchical object detection while maintaining privacy.

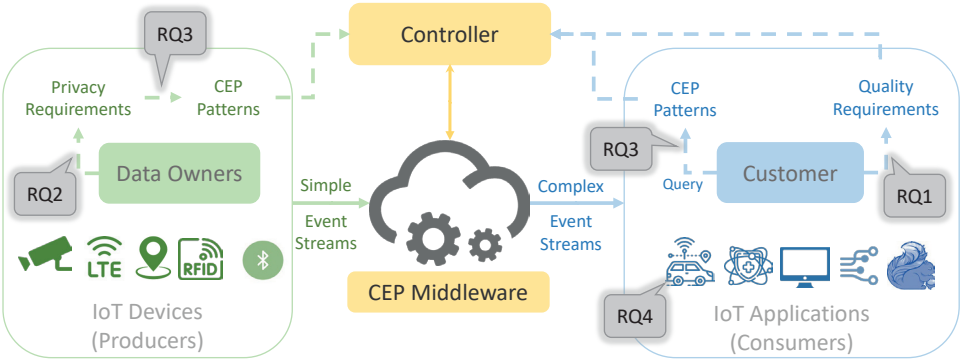


Figure 1.2: This thesis's research questions (RQ) and contributions are inspired by a publish-subscribe paradigm. On the right (i.e., Consumers), IoT application customers issue queries complemented by quality requirements (cf. Chapter 3 (RQ1)). On the left (i.e., Producers), IoT devices generate streams of simple events, and privacy demands of data owners are gathered (cf. Chapter 4 (RQ2)). On both sides, user-definable requirements and queries are represented as CEP patterns using the proposed autonomous rule generation approach (cf. Chapter 5 (RQ3)). Last, an application of established PUT is explored in a vehicular network scenario (cf. Chapter 6 (RQ4)).

2. This thesis introduces a novel strategy, *Dynamic Cluster Members Involvement*, which dynamically adjusts the number of participating clients within each cluster throughout the learning process.

3. This thesis develops two algorithms that leverage *Dynamic Cluster Members Involvement* to facilitate both inter-cluster and intra-cluster object detection within a hierarchical federated learning framework.

4. This thesis evaluates AR-CFL using a comprehensive benchmark based on three newly generated synthetic datasets, developed as part of this research using the *CARLA* simulator.

1.2.1 Mapping RQs and Contributions to the Pub-Sub Model

In Figure 1.2, we revisit the RQs and contributions of this thesis in the context of a publish-subscribe architecture, positioning them within the data and control flow of the system. This architecture captures both the consumer side, where IoT application users issue queries with specific quality requirements, and the producer side, where IoT devices generate event streams while data owners define privacy constraints. These two sides represent the user-driven adaptation of CEP systems, where our approach enables CEP pattern formulation through autonomous rule generation. Additionally, we demonstrate how the privacy-utility trade-off (PUT) is applied in a vehicular network scenario, highlighting how our contributions integrate into the broader event-driven architecture to enhance adaptability, privacy, and quality-aware processing.

To address the primary research question, we focus on the quality problem within the proposed method, AQuA-CEP, and present a solution in two key steps. (i) We propose a policy-driven specification of complex events by dynamically reconfiguring data sources while fulfilling customers' quality requirements. (ii) We explore strategies for configuring quality monitoring agents that trigger adaptation strategies upon any quality policy violation and address the impacts each configuration might have on the DCEP system's performance in terms of QoS and QoR. To address the second research question, we focus on the *privacy* problem in greater depth. Our contribution is the development of a method called APP-CEP, which addresses privacy concerns by capturing privacy requirements and situations of interest as event patterns. The system then determines the most effective obfuscation technique to conceal each private pattern, based on a graph structure representing pattern dependencies and a dynamic model of the adversary's potential background knowledge. To answer the third research question, we specifically address the *rule autonomy* challenge in CEP systems. Our contribution is the introduction of a method called GPT-CEP, which leverages the natural language processing capabilities of large language models (LLMs) to translate human-defined rules into CEP-like representations. The system then validates and refines these rules through a federated mechanism, ensuring adaptability and accuracy in dynamic environments. Finally, we explore the

application of establishing PUT in a vehicular network scenario by proposing a method called AR-CFL by employing a clusters federated learning approach to detect objects in the environment in an online learning approach.

1.3 Structure of the Thesis

The thesis is structured into six further chapters. Chapter 2 presents the preliminaries and related work analysis for the contributions of this thesis. Chapter 3 provides a comprehensive description and technical solutions for adaptive quality monitoring in CEP. Chapter 4 presents graph-based methods and algorithms addressing the privacy protection problem. Chapter 5 presents novel autonomous rule generation and federated refinement mechanisms targeted toward the rule autonomy problem. Chapter 6 describes an application of clustered federated learning for online object detection in vehicular networks, as an example of established PUT. Finally, Chapter 7 provides a summary of the contributions of this thesis and highlights future directions.

1.4 Publication Origins

The chapters in this thesis are based on the following papers:

- **Chapter 3**
  M. Lotfian Delouee, B. Koldehofe, V. Degeler – *"Towards Adaptive Quality-Aware Complex Event Processing in the Internet of Things."*, In Proceedings of the 18th International Conference on Mobility, Sensing and Networking (MSN'22), pp. 571-575, IEEE, doi: 10.1109/MSN57253.2022.00095.

  M. Lotfian Delouee, B. Koldehofe, V. Degeler, *"AQuA-CEP: Adaptive Quality-Aware Complex Event Processing in the Internet of Things"*, In Proceedings of the 17th ACM International Conference on Distributed and Event-Based Systems (DEBS'23), pp. 13–24, ACM press, doi: 10.1145/3583678.3596884.

  MLD conceptualized this study, the methodology, and the experiments,

wrote the original drafts, and reviewed and edited them. BK and VD aided in the methodology and in reviewing and editing the papers.

- **Chapter 4**
  M. Lotfian Delouee, B. Koldehofe, V. Degeler –*"Towards Pattern-Level Privacy Protection in Distributed Complex Event Processing"*, In Proceedings of the 17th ACM International Conference on Distributed and Event-Based Systems (DEBS'23), pp. 185–186, ACM press, doi: 10.1145.

  M. Lotfian Delouee, V. Degeler, P. Amthor, B. Koldehofe – *"APP-CEP: Adaptive Pattern-level Privacy Protection in Complex Event Processing Systems"*, In Proceedings of the 10th International Conference on Information Systems Security and Privacy (ICISSP'24), 12 pages, SCITEPRESS, doi: 10.5220/0012358700003648.

  MLD conceptualized this study, the methodology, and the experiments, wrote the original drafts, and reviewed and edited them. VD, PA, and BK aided in the methodology and in reviewing and editing the papers.

- **Chapter 5**
  Majid Lotfian Delouee, Daria G. Pernes, Victoria Degeler, Boris Koldehofe. Poster: Towards Federated LLM-Powered CEP Rule Generation and Refinement. In Proceedings of the 18th ACM International Conference on Distributed and Event-Based Systems (DEBS'24), 2 pages, ACM press,doi: 10.1145/3629104.3672429.

  Majid Lotfian Delouee, Victoria Degeler, Boris Koldehofe. GPT-CEP: Adaptive and Autonomous Rule Learning with LLMs in Federated Complex Event Processing Systems. Submitted to the 19th ACM International Conference on Distributed and Event-Based Systems (DEBS'25), ACM press.

  MLD conceptualized this study, the methodology, and the experiments, wrote the original drafts, and reviewed and edited them. DGP performed interactions with LLMs. VD and BK aided in the methodology and in reviewing and editing the papers.

- **Chapter 6**
  A. Khalil*[1], M. Lotfian Delouee*, V. Degeler, T. Meuser, AF. Anta, B. Koldehofe. Driving Towards Efficiency: Adaptive Resource-aware Clustered Federated Learning in Vehicular Networks. In Proceedings of the 22nd Mediterranean Communication and Computer Networking Conference (MedComNet'24), 10 pages, IEEE, doi: 10.1109/62012.2024.10578208

  MLD and AK conceptualized this study together through frequent discussions. They wrote the abstract, introduction, and literature review collaboratively. MLD wrote the case study section along with the system model and AR-CFL design, including figures, algorithms, and discussions. AK performed the experiments and wrote the evaluation section. Subsequent discussions were conducted between MLD and AK to interpret and justify the results and write the conclusion and future work. VD, TM, AFA, and BK aided in the methodology and scientific revision.

[1]Co-first authors (*)

Chapter 2

Fundamentals and State-of-the-art

> From wisdom springs boundless might, Knowledge rekindles the heart's old light.
>
> Ferdowsi (Revised by GPT4o)

This chapter provides background knowledge and related literature on the concepts discussed in this thesis. In the first part, we focus on Event-based Systems, especially Complex Event Processing systems, elaborating on satisfying user demands from both *Quality* and *Privacy* points of view as the main requirements for IoT applications' users. Based on this foundation, we provide an overview of related approaches to Quality Monitoring specified by application customers and Privacy Protection specified by data owners, which focused on the *quality problem* and *privacy problem*, respectively, as earlier discussed in Chapter 1. The second part discusses related concepts on how AI-powered approaches support complex tasks in stream processing systems. We present an overview of related work on CEP Rule Generation, which focused on *autonomy problem*, as well as the applications of Federated Learning in a vehicular network scenario, concerning the *PUT problem*.

2.1 Event-based Systems

An *event* represents the occurrence of a situation of interest in a specific period of time. The literature divides events into two categories, *simple* and *complex*, yet both are typically defined using a schema where each event consists of attribute-value pairs and is tagged with a timestamp. This standardized structure enables the reuse of outgoing events from one event-based system as input for another processing unit. Despite this, simple and complex events

are distinguished by the following definitions [52–55]:

- Simple event: Low-level events that are happening in a short period of time and could be labeled by a word or tiny phrase. Such events are self-contained in significance, unlike messages that may represent only fragments of a protocol interaction.

- Complex event: High-level events represented by a set of not necessarily consecutive simple events with temporal or spatial correlations that might occur over a specific (longer) period.

To better understand the definitions, let us consider an example of *shoplifting* in the e-commerce IoT domain. In this example, an event occurrence could be any activity shop customers perform, such as entering the shop. Thus, the low-level activity events contribute to generating a higher-level or complex event such as *shoplifting event*, derived by detecting a series (i.e., *sequence*) of activities related to a specific customer, e.g., picking a product followed by not paying this product in the counter. An event processing engine filters, aggregates, and combines events from event producers, such as cameras or motion detectors. Those complex events are then notified to the security monitoring applications, which act as event consumers.

An Event-Based System typically consists of three essential components: a *monitoring* component, a *communication* mechanism, and a *reactive* mechanism [56]. The monitoring component acts as the system's eye by observing the status of producers and consumers and detecting predefined situations in a stream of simple events. The communication component connects the selected producers to their corresponding consumers through *decoupling concept*. Consumers have no idea about the origin of their notifications and only express their situations of interest in the form of a continuous query and expect the system to notify them each time a match for the situation is detected. A crucial requirement from the consumer side is the quality of delivered results which we will discuss thoroughly in Section 2.1.2. On the other hand, producers are unaware of what happens to their generated simple events and who receives the final results. They are concerned about how their privacy is preserved. Such privacy-enhancing mechanisms will be elaborated separately in Section 2.1.3. Lastly, the reactive mechanism translates the customer-

defined queries into detection logic (i.e., rule) using a specification language. Such a translation is traditionally performed manually by domain experts, but several approaches offer autonomous translation mechanisms; both will be discussed in Section 2.2.1.

Having discussed the background information on Event-Based Systems, we will now detail one of the primary and important classes of Event-Based Systems, so-called Complex Event Processing, in the next subsection.

2.1.1 Complex Event Processing

Due to the enormous amount of data streams provided by sensor networks, stream processing (SP) has become a widespread challenge in various scenarios. Among these processing mechanisms, Complex Event Processing (CEP), which has attracted much attention, is a way of analyzing an incoming stream of low-level events and deriving correlations in real-time based on predefined event patterns. This paradigm was first introduced by David Luckham as a collection of tools and methods specified for contemporary distributed systems in order to examine and handle complex events [57]. Later, it became a trending mechanism for extracting insightful information from raw data in various application fields such as e-commerce, public healthcare, and the Internet of Things (IoT), and, more recently, in social networks such as Twitter.

Domain-specific terminologies should be elaborated on to help understanding CEP systems [11, 16, 58–60]. In this vein, we introduced these terms by utilizing the shoplifting example in Table 2.1.

Term	Definition	Nota.	Example
1. Query	The situation of interest specified by the system user in human language.	q	Shoplifting
2. Pattern	CEP-like presentation of a query	p	$Ent^{c_i} \rightarrow Pick^{c_i}_{p_j} \rightarrow \neg Pay^{c_i}_{p_j} \rightarrow Ex^{c_i}$
3. Window	A specific division of event stream that restricts the detection scope.	w	$within(1\ day)$
4. Dependency	The relationship between events	D	Relation between Ent^{c_i} and Ex^{c_i}
4.1. Temporal	Order relation of events	D_t	First observing Ent^{c_i} then Ex^{c_i}
4.2. Spatial	Location relation of events	D_t	events occur at the same store
4.3. Casual	Occurrence of the first event causes the second event's occurrence	D_c	Ent^{c_i} causes the occurrence of Ex^{c_i}
5. Operator	A specific processing function	ω	Extracting a person's events
5.1. Filter	Selecting events that satisfy a condition	ω_σ	Selecting events of a specific hour
5.2. Union	Merging streams by keeping the original events	$\omega_\cup$	Merging streams of two cameras
5.3. Join	Merging streams and composing new events	$\omega_\bowtie$	Merging a camera's and a motion detector's events
5.4. Sequence	Tracking events happen in an order	$\omega_\rightarrow$	$Pick^{c_i}_{p_j} \rightarrow Pay^{c_i}_{p_j}$
5.5. Conjunction	Tracking events occur in parallel	$\omega_\parallel$	$Pick^{c_i}_{p_j} \parallel Pick^{c_i}_{p_k}$
5.6. Negation	Tracking the absence of an event	$\omega_\neg$	$\neg Pay^{c_i}_{p_j}$
5.7. Aggregation	Performing aggregation functions over streams like max, min, etc.	ω_Σ	Count (Pay^{c_i})

Term	Definition	Nota.	Example
6. Operator Graph	A graph that shows the order of operators to generate complex events out of incoming input streams	G	$\omega_{\bowtie}(\text{streams}) \rightarrow \omega_{\sigma}$ to detect matches for the shoplifting pattern.
7. Operator Placement	The assignment of an operator to computing resources between time t_s and t_e.	P_{t_s,t_e}	Analyzing camera images of customer entrance on a specific resource
8. Partial Matching	The incomplete detection of a pattern within a window.	pm	Only detecting $Ent^{c_i} \rightarrow Pick_{p_j}^{c_i}$ in a window.
9. Total Order	an assumption that all events in the stream are sorted in the order of time	$<$	All events in the shop are ordered correctly.
10. Partial Order	An assumption that certain events are ordered correctly based on timestamps.	$\leqslant$	There are inconsistencies between sensor timings in the shop.
11. Watermark	A threshold that shows how much time is acceptable for an event to arrive late.	wm	two minutes for camera data
12. Tuple	An event represents a tuple of key-value pairs.	$t_{k,v}$	$Ent^{c_i} = \{(t(e), 12458751),\ (Store\_ID, 11), ...\}$

Table 2.1: Complex Event Processing Terminologies and Notations.

After introducing the fundamentals of CEP systems, we present the main types of event detection systems that are currently used. Detecting complex events typically involves using logical rules created by experts in the relevant fields. However, this approach may not always be efficient enough to meet the requirements of the users or the system as a whole. This is especially true in scenarios where there is a need to generate new rules quickly or for personalization, such as in healthcare systems. Traditional rule-based systems may not always be able to deliver the desired results in such cases.

Probabilistic approaches have been proposed to address this need. These mechanisms detect complex events based on statistical information and the relationship between events in the stream. Recently, deep learning-based models have emerged to fulfill the requirement of generating new rules on the fly and modeling more complex correlations in streams. Based on this, we categorize the event detection systems into three major classes that we will discuss in the later subsections.

Rule-based Event Detection ($\mathcal{R}$)

These mechanisms for detecting events are simple and have implementations that closely follow their specifications. However, they require domain knowledge to define detection rules. For instance, heuristic rules must be applied to determine which event attributes should be considered. These mechanisms perform well in capturing event attributes, particularly in dealing with noisy data and uncertainty [61, 62], in Fuzzy complex event processing-based decision-making systems [63], or in intrusion detection and prevention systems [64]. However, there are two drawbacks to the rule definition by experts. Firstly, rules cannot cover all the system details and provide maximum coverage. Secondly, rules should be updated over time, known as rule tuning. This process is complicated because of the dynamic nature of corresponding domains. Moreover, such mechanisms have the rule distribution and placement problem [65]. Thus, further support beyond domain experts is necessary for rule management to move from reactive towards proactive event processing, which might add complexity to the event detection procedure [30, 66].

Probabilistic Event Detection ($\mathcal{P}$)

Probabilistic approaches aim to learn the probability distribution of various parameters related to event datasets, such as the occurrence of events. These methods are suitable for applications where events have dynamic space. Although they estimate different aspects of events, their estimations may not be easily interpretable, and the results could depend on the degree of similarity between the sampling dataset and what actually happened in the real environment [61].

As one of the main classes of probabilistic-based event detection mechanisms, Bayesian Network Models (BN) are probabilistic graphical models that use Bayesian networks to calculate probabilities. They represent event dependencies as edges in the graphs. However, their weakness is that they should be attribute-independent, which means that all considered features must be independent of each other [67]. As an example, BN models have been employed to extract events dynamically in a real-time water management system, demonstrating that it is feasible to capture spatial and temporal dependencies of sensor networks [68].

On the other hand, Hidden Markov Models (HMMs) are very good at describing temporal dependencies between data and identifying significant data sequences as simple events from the environment. HMMs use random probability distribution models, in which data is used to model the transitions between states. They can describe the relationship between the current data and the previous data. However, HMM-based mechanisms require high computing power [67]. As an application of HMMs in event detection, research has been conducted on recognizing special human activities. The detection process can be done with a significantly low error rate in action recognition [69]. Detecting anomalous behaviors of a safety-critical system and consequently predicting a system failure before it happens is also studied by exploiting HMMs in CEP systems [70].

Besides these models, a probabilistic model called the Gaussian Mixture Model (GMM) can be used to identify the occurrence of an observed event. One approach to using this model in event detection methods is to associate a GMM with each event, and the final classification model would be designed based on the GMM with the highest probability. However, the most sig-

nificant weakness of GMM is the lack of guaranteeing the global minimum convergence. A walking event detection mechanism is presented as an example of utilizing GMM with IoT data. This mechanism uses a cell phone-based accelerometer sensor placed close to the body. The method recognizes temporal sequences in motion data and outperforms similar approaches [71].

Deep Learning-based Event Detection ($\mathcal{DL}$)

Deep Learning (DL) models are a popular type of Machine Learning mechanism that has gained significant interest in recent years. They are capable of detecting potential new and unknown events and adding them to the knowledge base, which expands the event detection functionality [72,73]. DL models extract abstract features automatically through hidden layers, reducing the burden of feature selection compared to traditional machine learning methods [61, 67]. Hidden features like temporal and spatial dependencies can be better understood and detected by employing DL models. DL models also perform better in large-scale data streams compared to simple learning models, which cannot handle a deluge of data [74]. However, the main drawback of DL methods is their tendency to overfit due to limited training data. Although data augmentation techniques have been introduced to alleviate this issue [61], more precise mechanisms are required to address this challenge fundamentally. With this information in mind that deep learning typically outputs a probabilistic result for a primitive event, the reasoning layer in the event processing framework aims at propagating the associated probability of a detected primitive event to the probability of a composed complex event. In an event processing system, DL methods can be used in the perception layer to capture simple events [75]. These models abstract raw sensor data and translate them into the form of event symbols. These symbols are part of a wider set called a label set, which users or the autonomous system can use to define complex event patterns. Applications of DL-based event processing range from abnormal event detection to image recognition in (industrial) IoT environments [76, 77].

Convolutional Neural Networks (CNNs) are currently the most used deep learning approaches, which are mainly suitable for processing spatial information (e.g., image features extractors) [78]. They are known as highly accu-

rate methods for identifying objects. In addition, they belong to supervised learning techniques; thereby, their layers are required to be fed by labeled datasets [79]. Example of employing CNN in event capturing expanded to human activity recognition [80, 81], image processing [77], anomaly detection [82], to autonomous driving [83].

Recurrent Neural Network (RNN) is a kind of neural network with a feedback loop that enables these models to process sequence data and recognize the time dependencies between them in several time periods [84]. In fact, these models have hidden layers by which they can memorize the previous incoming data. In each time step, they have the current incoming data as well as the previous incoming data to predict the output of the current layer. The main drawback of these models is that they are unsuitable for capturing long-term dependencies. Besides, they suffer from the vanishing gradient problem, where gradients become very small during backpropagation, preventing effective updates of earlier network layers [78]. Long Short Term Memory (LSTM) models are the state-of-the-art approaches that resolve the vanishing gradient problem by utilizing memory modules instead of ordinarily hidden nodes [84]. RNN can be employed in event capturing for activity recognition by exploiting the generated data from an inertial measurement unit (IMU) [85], detecting complex events based on subsymbolic data [86], security monitoring [87], and in healthcare applications [88].

Restricted Boltzmann Machines (RBMs) are the original deep learning model that generates data patterns by reconstructing incoming data with one visible and one hidden layer. These models are probabilistic undirected graphs in nature that can learn a probability distribution over its set of inputs [84]. Deep Belief Network (DBN) is a deep neural network model that utilizes multiple RBMs in order to extract features. In addition, DBN can handle both supervised and unsupervised data and extract features from input data recorded by different sensors, and then, detect temporal actions in cooperation with HMM [67]. A combination of DBN and CNN can be applied to recognize emergencies in the CEP system [89], and in the area of sound event recognition [90].

Deep Autoencoder (DA) is also a type of feedforward neural network in which the model's output is the same as the input and can extract unsuper-

vised features from the incoming data [84]. They have several layers; half perform encoding, while the remaining perform decoding. These models can be used to remove noise from the input data by learning useful data components. A smartphone-based human activity recognition showcases the application of DA in event-based systems [91] as well as a video-based abnormal event detection in [92].

In Table 2.2, we summarize and compare the state-of-the-art in terms of event capturing category and utilized algorithms, followed by a brief description of their pros and cons.

Year	Name, Ref.	Cat.	Algorithm	Key Contributions	Limitations
2009	Turchin et al. [66]	$\mathcal{R}$	Discrete Kalman Filters	Automatic Rule Generation, Rule Parameters Tuning	Extra Rule Tuning Delay
2011	Schilling et al. [65]	$\mathcal{R}$	Constraints Satisfaction Optimization	Minimizing Network Usage, Dynamic Rule Placement	Communication Overhead in Low Constraint Scenarios
2013	Ottenwalder et al. [93]	$\mathcal{R}$	PLAN-based Migration	Early Migration Planning, Fewer Migration Bandwidth Requirement	Unable to Migrate Overlapping Input Streams
2015	Baldoni et al. [70]	$\mathcal{P}$	HMM	Accurate Predictions of Failures	Fewer Failure Detection Accuracy Occurring Inside the Enclosure
2016	Mao et al. [68]	$\mathcal{P}$	BNM, HMM	Dependencies Modelling, Event Propagation Tracking, Scalability	Fixed Training Data Input, Dynamic Input Issue
2016	San et al. [69]	$\mathcal{P}$	HMM	Minimizing Recognition Error, Fewer Training Data	No Statistical Improvement
2016	Muaaz et al. [71]	$\mathcal{P}$	GMM	Novel Walking-style Authentication	Significant Computation Overhead, Poor Outcome
2017	Rivera et al. [85]	$\mathcal{DL}$	Long Short Term Memory	Fewer Pre-processing, Fewer Training Data	Low Recognition Accuracy, Fixed Overlapping Windows
2017	Almaslukh et al. [91]	$\mathcal{DL}$	Stacked Autoencoder	High Recognition Accuracy, Low Recognition Latency	Lack of Optimized Model Parameters

Year	Name, Ref.	Cat.	Algorithm	Key Contributions	Limitations
2017	Mousheimish et al. [30]	$\mathcal{DL}$	Early Classification on Time Series (ECTS)	Automatic Predictive Rule Generation, User Friendly/Independent	Lack of Supporting Big Data Flow and Learning Over Remote Clusters
2018	Ignatov et al. [80]	$\mathcal{DL}$	CNN	User/Platform Independent, Fewer Pre-processing Overhead, Fewer Runtime	User Inconvenient
2018	Wang et al. [90]	$\mathcal{DL}$	DBN	High Recognition Rate (99%), Noisy Conditions Robustness	Realisticity Issue Due to Using the Prior Knowledge of SNR
2020	Yin et al. [89]	$\mathcal{DL}$	CNN, DBN	Improving DBN by Incorporating CNN with Reduced Hidden Layers	Poor Interpretability for DDI Features
2020	Xing et al. [75]	$\mathcal{DL}$	Symbolic Human Knowledge + NN	Detect Events with Much Fewer and Sparse Annotations	Lack of Multi-modality
2021	Yadav et al. [79]	$\mathcal{P}$, $\mathcal{DL}$	YOLO, Single Regression	Near Real-time Detection, Multi-Stream Input	No Multi-modality, Limited Event Source Support
2021	Xiao et al. [63]	$\mathcal{R}$, $\mathcal{P}$	Fuzzy Set Theory	Cost-aware, Fault-tolerant and Reliable	Limited Modeling of Uncertainty Based on Triangular Fuzzy Number

Year	Name, Ref.	Cat.	Algorithm	Key Contributions	Limitations
2021	Ren et al. [77]	$\mathcal{DL}$, $\mathcal{R}$	CNN + Autoencoder NN	Improving NN Models by Learning from Data on Constrained IIoT Devices, High Inference Speed	Lack of Multi-modality
2021	Li et al. [92]	$\mathcal{DL}$	CNN + LSTM	Detecting Anomalous Behaviors, Noise Robustness	Not Able to Detect Real-time Surveillance Anomalies
2022	Rahman et al. [81]	$\mathcal{DL}$	CNN + LSTM	Satisfactory Activity Recognition, User Independent	Not Applicable to Other Data Types (Video)
2022	Lima et al. [64]	$\mathcal{R}$	Signature-based Intrusion Det.	Detecting Attacks on MQTT and CoAP Protocols	Restricted Evaluation to TCP
2022	He et al. [88]	$\mathcal{DL}$	Bidirectional Long Short Term Memory	Obtaining Event's Abundant Contextual Information, Multilevel Attention	No Multi-modality, Separate Approach for Simple and Complex Events
2022	Vosta et al. [87]	$\mathcal{DL}$	CNN + RNN	More Accurate Classification Compared to The State-of-The-Art	Not Able to Classify All Types of Anomalies in UCF-Crime Dataset
2023	Vilamala et al. [86]	$\mathcal{R}$, $\mathcal{DL}$	CNN	Multi-modality, Flexibility and Modularity in Rule Generation, Adversarial Robustness	User Dependent in Rule Definition

Table 2.2: Overview of related work on Complex Event Processing Systems.

2.1.2 Quality Monitoring

IoT applications rely on highly dynamic events, making it necessary for the characteristics of data sources to be continuously updated. This ensures that the most suitable data is selected, providing accurate and error-free information to the corresponding applications. However, the service quality of IoT applications can be susceptible to changes in the environment, such as battery levels and environmental conditions like air temperature, impacting the accuracy of sensor readings [94, 95]. Additionally, analyzing data streams from IoT environments raises concerns about the trustworthiness of sources and the freshness of extracted information [96]. Given this discussion, Definition 1 summarizes a quality monitoring mechanism employed by CEP systems as used in this thesis.

Definition 1. *Quality-aware CEP.*
A CEP system is defined to be quality-aware if it assesses the quality of the complex events it generates by computing specific quality properties and annotating each event with relevant quality dimensions, gradually improving event quality while retaining the system's core functionality [28, 97].

A CEP system typically measures quality in various steps in the event detection process. This can be done as soon as data is captured from the environment, when data is translated into simple events, during the detection of matches for patterns over simple event streams, or eventually by examining the generated complex events in regard to the user preferences. Considering this, we have categorized the literature on evaluating CEP quality into four categories based on the common CEP architecture layers: Quality of Data (QoD), Quality of Event (QoEv), Quality of Service (QoS), and Quality of Experience (QoE). In the upcoming sections, we will provide detailed explanations and examples of studies conducted in each category.

Quality of Data (QoD)

These studies mainly focus on designing algorithms to evaluate and increase the level of data quality before feeding the CEP system. During data collec-

Table 2.3: A Categorization of Data Quality Errors.

Error Name	Causes	Detection Algorithms
Incomplete Data (missing values)	Malfunctioned Sensor, Limited Battery Life, Physical Interference	Ontology/Knowledge-based
Inaccurate Data (uncertainty)	Low Sensor Accuracy, Environment-related Noise	Ontology/Knowledge-based, Artificial Neural Networks, Ensemble Classifiers
Data Inconsistency	Unstable Communication Links, Network Congestion	Deep Learning Methods, Ensemble Classifiers
Data Untimeliness	Unstable Communication Links, Network Congestion	Deep Neural Networks, Classification Methods
Data Outliers (anomalies, fault, bias, drift)	Malfunctioned Sensors	Principal Component Analysis (PCA), Artificial Neural Networks, Ensemble Classifiers

tion, due to cyber-physical attacks happening in the wireless medium, data might contain anomalies (e.g., missing data, redundant data, data failure, data outliers, touched data, etc.). Data pre-processing, as a way of enhancing data quality, validates data before being analyzed. In other words, useless data (e.g., records with missing fields or outliers) has to be removed from the data stream to prevent wasting time and resources. To implement pre-processing mechanisms, *stateless operators* can be employed to remove useless data (e.g., string value in an integer field or negative numbers for energy consumption). On the other hand, *stateful operators* (i.e., which keeps the state of input events before producing the complex events) can be used to keep track of input events and identify duplicate data records using *tumbling win-*

dows [98]. To better demonstrate the main causes for low-quality data, we categorize the main data errors in Table 2.3, as well as the algorithms that have been utilized in the literature to detect and quantify such errors [99].

Ontology/Knowledge-based. Ontology-based techniques provide a formal, structured representation of the domain knowledge surrounding sensors, their deployment environment, and the phenomena they measure. This structured knowledge includes defining concepts, relationships between them, and rules that govern their behavior. Knowledge bases store this ontological information along with factual data about specific sensors and their readings. These techniques work together to detect and correct potential data quality issues such as missing values, noise, or sensor malfunctions. By leveraging the relationships defined in the ontology, knowledge-based systems can reason about sensor readings, identify patterns that signal errors, and suggest appropriate corrections or flag the data as potentially unreliable [100–102].

Artificial Neural Networks (ANN). This method is a way of recognizing patterns in complex structures (i.e., pattern recognition). An ANN consists of many neurons, where each neuron receives multiple input values, processes them through an activation function, and produces an output. The network as a whole can have a single or multiple outputs, depending on the problem being addressed. In the feeding part, each input has a weight that determines the importance of that input value. The training process adjusts these weight coefficients to maximize the accuracy of output value, leading to modeling the functionality of a system. Using ANN, the sensor behavior is modeled considering its readings and an anomaly is detected by comparing the current sensed value and expected value produced by the ANN model [103–106].

Ensemble Classifiers. This category of fault detection mechanisms combines the results of multiple machine learning classifiers to better predict sensor data using methods such as algebraic combiners or heuristic rules. They finally detect anomalous data by setting the sensor reading and estimated data side by side. An ensemble classifier belongs to the supervised learning method, and one of its design complexities is the need to choose suitable base classifiers. Such a selection procedure might be complicated, especially depending on the application type; e.g., some classifiers might perform better

for anomaly detection while others perform better for sensor drifts. Moreover, the chosen classifiers must share similar characteristics, since some might require feature extraction and others might need fault-free training. Lastly, big datasets are required to train all selected classifiers [107–111].

Deep Learning (DL) Methods. DL methods offer a powerful approach to addressing data quality issues in sensor readings. These methods involve training complex neural networks on large datasets of sensor data, both with and without errors. By capturing complex dependencies within the data, deep learning models often outperform traditional rule-based systems in tasks such as anomaly detection, outlier identification, and sensor error detection. They can also be used for imputation, filling in missing values based on learned patterns from existing data. Additionally, deep learning can predict future sensor readings and compare these predictions with real-time readings to detect potential malfunctions. These methods offer the potential for greater adaptability and accuracy in data quality management compared to traditional approaches [112–116].

Principal Component Analysis (PCA). The most common method for detecting anomalies and faults in data is PCA, in which by putting the readings from m sensors captured in n time slots, a *Sensing Value Matrix*, namely X, is formed. The next step is to standardize each column (i.e., data related to a specific sensor) using a so-called *whitening* process (i.e., a well-known process in statistics). In this process, the following formula is applied to each value in a column.

$$(x_{i,j} - \mu_j)/\sigma_j$$

in which $x_{i,j}$ is the value of the sensor (column) j with the timestamp i, μ_j is the mean value of the sensor (column) j, and σ_j is the sensor's (column) standard deviation. Then, by following some statistical computations (e.g., calculating *Covariance Matrix*), a *residual subspace* will be extracted. When its magnitude is increased, exceeding a threshold, fault can detected in the sensing data [117–119].

Quality of Event (QoEv)

In an event-based system, events (both simple and complex) can be detected with various quality levels, ranging from detection uncertainty to event de-

tectability. The former refers to the lack of complete knowledge or confidence in the outcome of a decision or the state of a system and encompasses factors such as incomplete information, randomness, variability, and ambiguity in data or models. The latter measures how effectively the system can recognize predefined patterns or conditions in streaming data. High detectability means that the system can reliably identify relevant events or patterns with minimal *False Positives (FP)* (i.e., triggering a match for a complex event that does not happen in the environment) or *False Negatives (FN)* (i.e., not triggering a match for a complex event while it actually happened in the real world) [120].

Also, Specifying the metrics by which such quality specification could be determined is of great importance. In fact, to evaluate the aggregated quality of an event, quality metrics such as completeness, accuracy, FN, and FP should be taken into consideration [121]. In such methods, the overall quality of an event, which also *Quality of Results (QoR)*, is calculated by utilizing a utility function composed of the optimal value of these quality metrics, the constraints, and their weights, which are typically specified by the user or system designer, e.g., in location tracking domain [28, 122]. By categorizing the query properties into functional and non-functional, the user should be able to specify which type of events he is interested in and also can specify its constraints (e.g., a threshold value for the accuracy or requesting timely notifications). Moreover, the quality of the event can be represented by the utilization of an event ontology. By describing the different aspects of events, the system will be able to model the impacts of data streams on each other and, consequently, can check the actual occurrence of observed events [96].

Quality of Service (QoS)

The research works in this class of CEP systems mostly concentrate on ensuring that detected events are delivered reliably, accurately, and within acceptable timeframes to the applications or systems that need them. This involves various metrics such as latency (how quickly an event is processed and delivered), throughput (the rate at which events can be handled), and availability (ensuring the CEP system is operational). QoS requirements depend on the specific use case; for example, a stock trading application might prioritize low

latency, while a fraud detection system might emphasize accuracy. Maintaining high network-level QoS in CEP often requires careful configuration, load balancing, resource allocation, and the ability to scale the system in response to changing data flow volumes or event patterns [94].

An effective way to enhance the quality of service in event processing systems is to use *parallelization*, *replication*, and *elasticity* mechanisms [123]. To implement this, you can ask yourself two questions: How can you parallelize operator graph processing? And, how can you adapt parallelization methods to accommodate the dynamic data rate of the data source? For the former, it may be difficult to parallelize stateful operators as they need to be divided into different cores. For the latter, elasticity mechanisms should be deployed to manage the variations in the data source rate and to prevent the wastage of resources such as processing costs (e.g., using the pay-as-you-go model in the cloud).

CEP mechanisms rely on certain QoS metrics such as latency and throughput, which are referred to as Primary QoS metrics. In addition to these, Secondary QoS metrics such as load balancing, fault tolerance, and node utilization should also be taken into account [123]. In healthcare scenarios, adaptive QoS can be implemented by adopting key performance indicators like throughput, delay, transmission power, and duty cycle [124]. To meet user requirements, there must be a trade-off between QoR and QoS, which can be achieved by finding a degree of correlation between them and trading one for the other. For instance, a CEP system can switch to a data source with a lower end-to-end latency while maintaining or reducing the detection accuracy to an acceptable level [28, 122]. Besides those, numerous studies have been conducted to enhance QoS in other aspects of CEP, such as operator placement [11, 95, 125] or the CEP programming model [126], which significantly impact the QoS-related performance of the system.

Quality of Experience (QoE)

To enhance users' satisfaction, the concept of Quality of Experience (QoE) has gained popularity in various IoT network fields [127, 128]. Traditionally, questionnaires were used to determine the level to which user requirements were met. However, current methods rely on observing how users interact with

the system using the latest advancements in networking and communication technologies to obtain observable data [129].

One solution to address QoE requirements is to utilize modern networking platforms like SDN [130]. These platforms provide advantages such as central management, dynamic adaptation, programmability, and cost-effectiveness with network elements by separating the control plane and data plane, leading to guaranteeing QoE [131].

Determining the factors that contribute to user satisfaction can vary greatly depending on the domain, and it can be challenging to represent QoE metrics within a generalized framework. Additionally, to ensure user satisfaction, the system must be capable of accurately capturing the environment's dynamics and considering the user's changing needs. To achieve this, a step-by-step instruction set is needed to obtain sufficient information from users regarding the results and update the event processing procedures accordingly. One effective approach to properly respond to the dynamics and meet user requirements to achieve an acceptable level of QoE is to adopt the MAPE-K loop [132]. This feedback loop, which stands for Monitor, Analyze, Plan, Execute, is the most influential reference control model for autonomic and self-adaptive systems. By utilizing the MAPE-K loop based on human-centric requirements, the system will be able to improve its performance by meeting user preferences.

2.1.3 Privacy Protection

Today's world generates massive amounts of data that flow continuously. Think about social media updates, financial transactions, or sensor readings. Stream processing and Complex Event Processing (CEP) systems are designed to analyze these never-ending data streams in real-time. This lets us detect patterns, gain insights, and make immediate decisions. However, all this sensitive data raises serious privacy concerns. It's crucial to protect personal information from unauthorized access, misuse, or even inferences that could reveal things people want to keep private [36, 133, 134]. Privacy protection in stream processing and CEP systems is a complex challenge. Traditional security measures designed for stored data don't always work well with data that is constantly on the move [135, 136]. Researchers and developers are working

on innovative techniques like encryption, access control, and methods to protect data and deliberately reduce its precision. We also have to think about differential privacy, where the goal is to let us learn useful patterns from the data stream without revealing the specifics of individuals within it. It is also important to consider the evolving context of differential privacy, in which statistical and inference attacks have become major concerns [137]. Balancing privacy with the need for real-time insights is an ongoing challenge that engineers and privacy experts are constantly working to address [29].

Given this discussion, Definition 2 summarizes a privacy-enhanced CEP mechanism as used in this thesis.

Definition 2. *Privacy-enhanced CEP.*
A CEP system complemented by a set of tools and strategies that enable it to analyze data in real-time while safeguarding the privacy of sensitive information by restricting access to data or modifying original streams in order to prevent the identification of individuals or the misuse of their personal details [29].

Here's a list of the most important techniques used to protect privacy in Complex Event Processing (CEP) systems:

- **Data Minimization.** Limiting the collection of personal data to only what is strictly necessary for the CEP system's purpose. This reduces the overall privacy risk. It's like carefully packing a suitcase for a trip – you only include the items you truly need. By being selective with the data collected, we reduce the amount of sensitive information the CEP system handles, making it easier to secure and less likely to be misused. Data minimization is a proactive approach to privacy protection. It reduces the potential for data breaches and minimizes the negative consequences if something goes wrong. This principle demonstrates respect for users' privacy by showing a commitment to only collecting personal information when there's a clear and legitimate need. [138–140].

- **Data Anonymization.** This technique is like putting on a disguise for personal information. It involves removing or changing details that

directly identify individuals, such as names, addresses, and social security numbers. One way to do this is pseudonymization, where we swap real identifiers with fake ones (like codes) that still allow us to track patterns without revealing anyone's true identity. The beauty of data anonymization lies in its flexibility. Different levels of anonymization can be applied depending on how sensitive the data is and what the CEP system needs to do with it. For example, in some cases, simply removing names and addresses might be enough. In other situations, more advanced techniques like replacing data with realistic but fake information might be necessary. This lets the CEP system strike a balance between preserving privacy and maintaining the usefulness of the data for the analysis [141, 142].

- **Data Masking.** This acts like a clever filter for sensitive information within a CEP system. It changes the data to hide the real details but keeps the overall format the same, e.g., partially hiding a credit card number with asterisks ($****-1234$) or blurring faces in a video stream. The main advantage is that the structure is maintained, but the sensitive bits are obscured. This makes it less risky to analyze data in real-time. Data masking is particularly helpful in CEP systems where there is a need to see some pattern in the data without needing the exact, identifiable details, while still accounting for behavioral patterns that may emerge over time. For example, if a CEP system is analyzing financial transactions, masking could hide most of a credit card number while still revealing enough to identify the card type or bank. This lets the analysis happen while reducing the risk if the data is accidentally exposed. Another key advantage of data masking is that it often allows the CEP system to operate without major changes. Because the data retains its original format, existing analysis rules and queries may still work, unlike other techniques that might require more significant rewrites of the CEP system's logic. This makes it a practical option for protecting privacy while still getting value from the data stream [143, 144].

- **Encryption.** This method is like putting data in a secret lockbox. It uses complex mathematical algorithms to scramble the information, making it unreadable to anyone without the right key. This is essential

for CEP systems because it protects the data in two key ways: while it's traveling over networks (in transit) and when it's stored (at rest). This means even if someone intercepts the data, they won't be able to make sense of it. Two specialized types of encryption are particularly used in CEP systems; *Homomorphic Encryption*, which can be thought of as a safe equipped with internal machinery that processes data inside, without exposing its contents to the outside. It allows calculations directly on the encrypted data itself, without ever needing to unlock it. This is revolutionary for CEP because it means you can analyze sensitive information without exposing the raw, identifiable details [145, 146]. The second type is *Searchable Encryption*, which is an encryption method that enable the CEP system to search for keywords or patterns within encrypted data without having to decrypt it first, e.g., searching through a locked library without being able to open the books [147, 148].

- **Access Control.** AC is like having a security guard for data in a CEP system. It involves setting up strict rules about who can see or interact with the information and what specific actions they are allowed to take, creating different levels of clearance. In such mechanisms, some users may only have permission to view data, while others are authorized to modify it. This protects the data from unauthorized access or accidental changes. Two common types of access control utilized in CEP systems are *Role-Based Access Control* (RBAC) and *Attribute-Based Access Control* (ABAC). The former (i.e., RBAC) focuses on assigning permissions based on a person's job role within an organization. For example, an analyst might have read-only access to customer data, while a database administrator has the ability to modify it. This ensures people only have access to the data they need for their specific job functions. The latter (i.e., ABAC) offers even finer control by considering a wider range of factors (or attributes) beyond just a person's role. This could include things like the user's location, the time of the day, or the specific device they are using. ABAC provides a highly flexible way to tailor permissions, making it adaptable to even complex privacy and security needs within a CEP system [15, 149, 150].

- **Differential Privacy.** DP is a form of statistical coverage that works by adding a small amount of carefully designed "noise" to the results of analyses performed on a dataset. This noise obscures the contribution of any single individual, making it impossible to tell with certainty if someone's data was even included in the analysis, while also protecting against attempts to infer whether a particular match or correlation exists. The magic of differential privacy lies in its balance. While it protects individual privacy, it still allows analysts to extract meaningful statistical trends and patterns from the data within the CEP system. For example, a CEP system could determine overall spending habits in a region without revealing any specific person's purchase history. Differential privacy is a powerful tool for CEP systems that deal with sensitive data. It allows for responsible use of data analysis while providing a strong mathematical guarantee of privacy protection. This makes it particularly attractive when privacy regulations are strict, or the data involved is highly confidential [151–153].

- **Privacy-Preserving Data Mining.** PPDM enables CEP systems to find hidden insights in data while respecting privacy boundaries using a special set of tools carefully designed for data mining. It involves developing algorithms that can uncover patterns within a CEP system's data streams without revealing sensitive details about individual people. Instead of focusing on exact information tied to specific individuals, PPDM aims to identify broader trends and relationships. For instance, consider a medical researcher analyzing patient data. PPDM techniques could help him discover links between symptoms and diseases without needing to know the identity of each patient. This allows for valuable medical discoveries while safeguarding individuals' privacy. PPDM is constantly evolving, with researchers developing innovative methods to analyze data in a privacy-conscious way. These techniques often involve transforming or generalizing the data to reduce its specificity while still preserving enough information to be useful for the CEP system's purpose. PPDM offers a powerful way to balance the benefits of real-time data analysis with the crucial need to protect individual privacy (e.g., Randomization techniques [146, 154]).

- **Pattern-level Privacy.** While it's important to protect specific pieces of data like names or addresses, privacy threats in CEP systems can be even trickier. Sometimes, the combination of seemingly harmless pieces of information can reveal something sensitive. Pattern-level privacy is all about recognizing and safeguarding against these sneaky combinations, so-called *patterns*. One interesting example scenario is to consider a detective story. Individually, knowing someone bought groceries or visited a park might not be concerning. But if a CEP system detects a pattern of these events together, it could potentially reveal a sensitive location or activity. Pattern-level privacy focuses on identifying these risky combinations in advance and finding ways to protect them. Pattern-level privacy is especially crucial in CEP systems because they are built to spot patterns in real-time data. By understanding which patterns could compromise privacy, the CEP system can be designed to obfuscate those combinations or alert users before sensitive information is accidentally revealed. This proactive approach lets us get valuable insights from the data while still respecting people's privacy expectations [15, 16, 29, 36].

It is important to note that the best mix of techniques will depend on the specific CEP system, the sensitivity of the data, and the privacy regulations that apply in a given situation.

2.2 Employing AI for Stream Processing

Employing AI within stream processing opens up an exciting world of possibilities. Coupling a CEP system with AI not only allows it to analyze massive data streams in real-time but also to learn and adapt its behavior on the fly. This powerful combination could unlock unprecedented levels of efficiency and insight for businesses and organizations working with rapidly changing data [155]. A key area where AI shines is in shifting from reactive to proactive rule generation [156, 157]. Traditionally, CEP systems rely on manually defined rules by domain experts to detect patterns and trigger actions. With AI, these systems can analyze massive historical data and real-time patterns to discover complex relationships that might be invisible to humans. This means the CEP system can generate rules for situations that haven't even

been encountered before, transforming it into a proactive analysis of the data stream [158].

However, as AI is integrated into stream processing, privacy concerns become paramount. Federated Learning (FL) offers a promising solution. Instead of sending all raw data to a central location for AI training, federated learning distributes the learning process across multiple devices or nodes. Each node trains a local model on its own data and only shares model updates, not the raw data itself, with the central server. This collaborative approach preserves privacy while allowing the AI to benefit from the collective knowledge across the network, offering a powerful trade-off between quality and privacy protection [159].

2.2.1 CEP Rule Generation

In CEP systems, a rule defines a pattern or combination of events that trigger a specific action or alert. Initially, experts generated these rules based on their understanding of specific domains. These rules dictated how the system responded to patterns in a dynamic environment. However, this approach relied heavily on human expertise, often leading to time-consuming development and potential oversights [160].

To address these limitations, probabilistic methods were introduced. By analyzing past event data, these techniques could automatically generate rules along with the likelihood of those events occurring. This data-driven approach not only streamlined rule creation but also offered insights into the inherent uncertainties of complex event patterns. Additionally, the integration of AI, particularly deep learning methods [18], has revolutionized CEP rule generation. By using AI algorithms, CEP can autonomously learn complex patterns from vast datasets, leading to more accurate, adaptable, and efficient rule creation that dynamically responds to evolving environments and requirements. Given this discussion, Definition 3 summarizes a fully autonomous CEP mechanism as used in this thesis.

Definition 3. *Rule-Autonomous CEP.*
A CEP system with intelligent event processing engines that not only dynamically adapt to real-time environmental changes but also proactively learn from historical data and current streams to anticipate future patterns. They autonomously generate both reactive rules to handle past situations and proactive rules to predict and prepare for potential future scenarios, optimizing decision-making and responses in complex environments [18, 19].

We classified existing literature into two distinct categories based on the temporal aspect of rule generation, namely *Reactive* and *Proactive*. The former focuses on extracting patterns and correlations from historical events. The latter leverages historical data, current trends, and predictive models to anticipate potential future events and generate rules accordingly, enabling preemptive actions and responses.

Reactive

In Complex Event Processing (CEP), reactive rule generation leverages the analysis of historical event data to create or enhance rules automatically. This dynamic approach ensures that the system adapts to changing patterns and trends, improving the accuracy and effectiveness of event detection and response. By learning from past events, CEP systems can continuously optimize their rule sets, making them more responsive and relevant to the current context [18]. Depending on the level of human or AI involvement in rule generation, *reactive* rule generation mechanisms can be classified into the following categories.

Manual Reactive Rule Definition. In such mechanisms, domain experts create rules to tell the system what to do when specific things happen. These rules act like instructions, guiding the system's actions in real time, making the system flexible enough to handle complex situations as they arise. However, there are trade-offs to this approach. While the human touch allows for a deep understanding of the domain and fine-tuned responses, creating these rules can take a lot of time and effort. The manual nature of the

process also opens the door to human error, e.g., a misstep in rule creation could lead to unexpected system behavior. Therefore, while manual reactive rule definition offers a powerful tool for real-time decision-making in CEP, its implementation requires careful consideration of its potential drawbacks [19].

Data-Driven Rule Generation. Here, the system autonomously creates or refines rules by analyzing historical event data. This approach leverages AI-based algorithms, such as machine/Deep learning and data mining techniques, to identify patterns, correlations, and anomalies within the data. By extracting insights from past events, the system can automatically generate rules that accurately capture the underlying relationships and dependencies between different event types. This eliminates the need for manual rule definition and allows the system to adapt dynamically to evolving patterns and trends [161]. These methods offer several advantages, including increased accuracy, faster rule creation, and reduced human effort. They enable the system to identify complex patterns that may not be apparent to human experts, leading to more effective event detection and response. Additionally, they allow for continuous improvement of the rule set as new data becomes available, ensuring that the system remains up-to-date and relevant [20].

Hybrid Rule Generation. This group combines the strengths of both manual and data-driven approaches. In this approach, human experts provide initial rules based on their domain knowledge and experience. These rules serve as a starting point for the system, which then uses data-driven techniques to refine and optimize them. By analyzing historical event data, the system can identify areas where the rules can be improved and suggest modifications to enhance their accuracy and effectiveness. This synergistic relationship leads to more comprehensive and effective rule sets that can better capture the complexities of real-world events [75, 162, 163].

Proactive

Proactive rule generation in CEP shifts the paradigm from reactive event detection to predictive anticipation. Instead of relying solely on historical patterns, this approach leverages predictive models and simulations to generate rules for potential future events or scenarios that have not yet occurred. This forward-looking perspective allows CEP systems to proactively prepare

for and respond to emerging situations, potentially mitigating risks and optimizing outcomes [164–166].

While proactive rule generation offers the promise of enhanced responsiveness, it also presents significant challenges. The inherent uncertainty of predicting future events and the potential for rule definition inaccuracies can lead to the generation of irrelevant or ineffective rules. Furthermore, the computational complexity of generating and managing a vast number of potential rules can strain system resources. Striking a balance between proactivity and accuracy remains a key consideration in developing effective proactive rule-generation mechanisms [167].

Manual Proactive Rule Definition. In this category, domain experts leverage their knowledge and expertise to anticipate potential future events and create rules for CEP systems accordingly. This approach allows for the proactive detection of events that have not yet occurred but are deemed likely or significant based on their understanding. Experts can also merge existing rules to create new, more comprehensive ones. However, such methods are not without drawbacks. Domain experts may not possess complete information about all aspects of the system and its interactions, leading to uncertainties in rule creation. Additionally, human errors and biases can introduce inaccuracies, making the generated rules less reliable and potentially leading to false positives or missed events [18].

Deep Learning-based. By leveraging the power of neural networks, it is feasible to analyze historical event data and extend the application of CEP systems to detect potential future patterns. This approach can uncover complex relationships and dependencies that may not be evident through traditional rule-based methods, enabling the system to define rules for events that have not yet occurred proactively. However, deep learning models require substantial training data and computational resources, and their inherent "black box" nature can make it difficult to understand the reasoning behind generated rules. Additionally, overfitting to historical data can lead to inaccurate predictions and ineffective rules for future scenarios. Despite these challenges, deep learning offers a promising avenue for enhancing the proactivity and adaptability of CEP systems [164, 165].

2.2.2 Federated Learning in CEP

Federated learning (FL) is a decentralized machine learning approach that enables multiple participants to collaboratively train a shared model without directly exchanging raw events. This approach addresses privacy concerns by allowing each system to train a local model on its own data and then share model updates with a central server for aggregation [168].

Stream processing systems often struggle with conflicting user requirements, such as privacy and quality. Preserving the privacy of sensitive data is paramount, especially in domains like healthcare or finance, where data breaches can have severe consequences. However, maintaining high data quality is also essential for accurate analysis and decision-making. These two requirements often conflict, as stringent privacy measures, such as anonymization or data perturbation, can inadvertently degrade data quality. Federated learning presents a potential solution to this challenge by enabling collaborative model training without directly sharing raw data. By keeping sensitive information decentralized and only sharing model updates, federated learning can help preserve privacy while still allowing for the aggregation of knowledge from multiple sources, thus maintaining and potentially improving data quality through collaborative learning [169].

In CEP, federated learning can enhance the accuracy and applicability of event detection by utilizing diverse event streams from multiple sources. This involves comparing current rules against the different streams to understand better the rules' detection performance (e.g., in terms of false positives and false negatives) and to identify specific scenarios (i.e., corner cases) in which a rule needs exceptions or updates [170]. However, for successful implementation, challenges must be overcome, including communication overhead, the Heterogeneity of event distributions, and potential biases in rule updates generated locally. Additionally, it is crucial to address the security and confidentiality of updates during transmission and aggregation to maintain trust and privacy in federated CEP systems [159].

Federated learning in stream processing systems can be broadly classified into two categories: *horizontal* federated learning, *vertical* federated learning, and *Federated Transfer Learning*; based on how training data are distributed over the sample and feature spaces, which we elaborate on further [171].

Horizontal Federated Learning (HFL). In HFL, also known as *sample-based federated learning*, different stream processing systems or devices hold similar feature sets but have distinct data samples. This approach is suitable when multiple systems are processing similar types of data, but each system only has access to a subset of the overall data. HFL enables these systems to collaboratively train a shared machine learning model without directly exchanging raw data. Each system trains a local model on its own data, and then the model updates (e.g., gradients or model parameters) are aggregated to create a global model. This collaborative training process improves the model's performance while preserving data privacy [172].

Vertical Federated Learning (VFL). This category, which is also called *feature-based federated learning*, is employed when different stream processing systems or devices share the same data samples but have different feature sets. This is often the case when data is collected from various sources or sensors that capture different aspects of the same phenomenon. In VFL, each system trains a local model on its own feature set, and then the models are combined using a secure aggregation mechanism to create a global model that encompasses the information from all features. This approach enables the system to leverage the complementary information from different feature sets while maintaining data privacy and security [173].

Federated Transfer Learning (FTL). This technique is a hybrid approach that combines the principles of federated learning and transfer learning. In FTL, knowledge acquired from pre-trained models on a source domain is transferred and adapted to multiple local models trained on different but related target domains. This enables the collaborative learning process to benefit from existing knowledge, thereby reducing the need for extensive training data and computational resources in each individual domain. FTL is particularly useful in scenarios where data is distributed across different sources and labeled data is scarce in the target domains. By leveraging knowledge from a source domain, FTL can enhance the performance and efficiency of machine learning models in federated settings [174].

2.3 Summary

This chapter establishes the foundation for understanding this thesis's contributions. It begins by introducing key concepts and reviewing existing approaches in related fields, such as Event-Based Systems, Adaptive Systems, Privacy-focused Systems, the Internet of Things, and Natural Language Processing. This comprehensive review reveals four significant research gaps that this thesis aims to address, including *quality*, *privacy*, *autonomy*, and *PUT*. First, there is no quality monitoring mechanism that can fulfill the changing QoS and QoR requirements of users and dynamic IoT environment, which indicates the issues concerning the *quality* problem. Chapter 3 introduces our novel solution to this challenge. Second, current Complex Event Processing (CEP) systems primarily focus on protecting privacy at the attribute level, overlooking the potential leakage of sensitive information through patterns. Chapter 4 presents our innovative approach to address this *privacy* concern. Third, CEP mechanisms often rely on manually defined rules, a process that is time-consuming, prone to errors, and inherently reactive. Recent advances in deep learning have demonstrated potential in automating CEP rule generation, but existing methods are often domain-specific, inefficient, and not suitable for real-time rule generation based on user queries. Chapter 5 tackles this *autonomy* challenge by proposing a fully automated CEP rule generation system. Finally, by examination of the *PUT* problem in one of the most common IoT scenarios (i.e., vehicular networks), we investigate the challenges coupled with balancing these competing requirements in stream processing systems Chapter 6 investigates the potential of clustering techniques to overcome these challenges.

Chapter 3

Quality Monitoring in CEP

A voice without wisdom, though gilded with art, Holds no weight nor beauty, nor touches the heart. Fine words without meaning, though polished they be, Are empty as echoes adrift on the sea.

Ferdowsi (Revised by GPT4o)

Abstract

This chapter introduces the topic of dynamic quality monitoring and adaptation of complex event processing in distributed systems considering the dynamics of sensor data (the quality problem). In the proposed solution, called AQuA-CEP, consumer-defined quality policies guide the selection and configuration of suitable data sources. Different forms of expressing quality policies and their impact on quality monitoring are examined. Additionally, the evaluation of quality-related adaptations and their influence on correlation efficiency are studied. Experimental results in IoT scenarios show that AQuA-CEP enhances the quality of results while meeting consumer requirements and optimizing sensor energy consumption.

Reacting to unstable situations is a fundamental requirement in the Internet of Things (IoT) scenarios like traffic monitoring, healthcare systems, and smart homes. Distributed Complex Event Processing (DCEP) is a widely employed paradigm to support efficient situation detection based on a variety of different sensors and a step-wise transformation from *primary events* to situations of interest for consumers in the form of *complex events*. The resulting

quality, usually expressed in the form of Quality of Service (QoS) and Quality of Results (QoR), highly depends on the origin of primary events, especially in IoT scenarios where the primary events are generated often based on distributed sensor readings from the environment. These sensing deployments are vulnerable to the immense dynamicity in the environment (e.g., availability of the sensors), and more than a single sensing deployment is often needed to meet quality requirements determined by the system or its consumers.

An established solution to react appropriately to environmental dynamics is to adapt the detection logic's placement to the available computing resources, which are part of the DCEP framework. Also, such an adaptation needs to deal with the limitations that the allocated resources might have during the query execution (cf. [11,14,126,175–177]). This idea provides essential means to maintain or improve the QoS-related measures, e.g., by reducing the imposed end-to-end delay or regulating the bandwidth consumption. On the other hand, sacrificing QoR to keep QoS at an acceptable level can already benefit the DCEP systems by combining these mechanisms with other runtime approaches such as load shedding techniques [59, 178]. Although influencing QoR will lead to a degradation in consumer requirements' satisfaction, these techniques find it crucial to impact QoR as little as possible, e.g., by dropping events from partial matches in the event streams.

The state-of-the-art approaches decouple the detection procedure and adaptation strategies from sensing deployment configuration and operate only based on information existing in the design time. Such an idea limits the system's capabilities to react correctly and timely to dynamics in the sensing layer, e.g., the quality of sensor readings or their battery level. Hence, the degradation in QoR can be propagated due to these limitations, while the DCEP system cannot actively influence and prevent the consequences of hardwired sensing configuration. Moreover, even if the sensing deployment adaptation is considered at runtime, adaptation strategies' outcome can affect the QoR [179]. For example, in the case of updating the data sources from a camera to a motion detector, the motion event's accuracy would be degraded. In this vein, due to attaining the most elevated quality grade in the adaptation decisions, the inputs are mandated to have an acceptable level of quality. In this regard, the input data can be assumed of insufficient quality

if inaccurate, precise, fresh, or truthful. Events are also evaluated as inadequate quality if they do not hold a certain level of confidence, are received out of order, are wrongly detected, or are not detected. Therefore, measuring to what extent the consumer's quality requirements are met should be taken into account when applying any adaptation strategy.

In this chapter, we analyze and present concepts on expanding flexibility and adaptivity by proposing quality-aware event processing to enhance QoR and QoS. In particular, AQuA-CEP is a mechanism that is designed based on the idea of dynamically exchanging sensing deployment concerning the demarcated requirements by the consumer that influence how sensor data is processed [122]. We enhance DCEP with the concept of so-called quality policies and corresponding quality monitoring mechanisms. Upon any change occurring in the environment or in the sensing deployment observed by our designed DCEP system, AQuA-CEP will autonomously adapt the current sensing deployment according to the available sensing infrastructure, if required. It can be performed by defining sensing configuration restrictions (e.g., cost constraints) considering quality requirements expressed by consumers. Consequently, a utility metric concerning the defined restrictions performs an efficient sensing deployment assignment.

For more details, AQuA-CEP provides the following contributions:

1. We provide a new representation of the quality demands of a query in DCEP systems by proposing a policy-driven specification of complex events to boost data processing performance and more promising utilization of IoT resources.

2. We devise how quality monitoring can be applied in DCEP by presenting concepts allowing the dynamic reconfiguration of appropriate data sources while fulfilling consumer's quality requirements.

3. We explore strategies for configuring quality monitoring agents that trigger adaptation strategies upon any quality policy violation and address the impacts each configuration might have on the DCEP system's performance in terms of QoS and QoR.

4. We evaluate the performance of our proposed mechanism with a real-world dataset alongside using synthetic data to show the ability of

AQuA-CEP to boost the performance of the DCEP system in adapting the sensing deployment while observing consumer quality requirements.

3.1 Quality-Aware Event Source Selection Problem

We first introduce an IoT scenario that demonstrates the applicability of AQuA-CEP in the context of IoT security surveillance. In this scenario, a state-of-the-art DCEP system fails to couple the DCEP middleware with the available event sources by concepts for (i) quality requirement description, (ii) quality monitoring, and (iii) sensing deployment adaptation. In other words, by a hardwired sensing configuration in an IoT environment, most DCEP systems fail to exploit multiple sensing deployments. Consequently, they do not benefit from event sources adaptation to make a trade-off between QoR and QoS and react appropriately to the scenario dynamics, e.g., exceeding the coverage of a sensor due to the user's mobility. It indicates that DCEP systems must rethink how switching between event sources at runtime can benefit the system and improve the generated results.

We consider a continuous query to detect and warn any person approaching a Dangerous Area (DA) in an industrial zone depicted in Figure 3.1. Any consumer, e.g., a factory's manager, can issue a query to mark a specific point within the area as a DA. The query will check different conditions—the current location of each moving person or object and the list of DAs—observed within the industrial zone. Suppose a person is located closer than a predefined threshold to the Central Point (CP) of a dangerous area (i.e., within the Alarm Region (AR)). In that case, an alarm is triggered and sent to his device to inform him about the security violation. The location events are generated by five distinct sensing deployments, namely RFID, BLE, WIFI, Camera, and LTE signals available in the IoT infrastructures.

By employing AQuA-CEP, a consumer can specify its quality requirements, e.g., high accuracy of location detection close to DA. The location tracking of each target is performed by actively integrating the selected event source (i.e., turning on the mobile data to receive the LTE signals). Therefore, the assignment of any sensing deployment influences the battery life of each target's cellphone. On the other hand, due to the quality levels of each sensing

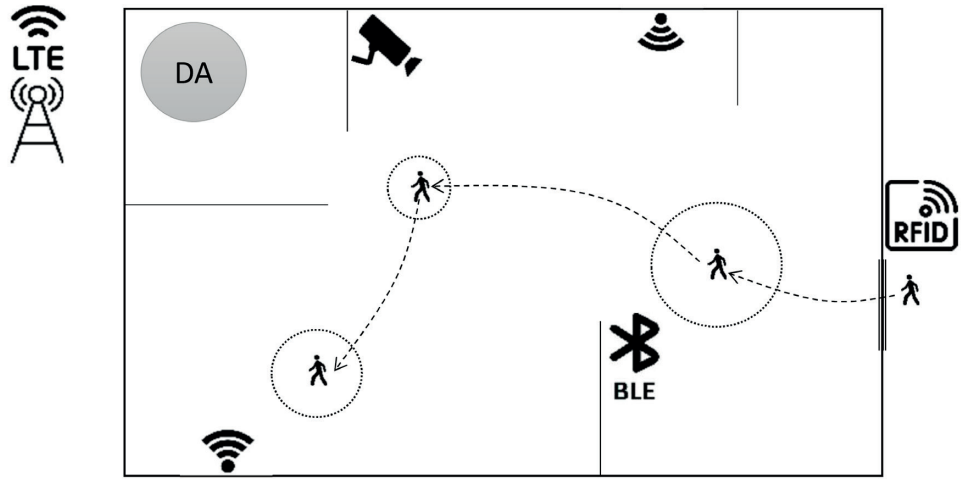


Figure 3.1: An example of event source adaptation at runtime in an IoT security monitoring scenario.

deployment, the location events can be generated with various quality levels that might not satisfy the consumer's expected quality of results. Hence, the DCEP system can monitor the quality and trade the QoR against QoS, e.g., reduce the accuracy of location detection in places far from the DA to achieve energy efficiency. Such a goal can be achieved by adapting the assigned sensing deployment, e.g., switching from highly accurate location sensors like LTE signals to WIFI signals at locations far from DA.

Consider the dynamic availability and multitude of event sources for the mentioned use case. Due to the mobility of each target or the environmental changes (e.g., a blocked camera), the quality of captured simple events changes over time; thereby, the quality of detected complex events (e.g., alarms) is different. Adapting event sources as a solution to fulfill the required QoR imposes costs that should be considered in adaptation strategies.

In this work, AQuA-CEP selects suitable data sources, i.e., a sensing deployment $\alpha(\mathcal{SD}) \subset \mathcal{SD}$, where $\boldsymbol{\alpha}$ determines which members of $\mathcal{SD}$ will be used. AQuA-CEP is required to meet the consumer constraints (e.g., high accuracy close to DA) or notify consumers when no proper sensing deployment is

feasible. Furthermore, each sensor source sd_i of a sensing deployment imposes a *System-Side Cost (SSC)* denoted by $C_{SSC}(sd_i)$ as well as the cost for performing *Quality Monitoring (QM)* for every query q_k denoted by $C_{QM}(sd_i, q_k)$. C_{SSC} includes those quality metrics that are more important for the system (i.e., energy consumption, reusability, resource utilization, etc.). C_{QM} is the cost imposed by the configuration model of the monitoring agent in terms of time (i.e., the delay related to the processing events in the operator and delay for performing the transition between sensing deployments) and computation (i.e., the required computational resources to conduct the monitoring process).

More formally, AQuA-CEP aims to find $\boldsymbol{\alpha}$ which minimizes the cost factors imposed by system-side costs and quality monitoring costs subject to the quality constraints of consumers, i.e.,

$$\begin{aligned} \min \quad & w_s \sum_{sd_i \in \mathcal{SD}} \alpha(sd_i) C_{SSC}(sd_i) \\ + \quad & w_q \sum_{sd_i \in \mathcal{SD}} \alpha(sd_i) \sum_{q_k \in \mathcal{Q}} C_{QM}(sd_i, q_k) \qquad (3.1) \\ s.t. \quad & \alpha(\mathcal{SD}) \text{ satisfies constraints in } \mathcal{G} \\ & \alpha(sd_i) = 1 \text{ iff } sd_i \text{ is selected.} \\ & \alpha(sd_i) \in \{0, 1\} \end{aligned}$$

Here, w_s indicates the weight related to system-side costs, and w_q is the weight associated with monitoring costs.

3.2 The AQuA-CEP System Design

We consider a DCEP system to consist of multiple *producers* (e.g., mobile phones, etc.) that generate streams of primary events from the received sensory data and announce them to the system as their advertisements. Correspondingly, *consumers* (e.g., users, applications, services, etc.) create situations of interest as subscriptions and submit them to the system as continuous queries where the set $\mathcal{Q} = \{q_1, \ldots, q_n\}$ denotes the set of currently deployed

queries. Moreover, a group of *brokers* (i.e., CEP engines) performs computational tasks (e.g., filter, join, etc.) by hosting a set of *operators* and forward the result to the next step that can be another operator or the consumers.

A query q_i explains the logic by which a complex event can be detected over primary event streams. It can be performed by applying standard CEP operators like pattern matching, aggregation, or windowing over primary events or their attributes. To do so, the imposed complex event detection logic should be applied to the specific brokers for execution. In the meantime, consumers are also allowed to specify their quality requirements as part of the query (e.g., the location accuracy of less than one meter). We denote the set of *consumer-side constraints* of all deployed queries in $\mathcal{Q}$ by $\mathcal{G} = \{g_1, g_2, ..., g_k\}$. Once the query registration is completed, AQuA-CEP is able to run a lookup service over the producers to discover those sensor(s) whose attributes conform to the query's quality requirements. A data source is considered an eligible candidate to feed the system if it can meet all related consumer-side constraints.

In AQuA-CEP, sensors are the origin of data that measure a specific phenomenon (e.g., temperature) in the environment. The sensory data sources (e.g., Bluetooth) that are used at a given time t form the set of active sensing deployments $\mathcal{SD} = \{sd_1, ..., sd_j\}$. Here, sd_i refers to a specific data source among all available options in the environment that can be participated reliably in the process of sensor assignment for a deployed query. The availability of data sources is dynamic, meaning the set $\mathcal{SD}$ might change over time. In the IoT environment, a data source can be mobile (e.g., sensors embedded in a smartphone) or stationary (e.g., a surveillance camera). We assume that the mobility status of data sources does not negatively or positively affect the quality of their readings.

In our scenario, IoT devices (e.g., smartphones carried by query targets) are interconnected to the system over a wireless sensor network and demanded to register their sensing deployment in the AQuA-CEP in advance. These devices represent the CEP producers who generate primary event streams from sensory data. Also, some of these devices are eligible to issue queries acting as CEP consumers. Moreover, CEP operators can be placed on IoT resources with sufficient computing capabilities, e.g., on the cloud or fog nodes.

For coordinating adaptation and selecting its correct triggers, AQuA-CEP needs to monitor the quality of produced events as well as the state of the sensing infrastructure. In this regard, we build on a sensing middleware (e.g., [180]) that offers the possibility to identify, configure, and access the physical sensors. The system samples the quality level for a subset of the produced events and evaluates potential alternative configurations.

In AQuA-CEP, the adaptation decisions need to serve multiple objectives, e.g., the cost for fulfilling the query's quality policies includes the expenses for utilizing the sensing infrastructure and completing reconfigurations. Besides, adaptation should guarantee a level of stability, which means how often the achieved quality of the detected event stays inside a predefined threshold region after applying the adaptation strategy. It would avoid oscillation and inessential switching costs. Also, with AQuA-CEP, we are looking to adapt the event processing to the environmental dynamics by switching the sensing deployment. For example, given a new set of sensors registered in the network by which some of the currently running queries would be answered. In this case, AQuA-CEP generates new query models considering these new sensors and checks for the costs imposed by transitioning from the current sensing deployments to the newly selected deployments.

In Figure 3.2, we depict the foundational components which are employed in AQuA-CEP. By adopting an SDN-like architectural approach, we use a *controller* to function as the coordinator module and enforce a quality-driven DCEP. This component is logically centralized but physically distributed and is in charge of exchanging control messages to synchronize the event detection procedure. To do so, the controller owns the principal role in matching the subscriptions to advertisements. A *query optimizer* component receives consumer queries, transforms the query description into a set of event types, and passes the list of required data sources to the *data source assignment engine*. A *look-up service* is triggered by the engine to explore the potential candidates for each event type in *data source database*, where the currently available data sources are previously registered themselves. The records in this database are dynamic and can be registered or canceled at runtime.

Moreover, the controller's functionality is enriched by employing *quality monitoring* agents in the DCEP layer. Upon any predefined situation (e.g.,

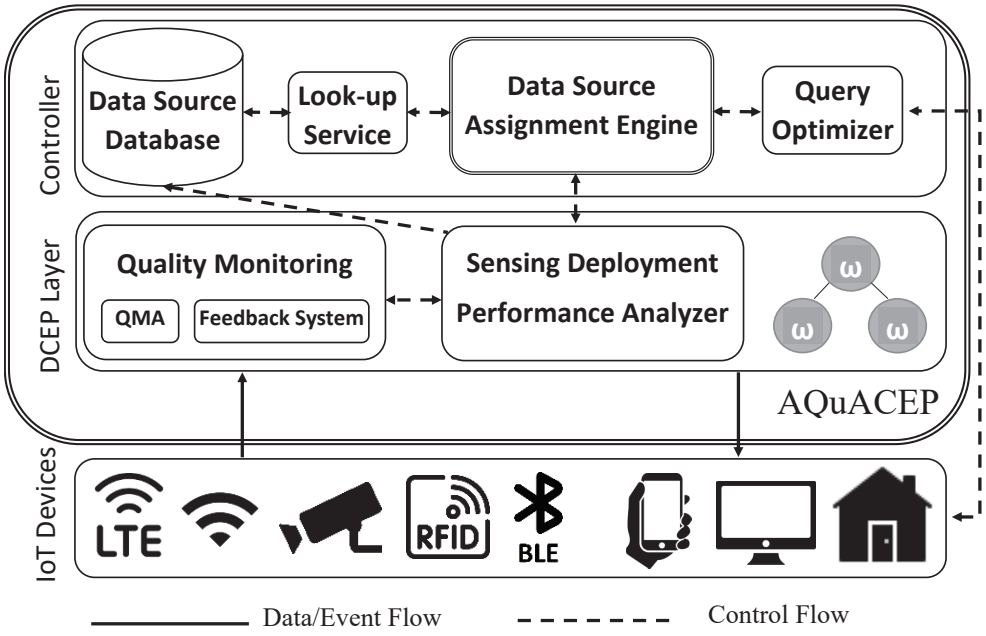


Figure 3.2: The AQuA-CEP system design.

data source disconnection), a so-called *control event* is created by the responsible agent. It notifies the controller to execute corresponding steps as adaptation scenarios in order to maintain the QoR. To do so, a *sensing deployment performance analyzer* investigates the current state of assigned data sources using information acquired by quality monitoring agents and updates the assignment engine to reconfigure sensing deployment, if necessary. The performance analyzer component also updates the data source database's records based on the quality monitoring results. It influences the characteristics of data sources or their availability.

Moreover, parts of our system are built on existing concepts for the flexible execution of event processing operators, as proposed in TCEP [11] and CEP-LESS [181]. This allows AQuA-CEP to modify the deployment and configuration of operators and integrate a wide range of additional event processing engines, e.g., Apache Flink. With such flexibility, AQuA-CEP can revise the

operators' deployment and influence QoS metrics, e.g., bandwidth usage.

3.2.1 Quality Requirement Description

The foremost step in acquiring the consumers' expectations in terms of quality is the technique by which they can elaborate on their requirements. Such a method must be not only easy to use for the consumers, but also sufficiently comprehensive to cover all aspects and flexible to fit different types of consumers' quality requirements. Hence, AQuA-CEP extends traditional query specification of event processing systems by providing the possibility to consumers to express their specific requirements related to a given query in the form of *quality policy*.

Definition 4 *Quality Policy (QP).*
A quality policy in a quality-aware CEP system is a set of rules that define how quality metrics are evaluated and enforced. It specifies key metrics, such as accuracy, along with *threshold levels* and *priorities* to balance conflicting query demands. Based on the threshold type, the policy can be *static* (fixed) or *dynamic* (adaptive). Additionally, it includes actions to adjust event processing or data sources to maintain quality requirements.

Quality requirements in the static type of quality policies are specified based on static thresholds as the exact amounts stated clearly in the query, e.g., the temperature data is requested to be delivered with an interval of 10 seconds. Only one type of quality metric can be involved in each static quality policy. Therefore, one expression is required in the query definition to assess multiple aspects of each event. For each data type, the system needs to provide a manner for the consumers to define acceptable values. For instance, to determine the resolution of images, the consumer should have the possibility to specify the thresholds based on PPI (i.e., pixels per inch). Thus, expressing quality policies for consumers will be as easy as possible.

Consider a situation in which a consumer is willing to specify various but related quality requirements based on a second parameter, e.g., various location accuracy thresholds based on the distance to the dangerous area for

the same query in our use case. In those circumstances, defining multiple static quality policies will enlarge the list of consumer constraints, increasing our problem's complexity. Since the system needs to fulfill only one of those multiple but related static quality policies at a time, we define a dynamic threshold that varies depending on a second factor which can be time or a context-related parameter.

For instance, a dynamic threshold based on the location factor looks like "the location accuracy of an object should be less than 2 meters if it is within 100 meters of a particular area. Otherwise, 10-meter accuracy would be sufficient". By utilizing dynamic thresholds, more intricate descriptions for quality requirements can be explainable, enhancing the flexibility of query definition. Such sort of flexibility will improve consumer satisfaction while optimizing our data analytic system to avoid utilizing more complex procedures to fulfill quality requirements.

In order to validate the admissibility of thresholds, the controller inspects the capability of the data sources and adjusts their characteristics based on the requested thresholds in the query, if applicable. In case there are no appropriate data sources available in the environment concerning the quality requirements, the controller revises the query model with the acceptable thresholds and notifies the consumer about the new query model. Then, based on the feedback obtained from the consumer, the controller will deploy the newly produced query model or cancel the query processing. On the other hand, priorities can also be determined in the query definition. It is worth noticing that a higher query priority will impose higher costs for the query issuer. Such costs can be defined by the system designer depending on the use case.

Moreover, the data source conditions and consumer's quality requirements might be varied over time. That is why the quality policies must be revised at runtime. For example, the consumer may need the result of a running query very urgently, such as the current blood pressure of a patient who may have an acute condition with lower intervals. Therefore, the consumer needs to inform the system about this change by raising the query's priority as well as changing the sensing interval's threshold. To do so, AQuA-CEP employs a feedback process for maintaining and updating the quality policies and re-

newing the policies whenever the quality requirements or the data sources' status is altered.

3.2.2 Quality Monitoring

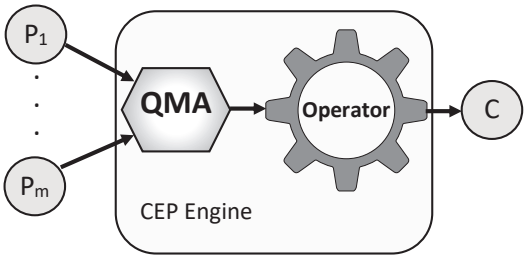
Quality observation should be performed with the lowest possible delay to ensure the expressed quality requirements can be met and adaptation decisions are conducted timely. On the other hand, the available resources for computation are usually bounded. Thus, an event monitoring process must ponder both of these aspects simultaneously.

Quality Management Agent (QMA)

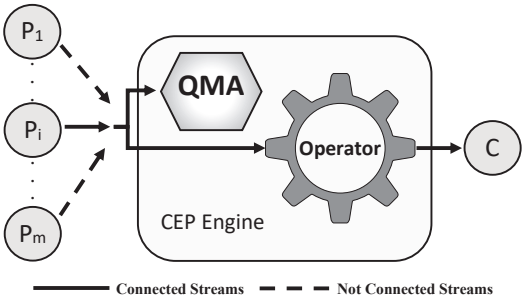
One of the novel traits of AQuA-CEP is to employ QMAs that are liable for inspecting the predefined quality metrics over the event streams and triggering warnings in the matter of quality degradation. On the other hand, the utilization of QMA and where it is hosted in the data plane might influence the quality of service. So, we defined the concept of QMA's configuration model and discussed how it improves the quality in various processing stages.

QMA Configuration Models

As we discussed earlier, the form of QMA configuration in the DCEP layer can yield different results. There are two kinds of configuration models; *sequential* and *parallel* as shown in Figure 3.3. In a sequential configuration model, the events from all producers that fulfill the requirements of a specific query are aggregated into one joint event stream and fed into the corresponding QMA for quality evaluation. In this case, QMA is in charge of filtering events and allows those events that possess the required level of quality to be delivered to the respective CEP engine. The primary advantage of this model is that the transition time (i.e., handover), the time required to swap data sources, is guaranteed to remain small, i.e., in the order of milliseconds. Moreover, joining events from two or more producers is feasible to boost the quality by utilizing redundancy. On the other hand, processing all events from the producers indeed imposes considerable latency in query processing that should be taken into account.



(a) QMA sequential configuration



(b) QMA Parallel Configuration

Figure 3.3: QMA Configuration Models.

The parallel configuration model will impose the minimum possible latency in processing since QMA analyzes the event attributes' quality in parallel. However, only one producer can be connected to the respective operator. Thus, if the quality of the event produced by this data source degrades, it takes time for the QMA to trigger an alert and request the controller to link another data source to the CEP engine that fulfills the quality requirements. Such a transition undoubtedly imposes a noticeable overhead on the event processing system. Nevertheless, both processing and transition delay should be considered as the major costs when the system wants to decide on the configuration model of QMAs.

Moreover, the QMA costs for monitoring each data source can be reviewed based on time and computation, directly dependent on the QMA's configuration model. In terms of time, if the parallel configuration model is chosen,

the cost is the delay related to switching between the current data source (i.e., sd_i) and the next option (i.e., sd_l) as ($C_s(sd_i, sd_l, q_k)$). On the contrary, if the sequential configuration model was chosen, the time overhead is the delay caused by quality analysis ($C_a(sd_i, q_k)$). In terms of computation costs, the overhead in both parallel and sequential models is almost the same. We called this ($C_w(sd_i, q_k)$) includes the required resources for analyzing an event stream using sliding windows and the computing resources for the data source assignment ($C_{ads}(q_k)$). Therefore, the total cost of monitoring the data source sd_i using a QMA is as follows:

$$
\begin{aligned}
C_{QM}(sd_i, q_k) = \quad & w_{seq}\ C_s(sd_i, sd_l, q_k) \\
+ \quad & w_{par}\ C_a(sd_i, q_k) \\
+ \quad & C_w(sd_i, q_k) + C_{ads}(q_k) \qquad (3.2) \\
s.t. \quad & w_{seq} = 1 \text{ iff } \text{“Sequential mode is selected.“} \\
& w_{par} = 1 \text{ iff } \text{“Parallel mode is selected.“} \\
& w_{seq} \in \{0, 1\} \text{ and } w_{par} \in \{0, 1\}
\end{aligned}
$$

3.2.3 Sensing Deployment Adaptation

Adaptation decisions in AQuA-CEP are performed following the MAPE-K feedback loop model [132] building on the previous two steps (i.e., quality description and monitoring). Consequently, the monitoring outcomes are analyzed within every loop to determine adaptation decisions and finally apply them to the stream processing infrastructure.

In Algorithm 3.1, we represent the controller's core functionality which is used in AQuA-CEP. The first step of query processing is to initialize the corresponding variables, i.e., the set of event types, their quality policies, and their thresholds. When the consumer with ID C_{cid} registers the query (e.g., security monitoring of DA), the controller initiates a subscription for each query's simple event (e.g., location event for each query's target). Such a subscription comprises details regarding the event type of simple event (i.e., et_i) and the respective query (i.e., Qid).

On the other hand, a producer P_{pid} registers its available sensing deployments (i.e., $\mathcal{SD}(P_{Pid})$), e.g., LTE signals for location tracking. Then, the controller generates an advertisement for each sensing deployment of the

Algorithm 3.1 Controller Functionality

1: **Initialization:**
   $Q_{Qid} = et_1, ..., et_l \leftarrow$ User Query Qid;
   $QP \leftarrow \{qp_1, ..., qp_n\}$;
   $SUB \leftarrow \varnothing$;
   $ADV \leftarrow \varnothing$;
2: **upon** (C_{Cid}.Submit(Q_{Qid})) **do**
3: **for** $et_i \in Q_{Qid}$ **do**
4: $SUB \leftarrow SUB \cup sub_{Qid}^{et_i}$;
5: AssignDataSource(ADV, SUB); ▷ Algorithm 3.2
6: **upon** (P_{Pid}.Register($\mathcal{SD}(P_{Pid})$)) **do**
7: **for** $sd_i \in \mathcal{SD}(P_{Pid})$ **do**
8: $ADV \leftarrow ADV \cup adv_{Pid}^{sd_i}$;
9: AssignDataSource(ADV, SUB); ▷ Algorithm 3.2
10: **upon** (QMA.Alarm) **do**
11: ProcessAlarm(QMA.Alarm); ▷ Algorithm 3.3

producer. It includes information about the respective data source and the producer's ID. Finally, the controller investigates possible solutions to match current advertisements to subscriptions (Refer to Algorithm 3.2).

Whenever a new dynamic occurs in the environment, a new alarm is issued to inform the controller. Consequently, according to the type of alarm, the controller will perform the corresponding actuation, elaborated on the Algorithm 3.3.

In Algorithm 3.2, we represent the process of assigning data sources. To match advertisements to subscriptions in Algorithm 3.2, the function AssignDataSource looks for potential candidates for each event type (e.g., location event) in the list of related advertisements. In this regard, a data source announced by an advertisement is examined by the MeetAllQP function, which compares the data source's current quality characteristics and the related quality policies' current thresholds (e.g., the accuracy level of fewer than 2 meters). If this data source meets all related quality policies, the correspond-

Algorithm 3.2 Data Source Management

1: **function** AssignDataSource(ADV, SUB)
2: **for** $sub^{et_i} \in SUB$ **do**
3: $QP_{sub^{et_i}} \leftarrow$ Related quality policies to et_i;
4: $M_{ADV^{et_i}} \leftarrow$ Matching advertisements to et_i;
5: **for** $adv^{sd_i} \in M_{ADV^{et_i}}$ **do**
6: **if** adv^{sd_i}.MeetAllQP($QP_{sub^{et_i}}$) **then**
7: $\mathcal{C}_{sd_i} \leftarrow C_{SSC}(sd_i) + C_{QMA}(sd_i, q_k)$;
8: $\mathcal{SD}(sub^{et_i}) \leftarrow \mathcal{SD}(sub^{et_i}) \cup (adv^{sd_i}, \mathcal{C}_{sd_i})$;
9: $\alpha_t(\mathcal{SD}) \leftarrow$ HACS($\mathcal{SD}$, $\mathcal{G}$); ▷ Solution
10: PerformTransition($\alpha_t(\mathcal{SD})$);
11: **function** PerformTransition(α_t)
12: **for** $(sub^{et_j}, adv^{sd_k}) \in \alpha_t$ **do**
13: **if** $sub^{et_j}.sd_{previous} = \varnothing$ **then**
14: $Immediate\_Transition(sd_k)$;
15: **else**
16: $Seamless\_Transition(sd_{previous}, sd_k)$

ing advertisement will be considered a qualified candidate for this subscription (e.g., LTE signal meets the 2-meter accuracy requirement).

The costs of applying this sensing deployment include systems-side and monitoring costs (e.g., energy consumption for utilizing LTE signal) which are calculated for each candidate and paired with its advertisement to form members of a list showing the eligible sensing deployments for each subscription (i.e., $\mathcal{SD}$). Then, a heuristic approach is applied on $\mathcal{SD}$ considering satisfying the constraints in $\mathcal{G}$ (i.e., HACS) to realize an approximate solution for the current situation. This approach checks the currently available sensing deployments and assigns them to the running queries considering the optimization parameter (e.g., maximizing the battery life of targets' cellphones). According to this newly generated solution, if the previous sensing deployment is changed for any query, the transition between data sources can be done in two ways using the function PerformTransition.

The *immediate* transition model occurs when the previous sensing deploy-

ment is not available anymore or notably unreliable (e.g., target goes out of the coverage of BLE). So, the controller should perform the transition as fast as possible with minimum delay (e.g., from BLE to WIFI). On the other hand, in *seamless* transition, the previous data source is still available. Still, it cannot satisfy all quality requirements due to changing the quality policy threshold (e.g., more accurate location event close to DA). Therefore, the controller performs the transition smoothly (e.g., from WIFI to LTE). For this type of transition, AQuA-CEP will process both data streams from previous and current producers concurrently in a period of β seconds in which the transition is happening from its invocation to its completion.

Adaptation Strategies

Algorithm 3.3 shows the capabilities of the AQuA-CEP to adapt dynamically to the changes in the environment, in the quality of data streams, or process the queries based on the dynamic quality policy thresholds. Hence, a feedback system continuously inspects the conditions in both the data plane and the control plane to trigger alarms upon any noteworthy changes. Each QMA alarm has attributes such as type, corresponding sensing deployment (i.e., *sd*), quality policy (i.e., *qp*), and query identifier (i.e., *Qid*).

Upon the arrival of a QMA alarm, if the alarm's type indicates that the connection to a data source is lost and this sensing deployment is not available anymore (i.e., SDUnavailability), the controller removes all the related advertisements and performs data source re-assignment using the global optimizer. The next type of alarm is triggered by a reduction in the quality of data streams concerning the current thresholds of quality policies (i.e., ReducedQuality), e.g., when an obstacle blocks part of a motion detector's vision. In this case, the system performs a *Wait-Monitor* procedure in a specific period, in which AQuA-CEP checks the quality of produced events. If the data source can recover from this situation promptly, our mechanism will continue with the current sensing deployment. Contrarily, suppose the lack of sufficient quality remains for the event stream. In that case, firstly, the related advertisements to this sensing deployment will be updated with the new quality characteristics, and then, a re-assignment procedure will start. The main goal of the Wait-Monitor procedure is to prevent oscillation between

Algorithm 3.3 Alarm Processing

```
1: function ProcessAlarm(A)
2:     switch (A.type) do
3:         case SDUnavailability:
4:             for adv^{sd_i} ∈ ADV do
5:                 if A.sd == sd_i then
6:                     ADV ← ADV − {adv^{sd_i}};
7:             AssignDataSource(ADV, SUB);
8:         case ReducedQuality:
9:             Wait-Monitor(A.sd);
10:            if A.sd Not Recovered then
11:                for adv^{sd_i} ∈ ADV do
12:                    if A.sd == sd_i then
13:                        Update(adv^{sd_i})
14:                AssignDataSource(ADV, SUB);
15:        case ChangedQualityThreshold:
16:            for sub^{et_i} ∈ SUB do
17:                if A.qp ∈ QP_{sub^{et_i}} then
18:                    Update(sub^{et_i})
19:            AssignDataSource(ADV, SUB);
20:        case QueryEnded:
21:            for sub^{et_i} ∈ SUB_{A.Qid} do
22:                SUB ← SUB − {sub^{et_i}};
23:            AssignDataSource(ADV, SUB);
```

data sources since it will lead to more switching costs and might produce a worse global solution.

Since the consumer is able to adjust the quality policy threshold at runtime, various ranges are possible for thresholds according to the query model. In such cases, an alarm is triggered (i.e., ChangedQualityThreshold) to indicate that a new threshold should be taken into consideration. Hence, each subscription related to the changed quality policy has to be updated, and a

new global re-assignment should be performed. Finally, if a query is finished on time or even ahead of time manually, the corresponding subscriptions will be removed from the set of subscriptions. In addition, the producers and CEP operators should disconnect from each other. Since the absence of those subscriptions may change the global solution, executing the AssignDataSource function on the available advertisements and subscriptions is necessary.

3.3 Evaluation

In this section, we experiment with different ways of monitoring the quality of event detection and its corresponding adaptation strategies. The main goals of the evaluation are to figure out 1) can dynamic event source assignment provide a trade-off between QoR (i.e., fulfilling consumer constraints) and QoS (i.e., minimizing the system-side costs) and 2) what limitations are involved in performing a transition among sensing deployments.

3.3.1 Simulation Setup

We created a single Virtual Machine (VM) in *Oracle VM Virtual Box Manager* [182] in which we installed *Ubuntu version 20* OS. We allocated 6 CPU cores with 100 percent execution capacity and 24 GB of main memory to the VM. We run complex event processing with multi-threading in this machine and create a thread for each of the issued queries; thereby, we could manage them simultaneously using Java. For future work, for making performance-related studies, we would need indeed to implement all components over separate computational resources.

For publish/subscribe communication of AQuA-CEP, we build on *Apache Kafka* [183] as a distributed platform. Furthermore, for detecting complex events, we build on *FlinkCEP* [184], which is a library implemented on top of Apache Flink. In our simulation[1], a Kafka server acts as an event broker that serves both data events and control events, as depicted in Figure 3.4. We monitor the quality of produced event streams in the QMA, which is located as an Apache Flink operator using the parallel QMA configuration model in the scenarios described below.

[1] https://github.com/majid-lotfian/AQuA-CEP-code

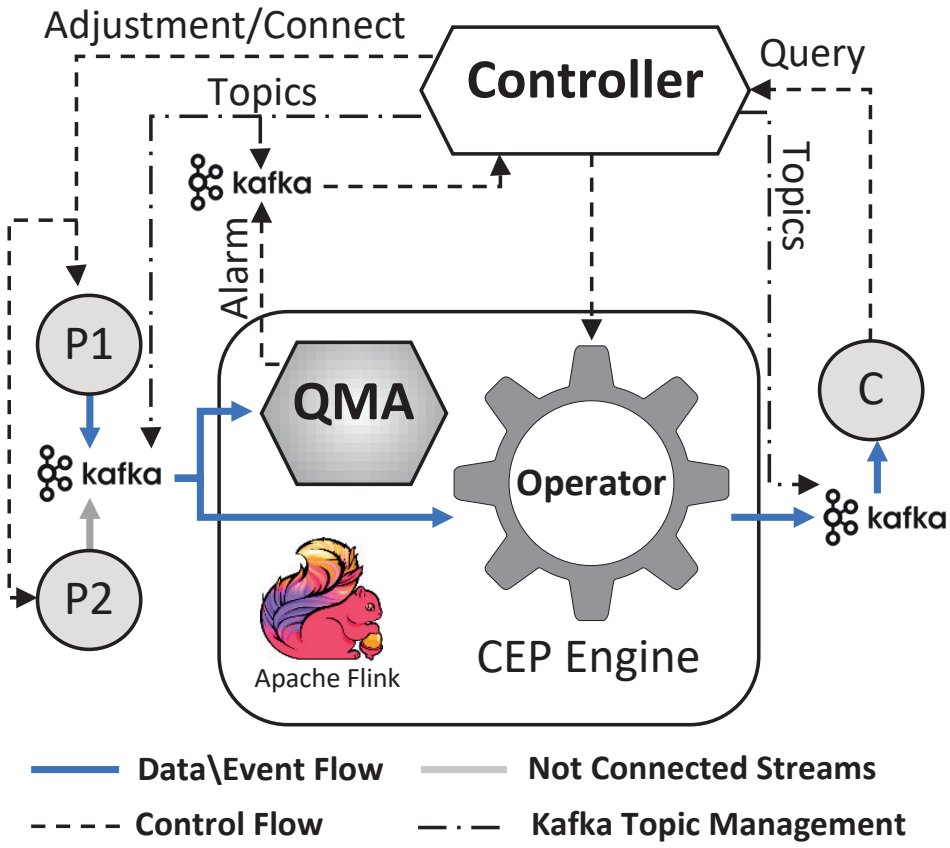


Figure 3.4: The evaluation model of AQuA-CEP.

Static Quality Policy Scenario

Consider an area where several temperature sensors are embedded in different locations shown in Figure 3.5. In such a scenario, the temperature in some locations can be captured by multiple sensors. Each of these sensors has its own characteristics (e.g., sensing accuracy, coverage, etc.) that influence the generated data. We consider a set of continuous queries, each detecting a high temperature in a specific location. We also generated random variations in each sensor's attributes, which simulate the environmental dynamics, acknowledging that such randomness only approximates real-world correlations

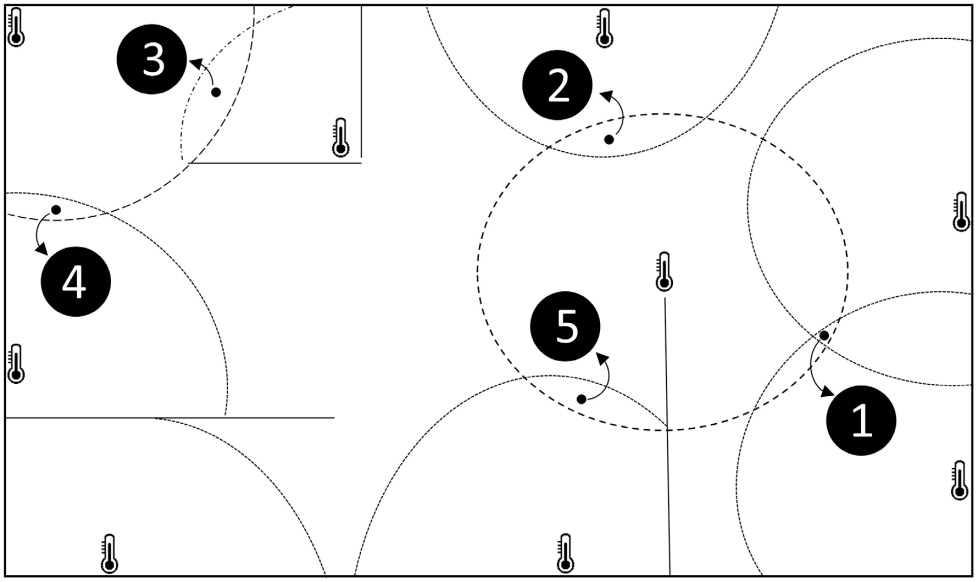


Figure 3.5: Modified coverage problem with static quality policy.

among sensors.

To make our motivation clear in this scenario, we began with a *coverage problem* scenario [185], in which the objective is to solve the sensing placement problem to maximize the coverage of m important points using n sensors. To adequately capture the capabilities of AQuA-CEP, we modified the coverage problem so that sensors are already located in the environment, and their location cannot be changed. Moreover, instead of a set of m critical points, we have a dynamic group of queries to be answered. The goal is to optimize the total energy consumption by activating the set of sensors that can cover all queries.

We analyze the performance of AQuA-CEP in terms of the event loss rate since assumed dynamics directly influence this quality metric. We compared our approach with two baseline mechanisms. The first approach is called *Optimal Dynamic Loss Rate (ODLR)*, which selects the best data source in terms of Loss Rate at the start of processing a query for each event type.

Table 3.1: Applied Queries in the Static Quality Policy Scenario

Q#	Location	Temp Threshold	Loss Rate Threshold (%)
Q1	42:18	> 20.8	< 10
Q2	31:7	> 20.1	< 15
Q3	11:6	> 21.1	< 12
Q4	2:11	> 19.1	< 15
Q5	29:21	> 22.5	< 10

Then, once the monitored loss ratio for produced event stream in runtime goes above its predefined characteristic, the controller first updates this feature with the new assessment and then performs a reassignment check. The second approach is called *Optimal Static Loss Rate (OSLR)*, which selects the best data source in terms of the event loss rate again. But, it stays with this data source until the end of the query and does not switch to another data source even in the case of an event loss ratio increase of its event stream.

```
SELECT  event.*
FROM
    // Selecting event stream
    SELECT  ds.stream
    FROM    DS
    PATTERN
        lossrate < QualityPolicy.threshold
    Within    window_size
    WHERE
        ds.type = 'Temperature'  AND
        ds.coverage(target_location) = True
WHERE  event.value > Query.value
```

Listing 3.1: Applied query with static quality policy.

We applied the same query with different threshold values, detailed in Table 3.1, on each approach with the same simulation setup. The points targeted in each query are marked in Figure 3.5, and two or more sensors can be hired for each. The query definition determines the temperature thresholds, and the minimum required Loss Rate is pinpointed in the respective quality policy. We acquire the consumer queries utilizing a designed JFrame on run-

time containing the definition and a set of quality policies. Then, the query is transformed into a CEP-enriched SQL format.

An example of a continuous query is shown in Listing 3.1, which aims to collect the temperature in a specific location. We defined a static quality policy by specifying a threshold value for the event loss rate characteristics of the sensing deployment in the *PATTERN* clause, which will be used in the data source assignment procedure.

In addition, the temperature data in [186] is used, which contains two datasets. We chose one that includes the sensory data about temperature, pH, and turbidity from 30 cm below the water surface. We utilized this dataset because it has a real-world distribution of temperature data that makes our calculation more realistic. In our mechanism, each functioning sensor takes the data from this dataset and transmits its own transformed data according to its predefined quality characteristics. It means that each sensor will produce a unique temperature data stream according to its own features.

Moreover, we estimated the amount of energy consumed by each sensor as described in [187]. This includes the energy for sensing the phenomena, processing the measurement, logging (i.e., reading data and writing it into the memory), communication, and transient energy (i.e., the transition energy to go from the idle state to the active state, and vice versa). We presumed that the distance between sensors to the gateway is the same, and they transmit packets of the same size.

Dynamic Quality Policy Scenario

Consider the earlier mentioned use case where the specified dynamic quality policy is "the quality of the person's location can degrade, as the person moves away from the borders of the dangerous area, but it should be as accurate as possible when it is in a proximity of the target location". In this policy, the quality metric is the location *accuracy*, and the second parameter is the target's *location* (i.e., the accuracy level is determined according to the target's current location). In this scenario, the consumer in the factory needs to submit a query to control people's access to the dangerous area. To do so, the query should consist of the dangerous area's details and the alarm region.

Table 3.2: Sensing Configurations and Their Characteristics

Name	Range (m)	EC (mW)	Accuracy (m)
BLE	70 - 100	426	1 - 3
RFID	1 - 12	375	0.1 - 2
WIFI	50 - 100	817	1 - 5
Camera	N/A	374	< 1
LTE	$>$ Km	1634	< 1

AQuA-CEP can access a sensing infrastructure based on a multitude of sensors deployed on a target's devices (e.g., smartphones), such as Bluetooth Low Energy (BLE), LTE, WiFi, and RFID sensors. Moreover, the system can also benefit from other positioning infrastructure embedded in the environment, like cameras. Each of these sensing deployments has its own characteristics, and in order to estimate the Energy Consumption (EC) in this scenario, we reuse the measurements collected from [188–192], as indicated in Table 3.2. Among all sensors, only *Camera* does not utilize the mobile phone's battery since it is a separate camera placed on the wall. Therefore, we assumed this sensor's energy consumption is equal to placing a target's mobile phone in airplane mode.

Since a publish/subscribe mechanism is used to manage people in this factory, a notification event of prohibition to approach a dangerous area is generated. All targets within the factory will be notified of the prohibition. That's because all targets in AQuA-CEP have already subscribed to these notification events when performing the admission process, and delivery reliability is ensured through acknowledgment and retransmission mechanisms. Moreover, during admission, each target explains its attributes, including its identity and role in the factory. Also, it describes its carrying devices with their sensing capability (e.g., smartphones, tablets, smartwatches, wearables, etc.). In addition, it should be clearly detailed what type of data each device can produce and what sensing and communication technologies it can provide. Also, we assume that each target will give continuous access to their

Table 3.3: An example of applied queries with dynamic quality policy

Q#	Definition		Quality Policies	
	CP	AR(m)	Q-Metric	Condition
Q1	60 : 185	70	Accu < 2m	0 < DTDA < 100
			Accu < 5m	100 < DTDA < 200
			Accu < 10m	200 < DTDA < 1000

registered sensing deployments and not deliberately block the connection.

The query correlates specific conditions—the target's position, the boundaries of the dangerous area, the alarm region, and the authorization status of the target—to detect a complex event of dangerous area violation. For this case, the perception of a location event could use different sensors to get an approximate position with the required quality level specified in the query's quality policies while optimizing the energy consumption of the application installed in the smartphone. Dependent on various aspects like coverage, location uncertainty, and sensing frequency, various query models can be utilized to meet the QoR and QoS trade-off requirements. In this specific use case, since we do not need a very accurate position when the person is far from the dangerous area, AQuA-CEP uses the sensor that first meets the quality level of the query and then has the smallest amount of energy consumption for the smartphone.

Alike to the static quality policy scenario, we applied the query to the simulation environment multiple times considering the dynamic quality policy and analyzed the results to gain more insights. With a fixed route for a target person, we randomly generated coordination for the dangerous area, including the details for *CP* and *AR*. That means if a person nears *CP* by less than *AR* meter, a violation will happen. Besides, such a location estimation should be done using a sensor with the accuracy (Accu) level denoted in Table 3.3, considering the conditions related to the Distance To the Dangerous Area (DTDA).

The CEP-enriched SQL format of the applied query with a dynamic quality policy is represented in Listing 3.2. The initial *WHERE* clause (line 9) performs preliminary stream selection based on location-specific quality policies. This pre-filtering step reduces the computational load on the subsequent

constraint satisfaction global optimization algorithm, which refines the list of candidate streams (e.g., LTE signals) by considering a broader range of system constraints and objectives to identify the optimal stream configuration.

In our simulation, we chose to apply *Choco-solver* [193], an open-source Java library for constraint programming. Employing this library, we can solve our optimization problem while considering consumer-side constraints. We expressed the event source assignment task as a variable, where the domain of all variables is the available sensing deployments. Therefore, the Choco-solver considers the constraints to find a global solution covering all variables. This optimization solution will determine which stream from the list of candidates should be assigned to each query.

```
SELECT  event.*
FROM
    // Selecting Event Stream
    SELECT  ds.stream
    FROM    DS
    PATTERN
      Accuracy < CurrentQualityPolicy.threshold
    Within    window_size
    WHERE   ds.type = 'Location'
          AND
          // Selecting Current Quality Policy
          CurrentQualityPolicy = (
                SELECT  qp
                FROM    Query.QP
                WHERE   qp.InRange(target_loc))
WHERE  Distance(event.loc, Query.DA) < Query.AR
```

Listing 3.2: Applied query with dynamic quality policy.

3.3.2 Simulation Results

In this section, we will analyze our findings compared to the other two mentioned baseline strategies in both categories of static and dynamic quality policies. It should be noted that the results regarding energy efficiency and quality of results do not depend on the number of consumers, but indeed on the number of queries only since a consumer can issue multiple queries. That's why increasing the number of involved queries can help evaluate the

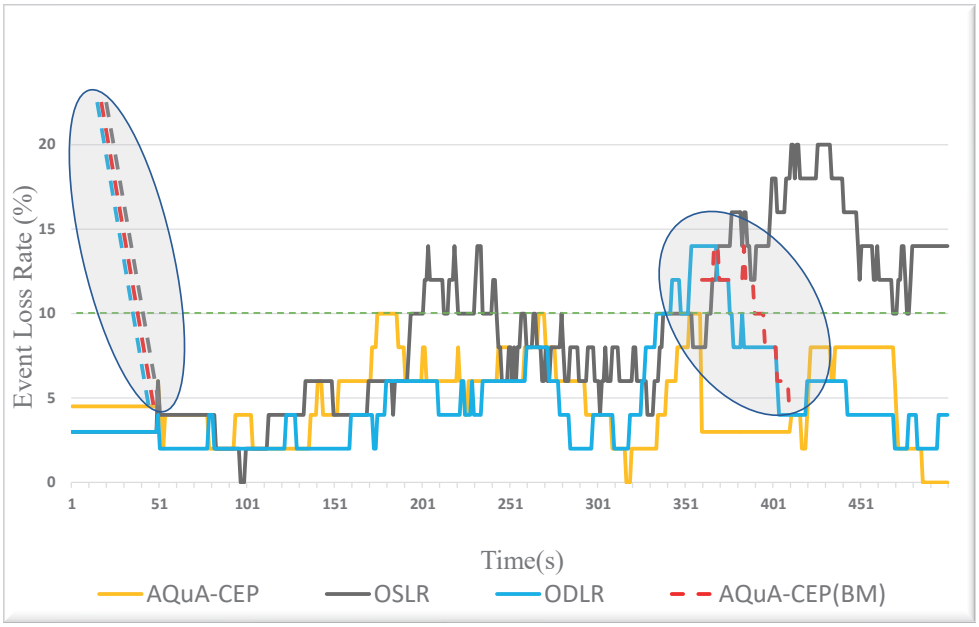


Figure 3.6: Event loss rate ratio during the execution for query 1 with the static quality policy.

scalability of each approach.

Static Quality Policy

The static quality policy scenario results have been illustrated in Figure 3.6 and Figure 3.8. We executed the simulation ten times for each query, and the results were approximately similar. To challenge our approach, we selected the results with AQuA-CEP's worst performance.

Regarding the event stream loss rate in Figure 3.6, the chart displays the event loss rate for each approach in one execution. It can be seen that the OSLR approach shows the worst performance and proves the idea that each mechanism requires adapting to the dynamics. Both AQuA-CEP and ODLR select those sensing deployments meeting the threshold, which is illustrated as a green dashed line. Only two times the proposed approach exceeds the threshold of the event loss rate highlighted by the gray circles.

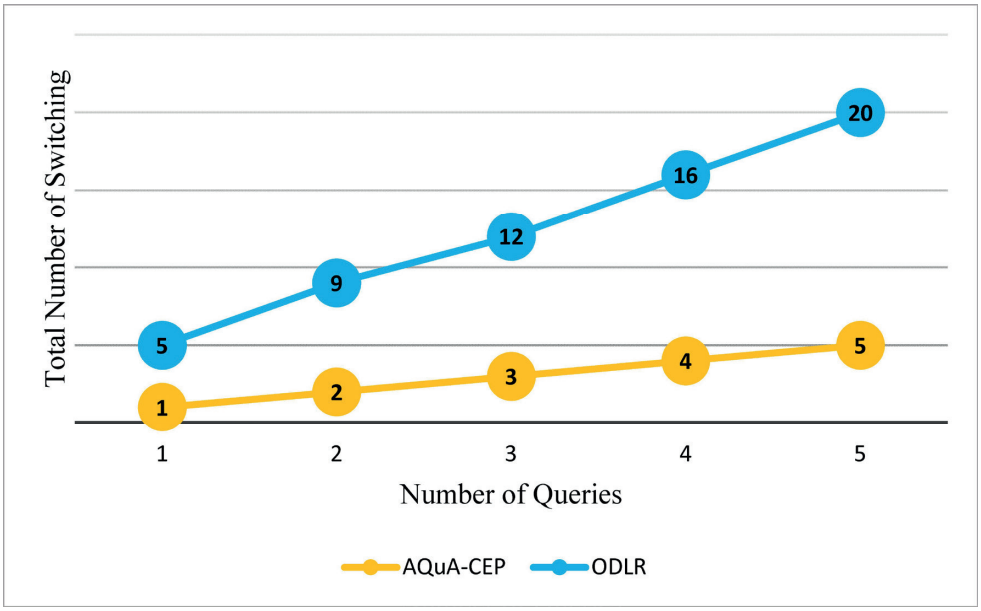


Figure 3.7: Switching counts between sensing deployments for different sets of queries with the static quality policy.

In the first quality policy violation, at the start of query execution, we must wait for the window to be completed since we are employing a 50-sec window size over the event stream. So, we can not rely on the monitoring results, and we called this period as *Blind Monitoring (BM)* period. Such a period started once we chose a new sensing deployment. We masked this period with data from the characteristic of the new sensor for all three approaches and showed the simulation values with dashed lines. The second gray oval indicates the sensing deployment switching time for AQuA-CEP. Again, we showed the simulation results with the red dashed line. We cannot rely on this window information because the window comprises events that partially belong to the previous sensor, and the rest belongs to the newly selected sensor until the BM period is ended. We can not perform adaptations within this period since the outputs are unreliable. Therefore, the event loss rate results can go above the predefined threshold, and no sensing deployment switching would be triggered.

Hence, a lower number of BM periods results in a higher percentage of query duration being monitored. Having this fact in mind, AQuA-CEP achieves better results than the ODLR since it has fewer switching counts. To better portray this advantage of AQuA-CEP, more queries are involved in the comparison, and the results for the number of sensing deployment switching are depicted in Figure 3.7. The chart shows that the difference between AQuA-CEP and ODLR is escalated considerably once more queries come to the system. That means the number of BM periods is increased significantly, leading to less reliability in quality monitoring that also influences the adaptation decisions.

In addition, a higher number of sensing deployment switching results in taking more actions to activate or deactivate sensors. This means that in the ODLR approach, more time overhead is imposed due to stopping the analysis of event streams and monitoring the event quality on the previous sensor. Moreover, initializing these two functionalities over the event stream of the new sensing configuration increases the time overhead.

From the energy consumption point of view, there is also a remarkable dissimilarity between these two approaches exhibited in Figure 3.8. The graph shows that the amount of energy consumed in AQuA-CEP is less than the ODLR approach. From four queries onward, the total consumed energy seems equal. The reason is one or more queries in ODLR stopped processing; thereby, the consumed energy for them was equal to zero. On the other hand, queries in AQuA-CEP keep being answered until the end of simulation time. To be fair, we illustrated the results for both approaches until 250 sec of execution, when all queries were still active. Hence, the consumed energy for all sensors in AQuA-CEP is dramatically lower than the ODLR approach, respectively, proving the ability of our approach to optimize the server-side costs. It also can be seen that the difference between the two approaches is increasing by involving more queries in the execution. This confirms that AQuA-CEP is more reliable than ODLR in dealing with involving more queries.

Dynamic Quality Policy

Figures 3.9 and 3.10 exhibit the simulation's outcome in the case of applying dynamic quality policy over the query processing.

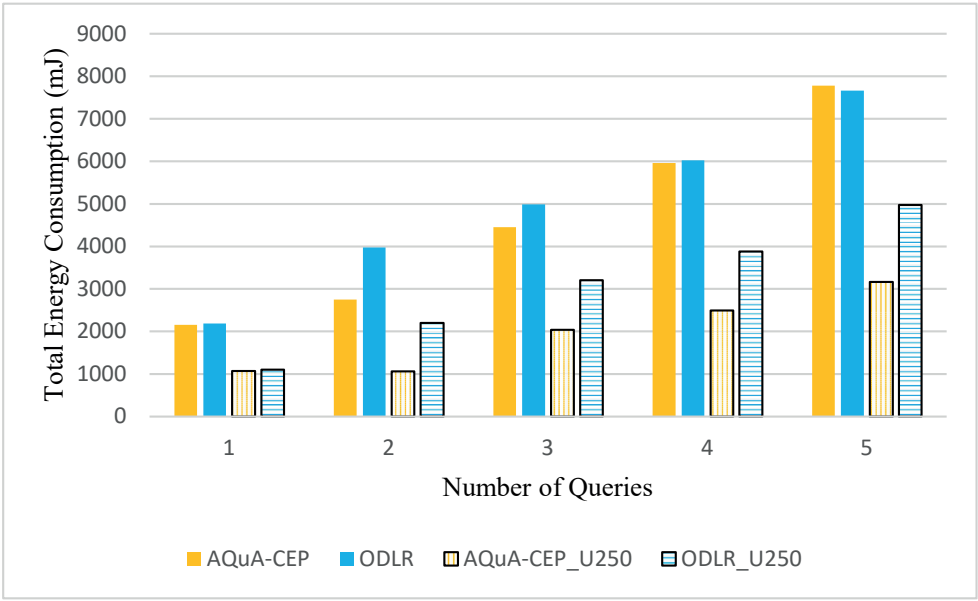


Figure 3.8: Total energy consumption for different sets of queries with the static quality policy.

With energy consumption in mind as the comparison parameter, one can observe from Figure 3.9 that there is a dramatic distinction between AQuA-CEP and ODA. The rationale is ODA prefers the LTE sensor at all times for location tracking since it delivers the most accurate results while consuming the highest amount of energy among all sensing deployments. From this point of view, it can be concluded that employing the ODA mechanism can quickly lead to the phone's battery exhaustion. On the other hand, although the ODE approach is assumed to be the optimal approach in terms of energy, it consumes slightly less energy than AQuA-CEP. That's why we can claim the performance of AQuA-CEP regarding energy consumption is near-optimal. The discrepancy between the results of the ODE approach and our mechanism is increasing slightly by initiating more queries, but it is still negligible.

In event-based systems, False Positives (FP) and False Negatives (FN) are widely used to compare detection accuracy. In our example, an FN denotes

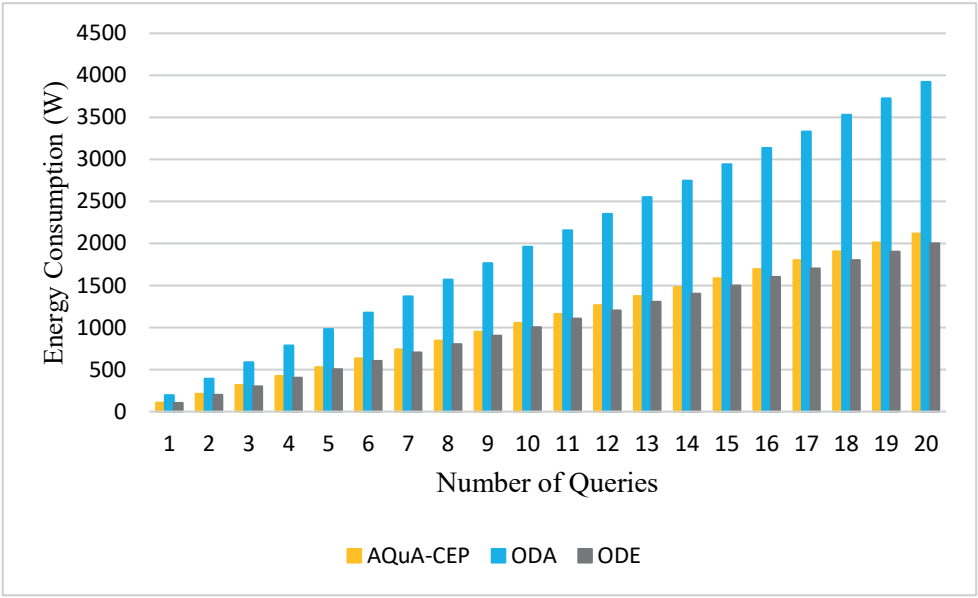


Figure 3.9: Total energy consumption for different sets of queries enriched by the dynamic quality policy.

a violation by entering a red-flagged area that occurred in the real world, but the event processing system could not catch it. Besides, an FP indicates a violation wrongly detected by the system. Since both of these errors are feasible in our use case with small counts, we form a single number of their summation that makes the differences more distinguishable, as illustrated in Figure 3.10.

Since ODA permanently answers queries with the most accurate sensors, it serves better than other methods. It could catch all the complex events without any FP or FN, but with the cost of draining the phone's battery. With fewer queries, ODE acts nearly the same as AQuA-CEP. But once more queries are involved, the situation becomes worse for this approach. It probably happens because of the lower detection capacity of data sources with minor energy consumption when the dangerous area is located in their close vicinity. The sensors with less energy consumption level have less room for consumers and might deliver less accurate data leading to more FPs and FNs.

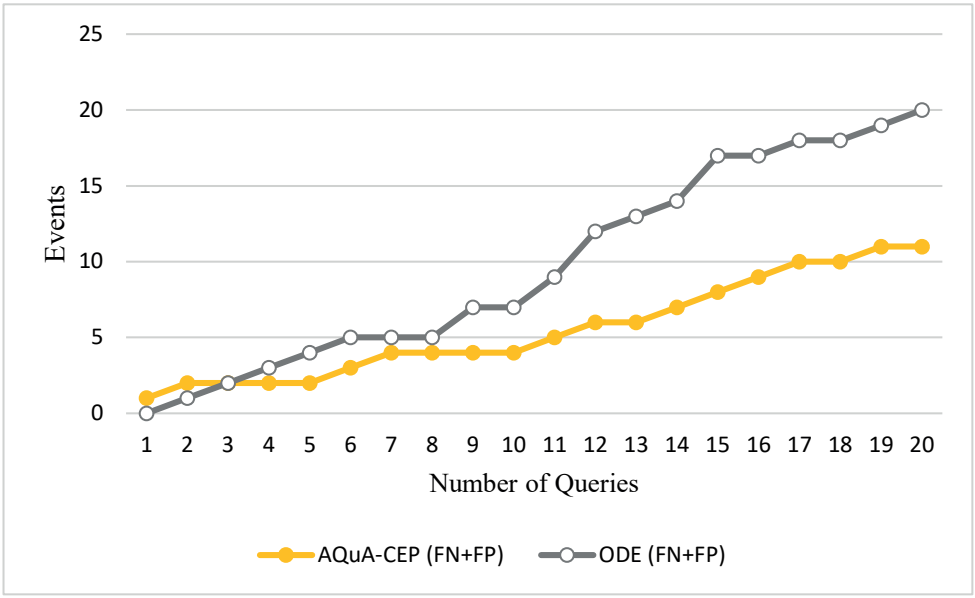


Figure 3.10: Sum of FP and FN for different sets of queries enriched by the dynamic quality policy.

Similar to FN and FP, *F-score* is a well-known performance measure in machine learning approaches that combines the other two measures, *precision* and *recall*, which are mainly employed to distinguish between classifiers [194] in terms of accuracy. Therefore, F-score can be used in stream processing to compare mechanisms in terms of accuracy. Analyzing the F-score shows an *ascending trend* in the reports, while the outcome for ODE displays a *descending trend*. This proves the ability of AQuA-CEP to deal with involving more queries while maintaining the accuracy of event detection. The possible reason is that involving more queries increases the overlap in data sources; thereby, multiple queries can benefit from our proposed switching mechanism. Consequently, the F-score values for our approach become better.

3.4 Related Work

Responding to environmental dynamics is highly important while using DCEP systems in IoT scenarios for maintaining the QoR and QoS at a satisfactory level based on consumers' quality requirements. To select the suitable sensing deployments for each query, AQuA-CEP performs a global optimization in sensing deployment configuration that takes into account the consumer-side constraints as their quality requirements and monitors the quality of produced events to assess and make adaptation decisions to maintain the quality of the produced outcomes. In this section, we compare our work to the related work in two key areas, sensor selection and quality monitoring.

3.4.1 Sensor Selection

In the context of IoT networks, the Sensor Selection Problem (SSP) is a leading research direction to select the best set of sensors to achieve energy efficiency due to power limitations in sensor nodes [195–197]. More precisely, the selection in SSP-oriented research works is performed by picking a set of homogeneous sensors and executing the sensor aggregation. In the context of stream processing (e.g., DCEP systems), most runtime adaptive approaches concentrate on the operator networks, including adaptation on topology, deployment, processing, overload, fault tolerance, and infrastructure [175]. There are a few approaches similar to AQuA-CEP which are called *data source switching* mechanisms [198, 199] and most of them only applied to video streaming applications [200], suggesting that while adaptive source switching is well explored for continuous media, its application to CEP remains under-investigated.

Although both of the mentioned related approaches study the dynamic selection of sensors in an IoT environment to optimize the QoR, such optimizations are performed respecting specific data attributes or a set of specific fused sensor sources. However, integrating such methods in the context of DCEP requires linking them dynamically to different configurations of heterogeneous sensors. Only in this way, the flexibility of current DCEP systems in reconfiguring and rewriting the detection logic of complex events can be used to optimize for QoR.

3.4.2 Quality Monitoring

In DCEP systems, quality assessment has been primarily studied in the placement of detection logic over the available computing resources (e.g., [11]), and only a few research works focus on the adaptation of sensing deployment to handle dynamics (e.g., [13]). Although these mechanisms are quite similar in performance to our proposed approach, they are focused only on one aspect of the system (e.g., CEP query language in [14]).

In the IoT environment, the *service composition* mechanisms consider the sensory data streams as services provided by the connected objects to be analyzed and deliver results to the corresponding applications and allow the interaction between consumers and smart objects of IoT environment [96,201,202]. Considering the vulnerability of IoT service quality to environmental dynamics, service composition techniques try to specify a set of quality metrics to analyze the quality of delivered streams to target applications. By employing heuristic and meta-heuristic techniques (e.g., [203]), several approaches attempt to find a global service composition solution while fulfilling QoS demands. These mechanisms tend to monitor and assess the quality of IoT services and adapt the system to maintain the quality, e.g., by training the Hidden Markov Models (HMM) to predict QoS. However, their quality expressivity is limited to defining the static thresholds causing inflexibility in acquiring more complex consumers' quality requirements. Moreover, AQuA-CEP is more efficient in energy consumption since it endeavors to minimize the number of active sensing deployments and eventually fulfills the requested quality requirements.

3.5 Summary

In this chapter, we presented AQuA-CEP, which proposes how to enable dynamic adaptation of sensing deployment configuration while observing the quality of produced events *and* their data sources. In addition, this mechanism can help save resources in the sensing infrastructure by proposing optimization criteria for sensor dynamic activation. Our evaluation results demonstrated that AQuA-CEP outperforms two baseline approaches in switching counts between sensing deployment and performed near-optimal concerning the total

energy consumed by the sensing network in a static quality policy scenario. Moreover, by applying a dynamic quality policy, AQuA-CEP achieved near-optimality in terms of energy consumption and quality measured in the form of FP and FN. Moreover, the F-score results proved that AQuA-CEP has sufficient capabilities to fulfill consumers' quality requirements when more queries are involved.

In our future work, we will consider *priority* in the quality policy definition and investigate its impacts to support concurrent queries. Estimating the switching overhead is another point of interest that requires further research. In addition, minimizing the blind monitoring periods can be attainable by predicting the data source switching time. We believe building on a statistical analysis of the data sources' performance will be a promising direction. Finally, we plan to extend our proposed research by considering dynamic factors, such as the degradation of data sources over time, in dynamic quality policies.

Chapter 4

Privacy Protection in CEP

Speak not words devoid of gain,
For from such flames, only smoke
shall remain.

Ferdowsi (Revised by GPT4o)

Abstract

This chapter discusses the topic of Privacy-preserving mechanisms to protect sensitive information revealed through event patterns in DCEP (the privacy problem). We introduce APP-CEP, a novel approach for pattern-level privacy in event-based systems, by selectively applying obfuscation models independent of the actual events. Utilizing CEP-like patterns, we generate dependency graphs to dynamically assign obfuscation models that minimize the impact on detecting other patterns while ensuring QoS. By modeling potential adversary knowledge, we enhance privacy protection. Evaluation in a real-world scenario demonstrates APP-CEP's effectiveness in achieving a privacy-utility trade-off and preventing adversaries from detecting stream modifications.

Interpreting the individual's data to generate insightful information has attracted interest in various application fields such as e-commerce, public healthcare, and the Internet of Things (IoT). Such applications typically involve distributed, timely data processing for preventive or predictive use, with constraints ranging from mere least-latency up to real-time requirements [122]. One of the state-of-the-art paradigms for analyzing data in a distributed manner in real-time is Distributed Complex Event Processing (DCEP): Streams of simple events (e.g., IoT raw data) are analyzed and transformed into complex events representing situations of interest (e.g., queries over sensor data

streams). This transformation is performed using a set of processing logic called CEP rules [28]. For example, a traffic monitoring system can infer a *road congestion* via simple events: Average_Vehicles_Speed $<$ 20 km/h and Vehicles_Density $>$ Normal_Density.
A key challenge in the analysis of such information is privacy [204]. It leads to a conflict of goals: On the one hand, users of a DCEP system should be provided with an optimal Quality of Service (QoS). As an example from the e-commerce domain, product managers should be able to detect and reason about varying sales. On the other hand, disclosing specific events (e.g., purchase details) might violate the privacy of data owners (e.g., customers of a webshop) [36]. Similar examples can be found for medical data of hospital patients or health insurance clients [15]. They share an honest-but-curious type of adversary, represented by users as well as nodes of a distributed CEP middleware. Hence, DCEP requires *Privacy Preserving Mechanisms (PPM)* to protect the privacy of data owners. Despite most PPMs (e.g., access control) operating on the level of single events (i.e., by protecting event attributes), privacy requirements are often represented by a combination of events through complex event patterns. A pattern is the CEP representation of a complex event with a set of one or more operations over simple events (e.g., filtering special events) to detect a situation. For instance, a *sequence* pattern equals the occurrence of specific events in a predefined order over single or joint streams, such as a purchase of a pregnancy test followed by children's toys.

Those privacy-sensitive patterns that data owners want to conceal are called *private patterns*. Conversely, *public patterns* are non-privacy-sensitive patterns that must be detected accurately and timely to deliver the promised services, e.g., query results. Regarding the accuracy of complex event detection, failing to detect a complex event that occurred in the real world (i.e., False Negative, FN) and false detection of events that did not actually happen (i.e., False Positive, FP) are the main QoS metrics by which the performance of a DCEP system can be evaluated.

This work demonstrates that a feasible trade-off between concealing private patterns and detecting public patterns cannot be comprehensively provided by a single, statically applied PPM [205]. To this end, our approach is based on a dynamic assignment of different obfuscation models to optimize

the following constraints.

First, complex event patterns share causal dependencies. This forbids a naïve solution based on just one obfuscation model, e.g., event reordering, which fails in case the reordered event stream cannot allow for any meaningful public pattern detection. Second, an adversary's background knowledge must be taken into account when choosing an obfuscation model. For example, for any event e_1 dropped in the result of one query, but not of another, an adversary could possibly infer parts of the original stream based on the combined knowledge of both. Third, DCEP systems are by nature dynamic: Public and private patterns can be dynamic due to changes in active queries or the data owners' privacy requirements. Input stream sources might vary spontaneously. Finally, an adversary's background knowledge monotonically increases over time.

In this chapter, we present APP-CEP as a solution to these challenges. Our contributions are as follows:

1. A PPM that obtains privacy requirements and situations of interest in the form of event patterns as input and specifies the most efficient obfuscation model to conceal each private pattern when delivering the results to the queries.

2. Obfuscation model selection in our proposed PPM based on a graph structure, used to model pattern dependencies and extract the input stream's features to predict the switching time between obfuscation models to avoid privacy violations and meet scalability requirements.

3. A structure for data owners to dynamically model the adversary's potential background knowledge based on an event dependency set and possible statistical event-related information gathered from the available event streams' history.

4. Performance evaluation based on two real-world datasets to show the ability of APP-CEP to boost the performance of the DCEP systems in real-world scenarios.

4.1 Pattern-level Privacy Protection Problem

In a recent project by the National Statistical Institute of Norway [206], several grocery chains have been ordered to share all their transaction data (receipts) with this statistics agency. To protect privacy, this institute claims that by performing *pseudonymization*, the account number, which can identify a person, will be changed to another unique number (i.e., attribute-level access control). However, pattern-level correlation plus background knowledge would enable the adversaries to realize customers' identities, violating their privacy.

In this section, we introduce a scenario based on the transaction set of an online retailer that demonstrates the application of APP-CEP in real-world situations similar to the mentioned Norwegian example (see Figure 4.1). In such a scenario, a state-of-the-art PPM fails to provide an acceptable level of privacy protection since they mainly protect at the level of events (e.g., by controlling the access using event attributes obfuscation). We consider all transactions of a retailer selling all-occasion gifts as the input stream and attempt to detect predefined public and private patterns. Any user (e.g., webshop's manager) can issue a query to detect situations of interest (e.g., top-selling products in a particular period) that are active for a specific time. On the other hand, data owners are customers of the webshop, and each of their purchases triggers an event in this scenario (i.e., using their customer ID) and is labeled with a timestamp. Lining up the purchase records according to their timestamp generates a transaction set event stream, which is analyzed to detect valuable real-time correlations.

By employing APP-CEP, a customer can specify the privacy requirements, e.g., concealing a Christmas party. In this scenario, we assume the sensitive information can be represented by CEP patterns (e.g., sequence, conjunction, negation, etc.). For instance, a Christmas party can be represented by *CONJ (Invitation Card, Christmas Ribbon, Christmas Decorations)*. To conceal such sensitive information, the chosen obfuscation model (e.g., drop event for *Invitation Cards*) might influence the results of non-sensitive queries (e.g., top-selling products). Moreover, various parameters (e.g., pattern dependencies, event distribution in the input stream, etc.) should be considered carefully. Hence, the goal of APP-CEP is to answer user queries intelligently in such

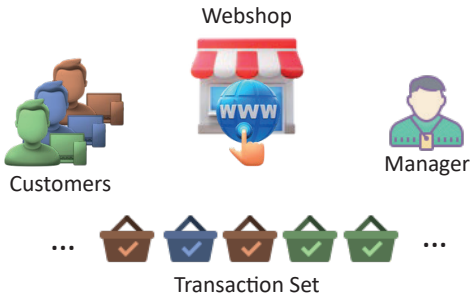


Figure 4.1: Webshop Scenario: The system aims to protect customers' privacy while answering webshop manager queries.

a way that the privacy of customers is not violated. Extending our analysis beyond the webshop scenario, we assessed the capabilities of APP-CEP on a medical dataset featuring more entangled private and public pattern sets. This dataset allowed for a more rigorous evaluation of our method under more complex conditions beyond the simple pattern characteristics in the webshop scenario. A detailed description of this medical dataset and its characteristics is provided in Section 4.3.

Consider as inputs of a stream processing system sets of public and private patterns. Basically, a PPM requires checking the actual events in the input stream to compare and assign obfuscation models on the fly (i.e., by limiting the stream dimension using windowing) [16]. Therefore, adaptive decisions for maintaining the privacy-QoS trade-off should be performed at run-time by exchanging between the best possible PPMs. Such run-time changes are also referred to as *transitions* [207]. Performing transitions between PPMs requires careful treatment regarding the required time and resources. Moreover, transitions can impose gaps during which PPM cannot meet the privacy requirements (i.e., periods with no applied obfuscation). Also, an oscillation between obfuscation models might occur, which will worsen as the pattern sets are scaled up since the number of switches will drastically increase. Consequently, finding a solution to predict transition points is necessary to minimize the overhead and prevent oscillation.

Moreover, the adversary's background knowledge level is determined in

such systems in the initial stage. However, an *Omnipresent adversary* can increase its knowledge by issuing extra queries to correlate the generated results. This indicates another weakness of current approaches that do not consider this dynamicity. In addition, it again approves the idea of adaptive assignment of OT models to private patterns since new knowledge obtained by adversaries should be considered in the assignment procedure. So, a comprehensive PPM should also provide enough means for dynamic modeling of the adversary's background knowledge.

Therefore, the main problem this Chapter solves is obtaining data owners' privacy requirements in runtime in the form of patterns and trying to conceal them in a way that leads to maximizing the Quality of Service (QoS) (i.e., utility) while considering potential adversary's knowledge. Here, the comparison metric is calculated based on the number of truly detected matches of public patterns (i.e., TP_{PUB}) and False Positives (FP) in detecting public patterns (i.e., FP_{PUB}). In addition, regarding private patterns, the number of truly obfuscated matches of private patterns (i.e., TO_{PRIV}) and wrongly created new private patterns after applying obfuscation models (i.e., FR_{PRIV}) are being involved. Finally, the last considered metric is the number of matches for private patterns that can be derived using the adversary's background knowledge.

Each event type (e.g., TP) is assigned a weight (e.g., W_{TP}) to reflect its relative importance. This weighting scheme acknowledges that, in certain applications, the consequences of different event types vary significantly. For instance, accurately detecting a critical event (true positive) might be prioritized over avoiding false positives, or vice versa. Similarly, in privacy preservation, it may be more important to conceal sensitive patterns (private patterns) than to avoid inadvertently creating new ones. Consider a healthcare example: the timely detection of life-threatening conditions (public patterns) is vital, even if it means compromising some patient privacy (private patterns). The assigned weights allow for this trade-off to be explicitly incorporated into the system's decision-making. Furthermore, the significance of individual patterns within their respective sets (private or public) is not uniform. Therefore, we assign unique weights to each pattern: w_{to}, w_{fr} for private patterns, and w_{tp}, w_{fp} for public patterns. Additionally, the relative

importance of revealed private patterns compared to other event types (e.g., false positives) can be adjusted using W_{Adv}. Furthermore, the priority among private patterns within the adversary list can be controlled with w_p.

Considering these event groups, a *Privacy-Utility Trade-off* (PUT) function is formulated as an optimization problem, enabling the performance comparison of obfuscation models. We extend the objective function formulation in [16] i) to deal with the dynamicity of matched private and public patterns, ii) and quantify the adversary's background knowledge. More formally, the resulting formula is:

$$\begin{aligned} Max \quad & \Bigg\{ \Big(W_{TP} \sum_{tp \in TP_{PUB}} w_{tp} \Big) - \Big(W_{FP} \sum_{fp \in FP_{PUB}} w_{fp} \Big) \\ & + \Big(W_{TO} \sum_{to \in TO_{PRIV}} w_{to} \Big) - \Big(W_{FR} \sum_{fr \in FR_{PRIV}} w_{fr} \Big) \\ & - \Big(W_{Adv} \sum_{p \in PRIV} w_p \Big) \Bigg\} \end{aligned} \tag{4.1}$$

This objective function can generally be used in any application context. In other words, the last part is calculated by estimating the number of already obfuscated private pattern matches, which can be revealed based on the adversary's background knowledge. However, if such knowledge is not sufficiently available, the last part of this objective function can be skipped to make a fair comparison.

4.2 The APP-CEP System Design

We consider a typical DCEP system with multiple *producers* (e.g., IoT sensors) that advertise the simple event streams they can provide. Each of these streams is related to one or more *Data Owners* (DO), meaning the provided data belongs to this data owner(s) from a privacy protection point of view. For example, the location event stream of a car belongs to its driver. Correspondingly, *consumers* (e.g., applications, services, etc.) submit their situations of interest as continuous queries to the system. Hence, the set $Q = \{q_1, ..., q_n\}$ denotes the set of running queries that must be responded to by processing the events of current input streams. In addition, a set of *brokers* (e.g., CEP

engines) perform event processing tasks (e.g., drop, reorder, union, tamper, etc.)by hosting corresponding *operators* and delivering the resulting stream(s) to the next step.

Notation	Description
C	Set of event consumers ($c \in C$)
B	Set of event brokers ($b \in B$)
Q	Set of issued queries ($q \in Q$)
Ω	Set of operators ($\omega \in \Omega$)
$\mathcal{O}$	Set of available obfuscation models ($o \in \mathcal{O}$)
$\mathcal{O}^{pr}$	Set of available obfuscation models applicable to private pattern pr
DO_i	The ith data owner
$PR(DO_i)$	Privacy requirements of ith data owner
(pr_i, w_i)	The pair of ith privacy requirement and its corresponding weight
$\mathcal{PU}$	Set of public patterns ($pu \in \mathcal{PU}$)
$\mathcal{PR}$	Set of private patterns ($pr \in \mathcal{PR}$)
$\mathcal{D}$	Set of event dependencies ($d \in \mathcal{D}$)
$\rightarrow$	Causal dependency between events
$\|\|$	Parallel occurrence of two or more events
$\neg$	No occurrence of an event
$\bar{X}$	Periodic occurrence of an event
$\nrightarrow$	Infeasible causal dependency between events
$\mathcal{G}$	Set of pattern dependency graphs ($G \in \mathcal{G}$)
g_o	The pattern dependency subgraph for the obfuscation model o
$\mathcal{F}$	Set of input stream features ($f \in \mathcal{F}$)
$deg^+_{g_o}(pr)$	The number of outgoing edges from private pattern pr in pattern dependency subgraph g_o
α	Mapping solution of obfuscation models to private patterns

Table 4.1: Notations and their meaning.

A query q_i determines the logic to detect complex events by applying standard CEP operators over simple events' attributes (e.g., pattern matching). To this end, each detection logic needs to be hosted by a specific operator

to be executed. Moreover, each data owner is able to describe its privacy requirements and deliver them to the system. Such requirements should be accompanied by a normalized weight that shows their importance. Hence, the set $PR(DO_i) = \{(pr_1, w_{pr_1}), ..., (pr_m, w_{pr_m})\}$ shows the privacy requirements specified by data owner DO_i along with their corresponding weights. As another query feature, the importance of queries might vary due to their priorities. That is why detecting such query patterns is of more significance to the system instead of concealing privacy requirements, e.g., life-related queries in a healthcare scenario. Table 4.1 presents the list of notations and their description used in our mechanism.

Producers, also called *event sources*, are the origin of the events that generate simple event streams. As an example, if we implement our mechanism in an IoT scenario, the sensors that measure a specific phenomenon (e.g., location of a target) act as event sources. Regarding the number of targets that can be covered, we categorized the event sources into two groups: *DO-Specific* and *Multi-DO*. For the former, the event source can only be used for queries related to the corresponding data owner. In other words, the generated stream will not reveal privacy-sensitive information about other targets (i.e., data owner). For instance, a cellphone's GPS signal can only be used for tracking a specific target. For the latter, information about multiple targets can be acquired by analyzing such event source streams. For instance, a stationary security camera delivers data related to various targets. These event sources should be used cautiously in any privacy-aware data analysis approaches.

In addition, the event sources here are interconnected to the system over a wireless network and must be registered in the system due to privacy concerns. Besides, each data owner can act as a CEP consumer by submitting queries. CEP operators can also be hosted in nodes with sufficient computing capabilities, e.g., on the cloud or fog node in an IoT scenario.

Definition 5 *Obfuscation Model.*
An obfuscation model is a real-time technique for modification in the order, occurrences, or values of event attributes that results in concealing sensitive information (i.e., private patterns). Such a modification can be applied to event streams, executed prior to the CEP middleware. More importantly, the obfuscation model should be placed on the earliest available, trustworthy resources to the producers, preventing inconsistencies in the delivered streams.

Among common obfuscation models, we consider a PPM is able to hire the following techniques:

- **Suppression:** Removing the occurrence of events by dropping them from the event stream, e.g., removing the event $HomePage\_Visit^{c_1}$ to conceal page navigation behavior of customer c_1.
- **Reordering:** Changing the order of two events by swapping their timestamps, e.g., changing the occurrence time of $Buy_i^{c_1}$ with the event $Buy_j^{c_1}$ to conceal the order of purchases for a specific customer c_1.
- **Injection:** Introducing fake events and inserting them in unique places in the stream, e.g., injecting the event $Buy_j^{c_1}$ multiple times to conceal a specific interest to item i. It shows the person c_1 interested in both items, i.e., i and j.
- **Tampering:** Changing attributes of an event to a wrong value or noise injection in event timestamps, e.g., changing the value of item type i in $Add\_To\_Basket_i^{c_1}$ to j which produces $Add\_To\_Basket_j^{c_1}$ in order to conceal unhealthily shopping habit of a diabetic customer from an insurance company.
- **Generalization:** Anonymizing the event's attributes by changing to a general value, e.g., generalizing an event $Buy_{apple}^{c_2}$ to $Buy_{fruit}^{c_2}$ to conceal a customer's shopping interests.
- **Hybrid:** A combination of thereof, e.g., suppression of event $Buy_{medicine}^{c_2}$ and injecting event $Buy_{drink}^{c_2}$ to conceal a customer's decease.

In Figure 4.2, we illustrate the main functionalities of APP-CEP. Parts of our system are built on existing concepts for converting the privacy requirements and situations of interest in natural language to a CEP-like pattern representation [208]. Therefore, we assume that data owners and users are able to feed the system with public and private patterns. The main goal of APP-CEP is to gather related information about ① patterns' dependencies, ② event dependency set, as well as ③ input stream features to opt for ④ obfuscation models which satisfy PUT, even without knowledge about the actual events in the input streams and adapt the selection based on the obfuscation results in execution time. This section will further elaborate on the functionality of these four components of APP-CEP.

Here, there are two dynamic databases, namely *stream database* and *obfuscation database*, which assist APP-CEP in the obfuscation procedure by providing up-to-date information. In the former, all previous events are collected (e.g., the transaction history of the webshop) for further analysis, such as extracting event dependencies and stream features. In the latter, all available models of applying obfuscation models (e.g., dropping the first event in a private pattern) are gathered to support performance analysis of the deployed obfuscation model and make adaptation decisions. Moreover, we presume a data owner has sufficient skills to support APP-CEP by dynamically providing event dependencies to facilitate the obfuscation procedure in the execution time. For example, a webshop customer can provide information about his preferred pick-up point for purchases that will be delivered in the working days since such information should be considered while obfuscating the patterns in location tracking applications.

In preliminary approaches on multiple pattern-type privacy protection [16] , a PPM is only able to calculate a utility value by inspecting the actual events in the input streams. Therefore, assigning the best obfuscation model for each private pattern is achievable by comparing the utility values. However, a significant disadvantage of such strategies is the complexity of designing a low-overhead adaptation procedure to maintain the privacy protection performance at an acceptable level concerning the system's dynamics, e.g., changing data owners' privacy requirements. To appropriately respond to dynamics, a greedy idea would be to switch between obfuscation models whenever a model

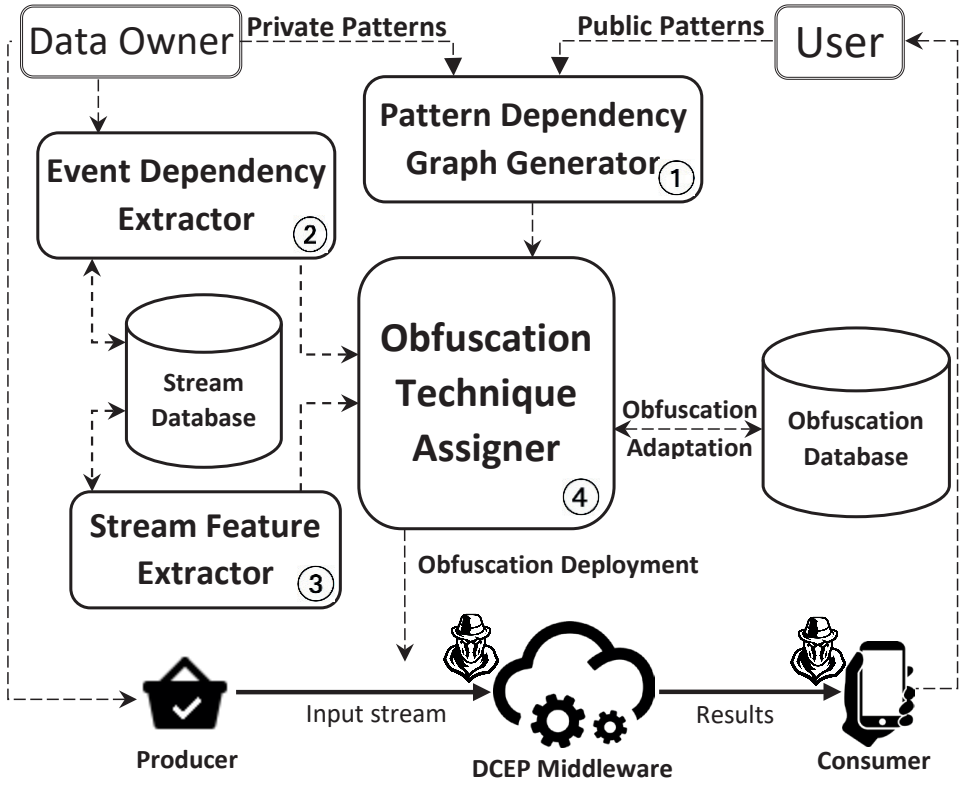


Figure 4.2: The APP-CEP proposed system design. The solid lines indicate the data/event flow, while the dashed lines indicate the control flow.

with a higher utility value is provided by the corresponding component (i.e., obfuscation model Assigner). The main pitfalls with this approach are, firstly, time and resource overhead for transitions and, secondly, oscillation between obfuscation models that downgrade the obfuscation performance.

To overcome the mentioned challenges, one could realize that relying only on the actual events appearing in the input streams causes transition overheads and oscillation issues. In other words, if a PPM is capable of predicting the obfuscation models' transition time, the data/operator migration can be performed ahead of time in order to minimize inefficiency in both time and resource utilization. Such prediction can be achieved by realizing the

correlation between patterns as well as events. Moreover, extracting input stream features will enable a PPM to prevent unnecessary obfuscation transitions, leading to performance efficiency. For example, the number of pattern matches can roughly be estimated by extracting the event distribution in each event stream. This will prevent oscillation in obfuscation assignment, which might be produced by a transition from dropping to reordering and quickly returning to dropping.

Nevertheless, generating more up-to-date insights about patterns and also events helps APP-CEP to wisely adapt the system to maximize the overall performance of the obfuscation procedure.

4.2.1 Pattern Dependency Graph

In an event-based system, the privacy requirements can be fulfilled by modifying the original event streams so that private patterns cannot be detected anymore. Such a modification might influence the detection of other public or private patterns that must be detected on the same streams. We call such impacts *pattern dependencies*, which can be defined as follows.

Definition 6 *Pattern Dependency (d).*
The positive or negative impacts of obfuscating a special private pattern u on the matches of pattern v are calculated based on common events in their corresponding patterns and formally defined as follows:

$$u \xrightarrow{d} v : \exists\, e \in u \mid (e \in v) \vee (\mathit{generalization}(e) \in v) \tag{4.2}$$

Predicting the obfuscation transition time is possible by determining the potential dependencies between private and public patterns.

Definition 7 *Pattern Dependency Graph (G).*
This graph illustrates the impacts of modifying streams to obfuscate any private pattern (i.e., data owners' privacy requirements) on detecting matches for public patterns (i.e., user queries). The dependency graph is defined as follows:

$$G(V,E) = \begin{cases} V = \{v \mid v \in \mathcal{PU} \cup \mathcal{PR}\} \\ E = \{(u,v) \mid u \in \mathcal{PR} \,\&\, v \in V \,\&\, u \xrightarrow{d} v\} \end{cases} \tag{4.3}$$

In Definition 7, a graph is produced by sets of vertices V and edges E. Each vertex v is a pattern of whether belongs to $\mathcal{PU}$ or $\mathcal{PR}$. On the other hand, an edge (u,v) in E refers to a unidirectional dependency d between a private pattern u and a pattern v that can be public or private. Such dependency is illustrated by $(\xrightarrow{d})$, which shows that applying at least one of the available obfuscation models on u will impact the detection of matches for v. Such influence might be positive (i.e., help obfuscate other private patterns) or negative (i.e., producing FP and FN for other public or private patterns).

Consider the following public and private patterns to understand the pattern dependency graph generation process better.

$$\begin{aligned} \text{PUBLIC} = \{\ & \{pu_1 \mid Buy^{p_i}_{CD} \rightarrow Return^{p_i}_{CD'}\}, \\ & \{pu_2 \mid Buy^{*}_{BC} \rightarrow Buy^{*}_{BC} \rightarrow Buy^{*}_{BC}\}, \\ & \{pu_3 \mid Buy^{p_i}_{AD} \parallel Buy^{p_i}_{FJ}\}\ \} \\ \text{PRIVATE} = \{\ & \{pr_1 \mid Buy^{p_1}_{IC} \parallel Buy^{p_1}_{CR} \parallel Buy^{p_1}_{CD}\}, \\ & \{pr_2 \mid Buy^{p_1}_{IC} \parallel Buy^{p_1}_{BC}\}, \\ & \{pr_3 \mid Buy^{p_1}_{AD} \rightarrow Buy^{p_1}_{DM}\}\ \} \end{aligned} \tag{4.4}$$

These sample pattern sets (i.e., public and private sets) indicate the following event correlations:

- pu_1: A causal relationship between buying and returning two types of Christmas decorations (CD) with the same customer ID, which shows the customers' interests.

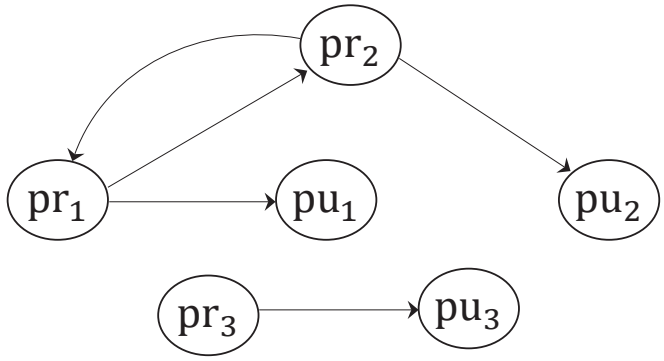


Figure 4.3: The global pattern dependency graph generated for pattern sets defined in Equation 4.4.

- pu_2: Multiple purchases of the same product (i.e., birthday candle (BC)) by different customers, which helps recommendation systems.
- pu_3: The concurrent ordering of alcoholic drinks (AD) and fruit juice (FJ) with the same customer ID, which detects customers' drinking interests.
- pr_1: Simultaneous purchasing of invitation cards (IC), Christmas ribbons (CR), and Christmas decorations, which reveals a Christmas party.
- pr_2: The concurrent purchase of invitation cards and birthday candles, which reveals a birthday party.
- pr_3: A causal ordering dependency of alcoholic drinks and diabetic medicine (DM) reveals a bad lifestyle for a diabetic person.

The respective pattern dependency graph for the mentioned patterns in Equation 4.4 is depicted in Figure 4.3. Here, the submitted privacy requirements belonging to an individual customer (i.e., p_1) have been used to define private patterns. Besides, queries represented as public patterns are generally defined, not for a specific customer ID. Signs ($\rightarrow$) and ($||$) indicate the causal dependency in a sequence pattern and parallel occurrence of events in a conjunction pattern, respectively.

To construct a pattern dependency graph, we assume that the graph's nodes are patterns. For each private pattern, outgoing edges are unidirectional arrows from this private pattern to all dependent public/private patterns. A pattern is called the *neighbor* of private pattern pr if an edge exists going from pr to this specific node. For instance, if we plan to obfuscate private pattern pr_1, suppression of event $Buy_{IC}^{p_1}$ helps to obfuscate private pattern pr_1 (positive impact). However, suppression of the event $Buy_{CD}^{p_1}$ will result in false negatives in the detection of public pattern pu_1 (negative impact).

The graph representation of pattern dependencies supports deriving insights that can be helpful in obfuscation model selection. For example, the outgoing edges of each node indicate the detection of how many patterns will be influenced. In a particular case, disconnected nodes (i.e., nodes with no neighbor) do not have any relationship with other nodes. Hence, the obfuscation selection is much easier and even static in such cases until their connectivity changes in the updated pattern dependency graph. Any available obfuscation model can be chosen in such situations since they all support the PUT goal.

Definition 8 *ZOD node.*
In a pattern dependency sub-graph g_o, the out-degree of a node (i.e., deg^+) is calculated by counting the number of its outgoing edges. A zero Out-Degree (ZOD) node represents a private pattern whose obfuscation procedure does not influence the detection of other patterns; its outgoing degree equals zero, formally defined as follows.

$$ZOD(g_o) = \{u \mid u \in \mathcal{PR} \cap V(g_o) \ \&\ deg^+(u) = 0\} \tag{4.5}$$

On the other side, if a private pattern has one or more outgoing edges in the global dependency graph, an alternative solution would be to break down the global dependency graph into obfuscation-specific sub-graphs (e.g., suppression(e_1) sub-graph, which drops the first event of all matches). This way, APP-CEP can find a sub-graph in which this specific node has no neighbors. For instance, Figure 4.4 shows the generated sub-graphs of four different obfuscation models. Here, nodes pr_1 and pr_3 are disconnected nodes in the Suppression(e_2), and pr_2 has no neighbors in the Suppression(e_1). Therefore,

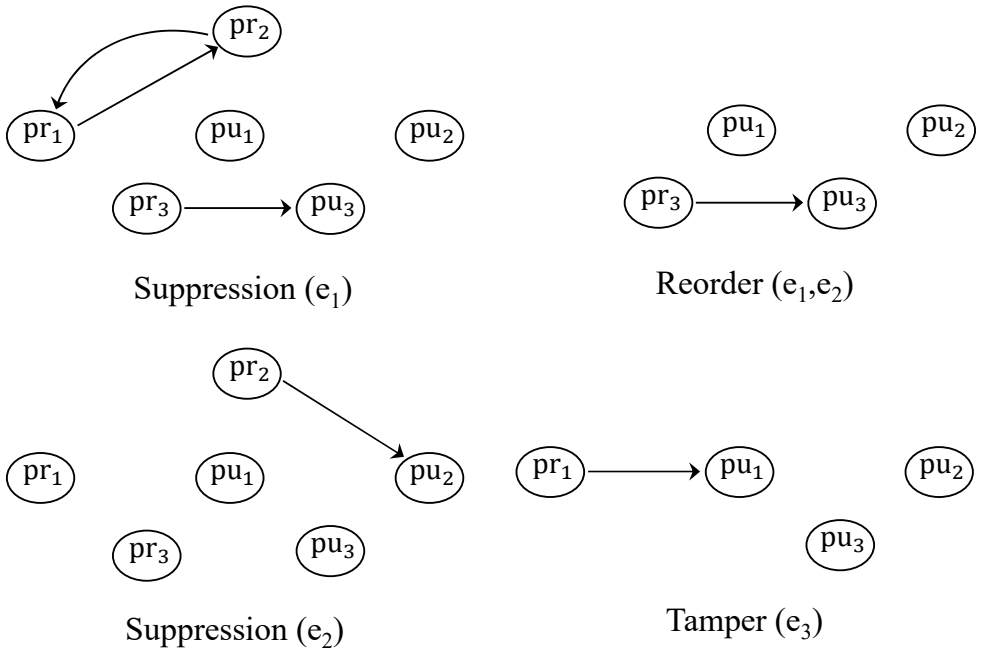


Figure 4.4: The alternative pattern dependency sub-graphs for the global dependency graph illustrated in Figure 4.3.

these obfuscation models are potential candidates for corresponding private patterns. Note that Reorder(e_1, e_2) is not among available obfuscation models for pr_1 and pr_3 because the conjunction operator is used in the definition of those patterns between the first and second events, and such obfuscation model is not able to conceal these patterns. Similarly, Tamper(e_3) is not applicable for patterns of length 2, e.g., pr_2 and pr_3.

4.2.2 Event Dependencies

Modeling the adversaries' background knowledge is a complicated task that has yet to be well-studied in this field. Some research works have been conducted to statistically predict special privacy attacks to model the adversary [16]. For example, an adversary might infer several event types' true mean inter-arrival time from stream history. Then, he can realize enforced

pattern obfuscation by computing statistical metrics like the suppression probability of this event type and reconstructing the original input stream. However, the potential data owners' information is not exploited to complement the final adversary model.

To this end, APP-CEP supports the runtime expression of event dependencies. This facilitates modeling the potential information that an adversary might use to derive insights related to data owners. For instance, the regular shopping habits of a customer, e.g., buying two products always together, give a hint to the system not to drop one while publishing the other. In more detail, four types of event dependencies can be introduced to be considered in the obfuscation assignment phase.

Causal Dependency (→)

In this category, a relationship between two events can be specified in two ways: firstly, one event is the reason for occurrences of the other (i.e., cause and effect), and secondly, one event always happens before the other. For example, in our use-case scenario, a causal dependency can be derived from history as: $CD\ (Submit\_Order_i^{c_1}\ ,\ Payment\_Successful_i^{c_1})$
which means the event of submitting an order is the cause, and the successful payment event is the effect. Also, a customer c_1 might express that he always checks the available balance before adding a product p to the shopping basket. Therefore, the customer c_1 can provide a causal dependency to the system as: $CD\ (Check\_Balance^{c_1}\ ,\ Add\_To\_Basket_i^{c_1})$

In these cases, APP-CEP considers the dependency, e.g., does not suppress the cause event and keeps the effect event or reorders these two events.

Parallel Occurrence (||)

Special events (two or more) happen always, or at least with a high probability, simultaneously or in a short period. This means an intelligent obfuscation assignment should consider them as a *single combined event*, e.g., reorder all events instead of one. As an example in the proposed webshop scenario, a customer c_1 might express that he always buys products p and q together. This leads to a parallel occurrence of the corresponding two events, represented as:

$PO\ (Add\_To\_Basket_p^{c_1}\ ,\ Add\_To\_Basket_q^{c_1})$.

Infeasible Events ($\nrightarrow$)

Another type of event relationship also shows the impossible occurrences of one or several events or patterns. For instance, when a customer c_1 is navigated to the bank payment page as a result of the $Submit\_Order$ event with id i, it is not possible to observe two events of $Payment\_Successful$ and $Payment\_Rejected$ afterward for this special order submission in the stream, that can be expressed by:

$IE\ [CD\ (Submit\_Order_i^{c_1}\ ,\ Payment\_Successful_i^{c_1}),$
$CD\ (Submit\_Order_i^{c_1}\ ,\ Payment\_Rejected_i^{c_1})]$

Such dependencies usually cannot be expressed by a data owner, but require help from domain experts in the application scenario.

Periodic Events ($\bar{X}$)

Apart from dependencies between events, this type of information shows insights can be derived as a result of statistical calculation over a specific event type for a unique data owner. We assume that the adversary has the required computing resources and access to the data history to correlate the statistical dependencies for a particular data owner. For example, a customer c_1 might have a shopping habit of buying a special product p each day. Hence, such a habit can be expressed by:

$PE\ (Buy_p^{c_1}\ ,\ day\ )$

Therefore, applying any obfuscation model related to dropping or altering such events (e.g., suppression, tampering, generalization, etc.) will lead to violating the privacy of data owner by revealing the stream modification on the adversary side, whereas obfuscation models like reordering still might be applicable in such situations.

4.2.3 Stream Features

Even if we know the privacy requirements of data owners and the dependencies between patterns and events, the input streams also bring up another dynamicity that should be considered. In more detail, the number of matches

for a pattern determines another vital dimension. For example, a Christmas party is more likely to be revealed during Christmas. Therefore, obfuscating a private pattern that increases the reveal of any Christmas party at another time of the year (e.g., in July) would be acceptable because the detection probability of such a party is very low in other periods.

Hence, extracting input stream features helps a PPM to assign the obfuscation model more efficiently. To do so, APP-CEP investigates the stream database and analyzes the input streams online to adapt the features. As mentioned earlier, one significant feature of streams is the average number of matches for each pattern. For instance, the number of matches for a Christmas party pattern is calculated weekly, which helps dynamically assign weight to this private pattern in the PUT function.

Apart from patterns, event-related stream features offer additional benefits by enabling more effective obfuscation model selection. For instance, the event distribution helps estimate a more accurate weight for each suppression model of a private pattern [24]. If the event e on a private pattern pr has a higher frequency in history, suppression of e tends to create more FN in detecting neighbors of pr which contain e. Such an idea can also be extended to subsets of each private pattern instead of a single event by checking its distribution for other types of obfuscation (e.g., a two-event subset for reordering).

4.2.4 Obfuscation Technique Assignment

In connection with the obfuscation database, the *obfuscation technique assigner* module, which is the core component of APP-CEP, acts as the system's brain. It aggregates the information collected from the other three modules (i.e., ① pattern dependency graph generator, ② event dependency extractor, and ③ stream feature extractor) to decide which obfuscation model should be chosen regardless of the current input streams for each private pattern.

In Algorithm 4.1, the sets of public and private patterns are initialized, and the event dependencies derived from streams' history alongside the available obfuscation operators are delivered as inputs (line 1). Upon receiving new public or private patterns, APP-CEP updates the obfuscation technique assignment (lines 2-7). Here, we also accept the event dependencies the data

Algorithm 4.1 Obfuscation Technique Assignment

Initialization:
  $\mathcal{PU}$, $\mathcal{PR} \leftarrow \varnothing$,
  $\mathcal{D} \leftarrow$ Event Dependencies,
  $\mathcal{O} \leftarrow$ Obfuscation Techniques;
upon ($Submit(Q_i)$) **do**
  $\mathcal{PU} \leftarrow \mathcal{PU} \cup$ CEP-Pattern(Q_i); ▷ Query
  Obfuscation_Assigner ($\mathcal{PU}$, $\mathcal{PR}$, $\mathcal{D}$);
upon ($Submit(Pr_i)$) **do**
  $\mathcal{PR} \leftarrow \mathcal{PR} \cup$ CEP-Pattern(Pr_i); ▷ Priv Req
  Obfuscation_Assigner ($\mathcal{PU}$, $\mathcal{PR}$, $\mathcal{D}$);
upon ($Submit(d_i)$) **do**
  $\mathcal{D} \leftarrow \mathcal{D} \cup d_i$; ▷ Event Dependency
  Obfuscation_Assigner ($\mathcal{PU}$, $\mathcal{PR}$, $\mathcal{D}$);
function OBFUSCATION_ASSIGNER($\mathcal{PU}$,$\mathcal{PR}$,$\mathcal{D}$)
  $\mathcal{G} \leftarrow$ Pattern Dependency Graphs;
  $\mathcal{F} \leftarrow$ Input Stream Features;
  for $pr \in \mathcal{PR}$ **do**
    $\mathcal{O}^{pr} \leftarrow \mathcal{O}$;
    for $o \in \mathcal{O}$ **do**
      if o violates $\mathcal{D}|\mathcal{F}$ **then**
        $\mathcal{O}^{pr} \leftarrow \mathcal{O}^{pr} - o$;
    $\widehat{\mathcal{O}}^{pr} \leftarrow \varnothing$;
    for $o \in \mathcal{O}^{pr}$ **do**
      if $deg^{+}_{g_o}(pr) = 0$ **then**
        $\widehat{\mathcal{O}}^{pr} \leftarrow \widehat{\mathcal{O}}^{pr} \cup o$;
    if $\widehat{\mathcal{O}}^{pr} \neq \varnothing$ **then**
      $o_{pr} \leftarrow \widehat{\mathcal{O}}^{pr}_1$;
    else
      $o_{pr} \leftarrow Random(\mathcal{O}^{pr})$;
  $\alpha \leftarrow \{(pr_1, o_{pr_1}), ..., (pr_n, o_{pr_n})\}$; ▷ Solution

owner enters at runtime (lines 8-10). The *OT_Assigner* function generates pattern dependency graphs for each obfuscation model and updates them on runtime based on changes in both public and private sets. Also, it extracts input stream features based on history and updates those features using the current streams (lines 12-13).

According to the event dependency set and stream features, for each private pattern pr, APP-CEP performs the following tasks: first, it removes those obfuscation models whose obfuscation can be realized by an adversary (lines 16-18). For example, if the result of reordering violates a causal dependency, such an obfuscation model will be removed from the list of available obfuscation models. In the remaining obfuscation model space (i.e., $\mathcal{O}^{pr}$), APP-CEP looks for a member whose dependency graph has a node with an out-degree equal to zero for pr, so-called *zero-out-degree* (ZOD) node (lines 20-22). One of the obfuscation models containing such a node will be assigned to the pr. If no ZOD node has been found for this private pattern, a random obfuscation model will be assigned to a pr (lines 23-26). Finally, the assigned obfuscation model for each private pattern is determined and will be updated upon any change in patterns or event dependencies (line 27).

More formally, the solution α can be represented as follows.

$$\alpha = \left\{ (pr, \Phi(pr)) \mid pr \in \mathcal{PR} \,\&\, o \in \mathcal{O} \,\&\, n = |\mathcal{PR}| \right\} \tag{4.6}$$

and,

$$\Phi(pr_i) = \begin{cases} o & \text{if } \exists\, o \in \mathcal{O} : \{\, pr_i \in ZOD(g_o) \,\& \\ & \nexists\, d \in \mathcal{D} : o \text{ violates } d \,\& \\ & \nexists\, f \in \mathcal{F} : o \text{ violates } f \,\} \\ Random(\mathcal{O}^{pr_i}) & \text{otherwise} \end{cases} \tag{4.7}$$

Although APP-CEP is a restrictive approach since the idea of selecting a ZOD node does not achieve the maximum QoS, finding these nodes guarantees both obfuscating each private pattern and trying to detect as many public patterns as possible. In addition, calculating the exact PUT value requires

considering the exact number of pattern matches in each input stream, whose disadvantages have been discussed earlier. If no ZOD node is found, we believe assigning the best obfuscation model is outside the scope of this study. Therefore, we randomly assign the obfuscation model in such a worst-case scenario.

4.3 Evaluation

The evaluation investigates the following questions:

1. Can graph-based pattern-level privacy provide a trade-off between privacy concerns and the system's promised QoS?

2. What limitations are involved in applying pattern-level privacy in real-world scenarios?

We established a single Virtual Machine (VM) running Ubuntu 20.04 on Oracle VM Virtual Box Manager, allocating 6 CPU cores and 24 GB of RAM. For event detection, we leveraged the *FlinkCEP* library [184], built on Apache Flink. To assess our approach, we analyzed two publicly available real-world datasets [1].

1. The first dataset is composed of transactions of an online retailer selling all-occasion gifts [209]. A part of the first dataset, including 12000 customer purchase records, was selected for this evaluation. We selected the 50 most frequent patterns (length 3) from which public and private pattern sets were randomly formed. We also performed another statistical analysis to extract the most parallel purchased items and periodically purchased items as dependencies considered for the adversary's background knowledge.

2. The second one is a medical dataset with information concerning hospital records of patients diagnosed with diabetes [210]. The dataset represents ten years (1999–2008) of clinical care at 130 US hospitals and integrated delivery networks. Each row concerns hospital records

[1] https://github.com/majid-lotfian/APP-CEP

of patients diagnosed with diabetes, who underwent laboratory tests, received medications, and stayed up to 14 days. We formed 20 patterns for public and 10 for private patterns by consulting with a medical expert and randomly selecting patterns from them for evaluation that are partly represented in Table 4.2. We also defined 8 event dependencies (two for each dependency type).

We executed the event detection process ten times for each combination of public and private patterns and stated the average results.

Given the novelty of modeling adversarial background knowledge and graph-based obfuscation assignment, we selected two baseline approaches for comparison.

- **Window-based:** This approach simulates existing state-of-the-art mechanisms (e.g., [16]) by dividing the input stream into fixed-size windows. It selects an obfuscation technique for each private pattern match based on its expected utility, considering the potential matches for all patterns after applying the technique.
- **Graph-based:** Similar to APP-CEP, this approach selects obfuscation techniques based on pattern dependencies. However, it does not factor in adversarial background knowledge, relying solely on the pattern dependency graph or OT-specific sub-graphs.

4.3.1 Evaluation Results for Webshop Dataset

Figure 4.5 indicates the total percentage of revealed private pattern matches (i.e., the summation of matches for data owners' privacy requirements added to the number of matches revealed by the adversary's background knowledge, totally divided by the actual matches) while the number of public patterns is increasing. Error bars also indicate the standard deviation of ten rounds of simulations for each approach. It can be seen that modeling and considering the adversary's background knowledge helps APP-CEP to significantly reveal less (i.e., conceal more) private patterns than other approaches. By adding more queries to the system, Graph-based and Window-based techniques fluctuate, but APP-CEP's results stay around ten percent. The random assignment of obfuscation techniques when APP-CEP cannot find a ZOD node in

Table 4.2: Partial list of public and private patterns used to analyze medical dataset: public patterns representing useful inquiries in the medical system, and private patterns highlighting situations that have consequences such as increased insurance premiums for patients.

#	Public/Private	Pattern Definition
1	Public	Outpatient Visit followed by Laboratory Test within 30 days.
2	Public	Outpatient Visit followed by Medication Prescription within 30 days.
3	Public	Emergency Visit followed by an Outpatient Visit within 30 days.
4	Public	A hospital admission followed by diagnostic imaging (CT, MRI) within the same stay.
5	Public	A medication prescription followed by a refill within 90 days.
6	Private	An outpatient visit followed by any kind of readmission within 30 days.
7	Private	An emergency visit followed by a diagnosis of a chronic disease within 30 days.
8	Private	An emergency visit followed by ICU admission within 30 days.
9	Private	Admission for surgery followed by readmission within 30 days post-discharge.
10	Private	An outpatient visit followed by a prescription of high-cost medication within 30 days.

the global dependency graph or obfuscation-specific sub-graphs shows its impacts in the last three bars. That is why complementing our algorithm with an intelligent solution for such situations would be challenging but necessary for our future work to solve the potential scalability issue.

Like FN and FP, *F1-score* is a well-known performance measure in machine learning approaches. It combines the other two measures, *precision* and

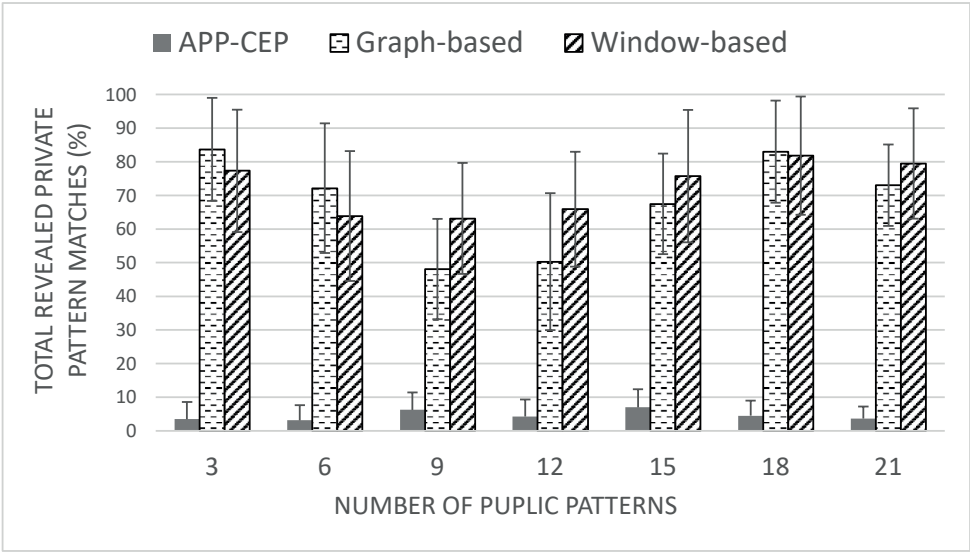


Figure 4.5: Percentage of total revealed private pattern matches Vs. the increasing number of public patterns for a set of 3 private patterns for the webshop dataset.

recall, which are mainly employed to distinguish between classifiers [194] in terms of accuracy. Therefore, the F1-score can be used in stream processing to compare mechanisms in terms of accuracy. From the public pattern detection's accuracy point of view, by increasing the number of involved queries, all three approaches fluctuate but gradually degrade; thereby, there is no specific front-runner, illustrated in Figure 4.6. However, APP-CEP's detection accuracy is mostly equal to or better than the Window-based approach when increasing the number of involved queries (i.e., public patterns), which confirms a slight improvement over the literature, even in this challenging comparison.

The effects of increasing the knowledge of the adversary are depicted in Figure 4.7. It can be seen that APP-CEP reveals significantly fewer matches than the other two approaches. By involving more dependencies in query processing, the percentage of expected revealed private pattern matches unexpectedly rises slightly but is still negligible compared to Graph-based and

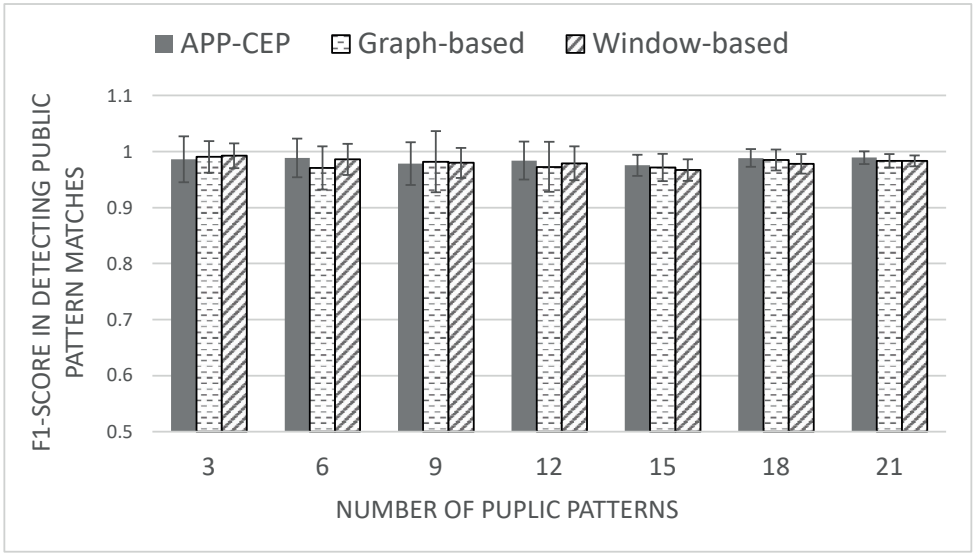


Figure 4.6: F1-score in detecting public pattern matches Vs. the increasing number of public patterns for a set of 3 private patterns for the webshop dataset.

Window-based approaches. Here, the graph-based approach reveals a few more private pattern matches than the window-based approach, but generally, both approaches show similar performance. Adding more dependencies corresponds to a more knowledgeable adversary that even impacts the performance of APP-CEP. Protecting against such an adversary is almost impossible without sacrificing utility in the system. However, the proposed approach shows such a trade-off can be achievable even independently of the actual events in the stream. Moreover, note that it becomes hard to compensate for the revealed matches without modeling the adversary's additional background knowledge.

Last, Figure 4.8 accumulates the previous results to compare the privacy-utility trade-off of the three approaches. As discussed earlier, the main goal of this Chapter is to provide a method to maximize the trade-off value (cf. PUT equation). The computation of such a value requires the system designer to opt for an optimized weights value. We decided to set equal weights

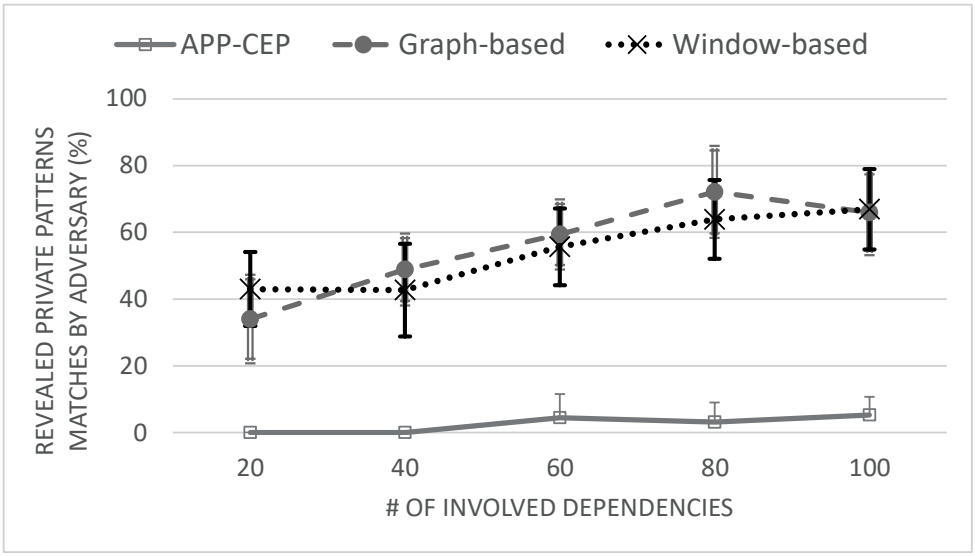


Figure 4.7: Percentage of revealed private pattern matches by adversary Vs. the increasing adversary's background knowledge (number of dependencies)

for revealed private pattern matches and expected revealed matches by the adversary. This weight is the ratio of the average of actual public matches to the average of actual private matches for each pattern set combination. For instance, the weight for the scenario in which we have 3 public patterns and 3 private patterns was calculated as 1.0301. Also, we set the weight for the detected public pattern matches equal to 1.

Here, the results indicate that APP-CEP is capable of achieving the expected PUT even by scaling up the system's involved queries. Although the Window-based approach had a good start, the bars fall once more public pattern matches are going to be detected, as the amount of revealed private matches by the adversary significantly impacts the PUT value. The graph-based approach seemed to perform better than the Window-based approach. However, its results also had a downward ending for the same reason.

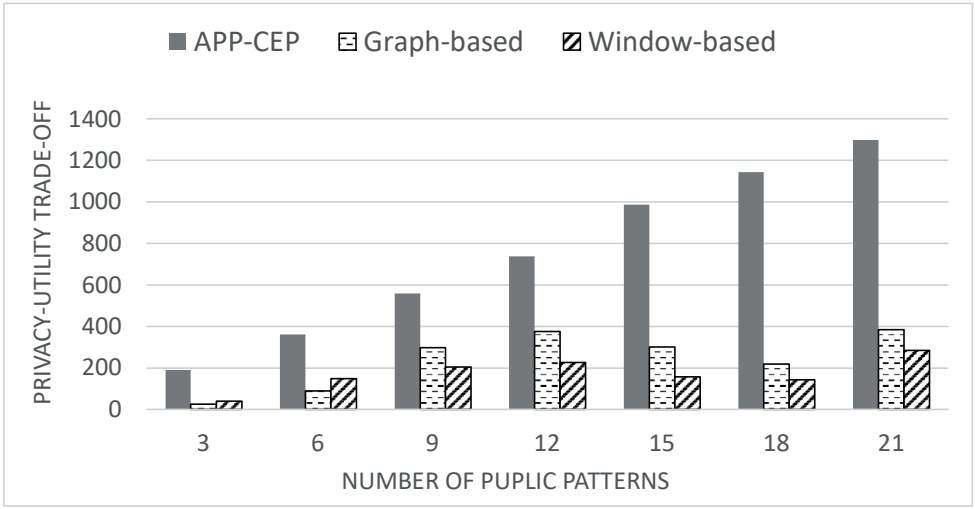


Figure 4.8: Privacy-Utility trade-off Vs. the increasing number of public patterns for a set of 3 private patterns for the webshop dataset.

4.3.2 Evaluation Results for Medical Dataset

We observed that the webshop dataset primarily comprised of unrelated simple patterns which limited the complexity of the pattern dependency graphs. By contrast, the second dataset features patterns with more shared common events, resulting in denser and more intricate dependency structures. This shift in dataset complexity allows us to rigorously evaluate our method's robustness and scalability under challenging conditions. It should be noted that for the sake of simplicity, we limited the available obfuscation techniques to *Suppression*, *Reordering*, and *Tampering*.

Figure 4.9 shows the percentage of revealed private pattern matches, including those inferred from the adversary's background knowledge, as the number of public patterns grows. Error bars represent simulation variability. Our APP-CEP significantly outperforms other methods by minimizing private pattern disclosure. While Graph-based and Window-based approaches fluctuate with increasing queries (between 55% and 82%), APP-CEP maintains a consistently low disclosure rate (below 13%). The random obfuscation

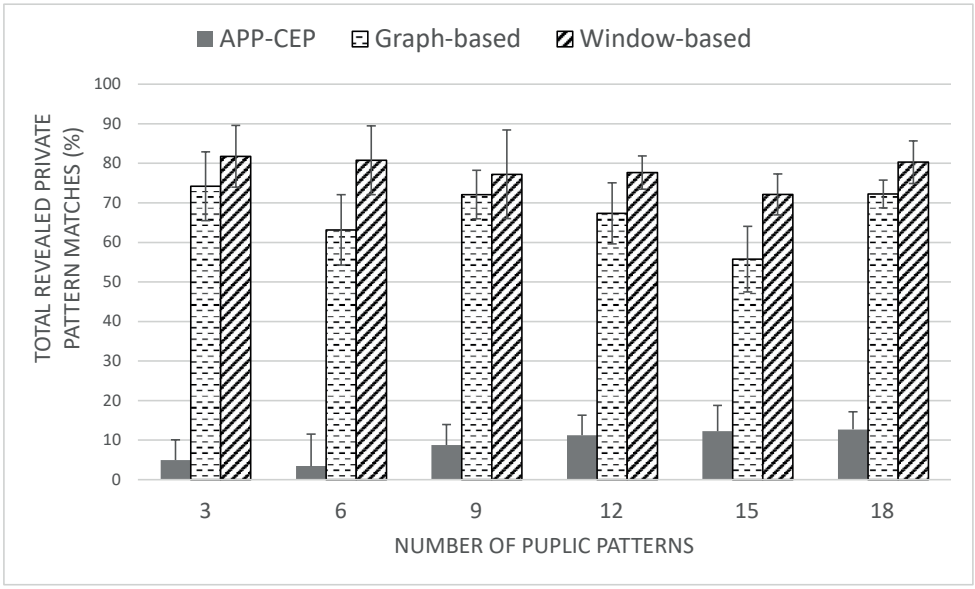


Figure 4.9: Percentage of total revealed private pattern matches Vs. the increasing number of public patterns for a set of 3 private patterns for the medical dataset.

strategy used when the optimal obfuscation method is unavailable impacts performance, as seen mostly in the last four bars. A direct comparison of Figure 4.9 with its counterpart for the first dataset (i.e., Figure 4.5) highlights the impact of increased pattern dependency complexity. While the medical dataset indeed produces denser dependency graphs, leading to a higher total rate of private pattern disclosure for our approach, it still maintains a significant advantage over the baseline strategies, namely Graph-based and Window-based. This indicates that our method's effectiveness in mitigating privacy leakage is robust even in the face of more challenging data conditions.

Figure 4.10 presents the accuracy rate, calculated as the ratio of true positive matches to the total number of matches, extracted as the ground truth. A notable limitation of our proposed method is a decreased accuracy rate due to the dense dependency graphs in the medical dataset. This reduction occurs because our approach prioritizes privacy preservation over maximizing

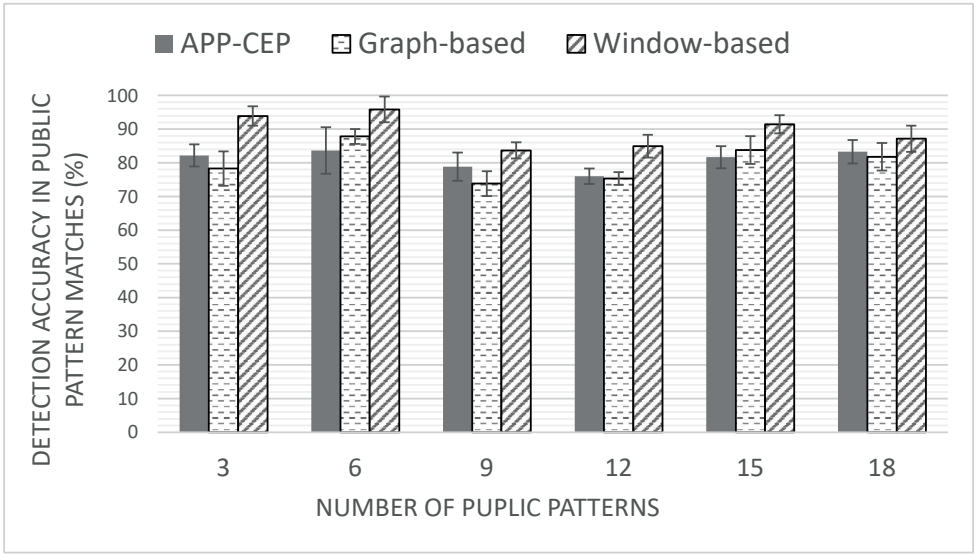


Figure 4.10: Accuracy of pattern detection for public patterns Vs. the increasing number of public patterns for a set of 3 private patterns for the medical dataset.

the detection of public patterns. Consequently, obfuscation models are selected primarily to protect sensitive information rather than optimize utility. In contrast, the Window-based approach demonstrates better performance in identifying public patterns, outperforming both graph-based alternatives.

Figure 4.11 provides a comprehensive evaluation of our method by employing the Privacy-Utility Trade-off (PUT) metric. Consistent with the findings from the first dataset, our approach demonstrates superior performance in balancing privacy and utility, surpassing both baseline strategies even with an increasing number of public patterns. While the PUT values of baseline methods exhibit fluctuations, our method consistently improves as more public patterns are incorporated. The notable decrease in PUT value for other methods when using 18 public patterns highlights the challenge of preserving privacy while maintaining utility in complex scenarios. This decline is primarily attributed to an increased rate of private pattern disclosure, emphasizing the importance of our method's privacy-centric approach. Although

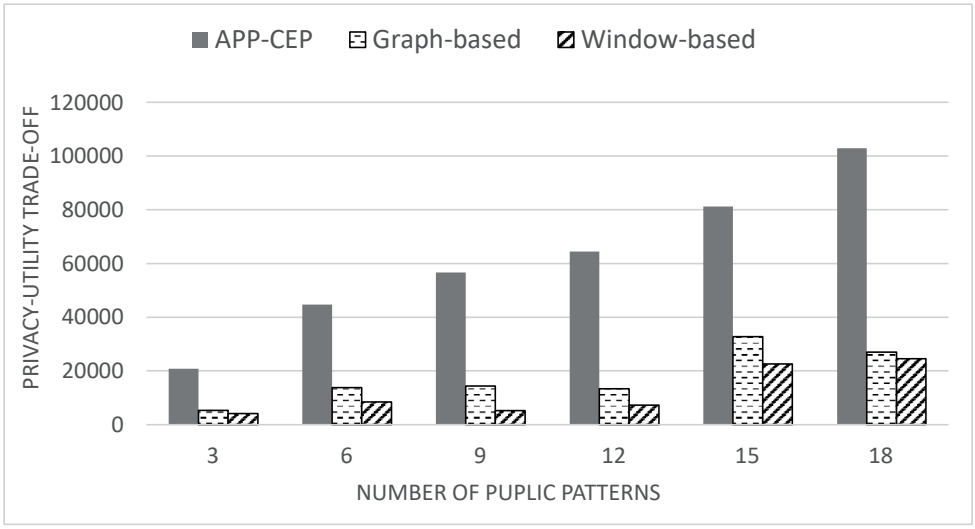


Figure 4.11: Privacy-Utility trade-off Vs. the increasing number of public patterns for a set of 3 private patterns for the medical dataset.

Window-based methods are excellent in detecting public patterns, their inability to effectively mitigate private pattern leakage renders them less suitable for privacy-sensitive domains like healthcare.

Ultimately, the evaluation results prove the expected improvements provided by graph-based pattern-level privacy protection, solving the literature's PUT gap. APP-CEP performs better in almost all aspects discussed in this section than approaches presented in previous studies due to considering both pattern dependencies and modeling the adversary's background knowledge.

4.4 Related Work

Although sharing data between stakeholders increases the utility of IoT applications, ensuring efficient preservation of privacy is still challenging [211]. Similarly, providing privacy in DCEP systems requires establishing a trade-off concerning QoS since protecting an individual's privacy should not sacrifice the application's functionality, which lays on more data to maximize the

quality [134,150]. Therefore, PPMs are required to be lightweight and do not prevent any application from detecting query patterns adequately. Moreover, although data owners have been recently becoming more aware of how their data will be analyzed and being able to control it [212–214], providing sufficient means for them to express their privacy concerns is still poorly studied.

The initial idea of preserving the individual's privacy by employing patterns was introduced more than a decade ago [24]. It was theoretically proved that by employing a probabilistic model, the choice of event suppression can be estimated to maximize the utility alongside a window-based approach, which makes the obfuscation decision according to the actual events. Similar methods computed the event distribution using advanced pattern match cardinality estimation techniques [45]. Other obfuscation operators (e.g., Reordering) have been discovered in the literature as an alternative to suppression operator [15]. The main drawback of the mentioned studies is they failed to obfuscate various pattern types since they focused only on sequence type. In addition, they were highly dependent on the input streams to apply obfuscation models, i.e., they failed to obfuscate various kinds of private patterns well in an environment with dynamic input event streams. Moreover, they have not supported simultaneous obfuscation models customized for each private pattern.

Recently, the idea of multi-operator multi-pattern privacy protection has been introduced [16]. Although this ILP-based approach maximizes the QoS while preserving privacy, it is limited to three pattern types and three obfuscation operators. Also, it suffers from oscillation between OT models since they change the deployment as soon as they detect a change in the input stream. This instability in the deployed OT models degrades performance by imposing unnecessary transition overhead. Moreover, the adversary's background knowledge, which might be used to realize the obfuscation, needs to be considered thoroughly.

4.5 Summary

In this chapter, we proposed APP-CEP, which represents how to enable dynamic integration of pattern-level privacy in event-based systems to protect

privacy while providing an acceptable level of QoS. In addition, by producing pattern dependency graphs, our mechanism is able to assign obfuscation models to conceal sensitive patterns selectively and make an obfuscation plan ahead of time. Our evaluation results demonstrated that APP-CEP outperforms two baseline techniques, namely Window-based and Graph-based approaches, in the total percentage of revealed private pattern matches and performed slightly better in the number of detection errors of public pattern matches. Moreover, by involving more background knowledge, APP-CEP achieves a significant performance in the expected revealed private matches by the adversary. Furthermore, the PUT results prove that APP-CEP successfully provides a solution to the gap mentioned in this study and envisions a new era in this field for adaptively assigning obfuscation models regardless of actual events in the streams effectively.

In our future work, we will consider designing an intelligent algorithm to select the best obfuscation model in case the dependency sub-graphs do not offer a ZOD node. That can be achievable by introducing a comparison metric by which both positive and negative effects of obfuscations can be computed and labeled in the sub-graphs. Enhancing the suitability for real-world applications requires thoughtful consideration of the stakeholders and their respective interests when selecting input representations. Moreover, we plan to extend our evaluation to different use cases to figure out other aspects of this problem as well as APP-CEP limitations in fulfilling various applications' requirements.

Chapter 5

CEP Rule Generation

> Wisdom thrives where effort's sown, No insight gained by ease alone. To light the path where knowledge lies, One must toil and analyze.
>
> Ferdowsi (Revised by GPT4o)

Abstract

Autonomous rule generation remains a critical challenge in complex event processing (CEP), particularly in distributed and federated environments where manual rule definition is inefficient and error-prone. This chapter introduces GPT-CEP, a novel framework that integrates large language models (LLMs) and prompt engineering to automate rule generation while ensuring adaptability and minimal human intervention. To enhance rule applicability and accuracy, GPT-CEP employs a federated learning schema, allowing local rule refinement using distributed data while preserving privacy and decentralization. To systematically evaluate its effectiveness, we propose the AGES Index, a composite metric assessing rule generation efficiency, accuracy, and model scalability. GPT-CEP further incorporates a cluster-based rule refinement mechanism, grouping related rules and applying simulated annealing for dynamic threshold optimization. Our experimental results demonstrate that GPT-CEP outperforms two baseline methods: Random Features and Threshold Values (RFV) and Random LLM and Prompting technique (RLP). GPT-CEP enhances rule generation efficiency, as measured by the AGES Index, while maintaining an F1-score of approximately 0.7 in activity recognition, representing a notable improvement over the baseline methods. This enables more effective rule generation and refinement while significantly reducing reliance on domain expertise.

Real-time data analysis is vital in today's fast-paced world, particularly in sectors like e-commerce, healthcare, and the Internet of Things (IoT). In these domains, the value of data rapidly diminishes if not analyzed promptly, as it becomes less effective in predicting or preventing problems or situations, so it needs to be analyzed almost immediately. DCEP is a well-established methodology for real-time data analysis. It employs CEP *rules*, which are essentially instructions that define the relationship between observed events and user-defined situations of interest, specified through CEP queries. However, a significant challenge arises due to the mismatch between how users conceptualize situations of interest and how CEP engines interpret queries. While CEP engines require queries in a structured syntax, users often struggle to translate their intended events into formal CEP queries, making it difficult to express queries in natural language or intuitive formats (e.g., [126]).

Traditionally, domain experts translate user queries into machine-readable CEP rules based on their knowledge of event-based systems. However, this manual process is time-consuming, prone to errors, and limited in capturing complex correlations. To address these challenges, researchers have integrated AI-based techniques into Distributed Complex Event Processing (DCEP) systems, leveraging historical data to automate rule generation and improve efficiency [18].

Despite these advancements, AI-driven approaches remain reactive, refining rules based on past events rather than dynamically adapting to new information, such as different sensor data or evolving system conditions. Large Language Models (LLMs), such as GPT, offer a promising alternative by generalizing from vast knowledge sources to generate adaptable rules. However, challenges remain, including ensuring interpretability, mitigating hallucinations, and aligning rules with real-world constraints. Overcoming these pitfalls is crucial for realizing the full potential of LLMs in autonomous rule generation. The next step is to develop a proactive framework that not only automates rule generation but also predicts and adapts to novel user queries and emerging patterns in real-time.

A key challenge in DCEP is validating rules across distributed data sources without centralized access to all event streams due to privacy constraints or data-sharing limitations. Federated rule refinement enables multiple clients to

collaboratively refine rules while keeping their data local, ensuring compliance with privacy regulations and preserving data confidentiality. By allowing event detection models to adapt to different environments without direct data exchange, this approach ensures that generated CEP rules remain adaptable and context-aware. Integrating LLMs with federated rule refinement allows rule generation to be continuously improved based on local observations across distributed nodes, leading to more accurate and reliable event detection.

The proposed DCEP architecture, GPT-CEP, combines reactive and proactive rule generation to enhance event detection. It leverages Large Language Models (LLMs) to generate rules dynamically while refining them based on past events. A key aspect of this approach is the use of prompt engineering, a technique for structuring inputs to guide LLMs in producing more accurate and relevant rules, and it is model-agnostic, so newer LLMs can be substituted without redesign, typically improving results.

GPT-CEP introduces several improvements, including better rule adaptability and enhanced detection accuracy against baselines. Its main contributions are as follows:

- We provide a novel mechanism for autonomous CEP rule generation and refinement powered by the NLP capabilities of LLMs and validate the rules over distributed data by a federated platform.
- We devise various levels of prompting using basic prompt engineering techniques and compare the performance of several LLMs in generating CEP-like rules.
- We propose a cluster-based strategy for validating the initial version of CEP rules using clients' local data and generating updates, leading to improving the accuracy of the initial rule version.
- We evaluate the performance of prompt engineering strategies and the federated rule validation and refinement phase, employing a *simulated annealing* optimization technique. over a real-world activity recognition dataset to realize potential limitations.

In the rest of this chapter, we first provide an overview of GPT-CEP by describing its modules and functionality in Section 5.2. Then, we propose and

investigate integrating large language models (LLMs) to automate and accelerate query translation and rule generation in event processing systems, including prompt engineering levels, elaborated on in Section 5.3. Furthermore, we introduce a federated learning schema to refine the initially generated rules by examining them over distributed event streams, ensuring greater accuracy and adaptability, which is discussed in Section 5.4. Finally, we display our evaluation setup and results in Section 5.5 and provide a literature review and conclude our chapter by wrapping up future works in Sections 5.6 and 5.7, respectively.

5.1 Autonomous Rule Generation Problem

In healthcare, real-time activity recognition plays a crucial role in monitoring patient health, detecting anomalies, and assisting in rehabilitation programs. Wearable IoT sensors embedded in smartwatches, chest straps, or motion-tracking devices continuously collect physiological and movement-related data, such as heart rate, accelerometer readings, gyroscope data, and skin temperature. By analyzing these real-time sensor streams, CEP systems can detect activities such as walking, sitting, running, or even abnormal conditions like falls or sudden inactivity. However, defining accurate and effective CEP rules for activity recognition has traditionally been a manual process, and while AI models can process raw sensor data directly, CEP offers interpretable, rule-based reasoning and guaranteed event-handling semantics essential for real-time decision systems.

In conventional CEP-based activity recognition systems, rules are typically formulated by domain experts who establish relationships between sensor data and predefined activity patterns. For example, a biomedical engineer may determine that an accelerometer threshold above a certain level indicates running, while a physiologist might refine the rule based on heart rate and energy expenditure data. These rules require extensive domain knowledge from both healthcare and sensor technology experts, making the rule definition process highly multidisciplinary and time-consuming. Additionally, predefined rules often fail to capture complex temporal dependencies, environmental factors, or personalized variations among different patients, leading

to high false-positive or false-negative rates in activity detection.

Given the dynamic nature of human activity and physiological responses, a static rule-based approach is insufficient for accurate event detection. Moreover, the heterogeneity of sensor types, data distributions, and patient-specific patterns makes manual rule engineering impractical at scale. A more adaptive and intelligent CEP framework is required, one that can automatically generate, refine, and validate rules based on real-time sensor data, without relying heavily on manual expertise.

Rules in CEP systems must balance reactivity (adapting to real-time environmental changes) and proactivity (anticipating unseen situations). However, existing rule generation methods remain limited to reactive adaptations, refining rules only based on historical data, failing to generalize effectively to emerging patterns. This leads to delays in response time and poor event detection performance when the system encounters new sensor inputs or previously unseen event correlations, although this conservative behavior can also enhance system stability.

Moreover, defining CEP rules has traditionally relied on domain experts, requiring multidisciplinary knowledge across sensor technology, event pattern analysis, and domain-specific expertise. This dependency slows down rule updates, introduces human inconsistencies, and limits scalability in dynamic IoT environments where event sources and conditions evolve continuously.

Ensuring high detection accuracy is another challenge, as static rule-based approaches struggle to maintain precision when sensor conditions shift or when new data distributions emerge. Without continuous adaptation, false positives and false negatives increase, reducing the overall reliability of the event detection system.

In this chapter, GPT-CEP combines proactive rule generation with reactive rule refinement, automating CEP rule updates while minimizing reliance on domain expertise. By generating a set of rules $R = \{r_1, r_2, ..., r_n\}$ in an IoT environment E with distributed sensors capturing event streams S, the objective is to maximize the following formula while meeting the necessary constraint:

$$
\begin{aligned}
\max \quad & w_{acc} \sum_{r_i \in R} Accuracy(r_i, S) \\
+ \quad & w_g \sum_{r_i \in R} Generalization(r_i, S) \\
+ \quad & w_{ra} \sum_{r_i \in R} RuleAutomationScore(r_i, S) \\
s.t. \quad & F1(r_i) \geqslant \tau_i, \quad \forall r_i \in \mathcal{R}, \\
& r_i(t+1) = f(r_i(t), S(t+1)), \quad \forall S(t+1) \in \mathcal{S}
\end{aligned}
$$

Here, w_{acc}, w_g, and w_{ra} are weighting factors that balance *accuracy*, *adaptability (generalization)*, and *automation*. The Rule Automation Score (RA) quantifies the level of human intervention required in defining and refining rules. To ensure robust performance, each rule r_i must satisfy an F1-score constraint, where its performance metric $F1(r_i)$ must remain above a predefined threshold τ_i. Additionally, the system must support dynamic rule adaptation, where each rule r_i updates over time based on new sensor stream s_t through an adaptive function f. GPT-CEP optimizes rule generation and refinement by ensuring high detection accuracy, adaptability to unseen scenarios, and reduced dependency on human experts.

5.2 The GPT-CEP System Design

The proposed mechanism automates and enhances reactive and proactive rule generation in DCEP systems. By dynamically creating, refining, and prioritizing rules based on real-time event patterns and system feedback, it improves efficiency and adaptability, enabling faster and more accurate decision-making in dynamic environments. As illustrated in Figure 5.1, our system bridges the gap in autonomous proactive rule generation by incorporating a refinement process that leverages distributed local streams.

The process begins with users defining queries, converting them to CEP-like patterns by Large Language Models (LLMs), and refining and adapting them based on local event streams provided by distributed clients (i.e., participants) in the rule optimization process. For instance, in traffic control,

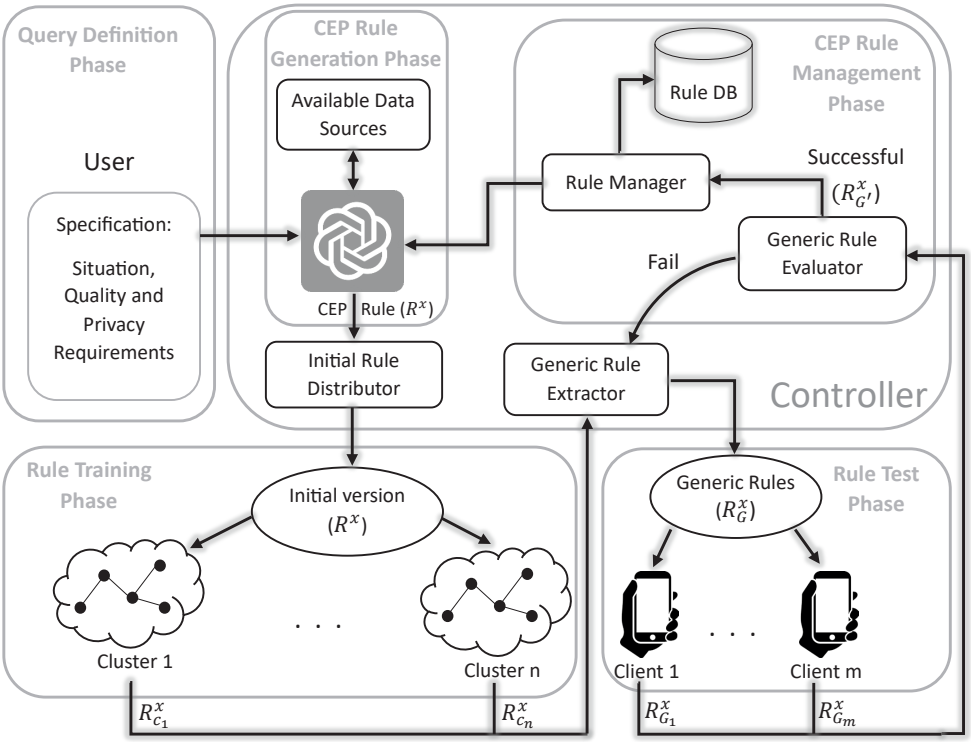


Figure 5.1: The GPT-CEP system design.

participants such as traffic cameras, sensor networks, and monitoring stations contribute real-time events to fine-tune rules and ensure alignment with dynamic traffic conditions.

However, manually converting these queries into rules which are CEP-compatible is challenging, requiring both expertise in event-based systems and knowledge of formal rule syntax. To address this, LLMs transform user-defined queries into structured CEP rules using natural language processing (NLP) capabilities. LLMs are particularly effective for this task as they understand and formalize user intent, converting ambiguous descriptions into structured rules. Unlike traditional rule-based systems, which rely on manually crafted templates, LLMs generalize from vast knowledge sources and adapt rules based on contextual nuances. Their ability to process diverse linguistic inputs makes them accessible to users with varying technical expertise.

Given these strengths, LLMs theoretically have the potential for automating and optimizing rule generation, minimizing human intervention while improving accuracy and adaptability.

LLMs process input through prompts, allowing users to submit natural language queries for structured responses. This makes them valuable for automating CEP rule generation, as they interpret user-defined queries and translate them into machine-readable rule formats. However, effective interaction with LLMs presents challenges. High-quality CEP rule generation requires carefully designed prompts to ensure clarity, relevance, and accuracy. We address key constraints, including prompt length limitations, multimodal input handling, API integration, response uncertainties, hallucinations, prompt efficiency optimization, and subscription costs. These challenges and solutions are discussed in Section 5.3.

Another challenge is adapting initial rules to specific deployment conditions using local context. Here, local context refers to patterns and conditions observed at individual participant nodes, such as traffic cameras in traffic control systems or IoT sensors in smart city environments. To ensure rules remain effective across different conditions, participants are clustered based on observed event characteristics, generating context-specific rule variations. For each cluster, a specialized aggregation method refines rules based on local updates. For example, when dealing with time-based thresholds, the method may prioritize the highest observed value to maximize rule applicability across local data streams. This ensures that refined rules remain relevant in real-world conditions. We detail this approach in Section 5.4.

To explain the proposed method in more detail, query processing in the proposed method initiates when a user submits a query (see Query Definition Phase in Figure 5.1). We assume the user possesses sufficient knowledge and resources to augment the query definition with quality and privacy requirements. Upon receiving the query, the *controller* immediately tasks the appropriate Large Language Model, potentially selected from multiple LLMs based on metrics such as availability, to leverage its NLP capabilities and domain knowledge to translate the user-defined query into CEP-like patterns (see CEP Rule Generation Phase). Notably, if feasible, the LLM will be granted access to a database containing comprehensive information about

registered event sources. This access enables the LLM to generate patterns based on simple events produced by these event sources. However, if the LLM lacks this capability, the user itself must provide sufficient information about available event sources within the query definition.

Once the LLM generates the initial version of the CEP rule (i.e., R^x in which R indicates it is a rule and x is the rule number), the controller proceeds to the next phase, focusing on refining rule features by applying the rule across distributed streams (see Rule Training Phase). In our approach, each client applies R^x to its simple event streams, encompassing both current and historical data. Proactively, each client iteratively adjusts rule characteristics (e.g., if-else thresholds, window size) to optimize specified quality metrics such as detection accuracy. For example, a client might incrementally increase the window size for pattern detection. This adjustment could potentially reduce partial matches and consequently identify more complete pattern matches. The client would then continue to refine the window size within a predefined limit of changes before selecting the optimal value that maximizes the desired quality metric.

Following the rule adjustment, each cluster (c_1 to c_n) submits its customized rule (e.g., $R^x_{c_1}$ to $R^x_{c_n}$) to a generic rule extractor module. This module evaluates the threshold updates and generates multiple versions (one to many) of a specific initial rule, referred to as R^x_G. The process of generating these generic rules varies depending on the use case and is designed to fit the specific application. For example, a congestion detection rule might behave differently on highways compared to residential areas. To address this, multiple tailored rule versions are created to suit the unique characteristics of each cluster.

The set of generic rules is then validated across additional test clients (i.e., a separate set of m clients) to assess its generalizability (see Rule Test Phase). This produces test results (i.e., $R^x_{G_1}$ to $R^x_{G_m}$ which show the test updates for client one to m) analyzed by the evaluator component. This iterative testing phase continues until the analysis demonstrates success, e.g., test clients for all generic rule versions achieve acceptable detection accuracy. Upon successful validation, the refined generic rule set (i.e., $R^x_{G'}$) is submitted to a rule manager module. This module updates the CEP rule database (see

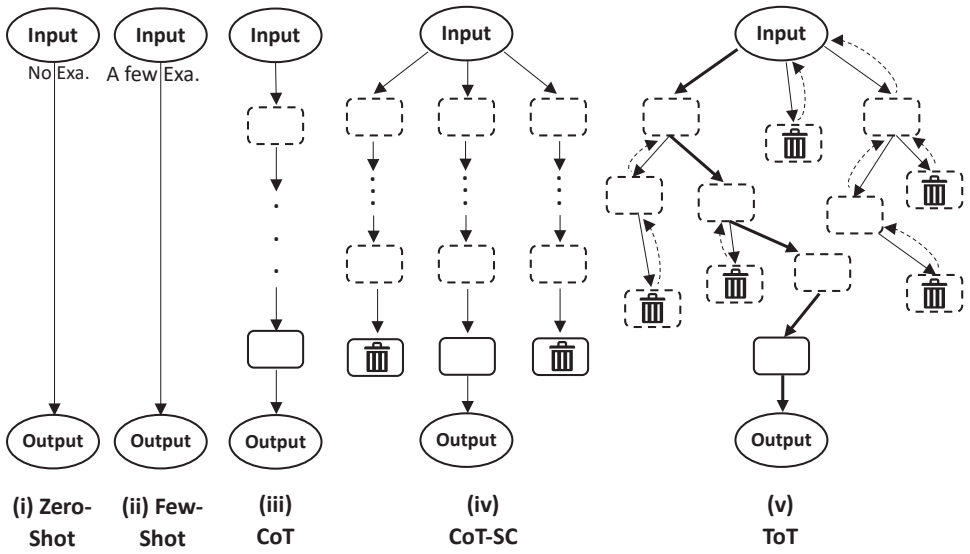


Figure 5.2: Basic prompt engineering techniques.

CEP Rule Management Phase). Concurrently, updates to the initial rule are forwarded to the LLM for future consideration, creating a feedback loop that continuously improves the rule database.

5.3 Autonomous LLM-based Rule Generation

To figure out what the true limitations of interacting with LLMs and finding or inventing solutions for them, we explore the concept of *prompt engineering* which is currently the most important technique that helps better transfer information to the LLM in order to generate to-the-point responses [215]. In Figure 5.2, we illustrate the *basic prompt engineering techniques*. In the accompanying diagrams, dashed rectangles encapsulate intermediate *thoughts* (i.e., coherent language sequences that contribute to the problem-solving process). Solid rectangles represent the final decision or output generated by the model. Additionally, rectangles marked with recycle bins represent abandoned thoughts discarded due to majority voting (as in CoT-SC) or backtracking (as in ToT), which will be explained further.

- **Zero-Shot:** In zero-shot prompting, the user directly tasks a language model with a request or question without providing examples of how they want it to respond. It relies on the LLM's inherent knowledge and pattern recognition capabilities to generate a relevant output. For example, asking a model, "Summarize the plot of 'The Great Gatsby'," without giving any hints about the novel's plot or desired summary format. This technique is extremely fast and flexible, requiring no extra data preparation. However, response accuracy can vary, especially for complex or nuanced tasks where examples would guide the LLM's response.

- **Few-Shot:** This prompting technique bridges the gap between zero-shot and full fine-tuning. The user provides a small number of examples (typically up to five) alongside the prompt itself to guide the LLM's output format and content. For example, one might ask a model to translate English to French, but first showing it a few examples of correct translations. Although this technique improves accuracy and consistency compared to zero-shot and requires minimal extra data, it still relies heavily on the LLM's pre-existing knowledge and may not be sufficient for very specific tasks.

- **Chain of Thought (CoT):** CoT prompting is a technique that explicitly encourages the LLM to break down its reasoning into a series of intermediate steps. Instead of just providing a final answer, the model also details its thought process, leading to more transparent and potentially more accurate results, e.g., asking a model to solve a math problem and instructing it to show each calculation step and the reasoning behind it. This technique enhances transparency, can improve accuracy, and provides insights into the LLM's reasoning. On the other hand, it can generate overly verbose responses and may not always be necessary for simple tasks.

- **Chain of Thought Self-Consistency (CoT-SC):** CoT-SC takes CoT further by generating multiple chains of thought for the same prompt and then applying a *majority voting* mechanism to the final result. This can improve accuracy by aggregating the insights from

different reasoning paths. This can further improve accuracy and robustness compared to standard CoT, but it is computationally more expensive than standard CoT, and the majority voting mechanism may not always lead to the best result.

- **Tree of Thoughts (ToT):** ToT expands on CoT by allowing the LLM to explore multiple reasoning paths simultaneously, like branches of a tree. This facilitates more creative and comprehensive problem-solving, as the model can backtrack and evaluate different options before settling on a final answer. It makes reasoning highly flexible and potentially more creative, making it suitable for complex or open-ended tasks. However, it can be computationally expensive and may generate overly verbose or disjointed responses.

Notably, the first group (i.e., zero-shot and few-shot prompting) primarily shapes the model's understanding of the task, while the second group (i.e., CoT, CoT-SC, and ToT) guides the model's approach to solving it. To maximize the accuracy and effectiveness of language model outputs, it's often beneficial to leverage a combination of techniques from both groups. For instance, the popular "Few-Shot CoT" approach merges the guidance of examples (few-shot) with the structured reasoning of the Chain of Thought, leading to more reliable and insightful results. To explore the full potential of prompt engineering, we developed and implemented seven hybrid techniques, strategically combining the fundamental approaches mentioned earlier, namely 1) Zero-Shot, 2) Zero-Shot with CoT, 3) Zero-Shot with ToT, 4) Few-Shot, 5) Few-Shot with CoT, 6) Few-Shot with CoT-SC, and 7) Few-Shot with ToT. It's important to acknowledge that the absence of examples in zero-shot prompting can increase the likelihood of incorrect solutions. Furthermore, when applied in a zero-shot setting, the CoT-SC (Chain of Thought Self-Consistency) technique may worsen this issue by potentially reinforcing consistently incorrect reasoning. Due to this risk, we intentionally excluded the Zero-Shot CoT-SC combination from our exploration of hybrid techniques.

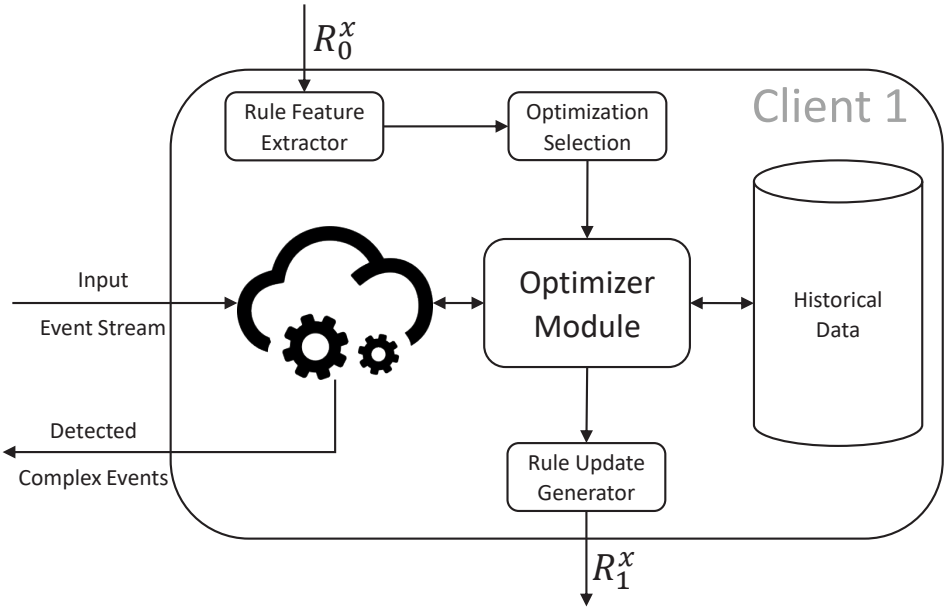


Figure 5.3: Cluster member's rule optimization design.

5.4 Federated Rule Refinement

Although LLMs can facilitate the generation of CEP rules, their responses might not be of sufficiently high quality to detect matches for situations of interest. To this end, we designed a rule refinement mechanism (see Figure 5.1) to optimize the rule thresholds according to the locally observed data.

CEP rule training begins by analyzing clients to identify shared characteristics and create distinct clusters for tailored rule customization. For example, a traffic congestion rule requires different vehicle speed thresholds on highways than in residential areas. Therefore, varying CEP rules must be generated to detect congestion accurately in both scenarios. This approach ensures rule adaptability and accuracy across diverse contexts, maximizing real-world effectiveness. Clusters are formed by performing similarity checking and grouping clients with similar features, such as geographical attributes, based on application-specific metrics. This yields non-overlapping client clusters for targeted rule generation.

Within each cluster, members receive the initial rule version from the LLM and experiment with its thresholds to find the optimal values for their local data. They first extract the rule's key features to determine the most suitable optimization technique. This chosen technique is then applied to detect rule matches in both real-time streams and historical data. Finally, a customized rule version tailored to the client's local data is forwarded as an update to the subsequent step. Figure 5.3 illustrates the control flow of this optimization method within each participating client.

5.5 Evaluation

Our proposed method underwent a two-part evaluation. Initially, we focused on a simplified, centralized scenario, omitting distributed streams and their aggregation functions. This streamlined setup allowed us to assess the performance of seven hybrid prompt engineering techniques on a dataset sample, effectively showcasing the potential of prompt engineering itself. A detailed analysis of these techniques can be found in Section 5.5.1. Secondly, we applied the generated CEP rules in the first phase over the whole dataset to refine the rules distributively, discussed in Section 5.5.2.

We evaluated GPT-CEP[1] by analyzing the *PAMAP2* dataset [216], which contains data on 18 different physical activities (e.g., walking) performed by 9 subjects wearing 3 inertial measurement units and one heart rate monitor.

5.5.1 LLM Rule Generation

To assess the impact of prompt engineering on Large Language Model (LLM) responses, we applied seven previously defined prompt models to the PAMAP2 activity recognition dataset. We further compared the performance of established LLMs, *OpenAI GPT4* [217] and *Google Gemini* [218], to the smaller, locally deployed model, *Codestral* [219], selected from *Hugging Face* [220] for its code generation capabilities.

Our primary evaluation goals are to:

[1] https://github.com/majid-lotfian/GPT-CEP

1. Investigate how accurately each LLM generates the initial version of a CEP rule.
2. Assess to what extent prompt engineering techniques can enhance the quality of generated rules.
3. Identify the limitations of our approach when applied to real-world datasets.

In the following, we first elaborate on our experiment results with the GPT4 model and then explain how Codestral can also be used in CEP rule generation.

GPT 4:

This AI model is a groundbreaking language model developed by OpenAI. It represents a significant leap forward in artificial intelligence, demonstrating remarkable abilities in understanding and generating human-quality text. Capable of handling complex tasks such as language translation, creative writing, and providing informative responses, GPT-4 has significantly advanced the capabilities of AI [221]. Recently, DeepSeek [222] has emerged as another promising model in this space. However, due to time constraints, we have not yet conducted a thorough evaluation of its capabilities. We plan to explore its performance in future work to assess its potential in comparison to existing models.

Simulation Setup: In the initial assessment, our experiments used the GPT-4. Attempts to obtain suitable responses from Gemini were unsuccessful since it delivered the rule generation's instructions, not the rule definitions, indicating that Gemini tended to explain the process of creating rules instead of synthesizing executable logic, likely due to its prompt interpretation bias. Each hybrid technique was applied at least five times, and the final results represent the average of these iterations. The error bars displayed illustrate the confidence intervals of our experimental findings. We tasked the GPT-4 model with generating CEP-like patterns to identify specific activities within the dataset.

To simplify the process, we provided the LLM with a sample from the PAMAP2 dataset and requested it classify the activity for each record based

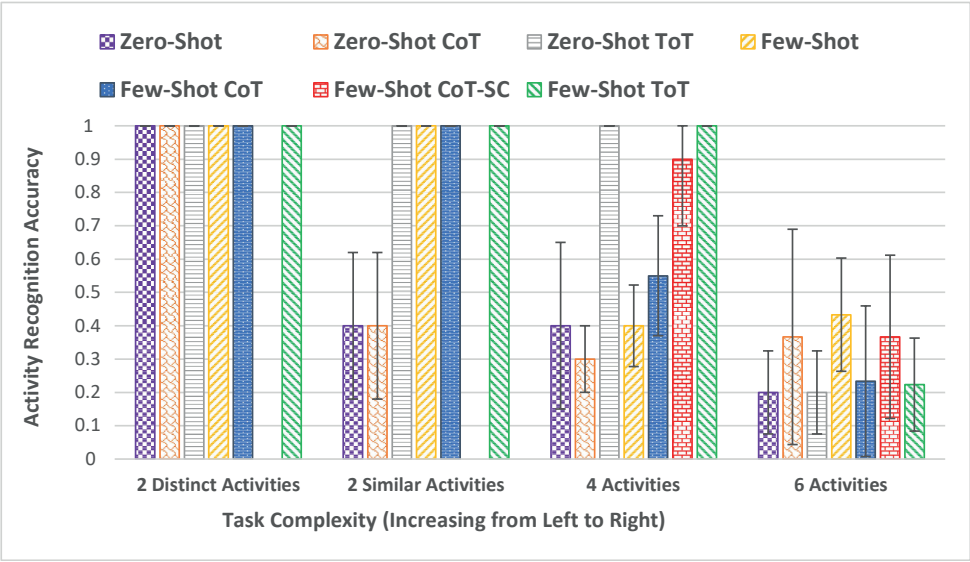


Figure 5.4: The initial evaluation results on activity recognition using a sample of PAMAP2 dataset on GPT4. None of the proposed prompt engineering models outperformed all other methods.

on the generated patterns. The activity recognition accuracy was then evaluated by comparing the LLM's results against the Ground Truth labels in the dataset, as follows:

$$Accuracy = \frac{\text{Number of True Positives}}{\text{Total Number of Activities}} \tag{5.1}$$

Note that in zero-shot, the LLM's lack of examples can lead to incorrect solutions, and CoT-SC may worsen this by favoring the most consistently wrong answer over the most accurate one. That's why we exclude Zero-Shot CoT-SC from our experiments.

Results Discussion: Figure 5.4 presents the initial results of utilizing the aforementioned techniques to generate CEP rules and classify activities.

The results demonstrate that GPT-4 generated rules possess adequate capacity to differentiate between two distinct activities. However, as the number of activities increased (i.e., increasing task complexity), GPT-4 tended to misclassify activities. This suggests that the generated rules may require further

refinement (e.g., adjusting thresholds), particularly for activities with greater similarity (e.g., sitting and lying).

While CoT, CoT-SC, and ToT were expected to enhance rule accuracy, this was not consistently observed. Although Few-Shot techniques generally outperformed Zero-Shots, the specific conditions favoring each group remain unclear. Additionally, the lack of labeled data in some use cases limits the applicability of Few-Shot approaches.

Codestral:

The second model is a cutting-edge language model specifically designed for code generation and manipulation. Developed by Mistral AI, Codestral excels at tasks such as code completion, writing functions, and debugging. It's proficient in a wide range of programming languages and offers a valuable tool for developers seeking to enhance their productivity and efficiency. With its ability to understand and generate code, Codestral is poised to revolutionize software development processes [219].

```
# Rule Definitions
df['sitting'] = ((df['Heartbeat'] >= 83) & (df['Heartbeat'] <= 109) &
                 (df['imu_hand_mean'] >= 25) & (df['imu_hand_mean'] <= 40) &
                 (df['imu_chest_mean'] >= 30) & (df['imu_chest_mean'] <= 36) &
                 (df['imu_ankle_mean'] >= 30) & (df['imu_ankle_mean'] <= 36))

df['running'] = ((df['Heartbeat'] > 109) |
                 (df['imu_hand_mean'] < 25) | (df['imu_hand_mean'] > 40) |
                 (df['imu_chest_mean'] < 30) | (df['imu_chest_mean'] > 36) |
                 (df['imu_ankle_mean'] < 30) | (df['imu_ankle_mean'] > 36))
```

Listing 5.1: An example of rules generated by Codestral. For readability, we present only the rule definition section.

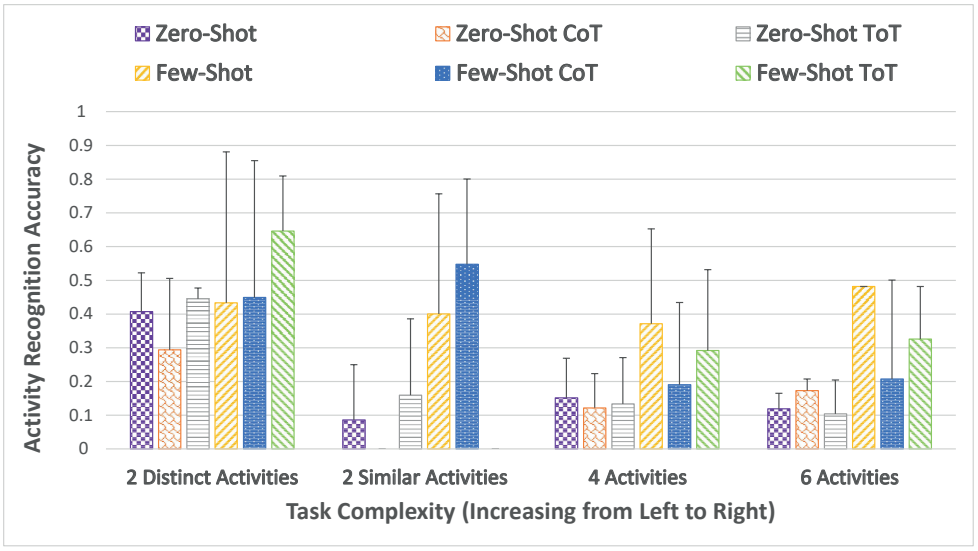


Figure 5.5: Activity recognition accuracy of Codestral LLM by involving more complex tasks (increasing the number of activities). Results have not proved the superiority of a specific prompt engineering method.

In Listing 5.1, we present a portion of the response generated by Codestral LLM, illustrating rule definitions for two distinct activities. The model processes feature titles from the input (e.g., *imu_hand_mean*) as dataset characteristics and identifies the most relevant features for each activity. It then uses its own knowledge to estimate initial threshold values, such as $['Heartbeat'] \geqslant 83$, to define the rules.

Compared to GPT-4, Codestral exhibited a slower initial performance in recognizing both distinct and similar pairs of activities, as well as sets of four activities. Nevertheless, its recognition accuracy for six activities proved comparable to GPT-4, especially considering the substantial disparity in model size. While GPT-4 boasts an impressive 1.76 trillion parameters [223], Codestral operates with a significantly smaller 22 billion parameters [219].

Solely relying on rule accuracy to assess LLM performance in this domain may present an incomplete view of their capabilities. For instance, our analysis of Codestral revealed a discrepancy between the number of rules with correct syntax and the overall quantity of responses. In this context, a cor-

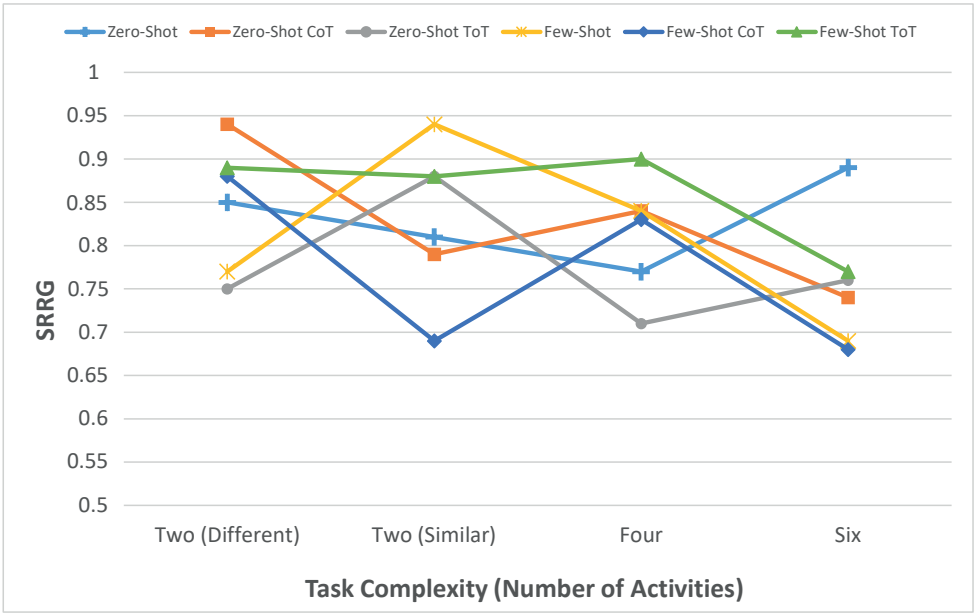


Figure 5.6: The success ratio in rule generation for GPT4 LLM by involving more complex tasks (increasing the number of activities).

rect syntax is one that adheres to the structure of Complex Event Processing (CEP) rules, which specify attribute-value pairs and thresholds for each attribute. We excluded responses that only outlined the steps for rule creation without generating a valid CEP rule. This detailed criterion highlighted an important observation in Codestral experiment, where the LLM provided us with a single, highly accurate response to a prompt but failed to respond effectively to the remaining prompts. While this single response demonstrated high accuracy, the overall low responsiveness underscored a potential overestimation of the LLM's performance when considering accuracy alone.

Enhancing Accuracy with Response Reliability:

To provide a clearer visualization of the challenge as mentioned earlier, we calculated the success ratio in rule generation (SRRG) as:

$$\text{SRRG} = \frac{\text{Number of Syntactically Valid Responses}}{\text{Total Number of Prompts}} \tag{5.2}$$

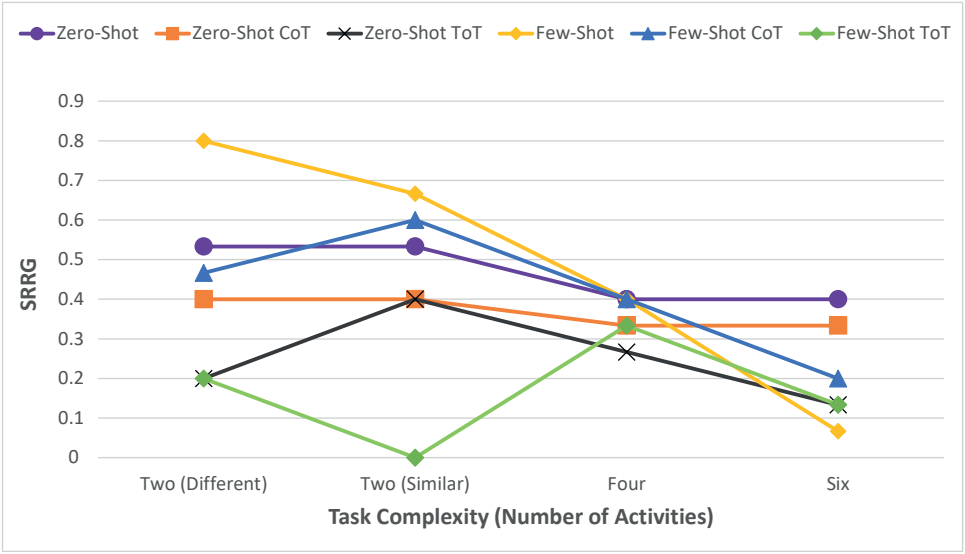


Figure 5.7: The success ratio in rule generation for Codestral LLM by involving more complex tasks (increasing the number of activities).

In this context, syntactically valid responses denote syntactically valid CEP rules that can be executed or evaluated, not necessarily rules that achieve high detection accuracy.

Figures 5.6 and 5.7 illustrate the SRRG metric across six prompt engineering techniques and four levels of task complexity for GPT4 and Codestral models, respectively. Cases such as Codestral's strong performance in the few-shot scenario with six activities (see Figure 5.5), which seems to have the best performance, show the importance of other metrics, such as SRRG and AGES-index which we introduce later. Although the initial assessment suggested promising results, a more in-depth analysis revealed a low SRRG value of only 0.06. Such a finding emphasizes the need for a comprehensive evaluation framework that incorporates additional metrics. By considering factors beyond accuracy, we can gain a more reliable understanding of LLM capabilities in rule generation and their suitability for downstream applications.

Rule Generation Efficiency:

While the success ratio (i.e., SRRG) provides a valuable measure of overall performance, it does not fully capture the potential overheads associated with incorrect LLM responses. While a low SRRG might suggest a high likelihood of incorrect responses, a single correct response can significantly contribute to the generation of high-quality CEP rules. Moreover, the rapid response times of state-of-the-art LLMs mitigate the potential drawbacks of requiring multiple prompts. To address these concerns, we have considered both the time overhead and monetary costs incurred by repeated prompts. To calculate the time overhead (TO), we consider three primary factors including the time required to compose the prompt (T_{write}), the duration the LLM takes to generate a response ($T_{response}$), and the time spent analyzing the response to determine its correctness (T_{check}). We multiplied the sum of these factors by the total number of prompts to obtain the overall TO, as follows:

$$TO = \big(T_{write} + T_{response} + T_{check}\big) \times \text{Total Number of Prompts} \tag{5.3}$$

Note that in case the prompting is done using LLM APIs, we consider the T_{write} equals to zero and only count the two last factors.

Furthermore, since to use LLMs we should pay the costs, we computed such a cost as the Monetary Cost Impact (MCI), by the below formula:

$$MCI = 1 + (\text{Prompting Charge} \times \text{Total Num of Prompts} \times \beta) \tag{5.4}$$

To account for the impact of LLM costs, we introduced a scaling factor in the MCI formula, β. This factor amplifies the monetary cost of using an LLM, ensuring that even small differences in cost have a noticeable effect on the metric ($1 \leqslant \beta$). Without scaling, the MCI might not be sensitive enough to distinguish between expensive and free LLMs. The appropriate value for β depends on the desired sensitivity and could be set, for instance, to 2 or 3 but it is always more than 1. The *Prompting Charge* represents the cost associated with interacting with an LLM. For free LLMs, the MCI value equals 1 (i.e., the minimum of MCI).

To evaluate the overall efficiency of using an LLM for rule generation, we combined the various factors into a unified metric: the rule Generation Efficiency (GE). The GE can be calculated as follows:

$$GE = \frac{N_{SRRG}}{N_{TO} \times N_{MCI}} \tag{5.5}$$

In this equation, the Normalized Success Ratio (i.e., N_{SRRG}) positively correlates with the rule Generation Efficiency (i.e., GE), while the Normalized Time Overhead (i.e., N_{TO}) and Normalized Monetary Cost Impact (i.e., N_{MCI}) hurt the value of GE. Here, normalization ensures that all metrics are scaled between 0 and 1, making them comparable and enabling a fair assessment of their contributions to the overall evaluation.

While LLM size can significantly influence performance metrics like accuracy and GE, it should be considered a secondary factor when comparing LLMs, especially when a CEP system tends to make the rule generation process completely local and run the model locally.

AGES Index: LLM's Composite Comparison metric

To comprehensively evaluate LLMs for selecting the optimal option to generate CEP rules, a composite metric integrating Accuracy, GE, and model size can be developed. Since model size holds relatively lower importance, it can serve as an adjustment to the final comparison metric rather than a primary factor.

To achieve a balanced evaluation, we first normalize the values of Accuracy and GE using the MaxMin normalization approach for both metrics. Rather than employing a simple weighted sum, we utilize the harmonic mean to penalize models that perform poorly in either metric, emphasizing a balanced performance between Accuracy and GE. The harmonic mean is particularly effective for creating composite metrics when balancing multiple factors that may have an inverse relationship.

A well-known example of the harmonic mean's effectiveness is the F1-Score in information retrieval and machine learning, where it balances precision and recall. This approach provides a conservative average, ensuring that a model with high precision but low recall, or vice versa, receives a lower F1-Score, accurately reflecting its suboptimal overall performance.

Similarly, in our context, both Accuracy and rule Generation Efficiency (GE) are equally important. The harmonic mean ensures a trade-off between normalized Accuracy (N_{Acc}) and normalized GE (N_{GE}), leading to a unified assessment metric called the *Accuracy-GE Score (AGE-Score)*, which is formulated as:

$$\text{AGE-Score} = 2 \times \left(\frac{N_{Acc} \times N_{GE}}{N_{Acc} + N_{GE}}\right) \tag{5.6}$$

To enhance the comprehensiveness of our evaluation, we incorporate model size as a supplementary factor in the AGE-Score calculation. By modulating the influence of model size with an α parameter, we ensure that this metric contributes minimally to the overall score while enriching the analysis. It should be noted that to make model size applicable in the previous formula, we used the normalized amounts for model size (N_{Model_Size}).

We call the proposed unified assessment metric *AGE-Size (AGES) Index*, and define it as follows:

$$\text{AGES Index} = \text{AGE-Score} \times \left[1 - (\alpha \times N_{Model\_Size})\right] \tag{5.7}$$

Where α is a weight ($0 \leqslant \alpha \leqslant 1$) that determines the extent to which model size penalizes the index. By fixing the value for α, a bigger model size for an LLM leads to a small reduction in the AGES Index, reflecting a balance between penalizing resource-intensive models and maintaining emphasis on accuracy and performance.

To compare GPT-4 and Codestral based on the AGES Index, we first calculated the GE value for all seven prompt engineering models for each of the two LLMs in four different task complexity. For the sake of simplicity, we assumed that the time overhead (TO) is similar for all models, setting TO to 1.

Regarding MCI, the cost of using Codestral was considered zero, resulting in MCI equal to 1. For GPT-4, the pricing was based on the OpenAI pricing model [224]: \$5 per 1 million input tokens and \$15 per 1 million output tokens. Given that our prompts averaged 500 tokens and outputs were approximately the same size, we calculated the cost per prompt as \$0.01. To appropriately amplify the monetary costs of GPT-4 against Codestral, we set the value of β to 1.5.

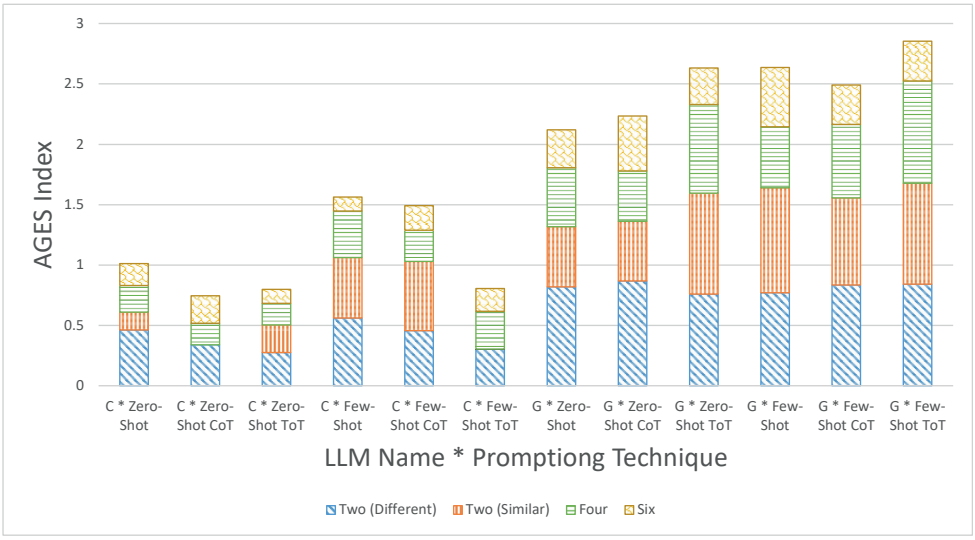


Figure 5.8: AGES Index values calculated for GPT4 (G) and Codesteral (C) models using different prompt engineering techniques in four different task complexities (two different activities, two similar activities, four activities, six activities). We increased the complexity of activity recognition by increasing the number of activities from two to six.

To assess the performance of LLMs across various prompting techniques, first, we calculated the MCI using Equation 5.4. Next, we determined the GE based on Equation 5.5. Finally, the AGE-Score was computed for each LLM using Equation 5.6.

In our evaluation, ChatGPT was used as an available application of the GPT-4 model for question-answering. While this introduces a potential limitation, it's important to note that the model size is not directly comparable between the two models. Therefore, to isolate the impact of prompting techniques and avoid unfair comparison, we set α to zero. This essentially equates the AGES Index to the AGE-Score, and the results are depicted in Figure 5.8.

The results depicted in Figure 5.8 demonstrate a clear superiority of the GPT-4 model over the Codestral model in terms of overall performance. While both models exhibited enhanced performance when employing few-shot prompting techniques compared to zero-shot approaches, the integration of

the ToT mechanism yielded distinct outcomes. In the Codestral model, ToT-based approaches not only cannot improve overall results but also lead to a reduction in the index when combined with few-shot prompting. Conversely, GPT-4 consistently demonstrated the highest performance across both zero-shot and few-shot prompting paradigms when incorporating ToT.

As expected, few-shot models outperformed zero-shot models due to the provision of illustrative examples. However, the exceptional performance of the base few-shot model in Cedesteral suggests that the model's architecture may be less adept at handling prompt engineering techniques. This could be attributed to a lack of alignment with reasoning skills or limitations in prompt responsiveness. Some LLMs excel at interpreting and following structured prompts, while others may struggle. If an LLM fails to adequately understand or apply the guidance embedded in prompts like CoT (e.g., breaking down complex problems step-by-step) or ToT (e.g., exploring multiple reasoning paths), it may produce suboptimal results.

As depicted in Figure 5.5, the accuracy metric for the *Few-Shot* prompting technique, where the task complexity is to recognize six activities applied to Codestral LLM, exhibits certain limitations (remember the situation where there is only one response out of fifteen prompts, but with high accuracy results). To provide a more comprehensive assessment that considers a wider range of evaluation metrics beyond raw accuracy, the AGES Index is introduced. Figure 5.5 demonstrates that when evaluating six activity tasks using the AGES Index, the *Few-Shot* technique no longer maintains its dominant position observed in the accuracy metric. This suggests that the AGES Index offers a more balanced and informative evaluation of the model's overall performance.

With the introduction of the AGES Index, we have completed the initial phase of our autonomous rule generation architecture. This index enables us to select the most suitable LLM for a given CEP system domain, based on its ability to generate accurate, efficient rules while considering resource constraints. By evaluating LLMs using the AGES Index, we can optimize rule generation performance tailored to specific application requirements.

5.5.2 Rule Refinement Evaluation

Once the most suitable LLM is identified for a specific domain, the generated rules for each query are distributed to clusters for further refinement. These clusters employ various techniques, such as rule-based approaches, machine learning algorithms, and human feedback, to enhance the quality and accuracy of the rules.

To assess the effectiveness of the proposed method, GPT-CEP, we conducted comparative evaluations against two baseline strategies. These baselines represent common approaches [225, 226] used in the field and serve as benchmarks for our system's performance.

- **Random Features and threshold Values (RFV)**: This strategy begins with no prior knowledge of the query. It is query-agnostic, randomly chooses a subset of features, and assigns arbitrary cutoff values.
- **Random LLM and Prompting technique (RLP)**: This strategy randomly selects LLM and prompt engineering techniques, neglecting the AGES Index and prompting techniques' performance in identifying rule patterns and thresholds.

In comparison with these two baseline strategies, GPT-CEP identifies the most effective approach based on its ability to generate accurate and comprehensive rule definitions, by comparing the AGES Index of each LLM-prompting technique combination.

Rule Accuracy Improvement

The primary objective of the cluster-based rule refinement phase in GPT-CEP is to enhance the detection accuracy of LLM-generated CEP rules. To achieve this, each member of the cluster iteratively adjusts the rule's thresholds using its locally stored data, employing a *Simulated Annealing* optimization technique [227].

Simulated annealing, a metaheuristic optimization algorithm inspired by the annealing process in metallurgy, is a suitable choice for this task due to its ability to efficiently explore the solution space and avoid becoming trapped in local optima. By gradually reducing the temperature parameter,

the algorithm progressively decreases the probability of accepting suboptimal solutions, thereby increasing the likelihood of finding a near-optimal solution [228, 229].

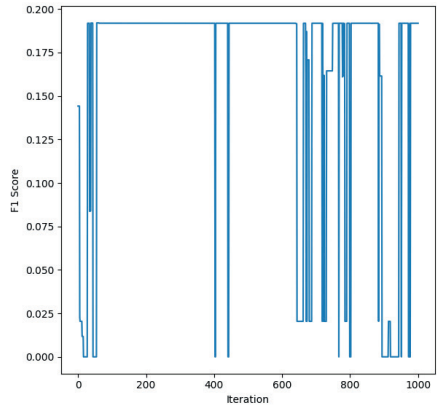
The selection of simulated annealing for rule threshold optimization in GPT-CEP is based on its balance of speed and effectiveness. While other optimization techniques might be considered, simulated annealing offers a robust and practical approach that can be tailored to the specific requirements of the system [230].

Simulated annealing is a probabilistic optimization technique that is inspired by the physical process of annealing, where a material is heated to a high temperature and then slowly cooled. The algorithm involves iteratively generating new solutions and accepting or rejecting them based on a probability distribution that is influenced by a temperature parameter. At high temperatures, the algorithm is more likely to accept suboptimal solutions, allowing it to explore a wider range of possibilities. As the temperature decreases, the probability of accepting suboptimal solutions also decreases, focusing the search on more promising regions of the solution space [231, 232].

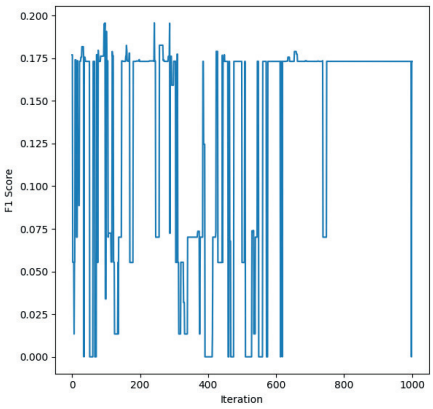
To evaluate the accuracy of event detection for each activity type among the six categories (Lying, Sitting, Standing, Walking, Running, and Cycling), we utilized the F1-score as our performance metric. This widely adopted quality assessment metric is commonly employed in stream processing applications to comprehensively consider both false positive and false negative errors, ensuring a balanced evaluation of event detection procedures.

The initiated rules, generated using three distinct strategies (i.e., RFV, RLP, and GPT-CEP), were applied to one of the objects (i.e., clients) in the PAMAP2 dataset. To simplify the evaluation process, we temporarily bypassed the rule aggregation module, which would typically merge updates from different cluster members. This allowed us to directly evaluate the performance of the rules on a single client. To ensure a comprehensive assessment, we divided the PAMAP2 dataset into training and testing sets. This enabled us to validate the performance of the rules after completing the simulated annealing optimization process.

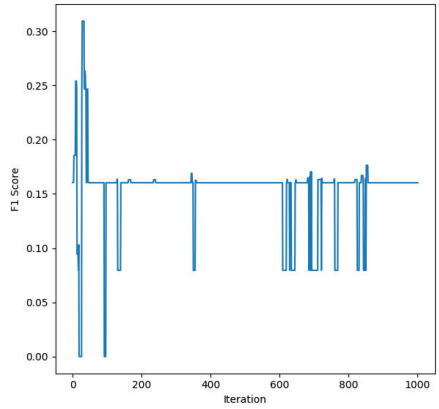
Figure 5.9 visually depicts the dynamic fluctuations in F1-score values for different activity types as the simulated annealing algorithm iteratively



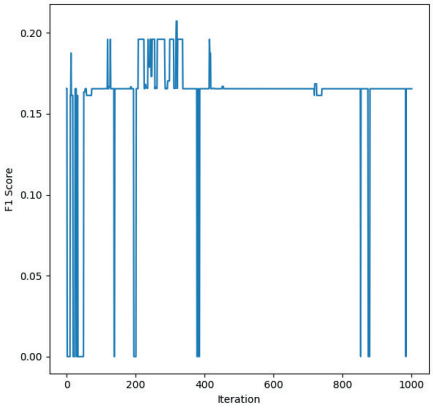
(a) F1-Score in detecting Lying down activity



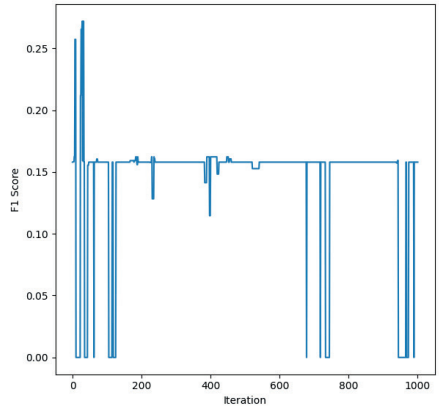
(b) F1-Score in detecting Sitting activity



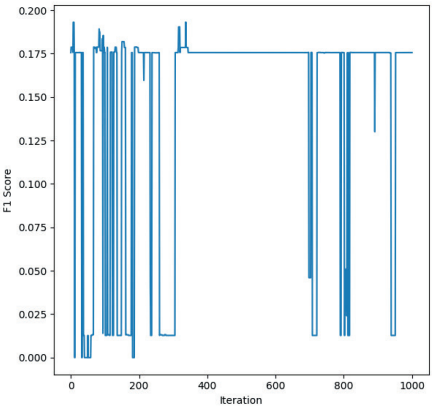
(c) F1-Score in detecting Standing activity



(d) F1-Score in detecting Walking activity



(e) F1-Score in detecting Running activity



(f) F1-Score in detecting Cycling activity

Figure 5.9: F1-Score convergence for six-activity detection in the RFV baseline as Simulated Annealing iteratively optimizes thresholds for one client.

adjusts rule thresholds to optimize performance within the RFV strategy. To mitigate the effects of random variations and enhance the reliability of our results, we executed the procedure with various initiated rules multiple times on the local data. The figure presents a representative sample of these executions, showcasing one instance for each activity.

As illustrated in Figure 5.9, simulated annealing effectively enhances F1-scores across all activity types by navigating the solution space and finding global optima. While the F1-scores fluctuate with local and global optima at different points, the algorithm successfully identifies solutions that maximize the F1-score overall, particularly in activities such as Walking and Running, demonstrating its capability to avoid suboptimal solutions and achieve globally optimal performance.

The observed F1-score of 0.17 primarily stems from a low precision value (below 0.1), indicating a high rate of false positives. This suggests that the model is incorrectly classifying activities. Conversely, the high recall value (above 0.9) implies that the model accurately detects almost all instances of activities. The combination of high recall and low precision suggests that the random selection process (i.e., RFV) is leading to the creation of overly general and inaccurate rules. These rules are likely detecting a broad range of activities as a target activity, resulting in a high recall rate but a significant number of false positives (low precision).

While simulated annealing enhances detection accuracy (e.g., by up to 100% in Standing activity), the current F1-Score range of 0.18 to 0.3 remains insufficient for a real-time event detection system. This underscores the criticality of carefully crafting CEP rules that select the most pertinent features and assign appropriate initial threshold values. To achieve acceptable detection accuracy, we propose leveraging the combined power of LLMs for rule generation and prompt engineering techniques to obtain more precise and focused rule initiation.

To address the limitations of random rule initiation, we introduced another baseline strategy, termed *RLP*, which randomly selects a combination of LLMs and prompt engineering techniques. The RLP strategy significantly improved the detection accuracy of the initial rule version, in most cases even doubling the detection accuracy of the RFV strategy after applying simulated

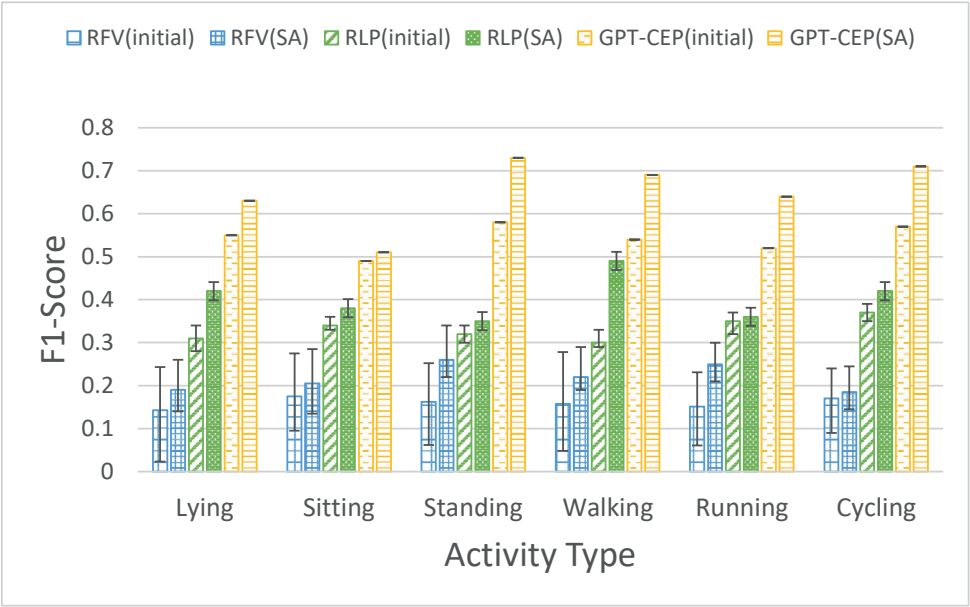


Figure 5.10: Improving the F1-Score for detecting multiple activities by using Simulated Annealing (SA) to adjust thresholds for three strategies: RFV, RLP, and GPT-CEP, in a single client. The results show that GPT-CEP outperforms the baseline strategies, achieving a higher F1-score, thanks to the advantages provided by the AGES Index.

annealing. By incorporating domain knowledge through the use of LLMs, RLP mitigates the pitfalls of random rule generation. This CEP rule generation strategy ensures that more relevant features are selected while irrelevant ones are filtered out, bringing us closer to achieving accurate autonomous rule generation.

However, one might question whether a random selection of LLMs and prompting techniques could lead to opting for zero-shot approaches that significantly underperform compared to their few-shot counterparts. Despite the improvements achieved through the RLP strategy, the average F1-score of 0.3 for initial rules and 0.4 after simulated annealing remains insufficient for real-time processing systems. This signifies the importance of leveraging the AGES Index value. Figure 5.10 illustrates the F1-scores of three mentioned

rule initiation strategies.

For overall activity recognition accuracy, GPT-CEP consistently outperforms the baselines, demonstrating the benefits of incorporating domain knowledge through LLMs and leveraging few-shot learning techniques as captured by the AGES Index metric. As observed in Figure 5.10, in cases like Walking activity, RLP might occasionally match the performance of GPT-CEP before simulated annealing. This is often due to the frequent selection of few-shot approaches. However, on average, the AGES Index consistently chooses the optimal option without the randomness of random selection.

5.6 Related Work

Autonomous rule generation in complex event processing (CEP) systems has emerged as a promising research area, aiming to alleviate the burden of manual rule definition by domain experts. Recent advancements in large language models (LLMs) and deep learning techniques have paved the way for innovative approaches in this field [233].

Early research efforts in autonomous rule generation focused on rule extraction from existing event logs or data repositories [17, 31]. These methods, often based on association rule mining or frequent pattern analysis [32], were limited in their ability to generate complex and context-aware rules and struggled to handle temporal dependencies and complex event [20, 162, 234].

More recent studies have leveraged deep learning architectures to address the limitations of earlier approaches [19, 35, 170]. Recurrent neural networks (RNNs) and long short-term memory (LSTM) networks have been employed to capture temporal dependencies in event sequences and generate rules based on learned patterns [18]. Several studies [161,235] have been proposing LSTM-based models for rule generation, demonstrating improved performance compared to traditional rule extraction methods. In addition to deep learning approaches, heuristic-based methods such as swarm intelligence algorithms have been explored to automate CEP rule mining, leveraging biologically inspired techniques to uncover hidden causal and temporal relationships in streaming data [33,236]. However, these methods often require large amounts of labeled data for training, which can be challenging to obtain in certain domains.

The emergence of Transformer-based LLMs has revolutionized the field of natural language processing, demonstrating significant promise for autonomous rule generation. LLMs can generate human-readable rules by learning from vast amounts of text data. An LLM-based approach for emotion recognition based on rule-based systems has been proposed in the literature [237], demonstrating its ability to generate complex and context-aware rules. However, LLMs can be susceptible to biases present in the training data, leading to the generation of unfair or discriminatory rules.

While significant progress has been made in autonomous rule generation for CEP systems, several challenges remain. One of the key challenges is ensuring the quality and interpretability of the generated rules. Techniques such as rule evaluation and explanation are essential for verifying the correctness and understanding the rationale behind the generated rules. Additionally, addressing issues such as bias and fairness in rule generation is crucial to ensure the ethical and responsible deployment of these systems [233], especially as LLMs may inherit biases from their training data that could propagate into CEP decision-making.

5.7 Summary

This chapter introduces GPT-CEP, an adaptive framework for autonomous rule generation in complex event processing (CEP) systems. By integrating large language models (LLMs) and prompt engineering, GPT-CEP automates rule initiation, reducing reliance on domain expertise while achieving initially promising accuracy without model fine-tuning. To systematically evaluate its effectiveness, we introduce the AGES Index, a composite metric that assesses rule generation efficiency, accuracy, and scalability. Beyond rule initiation, GPT-CEP employs a cluster-based rule refinement mechanism, grouping related rules and optimizing their thresholds through simulated annealing. This adaptive approach improves detection accuracy while maintaining computational efficiency by iteratively refining rule parameters in response to local updates. Experimental results show that we could accomplish acceptable performance in the form of the F1-score, demonstrating GPT-CEP's effectiveness in autonomous rule generation. The achieved F1-score of approximately

0.7 reflects the system's ability to refine rules dynamically through local updates, improving detection accuracy. The performance gains over baseline approaches highlight GPT-CEP's adaptive rule refinement, suggesting further potential for optimization through fine-tuning and additional adaptation strategies.

Future research will focus on expanding the scope of LLM exploration, including task-specific model fine-tuning to improve rule coherence and domain relevance. Additionally, meta-learning techniques could optimize prompt engineering dynamically, ensuring adaptability to different event processing tasks. In rule refinement, alternative optimization techniques beyond simulated annealing, such as genetic algorithms and reinforcement learning, will be explored to further improve rule accuracy. Another key direction is federated rule optimization, where models learn and refine rules collaboratively across distributed environments while preserving privacy. To address scalability, graph-based rule aggregation will be investigated to consolidate redundant or overlapping rules more effectively. Lastly, integrating real-time adaptive tuning of rules based on streaming data characteristics will be a priority, enhancing GPT-CEP's responsiveness to evolving event patterns in large-scale deployments.

Chapter 6

Federated Stream Processing: An application in Vehicular Networks

When trials loom and shadows grow, Together stand, let courage show. For hearts in union bear the strain, And turn to gold the toil and pain.

Ferdowsi (Revised by GPT4o)

Abstract

This chapter introduces the concept of object detection model training in vehicular networks using Clustered Federated Learning (FL) as an application of AI in stream processing systems (the PUT problem). Here, we propose AR-CFL, an Adaptive Resource-aware Clustered Federated Learning framework. AR-CFL enhances system efficiency through dynamic adjustments and universal access to refined detection models while preserving data privacy. Evaluation results show robust detection performance despite limited storage, with superior training performance compared to certain classical FL scenarios.

Autonomous driving comes with the promise of making vehicles' movements more predictable and less reliant on the drivers' decisions, hence increasing road safety and traffic efficiency. However, today's vehicles have a narrow perception of the environment due to limited onboard sensing [50]. To cope with this, exchanging collected data among vehicles (and all road users) can help achieve a better perception of the environment [51]. The evolving Vehicle-to-everything (V2X) technologies provide a means of communication between road users and enable them to collect and aggregate perception data cooperatively, in the so-called Collective Perception [238].

Deep Neural Networks (DNNs) have a pivotal role in individual perception and object detection in autonomous driving. These networks usually undergo centralized training before deployment, utilizing data that is limited in its coverage. Consequently, DNN models trained on such data may exhibit low performance in object detection, even with quality management techniques in place [28, 122]. To overcome this limitation, continuous online model training can be leveraged to enhance adaptability and ensure robust object detection performance for fully autonomous driving across diverse conditions [51].

FL is a technique to train DNN models from distributed data sources (e.g., using the computational resources of each road user). With FL, a central server maintains DNN models that are updated with the incremental changes in its parameters provided by the participants/clients. Since the data sent to the central server is usually much smaller than the raw training data, FL reduces communication requirements. Additionally, it protects (up to some level) the data owners' privacy, which is important in many IoT applications (e.g., [29, 36]), by keeping the raw data stored locally.

Recent studies demonstrate that FL techniques can train DNN with optimized client-server interactions [239]. Moreover, FL is a good paradigm for implementing collective perception techniques among heterogeneous resources with non-IID data (i.e., Not Independent and Identically Distributed, data gathered by each vehicle has unique characteristics, distributions, or biases due to factors like diverse routes, driving behaviors, or environmental conditions). With these benefits in mind, continually training a DNN employing FL may solve some of the challenges of autonomous driving.

Nevertheless, the communication requirements between participants and the central server in FL may still be too high in certain conditions. Hence, *clustering* can be used to alleviate communication overhead. *Clustered FL* works by grouping clients nearby into a cluster, which requires extensive coordination and synchronization between the involved entities. But, it reduces the communication requirements between clients and the central server, since most of the training data exchange occurs inside clusters [240]. Moreover, it can help enhance the performance of the FL approach by decreasing the time required to train the model with a considerable level of perception capability.

To leverage the benefits of Clustered FL at its fullest, we present an Adapt-

ive Resource-aware Clustered Federated Learning framework, referred to as AR-CFL, specifically designed to explore and optimize factors impacting online learning and communication comprehensively needs within vehicular environments. Our innovative framework incorporates adaptive mechanisms to optimize system efficiency dynamically. Additionally, leveraging AR-CFL, we investigate the training of a DNN vehicle detection model on non-IID data under diverse conditions. We systematically compare and discuss the outcomes obtained under different design decisions and configuration options. In summary, this chapter's contributions are:

1. A novel framework (AR-CFL) that extends the capabilities of FL with adaptive clustering to provide hierarchical FL (improving existing Clustered FL solutions). This leads to boosting environment perception capability (i.e., detection performance), reducing the volume of the exchanged data, and providing a fast-converging training process.

2. A novel *Dynamic Sampling* concept, introduced to more realistically consider the storage limitation of vehicles in V2X networks.

3. A new *Dynamic Cluster Members Involvement* strategy to dynamically adjust the number of clients participating in the learning process in each cluster.

4. Developing two algorithms that utilize *Dynamic Cluster Members Involvement* to enable both inter-cluster and intra-cluster object detection within a hierarchical federated learning framework.

5. The evaluation of AR-CFL with the generated synthetic datasets provides interesting results and conclusions.

6.1 Related Work

In this section, we review the literature in two key areas, *Object Detection* and *Clustering*, both using FL in vehicular context, identifying the research gaps.

6.1.1 FL-based Object Detection in Vehicular Context

How FL affects vehicular environments has been explored vastly in the literature showing comparable performance with traditional centralized learning while preserving user's privacy [241]. The benefits can vary from improving the detection performance [242] to training models over heterogeneous data sources (i.e., vehicle's onboard sensors) [243] by dynamically adjusting local training iterations and using model compression to reduce communication overhead during model exchanges [244]. However, resource allocation remains challenging and can be alleviated by optimizing network management with a multi-layer graph (e.g., [245]). On the other hand, trading some privacy for higher utility can be achieved by opting for clients with sufficient resources, while others send their datasets to the central server [246]. Even with FL, privacy still might be violated in the model exchange or client selection phases. However, initial solutions such as multi-layer context-aware client selection and aggregation [247] degrade privacy violations.

6.1.2 Clustered Federated Learning in Vehicular Context

Adapting to the dynamic client diversity in different vehicular network topologies can be achieved by a weighted inter-cluster cycling update algorithm [248]. In addition, imbalanced and distribution-shifted training data was handled by a flexible CFL framework that groups clients based on optimization direction similarities to reduce training divergence [249]. Cluster formation in FL can be done based on client data distribution, e.g., by incorporating game theory principles [250] or considering the benefits of platooning [251]. Also, CFL-based object detection techniques in vehicular networks have been studied briefly by utilizing vehicular-to-vehicular (V2V) resources to bypass the communication bottleneck (cf., [252]). Lack of adaptivity by the static formation of clusters (e.g., vehicles under a base station's coverage) and the limitation of dealing with mobility (i.e., vehicle handover) to enable continuous training in similar research works indicate that further studies are required.

6.1.3 Research Gaps

Several noteworthy research gaps necessitate further investigation, outlined as follows:

Application-Communication Network Integration: A notable gap exists in the field of CFL, particularly in examining the relationship between exchanged data volume, influencing communication overhead, and the associated impact on application-related performance.

Limited Storage Consideration: Existing CFL techniques overlook the consideration of limited storage on vehicles, which leads to unrealistic performance evaluation.

Training on Freshly Collected Data: The imperative need for training models with freshly collected data over successive iterations rather than static datasets is insufficiently addressed in current CFL approaches.

Influence of Varied Traffic Densities: The literature lacks exploration into the impact of varying traffic densities on online training systems within the context of CFL.

Dynamic Clustering Participation: The evaluation of CFL approaches has not encompassed the exploration of varying cluster counts and the involvement of a diverse number of vehicles within clusters.

6.2 Object Detection Model Training in Vehicular Networks

We demonstrate how to benefit from involving the clustering concept in designing a two-level FL approach in the online model training of object detection scenarios applicable to autonomous driving. Although FL is proven capable of assisting object detection [51], the amount of exchanged data and the convergence time are considerably increased to achieve an acceptable detection capability level. Hence, achieving optimum values for both factors requires modifying the conventional FL mechanism. When a cluster of vehicles is selected as an FL client, a new set of learning rounds will be initiated within the cluster, called Intra-Cluster Federated Learning. The trained model is exchanged directly between cluster members using V2V communication. It experiences multiple rounds of training within the cluster before

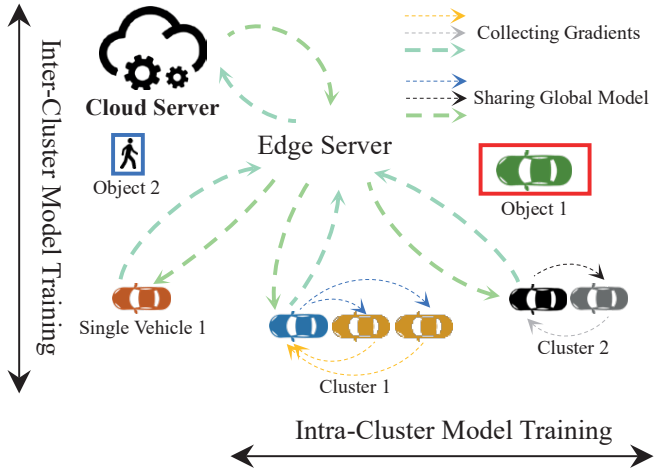


Figure 6.1: An example of improving object detection's performance by employing a two-level federated learning.

being shared with the server. This approach reduces data exchange via cellular communication. Additionally, stable direct V2V communication within the cluster decreases convergence time. As depicted in Figure 6.1, the training involves three primary entities:

Road User. A vehicle or any other entity participating in the process that generates data streams about the surrounding environment is called a *road user*. Each road user has its own onboard sensing configurations (e.g., a *camera* or *LIDAR* sensor) [253]. By merging the list of detected objects with the spatio-temporal information, a road user can build its own *local environment model* that will be used in subsequent decision-making processes. Also, vehicles can exchange perception data, extending spatial awareness above their limited perception. Such perception data is encapsulated in collective perception messages [254] that can be exchanged directly between road users or through any intermediate node (e.g., an edge server) [255]. For the sake of simplicity, we assume that all road users have similar sensing deployment to generate the same data type and operate with identical environment models.

Edge Server. In a conventional scenario, an *edge server* acts as a simple base station that only forwards data. Besides communicating with road users, edge servers can also exchange data with each other [256, 257]. That is why our mechanism not only provides a collective perception for road users who are within the communication range of each other but also beyond such a spatial limitation.

Cloud Server. The principal functionality of this component is to orchestrate the whole process of collective perception in object detection. It is responsible for initiating the detection task that should be performed based on the road users' collective perception. In this regard, the cloud server is working closely with the edge servers to distribute the detection models among them. In addition, it benefits from the results of the learning process performed in the edge servers by combining the learning parameters.

Each cluster can be formed by moving road users (e.g., a combination of moving vehicles) and by a set of non-moving users (e.g., a group of parked vehicles plus those waiting behind a traffic light). Notice that increasing the number of cluster members is a double-edged sword: It increases the computing capabilities and membership dynamicity. Each cluster has a Cluster Head (CH), that manages and coordinates the rest of the Cluster Members (CMs) and exchanges information with them via the Wi-Fi Direct (ad-hoc) network.

6.3 AR-CFL System Design

In Figure 6.2, we illustrate the main components of AR-CFL. When a user issues a *task* to the central server, a global model is generated to be trained online. In our case study, a user could be a car manufacturer who aims to improve the perception models of his vehicles. For example, if the *task* aims to detect *object 1* in Figure 6.1, the model concerns the dimensions and position of this object. Then, the global model will be pre-trained by a random sample of vehicles' data. Next, the central server decides the required vehicles based on this specific task. Finally, the new global model will be generated by aggregating the model updates trained in the selected clients. The entire learning process in this step is called Inter-Cluster Federated Learning

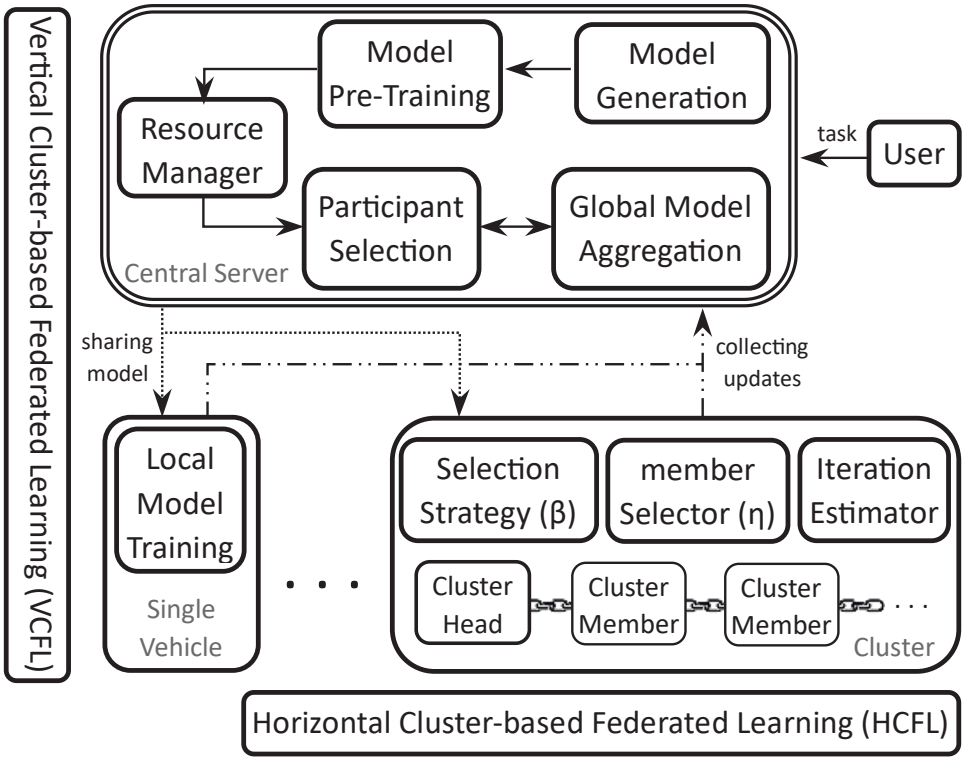


Figure 6.2: The AR-CFL system design, which includes the main components of both levels of federated learning.

(RCFL), and it includes the processing flow from the central server (i.e., in the cloud layer) to the participants (i.e., vehicles in the edge layer). The second level of learning is within each cluster, called Intra-Cluster Federated Learning (ICFL). Here, the cluster head decides the required number of local iterations (i.e., training rounds within the cluster) according to the cluster's available computing resources.

Each task requires a particular amount of data and computing resources to be executed. Therefore, the *Resource Manager* module aims to adjust control variables β (in-cluster participant selection strategy) and η (number of in-cluster participants) to match the required computing resources for each task.

These parameters collectively determine the total number of participants N_n. Please refer to Table 6.1 for a comprehensive list of the evaluation parameters utilized in our study.

6.3.1 RCFL Framework

In Algorithm 6.1, we present the procedure of client selection and synchronized online learning from the cloud server point of view. We chose Federate Averaging (FedAvg) [258], a pioneering aggregation approach that achieves better accuracy in previous studies [259], in which a central server (e.g., cloud server in the proposed scenario) hosts the shared global model ω_g, where g stands for the global iteration number in the first level of FL.

At the start of RCFL, the list of available clients and minimum required resources for the task are initialized. Besides, the central server pre-trains the global model (i.e., ω_0) by utilizing a small set of data gathered from all potential participants. Moreover, the participant selection strategy within the cluster (i.e., β) is determined here by the central server and sent to the cluster heads so that all clusters perform the selection uniformly. Various selection strategies have been introduced in the proposed approach that will be elaborated on further. Also, the number of participants that should be chosen in the specified selection strategy is determined here using the η value.

Unlike FedAvg, which chooses random clients, this approach selects as many cluster participants as possible. To this end, it selects clients with the highest computing resource values (i.e., $CR_{c_{max}}$), as clusters usually have higher computing resources. It continues selecting clients until their computing summation (i.e., R_g) reaches the threshold required for the task. Then, each chosen participant receives the global model ω_g and replaces its current local model ω_g^c.

Suppose the client is of a type of cluster. In that case, the second level of FL is started within the cluster (see Algorithm 6.2). In the end, the cluster head uploads the cluster's aggregated update to the central server. Otherwise, if the client is a single vehicle, it partitions the local data into batches of size B and repeatedly applies the model to these data blocks for E number of iterations, e.g.,, using Stochastic Gradient Decent (SGD). This generates the updated local model ω_{g+1}^c, which is uploaded to the central server. Finally,

Algorithm 6.1 RCFL Procedure

1: **Initialization:**
  $\mathcal{C} \leftarrow \{[c_1, CR_{c_1}], ..., [c_n, CR_{c_n}]\}$; ▷ clients
  $CR_{task} \leftarrow$ Minimum Required Resources for *task*;
  Pre-trained ω_0; ▷ Initial model
  $\beta \leftarrow$ In-Cluster Participant Selection Strategy;
  $\eta \leftarrow$ Number of In-Cluster Participants;
2: **for** global iteration $g = 0, 1, ...$ **do**
3: $Update(\mathcal{C})$;
4: $R_g \leftarrow 0$; ▷ Sum of Computing Resources
5: $E_\sigma \leftarrow 0$; ▷ Sum of Client's Weights
6: $C_g \leftarrow \varnothing$; ▷ Selected Clients
7: $\mathcal{C}' \leftarrow$ Sort $(\mathcal{C}, CR)$; ▷ Descending Sorted Client Set
8: **while** $R_g \leqslant CR_{task}$ **do**
9: $c_{max} \leftarrow \mathcal{C}'[0]$; ▷ Client with the highest CR value
10: $C_g \leftarrow C_g \cup c_{max}$;
11: $R_g \leftarrow R_g + CR_{c_{max}}$
12: $\mathcal{C}' \leftarrow \mathcal{C}' - \mathcal{C}'[0]$;
13: Distribute ω_g to clients in C_g; ▷ Global model
14: **for** client $c \in C_g$ **do** ▷ In Parallel
15: $\omega_g^c \leftarrow \omega_g$;
16: **if** c is a cluster **then** ▷ In-cluster training
17: $\omega_{g+1}^c \leftarrow$ ICFL(ω_g^c, β, η); ▷ Algorithm 6.2
18: **else** ▷ Single client training
19: $P_c \leftarrow$ batches of size B; ▷ Data Partitions
20: $E_c \leftarrow |P_c|$; ▷ Number of training rounds for c
21: **for** partition $p \in P_c$ **do**
22: $\omega_g^c \leftarrow$ LocalTraining (ω_g^c, p);
23: $\omega_{g+1}^c \leftarrow \omega_g^c$;
24: $E_\sigma \leftarrow \sum_{c \in C_g} E_c$;
25: $\omega_{g+1} \leftarrow \sum_{c \in C_g} (E_c / E_\sigma) \times \omega_{g+1}^c$; ▷ Trained model

the received trained local models are aggregated in the central server using a weighted sum into the new global shared model ω_{g+1}. Notice that the weight for each locally trained model is calculated based on the number of performed iterations for each client c (i.e., E_c) over the total iterations in this global training round (i.e., E_σ). This way, more importance is given to the cluster

updates, which were trained with more local training iterations.

6.3.2 ICFL Framework

The second level of FL is illustrated in Algorithm 6.2. After initializing the cluster member set and the local training model, in each local training iteration, the cluster head opts for the set of participants according to the in-cluster participant selection strategy (i.e., β) and also the η value. There are three selection strategies for selecting in-cluster participants β: *Full Aggregation*, *Random*, and *MaxLabel*. When $\beta = FullAggregation$, all cluster members would participate, and models are aggregated at the cluster head. On the other hand, when $\beta = Random$, FL clients are randomly selected in each iteration. Finally, when $\beta = MaxLabels$, cluster members with the most labels (data-rich) are selected (See Figure 6.3).

In both *Random,* and *MaxLabel* strategies, η specifies how many members will be involved in the local training within each cluster. The central server determines η based on various conditions (e.g., task, vehicle's computation capabilities, etc.). E.g., $\eta = 2$ means that two clients from each cluster will be selected to participate in this local iteration. When having $\eta = 2$ and the $\beta = MaxLabels$, two cluster members with the largest label count in that specific cluster are selected. In the case of *FullAggregation* setup, η equals the total number of cluster members, including CH. In the next step, each selected member trains the model with fresh data and returns its update to the cluster head. The collected updates will be aggregated into a new model and be used for the next local iteration in this cluster. Once the training rounds are finished, the cluster head returns the last aggregated update to the central server as a result of this round of global training procedure (i.e., RCFL).

6.3.3 Handling The Limited Storage Challenge

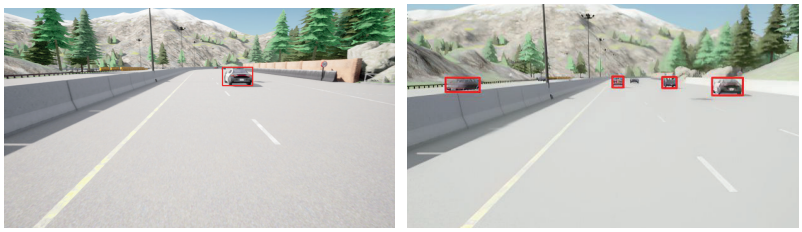
Unlike the conventional FL approaches, we take into account the constrained storage capacity of the participating vehicles by purging the utilized data in each training round, making room for the acquisition of fresh data. This mechanism reflects a heightened degree of realism and effectiveness, particularly in the context of vehicular Federated Learning scenarios, where we factor in the

Algorithm 6.2 ICFL Procedure

Initialization:
  $Cluster \leftarrow \{[m_1, CR_{m_1}], ..., [m_n, CR_{m_n}]\}$;
  $\omega_0^{Cluster} \leftarrow \omega_g^c$; ▷ Initial model in cluster
for local iteration l = 0, 1, ..., k **do**
  $C_l \leftarrow \varnothing$; ▷ Selected members for training
  if $\beta == Full\_Aggregation$ **then**
    $C_l \leftarrow Cluster$;
  else if $\beta == Random$ **then**
    for $i \leftarrow 1$ to η **do**
      $c_i \leftarrow Random(Cluster)$;
      $C_l \leftarrow c_i$;
      $Cluster \leftarrow Cluster - c_i$;
  else if $\beta == MaxLabel$ **then**
    for $i \leftarrow 1$ to η **do**
      $c_i \leftarrow \max_{m_j \in Cluster} LabelCount_{m_j}$;
      $C_l \leftarrow c_i$;
      $Cluster \leftarrow Cluster - c_i$;
  Distribute $\omega_l^{Cluster}$ to clients in C_l;
  for $c' \in C_l$ **do** ▷ In Parallel
    $\omega_l^{c'} \leftarrow \omega_l^{Cluster}$;
    $\omega_{l+1}^{c'} \leftarrow$ LocalTraining $(\omega_l^{c'})$;
    $\Omega \leftarrow \Omega \cup \omega_{l+1}^{c'}$; ▷ Set of Collected Updates
  $\omega_{l+1}^{Cluster} \leftarrow Aggregate(\Omega)$;
return $\omega_k^{Cluster}$; ▷ Final in-cluster trained model

inherent limitations of onboard storage and the dynamic real-time conditions surrounding the participating vehicles.

Besides, *Dynamic Involved Members* concept helps the central server to be flexible in the number of cluster members that would participate in each round of training. By combining parameters β and η, clients of the second level of learning can be adjusted according to the learned insights from the previous iterations. For example, the size of the FL participant set can be decreased to save resources in case the detection performance is not improved by involving more cluster members.



(a) Image contains one label. (b) Image contains four labels.

Figure 6.3: Example of two image samples. Here, the image in (b) is more data-rich than the image in (a).

6.4 Evaluation

In this section, we explore the scenario of online car detection model training using our AR-CFL framework. Our investigation focuses on two key aspects: *online learning efficiency* and *communication overhead*. The evaluation covers the following:

1. We analyze how varying traffic density influences the overall system performance.
2. We investigate how CFL enhances the efficiency of online learning in terms of communication overhead and learning efficiency, as compared to the centralized learning approach and to the classical FL approach.
3. We explore the influence of various selection strategies (i.e., β), the number of participants in the cluster (i.e., η), and the number of clusters (i.e., N_{cls}) on the overall performance.

Our evaluation considers several restrictions regarding vehicle equipment and data distribution:

1. **Non-IID Data:** We used non-IID data, with a clear characterization of data heterogeneity across system members.
2. **Environmental Considerations:** The evaluation is carried out under various environmental conditions to assess the robustness and adaptability of the CFL approach.

Table 6.1: Evaluation parameters with descriptions.

Parameter	Description
β	In-cluster member selection strategy
η	Number of participants in cluster
N_v	Total number of vehicles
N_n	Number participating vehicles
α	Traffic density (30, 50, 100)
N_{cls}	Number of clusters
E_g	Number of global iterations (epochs)
g	Global iterations index
E_l	Number of local training iterations
b_s	Batch size
$lr0, lrf$	Learning rate parameters
tr_t	Total training time
e_d	Total volume of exchanged data for training
ed_l	Data volume (cellular communication)
ed_s	Data volume (direct V2V communication)

3. **Communication Assumptions:** We assume that communication between vehicles within the same cluster is easier to establish and less costly than communicating with edge or cloud servers (cellular communication).

6.4.1 Evaluation Scenario and Experimental Setup

We outline now the evaluation scenario and experimental setup aimed at assessing our approach. Our focus lies in training a car detection model using image data captured from the participating vehicles. To meet the requirement of having image data from multiple vehicles in similar conditions, we created a synthetic dataset using the Carla simulator [260]. The experiments employed the *YoloV8n* model [261] as a car detection model. We use a Linux server with an NVidia RTX3090 Ti GPU for running the experiments. We list all evaluation parameters with their descriptions in Table 6.1.

Our study involves data collection in various weather and lighting scenar-

(a) clear day-time

(b) rainy day-time

(c) clear night-time

(d) rainy night-time

Figure 6.4: Examples of weather and lighting conditions considered in our study. The generated datasets are characterized by well-balanced distributions, ensuring that each condition constitutes approximately 25% of the total dataset samples.

ios (as illustrated in Figure 6.4). These include clear weather day-time, rainy weather day-time, clear weather night-time, and rainy weather night-time. Experiments were conducted using a combination of these conditions.

The total vehicle count is set at ($N_v = 12$). We vary traffic density (α) with values of 30, 50, and 100, where $\alpha = 50$ indicates the presence of 50 vehicles. These vehicles differ from the aforementioned 12 data collector vehicles participating in FL model training. Furthermore, we investigate various scenarios by adjusting the number of clusters (N_{cls}) to either 2 or 4, evenly distributing the clients across the clusters.

We benchmark our approach against two main baselines:

- *Centralized*: Represents the optimal oracle scenario where all data collected from vehicles is centrally stored and used for model training.

- *ClassicalFL*: In this case, no clustering is considered, and all FL clients

(vehicles) are at the same level, and communicating directly with the central aggregation server using cellular communication.

Federated Learning Hyper-parameters

The total number of global iterations was $E_g = 50$. Upon receiving the model, each client engaged in $E_l = 100$ local iterations on the currently available chunk of the local data. The batch size was set to $b_s = 16$. We established the learning rate parameters with $lr0$ and lrf, configured at their default values of $lr0 = lrf = 0.01$. Also, we set $optimizer = auto$ while maintaining default values for all model training and validation parameters [262].

We considered the following evaluation metrics to compare approaches.

Detection performance. The two key metrics often used to evaluate the detection model's performance are:

- mean Average Precision (mAP): This metric considers precision and recall across multiple object classes [263]. mAP is particularly valuable because it considers the object detection performance at different confidence score thresholds, making it a robust evaluation metric. In our study, we measure mAP50 (IoU$\geqslant$0.5).

- F1 Score: We use the F1 score as a supportive metric to measure the trained model's detection performance.

Training time. The total training time is denoted as tr_t and measured in minutes. We omitted the model exchange time for the sake of simplification. Moreover, we excluded the selection time for participating clients. We considered the actual model training time and the model aggregation time.

Communication Overhead. We define e_d to measure the size of the exchanged data while neglecting the generated traffic to select the participating clients in the CFL setups. In addition, we omitted all the other Collective Perception Message loads for simplicity. In the case of *Centralized* setup, e_d is calculated by measuring the size of the data samples (images) that are sent

from the N_v vehicles to the server, as follows:

$$e_d = \sum_{i=1}^{N_v} \sum_{j=1}^{k_i} data_s(i,j)$$

where k_i is the number of data chunks collected in vehicle i, and $data_s(i,j)$ is the data size j from the vehicle i. On the other hand, for *ClassicalFL* setup, we exchange the models instead of raw data. The exchanged data volume here is relevant to the number of selected clients N_n in each global iteration g, Upon finishing the training on the number of local iterations E_l, each selected vehicle sends the model back to the server. In this case, the final formula to calculate e_d is as follows:

$$e_d = \sum_{g=1}^{E_g} 2 \times N_n \times model_s$$

where $model_s$ indicates the model size.

Finally, we consider two-level aggregation while computing data exchange volumes in the different CFL setups. This involves two communication types, cellular and direct V2V communication. Cellular communication is required between the server and cluster heads. By minimizing data exchange in this costly and delayed communication type, the overall system efficiency improves. The bandwidth cost, denoted as ed_l, is computed as

$$ed_l = \sum_{g=1}^{E_g} 2 \times N_{cls} \times model_s$$

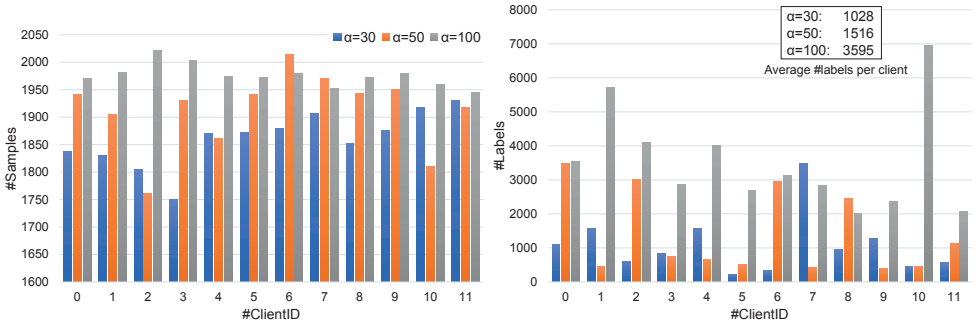
replacing N_n with N_{cls}.

On the other hand, direct V2V communication is required between cluster heads and members is faster and less costly. The bandwidth cost for this communication type, denoted as ed_s, is calculated as

$$ed_s = \sum_{g=1}^{E_g} 2 \times N_{cls} \times \eta \times model_s$$

Finally, the value of e_d is computed by summing the ed_l and ed_s values as follows:

$$e_d = ed_l + ed_s$$



(a) Sample distribution over vehicles. (b) Label distribution over vehicles.

Figure 6.5: Data distribution statistics of the clients $N_v = 12$. We generated three datasets with varying traffic densities $\alpha = 30, 50, 100$.

6.4.2 Data Generation

We used the Carla simulator [260] to generate training and validation data, building upon [264] for concurrent image and ground truth data generation from multiple vehicles. Simulations used the pre-built Map-Town04 [265], and data for each detection task was uniformly generated.

Data Distribution Statistics

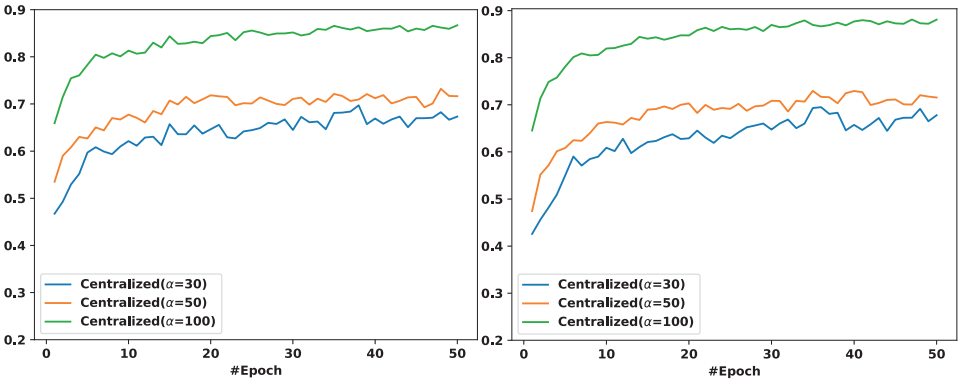
Figure 6.5a visually illustrates sample (image) distribution across clients for different traffic densities α, totaling 22,327 for $\alpha = 30$, 22,948 for $\alpha = 50$, and 23,715 for $\alpha = 100$. Differences in sample sizes are negligible relative to the complete dataset.

Figure 6.5b depicts label (car bounding box) distribution across the clients' samples for different traffic densities across all iterations. Higher α values result in increased total label count.

What is important to be observed from Figure 6.5 is the minimal data quantity skew among clients, and thus, the difference in sample size can be neglected. However, a noticeable label distribution imbalance across clients highlights our consideration of non-IID data handling and illustrates data heterogeneity across clients [266, 267].

Table 6.2: Total training time tr_t of different approaches

Approach	Training time ($\mathbf{tr}_t$)
Centralized	207 minutes
ClassicalFL	101 minutes
Clustering	101 minutes



(a) F1 Score (b) Mean Average Precision

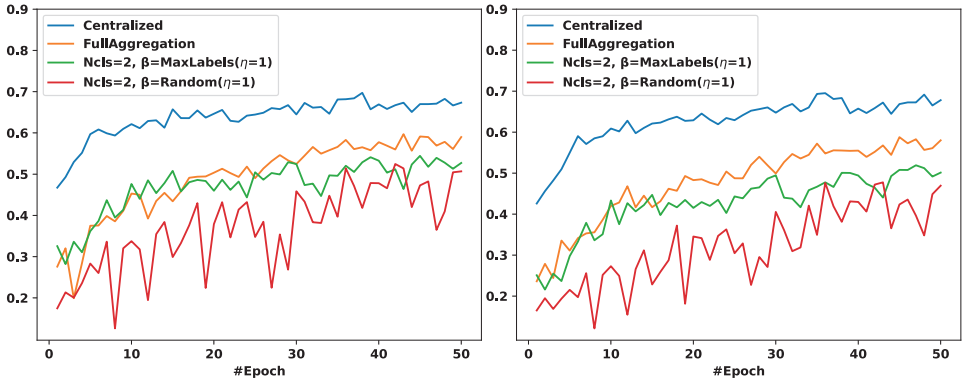
Figure 6.6: The detection performance of the *Centralized* approach is evaluated across different traffic densities with values $\alpha = 30, 50, 100$. #Epoch refers to the number of global iterations. A noticeable enhancement in performance is evident with the increase in traffic density.

6.4.3 Results and Discussion

Analyzing Traffic Density Impact on Performance

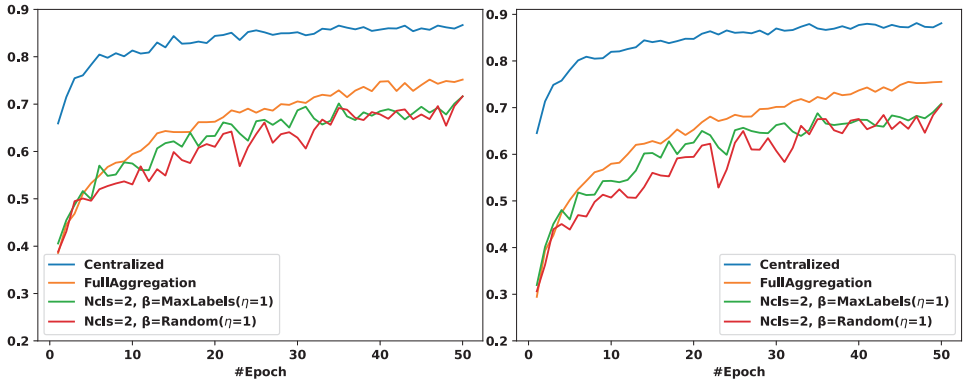
We explore the influence of traffic density on the performance, considering three different values $\alpha = 30, 50, 100$. Figure 6.6 illustrates some detection performance values in the *Centralized* approach, revealing an evident trend: as traffic density rises, there's more consistent model performance and an overall enhancement in the detection performance. Increased traffic density results in capturing more objects within the generated images, thereby enhancing the model training performance.

Notably, heightened traffic density correlates with increased detection performance without influencing training time or data exchange volume. These



(a) F1 Score (b) Mean Average Precision

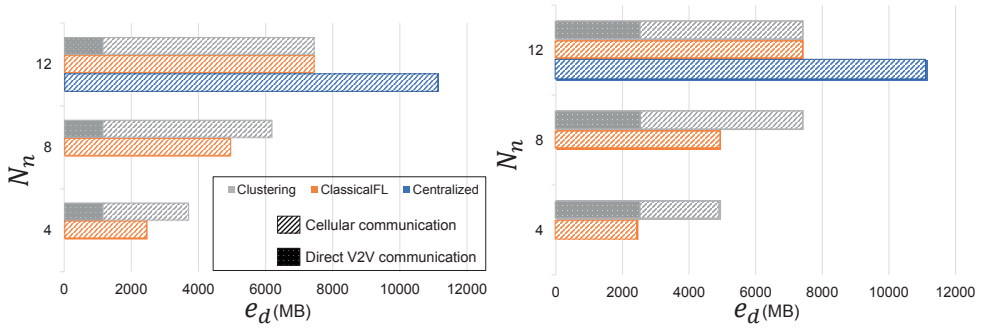
Figure 6.7: Comparing the detection performance between the *Centralized* approach and selected *Clustering* approaches under a traffic density of $\alpha = 30$.



(a) F1 Score (b) Mean Average Precision

Figure 6.8: Comparing the detection performance between the *Centralized* approach and selected *Clustering* approaches under a traffic density of $\alpha = 100$.

aspects depend solely on sample size and not on the characteristics within the samples. These observations go beyond the *Centralized* approach and are applicable across all other approaches.



(a) $N_{cls} = 2$ (b) $N_{cls} = 4$

Figure 6.9: Comparison of exchanged data volume across different approaches relative to the number of participating clients N_n and the number of clusters N_{cls}. All CFL setups exhibit consistent values across varying traffic densities.

Influence of Clustering vs. *Centralized* Approach on Online Learning Efficiency

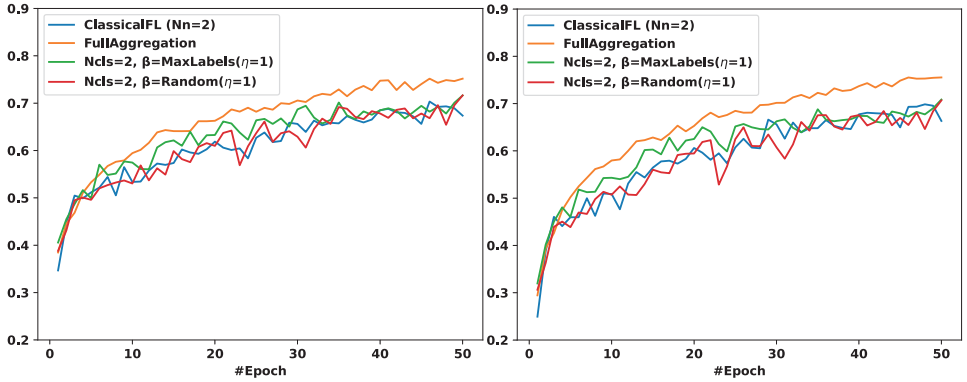
Examining the impact of CFL on online learning efficiency versus centralized learning, we emphasize the communication overhead and application-related performance. Figures 6.7 and 6.8 reveal that the *Centralized* approach consistently outperforms CFL setups in detection performance. Despite this, the gap remains constant across different α values.

The true advantage of both FL and CFL emerges in reduced training time. As illustrated in Table 6.2, both approaches demonstrate an impressive 52% decrease in training time compared to *Centralized* training approach. Furthermore, as illustrated in Figure 6.9, the CFL setups demonstrate a significant reduction of approximately 30% in the exchanged data volume compared to the *Centralized* approach when involving all clients in the training process.

Clustering vs. *ClassicalFL*

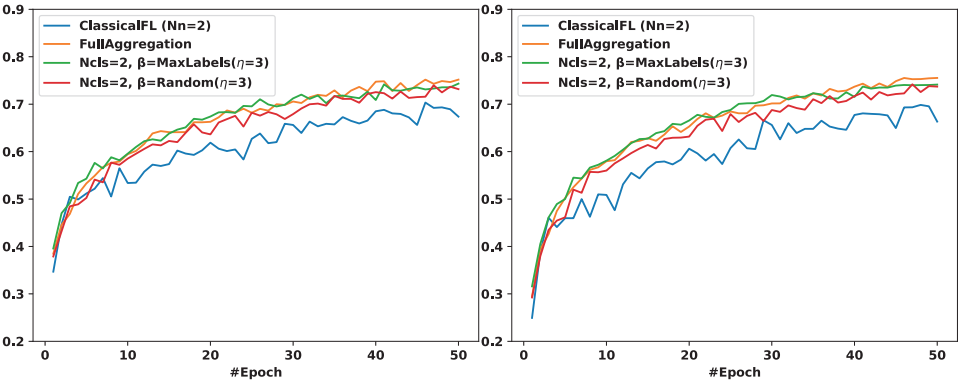
We analyze how CFL impacts online learning efficiency compared to the classical federated learning approach.

Figures 6.10 and 6.11 offer nuanced insights into online learning efficiency. *FullAggregation* and *MaxLabels* CFL strategies outperform traditional *Clas-*



(a) F1 Score (b) Mean Average Precision

Figure 6.10: Comparing the detection performance between the *ClassicalFL* approach and selected *Clustering* approaches under a traffic density of $\alpha = 100$ with one selected client at each cluster ($\eta = 1$).

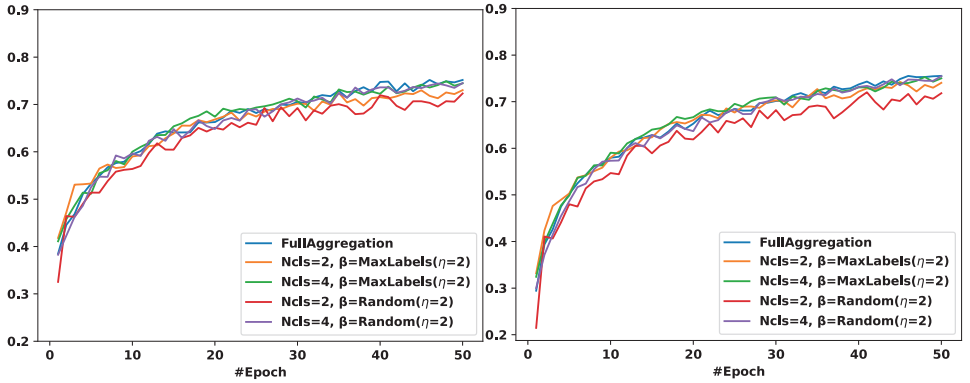


(a) F1 Score (b) Mean Average Precision

Figure 6.11: Comparing the detection performance between the *ClassicalFL* approach and selected *Clustering* approaches under a traffic density of $\alpha = 100$ with three selected client at each cluster ($\eta = 3$).

sicalFL in detection performance. *FullAggregation* involves all clients in each iteration, contrasting with *ClassicalFL*, which randomly selects a subset (N_n) of clients per iteration. However, *FullAggregation* introduces increased short-range communication (ed_s) compared to *ClassicalFL*.

MaxLabels surpasses *ClassicalFL* by selecting clients within each cluster



(a) F1 Score (b) Mean Average Precision

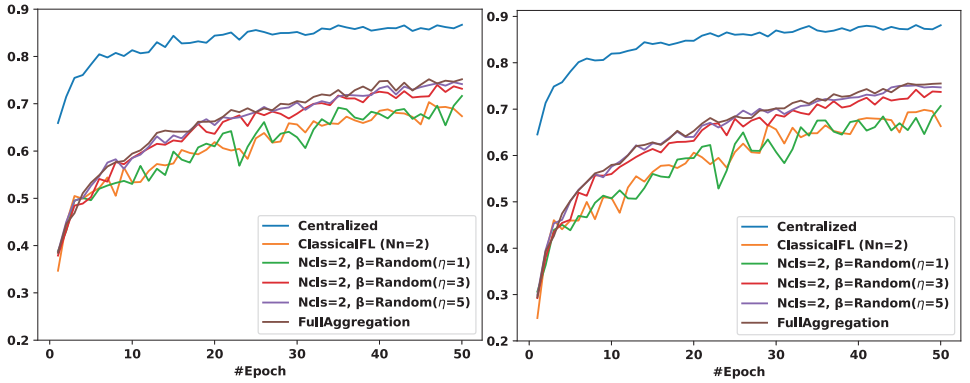
Figure 6.12: The detection performance of various *Clustering* approaches under a traffic density of $\alpha = 100$, with two selected clients in each cluster ($\eta = 2$), and varying cluster counts N_{cls}.

with the maximum labels per iteration, enhancing the detection performance and convergence. However, as illustrated in Figure 6.9, CFL introduces additional communication overhead. Nevertheless, with the increased number of participants, there is a decrease in ed_l and an increase in ed_s, highlighting the benefits of CFL over *ClassicalFL*. The *Random* CFL setup with $\eta = 1$ shows comparable performance to *ClassicalFL*, randomly selecting clients in each iteration. With increased η (e.g., $\eta = 3$), *Random* CFL outperforms *ClassicalFL* in detection performance due to more participating clients.

Impact of In-cluster Member Selection Strategy & Varying Cluster Numbers on Overall CFL Performance

We examine the influence of changing the total number of clusters (N_{cls}), diverse selection strategy (β), and involved cluster members (η) on overall system performance, using $\beta = FullAggregation$ as a baseline for comparison.

Varying Cluster Numbers (N_{cls}): As shown in Table 6.2, the training time consistently stands at $tr_t = 101$ minutes across varying numbers of clusters N_{cls}. Examining the communication overhead illustrated in Figure 6.9, we observe an increase in long-range communication overhead corresponding to the increased values of N_{cls}. This can be attributed to the increased com-



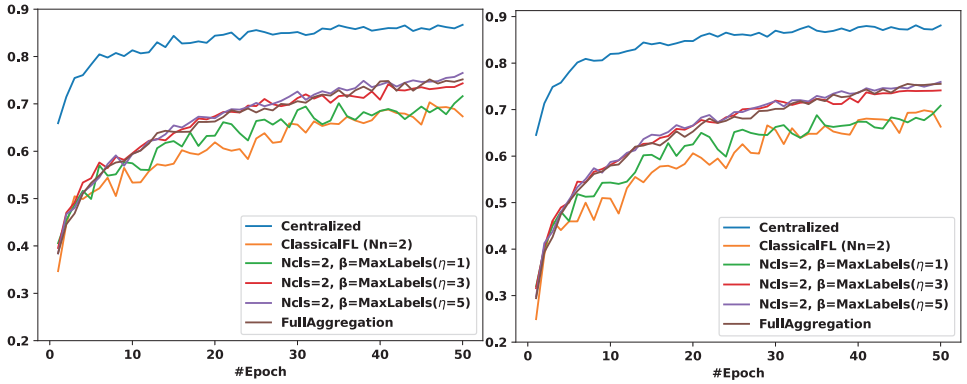
(a) F1 Score (b) Mean Average Precision

Figure 6.13: Comparing the detection performance between the *FullAggregation* and *Random* CFL approaches under a traffic density of $\alpha = 100$, with a varying number of selected clients at each cluster ($\eta = 1, 3, 5$), and two clusters ($N_{cls} = 2$).

munication overhead between the head nodes of clusters and the server. Similarly, the short-range communication overhead exhibits a rising trend with an increased number of clusters. This trend indicates a broader engagement of cluster nodes in online learning.

Turning our attention to detection performance, as depicted in Figure 6.12, we found that when $\beta = FullAggregation$, the detection performance remains constant across different N_{cls} values. This observation aligns with the intuitive expectation that all cluster nodes, including head nodes, participate in online learning regardless of the cluster count. On the other hand, when β takes values of either $\beta = Random$ or $\beta = MaxLabels$, detection performance becomes intricately linked to the parameter η. For instance, with $\eta = 2$, a $N_{cls} = 2$ configuration implies the participation of four nodes in online learning. In contrast, for $N_{cls} = 4$, eight nodes engage in the learning process. This correlation results in an enhanced detection performance with an increased number of clusters.

Different Selection Strategy (β) with Varying (η): As illustrated in Figures 6.13, 6.14, when $\beta = Random$, increasing η slightly enhances detection performance but consistently falls short of the detection performance



(a) F1 Score (b) Mean Average Precision

Figure 6.14: Comparing the detection performance between the *FullAggregation* and *MaxLabels* CFL approaches under a traffic density of $\alpha = 100$, with a varying number of selected clients at each cluster ($\eta = 1, .., 5$), and two clusters ($N_{cls} = 2$).

achieved with $\beta = FullAggregation$. Similarly, with $\beta = MaxLabels$, increasing η notably improves detection performance. Furthermore, we observed that with 16-25% fewer participating nodes, $\beta = MaxLabels$ outperforms $\beta = FullAggregation$. This threshold's variability, contingent on traffic density, is evident in the transition from $\eta = 4$ to $\eta = 5$ under $\alpha = 30$, $N_{cls} = 2$, where detection performance drops, compared to the continuous increase with $\alpha = 100$, $N_{cls} = 2$.

6.4.4 Limitations of the Study

In our study, we exchanged the complete detection model (6.2MB) during online model training for both *ClassicalFL* and *CFL* approaches. However, in practical settings, object detection models may exhibit larger sizes, prompting the necessity for model compression to enhance efficiency. One limitation involves the requirement for image data captured by multiple vehicles in close proximity under similar environmental conditions. We addressed this limitation by generating synthetic datasets using the Carla simulator [260], although real-world data would offer a more precise representation. To ensure greater traceability, we restricted the total number of participating vehicles to 12, but

evaluating AR-CFL in a broader scenario would be recommended.

Notably, security or privacy-preserving mechanisms were not incorporated into this work. For a more comprehensive approach, integrating encryption and privacy-preserving techniques such as differential privacy is advisable. These considerations represent important directions for future research and refinement of the proposed framework.

6.5 Summary

In this Chapter, we introduced AR-CFL, a framework for adaptive and resource-aware Clustered Federated Learning tailored to the challenges of continuous online learning in vehicular environments, extending the thesis's broader aim of enabling distributed, context-aware intelligence in event-driven environments. Using the CARLA simulator, we created synthetic datasets to simulate different traffic densities and trained object detection models on non-IID data. Unlike existing studies, we trained local models on freshly captured data at each iteration, finding that higher traffic density improved model performance.

A critical aspect of such systems is balancing utility and privacy, as data owners are often reluctant to share large datasets due to privacy concerns. This introduces the need for a privacy-utility trade-off (PUT), where privacy may be compromised to maintain utility. Our framework integrates this trade-off, considering the benefits of federated learning. We found that the *MaxLabels* participant selection strategy reduced node participation by 25% and long-range communication by 33%, improving efficiency while maintaining performance.

Chapter 7

Conclusion and Future Work

> Strive now and clear away hardship's haze, Lest regret cloud tomorrow's gaze. When fortune wakes, do not ignore, Wisdom's ally opens every door.
>
> Ferdowsi (Revised by GPT4o)

This thesis presented four key contributions to solve problems of *quality* (RQ1), *privacy* (RQ2), *autonomy* (RQ3), and *privacy-utility trade-off (PUT)* (RQ4) in the context of Complex Event Processing (cf. Figure 7.1).

In particular, we developed (i) a methodology for selecting optimal sensing deployments based on user-defined quality metrics and resource constraints, (ii) a graph-based mechanism for enhanced pattern-level privacy protection by leveraging graph theory to preserve sensitive patterns, (iii) a novel LLM-based CEP rule generation and refinement approach that utilizes LLMs to generate initial rules and then refines them through a federated process, and (iv) an investigation of PUT in vehicular scenarios explores the trade-offs between privacy and utility in real-world applications. These contributions are implemented in AQuA-CEP[1], APP-CEP[2], GPT-CEP[3], and AR-CFL, whose source code is publicly available.

This chapter provides a comprehensive overview of our research contributions, addresses the research questions posed in Section 7.1, presents our conclusions based on the findings in Section 7.2, and highlights potential avenues for future research in Section 7.3.

[1] https://github.com/majid-lotfian/AQuA-CEP-code
[2] https://github.com/majid-lotfian/APP-CEP
[3] https://github.com/majid-lotfian/GPT-CEP

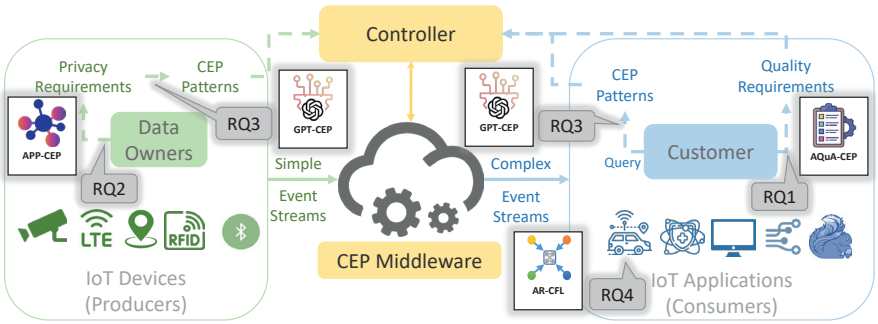


Figure 7.1: Summary of all contributions as a single architecture. Here, AQuA-CEP contributed a novel quality-aware query processing model for obtaining users' quality demands, APP-CEP contributed a unique pattern-level privacy protection model to acquire users' privacy requirements, GPT-CEP proposed a novel autonomous LLM-based CEP rule generation and refinement strategy, and AR-CFL investigated the application of federated learning in online object detection in a vehicular network scenario to provide privacy-utility trade-off in real-world scenarios.

7.1 Contributions Revisited

This thesis addresses four key challenges within CEP systems: quality assurance, privacy protection, autonomy, and the privacy-utility trade-off, as outlined in Chapter 1. We identify four research gaps in the literature in Chapter 2, namely, (i) lack of effective event source utilization to adapt to dynamic environments, compromising quality in the face of changing conditions, (ii) lack of pattern-level privacy due to attribute-level focus, leaving sensitive information vulnerable, (iii) lack of methodologies for employing LLMs to autonomously define, validate, and refine CEP rules over distributed data, and (iv) missing PUT-aware algorithm generation and limitation investigation for IoT scenarios.

To this end, In Chapter 3, we introduced AQuA-CEP, a novel adaptive and quality-aware event source selection mechanism. This mechanism ensures that event sources are chosen to meet the evolving quality requirements of applications and consumers. AQuA-CEP incorporates concepts for defining

user quality requirements, monitoring quality, and sensing configuration deployment algorithms to support these objectives. In Chapter 4, we introduced APP-CEP, a unified privacy protection model that safeguarded sensitive information in continuous data streams. This model employs a pattern-based approach, utilizing *innovative* pattern dependency graphs and event dependencies to select and adapt obfuscation techniques tailored to specific privacy demands. In Chapter 5, we proposed GPT-CEP, a pioneering approach to autonomously initiating and refining CEP rules. This approach leverages the capabilities of large language models to replace the need for domain expert knowledge in rule creation and eliminate human errors. Additionally, we employed a federated cluster-based architecture to refine rules using locally stored data. In Chapter 6, we introduced AR-CFL, a PUT-aware algorithm deployed on vehicular networks. This algorithm investigates the benefits and drawbacks of federated learning-based stream processing systems. AR-CFL includes algorithms for online situation detection that consider the privacy-utility trade-off.

7.2 Key Results

We show the benefits of the proposed contributions by performing extensive real-world system evaluations for each of our four contributions (cf. Figure 7.2).

First, in AQuA-CEP, we introduced consumer-definable quality policies that guide the detection of complex events. These policies were evaluated to autonomously select or configure appropriate data sources within the sensing infrastructure. We explored different ways to express quality policies and analyzed their impact on the quality monitoring process. Additionally, we examined various methods for evaluating and applying quality-related adaptations and their effects on correlation efficiency.

We evaluated AQuA-CEP in IoT scenarios by assessing its performance based on quality policies and query processing adaptation using knowledge gained from quality monitoring. Our results demonstrated that AQuA-CEP can enhance the performance of DCEP systems in terms of result quality while satisfying consumer quality requirements (QR). Quality-based adaptation can

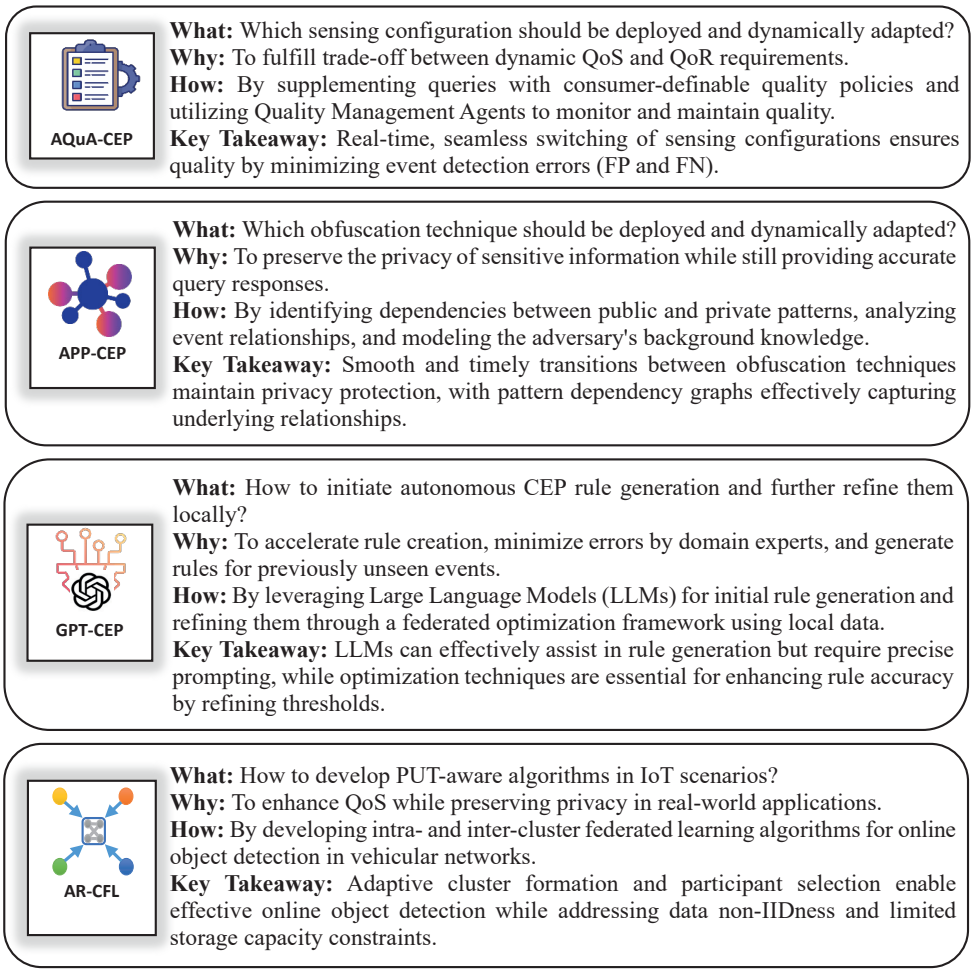


Figure 7.2: Core insights gleaned from our research, informed by our motivation, methodologies, and evaluation outcomes.

also extend network lifetime by optimizing sensor energy consumption through efficient data source selection.

Second, within the APP-CEP framework, we addressed the challenge of integrating pattern-level privacy into event-based systems. APP-CEP involves selectively applying obfuscation techniques to conceal private information. Unlike existing methods, we aimed to enforce privacy without relying on ac-

tual events in streams. To achieve this, we utilized CEP-like patterns to capture user queries and privacy requirements.

We protected privacy by generating pattern dependency graphs and dynamically selecting obfuscation techniques that minimize interference with the detection of both sensitive and non-sensitive patterns essential for maintaining acceptable Quality of Service. Furthermore, we modeled potential adversaries' knowledge to assess their ability to compromise privacy and refine our obfuscation procedures accordingly. Through real-world evaluation with online retailer's transactions and a medical dataset, we demonstrated APP-CEP's effectiveness in balancing privacy and utility. Our approach on modeling background knowledge effectively prevents adversaries from detecting modifications to the input streams.

Third, in the context of GPT-CEP, we employed large language models and prompt engineering to generate accurate and efficient CEP rules. The AGES Index, a comprehensive metric evaluating model size, rule generation efficiency, and rule accuracy, was introduced to assess GPT-CEP's performance against baseline strategies. GPT-CEP consistently outperformed baselines, demonstrating its effectiveness, where the AGES Index was instrumental in selecting the optimal LLM and prompt engineering techniques.

GPT-CEP also incorporated a cluster-based rule refinement approach to optimize initial LLM-generated rules. Simulated annealing was used to refine rule thresholds, further enhancing detection accuracy. Overall, GPT-CEP represents a significant step forward in autonomous rule generation for CEP systems. By combining LLMs, prompt engineering, and rule refinement, it offers a reasonable solution for automating rule creation and improving CEP application performance.

Finally, in the context of AR-CFL, we investigated the application of providing privacy-utility trade-off in stream processing systems, particularly focusing on autonomous vehicular networks. AR-CFL offers a compelling solution by allowing model training across distributed vehicles without the need to share raw sensory data, thereby addressing privacy concerns. In vehicular networks, continuous adaptation and real-time model updates are essential due to dynamic driving conditions and constant data generation. AR-CFL facilitates this adaptation through clustered federated learning, which reduces

communication overhead and ensures efficient resource utilization by clustering participants based on available bandwidth, storage, and computational capacity—key considerations in stream processing systems.

AR-CFL serves as a trade-off between privacy and utility, crucial for managing continuous data streams generated by vehicles while safeguarding sensitive information. Our framework dynamically optimizes the number of clusters and selects participants based on resource availability, ensuring efficient event processing. Evaluation results indicated that AR-CFL achieved a robust detection performance for online perception model training even with non-IID data across varied traffic densities. Furthermore, the training efficiency of participating nodes improved by up to 25%, while communication overhead decreased by 33% compared to conventional federated learning methods in addition to preserving privacy by local data analysis. This demonstrated that AR-CFL can effectively support the requirements of stream processing in vehicular networks, balancing the trade-off between privacy preservation and model utility.

7.3 Future Work

In the following, we discuss potential avenues for future work stemming from our research.

AQuA-CEP

The potential future research for AQuA-CEP can delve deeper into the concept of *priority* within the quality policy framework. By assigning priorities to different queries, the aim will be to optimize the handling of concurrent queries, ensuring that critical data is accessed and processed efficiently. This will involve a thorough investigation of the potential benefits and trade-offs associated with prioritizing data sources in various scenarios. Furthermore, we recognize the significance of minimizing the switching overhead between data sources, influencing query performance. To address this, in-depth research can be conducted to estimate and reduce the associated costs.

Also, the possibility of predicting data source switching times to minimize

the duration of blind monitoring periods can be explored. This predictive capability can be achieved by leveraging statistical analysis of the data sources' performance characteristics, enabling more proactive and efficient resource management. Finally, another future research direction can be to incorporate dynamic quality policies that account for factors such as data source quality degradation over time. This will allow us to adapt our quality policies in real-time, ensuring optimal data selection and usage in evolving environments.

APP-CEP

In the future, a sophisticated algorithm can be developed built on APP-CEP, which intelligently selects the most effective obfuscation technique for scenarios where the dependency sub-graphs lack a ZOD node. This algorithm will achieve this by introducing a comprehensive comparison metric that can quantitatively assess both the positive and negative effects of various obfuscation methods within the sub-graphs. This metric will allow making informed decisions about the optimal obfuscation strategy based on the specific characteristics of the given application.

Furthermore, to enhance the practical applicability of APP-CEP, the diverse stakeholders involved in real-world applications and their respective interests can be thoroughly considered when selecting appropriate input representations. By carefully aligning our input representations with the needs and concerns of all stakeholders, one can ensure that our obfuscation techniques are both effective and acceptable in real-world contexts. Additionally, our evaluation efforts can be expanded to encompass a wider range of use cases, allowing the identification of additional aspects of this problem and uncovering any limitations that may arise in fulfilling the requirements of various applications. This comprehensive evaluation will provide valuable insights for further refining APP-CEP and ensuring its broad applicability.

GPT-CEP

While GPT-CEP demonstrates commendable performance in rule initiation and refinement, there is still significant potential for improvement. In the rule

initiation phase, the synergy between LLMs and prompt engineering techniques has proven effective in enhancing rule quality. However, exploring a broader range of LLMs could further elevate performance. Additionally, developing novel prompt engineering techniques tailored to the unique characteristics of CEP systems presents a promising avenue for advancement. Ideally, the ultimate goal would be to create a specialized LLM that is optimally suited to the demands of stream processing systems.

On the other hand, in the refinement phase of GPT-CEP, beyond simulated annealing, a multitude of optimization techniques exist that could be employed based on the specific characteristics of the application scenario and its data. Consequently, the wise selection of an optimization technique plays a crucial role in refining the accuracy of CEP rules. Furthermore, certain concepts that were simplified or omitted in GPT-CEP, such as rule update aggregation mechanisms, rule versioning, rule personalization, and rule merging within or across clusters, require further investigation to explore potential avenues for improvement.

AR-CFL

To further advance the capabilities of AR-CFL algorithms, incorporating model compression techniques presents a promising avenue. By reducing the model's size and computational complexity, one can enhance algorithm efficiency and potentially improve its utility values. This would make our algorithms more practical for deployment in resource-constrained environments and accelerate inference times.

Moreover, validating AR-CFL with real-world datasets would be crucial to ensure its generalizability and robustness. Such evaluations would provide invaluable insights into the practical limitations and challenges that may arise in establishing PUT in real-world scenarios. Additionally, investigating potential conflicts of interest between data owners and end-users is essential to understand the complexities involved in balancing privacy and utility. Finally, integrating advanced encryption and privacy-preserving mechanisms, such as differential privacy, can significantly strengthen the security and privacy guarantees of our framework. This would make it more difficult for adversaries

to exploit vulnerabilities and ensure that the benefits of PUT in improving utility are realized while protecting user privacy.

Intelligent Event Processing

Beyond the specific future research directions for AQuA-CEP, APP-CEP, GPT-CEP, and AR-CFL, future work on DCEP systems and their integration with AI, privacy, and data quality should take a broader and more transformative approach. The field is rapidly evolving with the emergence of autonomous, self-adaptive CEP systems, integrating AI-driven decision-making, real-time learning, and edge intelligence. One promising direction is the exploration of fully decentralized and privacy-preserving CEP architectures, where federated learning and secure multi-party computation can enhance both data confidentiality and distributed intelligence. Additionally, the increasing adoption of event-driven AI applications in smart cities, autonomous systems, and large-scale industrial automation highlights the need for more explainable and interpretable event processing models to ensure trust and transparency in decision-making. Another major challenge is the seamless integration of event processing across heterogeneous data ecosystems, requiring standardized interoperability frameworks that can bridge real-time data streaming with knowledge graphs, digital twins, and large-scale simulation environments. Addressing these challenges will not only advance CEP research but also enable next-generation intelligent systems that can autonomously adapt to complex, dynamic environments.

Bibliography

[1] M. H. Kashani, M. Madanipour, M. Nikravan, P. Asghari, and E. Mahdipour, "A systematic review of IoT in healthcare: Applications, techniques, and trends," *Journal of Network and Computer Applications*, vol. 192, p. 103164, 2021.

[2] S. Singh and N. Singh, "Internet of Things (IoT): Security challenges, business opportunities & reference architecture for e-commerce," in *2015 International conference on green computing and internet of things (ICGCIoT)*, pp. 1577–1581, IEEE, 2015.

[3] Techinformed, "IoT in 2023 and beyond (2023-03)," 2023. Accessed: May 1st, 2024.

[4] D. Luckham, "The power of events: An introduction to complex event processing in distributed enterprise systems," in *Proceedings of the International Symposium on Rule Representation, Interchange and Reasoning on the Web*, p. 3, 2008.

[5] S. Chakravarthy and Q. Jiang, "Stream data processing: A quality of service perspective modeling, scheduling, load shedding, and complex event processing," 2009.

[6] X, "Processing billions of events in real time at Twitter," 2021.

[7] Confluent, "Confluent inducted into JPMorgan chase hall of innovation," 2019.

[8] AmazonMSK, "Amazon managed service for apache flink," 2023.

[9] T. Karnagel, D. Habich, and W. Lehner, "Adaptive work placement for query processing on heterogeneous computing resources," *Proceedings of the VLDB Endowment*, vol. 10, no. 7, pp. 733–744, 2017.

[10] S. Liu, J. Weng, J. H. Wang, C. An, Y. Zhou, and J. Wang, "An adaptive online scheme for scheduling and resource enforcement in storm," *IEEE/ACM Transactions on Networking*, vol. 27, no. 4, pp. 1373–1386, 2019.

[11] M. Luthra, B. Koldehofe, P. Weisenburger, G. Salvaneschi, and R. Arif, "TCEP: Adapting to dynamic user environments by enabling transitions between operator placement mechanisms," in *Proceedings of the 12th ACM International Conference on Distributed and Event-Based Systems (DEBS)*, pp. pp. 136–147, ACM, 2018.

[12] V. Cardellini, V. Grassi, F. Lo Presti, and M. Nardelli, "Optimal operator placement for distributed stream processing applications," in *Proceedings of the 10th ACM International Conference on Distributed and Event-based Systems*, pp. 69–80, 2016.

[13] S. Frischbier, P. Pietzuch, and A. Buchmann, "Managing expectations: Runtime negotiation of information quality requirements in event-based systems," in *Proceedings of the 12th International Conference on Service Oriented Computing (ICSOC)*, pp. pp. 199–213, Springer Berlin Heidelberg, 2014.

[14] P. Weisenburger, M. Luthra, B. Koldehofe, and G. Salvaneschi, "Quality-aware runtime adaptation in complex event processing," in *Proceedings of 12th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS)*, pp. pp. 140–151, IEEE/ACM, 2017.

[15] S. M. Palanisamy, F. Dürr, M. A. Tariq, and K. Rothermel, "Preserving privacy and quality of service in complex event processing through event reordering," in *Proceedings of the 12th ACM International Conference on Distributed and Event-based Systems*, pp. 40–51, 2018.

[16] S. M. Palanisamy, "Towards multiple pattern type privacy protection in complex event processing through event obfuscation strategies," in *Data Privacy Management, Cryptocurrencies and Blockchain Technology: ESORICS 2020 International Workshops, DPM 2020 and CBT 2020, Guildford, UK, September 17–18, 2020, Revised Selected Papers 15*, pp. 178–194, Springer, 2020.

[17] A. Margara, G. Cugola, and G. Tamburrelli, "Learning from the Past: Automated Rule Generation for Complex Event Processing," in *Proceedings of the 8th ACM International Conference on Distributed Event-Based Systems*, DEBS '14, p. 47–58, 2014.

[18] M. U. Simsek, F. Yildirim Okay, and S. Ozdemir, "A Deep Learning-based CEP Rule Extraction Framework for IoT Data," *The Journal of Supercomputing*, vol. 77, pp. 8563–8592, 2021.

[19] J. Roldán-Gómez, J. Boubeta-Puig, J. Carrillo-Mondéjar, J. M. C. Gómez, and J. M. del Rincón, "An automatic complex event processing rules generation system for the recognition of real-time IoT attack patterns," *Engineering Applications of Artificial Intelligence*, vol. 123, p. 106344, 2023.

[20] J. Lv, B. Yu, and H. Sun, "CEP rule extraction framework based on evolutionary algorithm," in *2022 11th International Conference of Information and Communication Technology (ICTech))*, pp. 245–249, IEEE, 2022.

[21] A. Friedman, I. Sharfman, D. Keren, and A. Schuster, "Privacy-preserving distributed stream monitoring.," in *NDSS*, pp. 1–12, 2014.

[22] M. A. Erdogdu and N. Fawaz, "Privacy-utility trade-off under continual observation," in *2015 IEEE International Symposium on Information Theory (ISIT)*, pp. 1801–1805, IEEE, 2015.

[23] S. Cerf, V. Primault, A. Boutet, S. B. Mokhtar, R. Birke, S. Bouchenak, L. Y. Chen, N. Marchand, and B. Robu, "Pulp: Achieving privacy and utility trade-off in user

mobility data," in *2017 IEEE 36th symposium on reliable distributed systems (SRDS)*, pp. 164–173, IEEE, 2017.

[24] Y. He, S. Barman, D. Wang, and J. F. Naughton, "On the complexity of privacy-preserving complex event processing," in *Proceedings of the 13th ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pp. 165–174, 2011.

[25] E. Erdemir, P. L. Dragotti, and D. Gündüz, "Active privacy-utility trade-off against inference in time-series data sharing," *IEEE Journal on Selected Areas in Information Theory*, 2023.

[26] Z. Gao, A. Sepahi, S. Pouryousef, L. Zhou, and H. Zhu, "Trade-off between privacy and utility for location-based recommendation services," in *ICC 2022-IEEE International Conference on Communications*, pp. 4396–4401, IEEE, 2022.

[27] S. Cerf, S. Bouchenak, B. Robu, N. Marchand, V. Primault, S. B. Mokhtar, A. Boutet, and L. Y. Chen, "Automatic privacy and utility preservation for mobility data: A nonlinear model-based approach," *IEEE Transactions on Dependable and Secure Computing*, vol. 18, no. 1, pp. 269–282, 2018.

[28] M. Lotfian Delouee, B. Koldehofe, and V. Degeler, "AQuA-CEP: Adaptive quality-aware complex event processing in the internet of things," in *Proceedings of the 17th ACM International Conference on Distributed and Event-Based Systems (DEBS'23)*, p. 13–24, ACM, 2023.

[29] M. Lotfian Delouee, V. Degeler, P. Amthor, and B. Koldehofe, "APP-CEP: Adaptive pattern-level privacy protection in complex event processing systems," in *Proceedings of the 10th International Conference on Information Systems Security and Privacy (ICISSP'24)*, p. 12 pages, SCITEPRESS–Science and Technology Publications, 2024.

[30] R. Mousheimish, Y. Taher, and K. Zeitouni, "Automatic learning of predictive cep rules: bridging the gap between data mining and complex event processing," in *Proceedings of the 11th ACM international conference on distributed and event-based systems*, pp. 158–169, 2017.

[31] E. Petersen, M. A. To, and S. Maag, "An online learning based approach for CEP rule generation," in *2016 8th IEEE Latin-American Conference on Communications (LATINCOM)*, pp. 1–6, IEEE, 2016.

[32] E. Petersen, M. A. To, S. Maag, and T. Yamga, "An unsupervised rule generation approach for online complex event processing," in *2018 IEEE 17th International Symposium on Network Computing and Applications (NCA)*, pp. 1–8, IEEE, 2018.

[33] R. Bruns, J. Dunkel, and N. Offel, "Learning of complex event processing rules with genetic programming," *Expert Systems with Applications*, vol. 129, pp. 186–199, 2019.

[34] M. Gholami, B. Zamani, and B. S. Ghahfarokhi, "Automatic generation of CEP rules using data analysis techniques and model-driven engineering," in *2023 7th International Conference on Internet of Things and Applications (IoT)*, pp. 1–5, IEEE, 2023.

[35] Y. Sun, G. Li, and B. Ning, "Automatic rule updating based on machine learning in complex event processing," in *2020 IEEE 40th International Conference on Distributed Computing Systems (ICDCS)*, pp. 1338–1343, IEEE, 2020.

[36] M. Lotfian Delouee, B. Koldehofe, and V. Degeler, "Poster: Towards pattern-level privacy protection in distributed complex event processing," in *Proceedings of the 17th ACM International Conference on Distributed and Event-Based Systems*, p. 185–186, ACM, 2023.

[37] A. Zamani, T. J. Oechtering, and M. Skoglund, "On the privacy-utility trade-off with and without direct access to the private data," *IEEE Transactions on Information Theory*, 2023.

[38] X. He, A. Machanavajjhala, and B. Ding, "Blowfish privacy: Tuning privacy-utility trade-offs using policies," in *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*, pp. 1447–1458, 2014.

[39] R. Dong, L. J. Ratliff, A. A. Cárdenas, H. Ohlsson, and S. S. Sastry, "Quantifying the utility–privacy trade-off in the internet of things," *ACM Transactions on Cyber-Physical Systems*, vol. 2, no. 2, pp. 1–28, 2018.

[40] A. C. Valdez and M. Ziefle, "The users' perspective on the privacy-utility trade-offs in health recommender systems," *International Journal of Human-Computer Studies*, vol. 121, pp. 108–121, 2019.

[41] H. Cho, S. Simmons, R. Kim, and B. Berger, "Privacy-preserving biomedical database queries with optimal privacy-utility trade-offs," *Cell systems*, vol. 10, no. 5, pp. 408–416, 2020.

[42] B. Z. H. Zhao, M. A. Kaafar, and N. Kourtellis, "Not one but many trade-offs: Privacy vs. utility in differentially private machine learning," in *Proceedings of the 2020 ACM SIGSAC Conference on Cloud Computing Security Workshop*, pp. 15–26, 2020.

[43] M. Kim, O. Günlü, and R. F. Schaefer, "Federated learning with local differential privacy: Trade-offs between privacy, utility, and communication," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2650–2654, IEEE, 2021.

[44] X. Zhang, Y. Kang, K. Chen, L. Fan, and Q. Yang, "Trading off privacy, utility, and efficiency in federated learning," *ACM Transactions on Intelligent Systems and Technology*, vol. 14, no. 6, pp. 1–32, 2023.

[45] D. Wang, Y. He, E. Rundensteiner, and J. F. Naughton, "Utility-maximizing event stream suppression," in *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*, pp. 589–600, 2013.

[46] Y. Yao, J. Duan, K. Xu, Y. Cai, Z. Sun, and Y. Zhang, "A survey on large language model (llm) security and privacy: The good, the bad, and the ugly," *High-Confidence Computing*, p. 100211, 2024.

[47] J. Zamfirescu-Pereira, R. Y. Wong, B. Hartmann, and Q. Yang, "Why johnny can't prompt: how non-AI experts try (and fail) to design LLM prompts," in *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, pp. 1–21, 2023.

[48] OpenAI, "Openai chatgpt," 2022. https://chat.openai.com/, Accessed on May 15th, 2024.

[49] Google, "Google gemini," 2023. https://gemini.google.com/app, Accessed on May 15th, 2024.

[50] Y. Zhang, A. Carballo, H. Yang, and K. Takeda, "Perception and sensing for autonomous vehicles under adverse weather conditions: A survey," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 196, pp. 146–177, 2023.

[51] A. Khalil, T. Meuser, Y. Alkhalili, A. Fernández Anta, L. Staecker, and R. Steinmetz, "Situational collective perception: Adaptive and efficient collective perception in future vehicular systems," in *International Conference on Vehicle Technology and Intelligent Transport Systems (VEHITS'22)*, pp. 346–352, 2022.

[52] G. Barquero, L. Burgueño, J. Troya, and A. Vallecillo, "Extending complex event processing to graph-structured information," in *Proceedings of the 21th International Conference on Model Driven Engineering Languages and Systems*, pp. pp. 166–175, ACM/IEEE, 2018.

[53] J. Krämer and B. Seeger, "Semantics and implementation of continuous sliding window queries over data streams," *ACM Transactions on Database Systems (TODS)*, vol. 34, no. 1, pp. 1–49, 2009.

[54] G. Cugola and A. Margara, "Processing flows of information: From data stream to complex event processing," *ACM Computing Surveys (CSUR)*, vol. 44, no. 3, pp. 1–62, 2012.

[55] I. Flouris, N. Giatrakos, A. Deligiannakis, M. Garofalakis, M. Kamp, and M. Mock, "Issues in complex event processing: Status and prospects in the big data era," *Journal of Systems and Software*, vol. 127, pp. 217–236, 2017.

[56] A. Hinze, K. Sachs, and A. Buchmann, "Event-based applications and enabling technologies," in *Proceedings of the Third ACM International Conference on Distributed Event-Based Systems*, pp. 1–15, 2009.

[57] D. C. Luckham and B. Frasca, "Complex event processing in distributed systems," *Computer Systems Laboratory Technical Report CSL-TR-98-754. Stanford University, Stanford*, vol. 28, p. 16, 1998.

[58] B. Ottenwälder, B. Koldehofe, K. Rothermel, and U. Ramachandran, "Migcep: Operator migration for mobility driven distributed complex event processing," in *Proceedings of the 7th International Conference on Distributed and Event-based Systems (DEBS)*, ACM, 2013.

[59] A. Slo, S. Bhowmik, and K. Rothermel, "hSPICE: State-aware event shedding in complex event processing," in *Proceedings of the 14th International Conference on Distributed and Event-based Systems (DEBS)*, pp. pp. 109–120, ACM, 2020.

[60] H. v. d. Aa, A. Artikis, and M. Weidlich, "Complex event processing methods for process querying," in *Process Querying Methods*, pp. 479–510, Springer, 2021.

[61] M. Yu, M. Bambacus, G. Cervone, K. Clarke, D. Duffy, Q. Huang, J. Li, W. Li, Z. Li, Q. Liu, *et al.*, "Spatiotemporal event detection: A review," *International Journal of Digital Earth*, vol. 13, no. 12, pp. 1339–1365, 2020.

[62] N. Moreno, M. F. Bertoa, L. Burgueño, and A. Vallecillo, "Managing measurement and occurrence uncertainty in complex event processing systems," *IEEE Access*, vol. 7, pp. 88026–88048, 2019.

[63] F. Xiao, "CaFtR: A fuzzy complex event processing method," *International Journal of Fuzzy Systems*, vol. 24, no. 2, pp. 1098–1111, 2022.

[64] M. Lima, R. Lima, F. Lins, and M. Bonfim, "Beholder – A CEP-based intrusion detection and prevention systems for IoT environments," *Computers & Security*, vol. 120, p. 102824, 2022.

[65] B. Schilling, B. Koldehofe, and K. Rothermel, "Efficient and distributed rule placement in heavy constraint-driven event systems," in *2011 IEEE International Conference on High Performance Computing and Communications*, pp. 355–364, IEEE, 2011.

[66] Y. Turchin, A. Gal, and S. Wasserkrug, "Tuning complex event processing rules using the prediction-correction paradigm," in *Proceedings of the Third ACM International Conference on Distributed Event-Based Systems*, pp. 1–12, 2009.

[67] L. M. Dang, K. Min, H. Wang, M. J. Piran, C. H. Lee, and H. Moon, "Sensor-based and vision-based human activity recognition: A comprehensive survey," *Pattern Recognition*, vol. 108, p. 107561, 2020.

[68] Y. Mao, X. Chen, and Z. Xu, "Real-time event detection with water sensor networks using a spatio-temporal model," in *Database Systems for Advanced Applications: DASFAA 2016 International Workshops: BDMS, BDQM, MoI, and SeCoP, Dallas, TX, USA, April 16-19, 2016, Proceedings 21*, pp. 194–208, Springer, 2016.

[69] R. San-Segundo, J. D. Echeverry-Correa, C. Salamea, and J. M. Pardo, "Human activity monitoring based on hidden markov models using a smartphone," *IEEE Instrumentation & Measurement Magazine*, vol. 19, no. 6, pp. 27–31, 2016.

[70] R. Baldoni, L. Montanari, and M. Rizzuto, "On-line failure prediction in safety-critical systems," *Future Generation Computer Systems*, vol. 45, pp. 123–132, 2015.

[71] M. Muaaz and R. Mayrhofer, "Accelerometer based gait recognition using adapted gaussian mixture models," in *Proceedings of the 14th international conference on advances in mobile computing and multi media*, pp. 288–291, 2016.

[72] E. Adi, A. Anwar, Z. Baig, and S. Zeadally, "Machine learning and data analytics for the iot," *Neural computing and applications*, vol. 32, pp. 16205–16233, 2020.

[73] R. Sahal, J. G. Breslin, and M. I. Ali, "Big data and stream processing platforms for industry 4.0 requirements mapping for a predictive maintenance use case," *Journal of manufacturing systems*, vol. 54, pp. 138–151, 2020.

[74] X. Ma, T. Yao, M. Hu, Y. Dong, W. Liu, F. Wang, and J. Liu, "A survey on deep learning empowered IoT applications," *IEEE Access*, vol. 7, pp. 181721–181732, 2019.

[75] T. Xing, L. Garcia, M. R. Vilamala, F. Cerutti, L. Kaplan, A. Preece, and M. Srivastava, "Neuroplex: learning to detect complex events in sensor networks through knowledge injection," in *Proceedings of the 18th Conference on Embedded Networked Sensor Systems*, pp. pp. 489–502, ACM, 2020.

[76] N. Koroniotis, N. Moustafa, and E. Sitnikova, "A new network forensic framework based on deep learning for internet of things networks: A particle deep framework," *Future Generation Computer Systems*, vol. 110, pp. 91–106, 2020.

[77] H. Ren, D. Anicic, and T. A. Runkler, "The synergy of complex event processing and tiny machine learning in industrial iot," in *Proceedings of the 15th ACM International Conference on Distributed and Event-based Systems*, pp. 126–135, 2021.

[78] S. Grigorescu, B. Trasnea, T. Cocias, and G. Macesanu, "A survey of deep learning techniques for autonomous driving," *Journal of field robotics*, vol. 37, no. 3, pp. 362–386, 2020.

[79] P. Yadav, D. Sarkar, D. Salwala, and E. Curry, "Traffic prediction framework for openstreetmap using deep learning based complex event processing and open traffic cameras," *arXiv preprint arXiv:2008.00928*, 2020.

[80] A. Ignatov, "Real-time human activity recognition from accelerometer data using convolutional neural networks," *Applied Soft Computing*, vol. 62, pp. 915–922, 2018.

[81] M. A. Rahman, Y. Mia, M. R. Masum, D. M. H. Abid, and T. Islam, "Real time human activity recognition from accelerometer data using convolutional neural networks," in *2022 7th International Conference on Communication and Electronics Systems (ICCES)*, pp. 1394–1397, IEEE, 2022.

[82] P. Schneider and F. Xhafa, *Anomaly detection and complex event processing over IoT data streams: with application to EHealth and patient data monitoring*. Academic Press, 2022.

[83] M. Jouhari, A. K. Al-Ali, E. Baccour, A. Mohamed, A. Erbad, M. Guizani, and M. Hamdi, "Distributed CNN inference on resource-constrained uavs for surveillance systems: Design and optimization," *IEEE Internet of Things Journal*, vol. 9, no. 2, pp. 1227–1242, 2021.

[84] L. Yue, D. Tian, W. Chen, X. Han, and M. Yin, "Deep learning for heterogeneous medical data analysis," *World Wide Web*, vol. 23, pp. 2715–2737, 2020.

[85] P. Rivera, E. Valarezo, M.-T. Choi, and T.-S. Kim, "Recognition of human hand activities based on a single wrist imu using recurrent neural networks," *Int. J. Pharma Med. Biol. Sci*, vol. 6, no. 4, pp. 114–118, 2017.

[86] M. R. Vilamala, T. Xing, H. Taylor, L. Garcia, M. Srivastava, L. Kaplan, A. Preece, A. Kimmig, and F. Cerutti, "Deepprobcep: A neuro-symbolic approach for complex event processing in adversarial settings," *Expert Systems with Applications*, vol. 215, p. 119376, 2023.

[87] S. Vosta and K.-C. Yow, "A cnn-rnn combined structure for real-world violence detection in surveillance cameras," *Applied Sciences*, vol. 12, no. 3, p. 1021, 2022.

[88] X. He, P. Tai, H. Lu, X. Huang, and Y. Ren, "A biomedical event extraction method based on fine-grained and attention mechanism," *BMC bioinformatics*, vol. 23, no. 1, p. 308, 2022.

[89] H. Yin, J. Cao, L. Cao, and G. Wang, "An improved deep belief network for chinese emergency recognition," in *Journal of Physics: Conference Series*, vol. 1549, p. 022065, IOP Publishing, 2020.

[90] C.-Y. Wang, J.-C. Wang, A. Santoso, C.-C. Chiang, and C.-H. Wu, "Sound event recognition using auditory-receptive-field binary pattern and hierarchical-diving deep belief network," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 26, no. 8, pp. 1336–1351, 2018.

[91] B. Almaslukh, J. AlMuhtadi, and A. Artoli, "An effective deep autoencoder approach for online smartphone-based human activity recognition," *Int. J. Comput. Sci. Netw. Secur*, vol. 17, no. 4, pp. 160–165, 2017.

[92] T. Li, X. Chen, F. Zhu, Z. Zhang, and H. Yan, "Two-stream deep spatial-temporal auto-encoder for surveillance video abnormal event detection," *Neurocomputing*, vol. 439, pp. 256–270, 2021.

[93] B. Ottenwälder, B. Koldehofe, K. Rothermel, and U. Ramachandran, "Migcep: Operator migration for mobility driven distributed complex event processing," in *Proc of the 7th ACM international DEBS conference*, 2013.

[94] F. Gao, M. I. Ali, E. Curry, and A. Mileo, "QoS-aware adaptation for complex event service," in *Proceedings of the 31st Annual ACM Symposium on Applied Computing*, pp. 1597–1604, 2016.

[95] M. Luthra, B. Koldehofe, N. Danger, P. Weisenberger, G. Salvaneschi, and I. Stavrakakis, "TCEP: Transitions in operator placement to adapt to dynamic network environments," *Journal of Computer and System Sciences*, vol. 122, pp. 94–125, 2021.

[96] Ş. Kolozali, M. Bermudez-Edo, N. FarajiDavar, P. Barnaghi, F. Gao, M. I. Ali, A. Mileo, M. Fischer, T. Iggena, D. Kuemper, *et al.*, "Observing the pulse of a city: A smart city framework for real-time discovery, federation, and aggregation of data streams," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2651–2668, 2018.

[97] D. Gkoulis, C. Bardaki, M. Nikolaidou, G. Kousiouris, and A. Tsadimas, "A hybrid simulation platform for quality-aware evaluation of complex events in an IoT environment," *Simulation Modelling Practice and Theory*, p. 102919, 2024.

[98] P. Baban, "Pre-processing and data validation in iot data streams," in *Proceedings of the 14th ACM International Conference on Distributed and Event-based Systems*, pp. 226–229, 2020.

[99] H. Y. Teh, A. W. Kempa-Liehr, and K. I.-K. Wang, "Sensor data quality: A systematic review," *Journal of Big Data*, vol. 7, no. 1, p. 11, 2020.

[100] K. Moulouel, A. Chibani, and Y. Amirat, "Ontology-based hybrid commonsense reasoning framework for handling context abnormalities in uncertain and partially observable environments," *Information Sciences*, vol. 631, pp. 468–486, 2023.

[101] K. Ip, A. Asok, Y. Xu, D. Le, N. Mionis, and R. Batoukov, "Ml-assisted monitoring and characterization of IoT sensor networks," in *2020 IEEE Conference on Evolving and Adaptive Intelligent Systems (EAIS)*, pp. 1–8, IEEE, 2020.

[102] T. Li, Y. Zhao, C. Zhang, K. Zhou, and X. Zhang, "A semantic model-based fault detection approach for building energy systems," *Building and Environment*, vol. 207, p. 108548, 2022.

[103] G. Jäger, S. Zug, T. Brade, A. Dietrich, C. Steup, C. Moewes, and A.-M. Cretu, "Assessing neural networks for sensor fault detection," in *2014 IEEE international conference on computational intelligence and virtual environments for measurement systems and applications (CIVEMSA)*, pp. 70–75, IEEE, 2014.

[104] L. Erhan, M. Ndubuaku, M. Di Mauro, W. Song, M. Chen, G. Fortino, O. Bagdasar, and A. Liotta, "Smart anomaly detection in sensor systems: A multi-perspective review," *Information Fusion*, vol. 67, pp. 64–79, 2021.

[105] G. Sivapalan, K. K. Nundy, S. Dev, B. Cardiff, and D. John, "Annet: A lightweight neural network for ecg anomaly detection in IoT edge sensors," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 16, no. 1, pp. 24–35, 2022.

[106] M. N. Hasan and I. Koo, "Machine learning-based sensor drift fault classification using discrete cosine transform," in *2021 International Conference on Electronics, Communications and Information Technology (ICECIT)*, pp. 1–4, IEEE, 2021.

[107] P. Das, A. Manna, and S. Ghoshal, "Gas sensor drift compensation by ensemble of classifiers using extreme learning machine," in *2020 International Conference on Renewable Energy Integration into Smart Grids: A Multidisciplinary Approach to Technology Modelling and Simulation (ICREISG)*, pp. 197–201, IEEE, 2020.

[108] P. Biswas and T. Samanta, "Anomaly detection using ensemble random forest in wireless sensor network," *International Journal of Information Technology*, vol. 13, no. 5, pp. 2043–2052, 2021.

[109] N. Iftikhar, T. Baattrup-Andersen, F. E. Nordbjerg, and K. Jeppesen, "Outlier detection in sensor data using ensemble learning," *Procedia Computer Science*, vol. 176, pp. 1160–1169, 2020.

[110] Y. K. Saheed, O. H. Abdulganiyu, and T. A. Tchakoucht, "A novel hybrid ensemble learning for anomaly detection in industrial sensor networks and scada systems for smart city infrastructures," *Journal of King Saud University-Computer and Information Sciences*, vol. 35, no. 5, p. 101532, 2023.

[111] J. Park, Y. Seo, and J. Cho, "Unsupervised outlier detection for time-series data of indoor air quality using lstm autoencoder with ensemble method," *Journal of Big Data*, vol. 10, no. 1, p. 66, 2023.

[112] V. S. R. Kosuru and A. Kavasseri Venkitaraman, "A smart battery management system for electric vehicles using deep learning-based sensor fault detection," *World Electric Vehicle Journal*, vol. 14, no. 4, p. 101, 2023.

[113] H. Ruan, B. Dorneanu, H. Arellano-Garcia, P. Xiao, and L. Zhang, "Deep learning-based fault prediction in wireless sensor network embedded cyber-physical systems for industrial processes," *Ieee Access*, vol. 10, pp. 10867–10879, 2022.

[114] H. Pan, W. Sun, Q. Sun, and H. Gao, "Deep learning based data fusion for sensor fault diagnosis and tolerance in autonomous vehicles," *Chinese Journal of Mechanical Engineering*, vol. 34, no. 1, p. 72, 2021.

[115] H. Son, Y. Jang, S.-E. Kim, D. Kim, and J.-W. Park, "Deep learning-based anomaly detection to classify inaccurate data and damaged condition of a cable-stayed bridge," *IEEE Access*, vol. 9, pp. 124549–124559, 2021.

[116] M. Ameli, V. Pfanschilling, A. Amirli, W. Maaß, and K. Kersting, "Unsupervised multi-sensor anomaly localization with explainable ai," in *IFIP International Conference on Artificial Intelligence Applications and Innovations*, pp. 507–519, Springer, 2022.

[117] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. USA: Cambridge University Press, 3 ed., 2007.

[118] R. Dunia, S. J. Qin, T. F. Edgar, and T. J. McAvoy, "Use of principal component analysis for sensor fault identification," *Computers & Chemical Engineering*, vol. 20, pp. S713–S718, 1996.

[119] M. A. Rassam, M. A. Maarof, and A. Zainal, "Adaptive and online data anomaly detection for wireless sensor systems," *Knowledge-Based Systems*, vol. 60, pp. 44–57, 2014.

[120] Y. Hu and A. Liu, "An efficient heuristic subtraction deployment strategy to guarantee quality of event detection for wsns," *The Computer Journal*, vol. 58, no. 8, pp. 1747–1762, 2015.

[121] M. Kumar, P. K. Singh, M. K. Maurya, and A. Shivhare, "A survey on event detection approaches for sensor based iot," *Internet of Things*, vol. 22, p. 100720, 2023.

[122] M. Lotfian Delouee, B. Koldehofe, and V. Degeler, "Towards adaptive quality-aware complex event processing in the internet of things," in *Proceedings of the 18th International Conference on Mobility, Sensing and Networking (MSN)*, pp. pp. 571–575, IEEE, 2022.

[123] H. Röger and R. Mayer, "A comprehensive survey on parallelization and elasticity in stream processing," *ACM Computing Surveys (CSUR)*, vol. 52, no. 2, pp. 1–37, 2019.

[124] A. H. Sodhro, A. S. Malokani, G. H. Sodhro, M. Muzammal, and L. Zongwei, "An adaptive QoS computation for medical data processing in intelligent healthcare applications," *Neural computing and applications*, vol. 32, pp. 723–734, 2020.

[125] Y. Wang, H. Hu, H. Kuang, C. Fan, L. Wang, and X. Tao, "Rl-based CEP operator placement method on edge networks using response time feedback," in *International Conference on Web Information Systems and Applications*, pp. 559–571, Springer, 2023.

[126] M. Luthra and B. Koldehofe, "Progcep: A programming model for complex event processing over fog infrastructure," in *Proceedings of the 2nd International Workshop on Distributed Fog Services Design (DFSD)*, pp. pp. 7–12, ACM, 2019.

[127] M. R. dos Santos, A. P. Batista, R. L. Rosa, M. Saadi, D. C. Melgarejo, and D. Z. Rodríguez, "Asqm: Audio streaming quality metric based on network impairments and user preferences," *IEEE Transactions on Consumer Electronics*, 2023.

[128] S. Dilek, K. Irgan, M. Guzel, S. Ozdemir, S. Baydere, and C. Charnsripinyo, "QoS-aware IoT networks and protocols: A comprehensive survey," *International Journal of Communication Systems*, vol. 35, no. 10, p. e5156, 2022.

[129] L. Zhou, D. Wu, X. Wei, and Z. Dong, "Seeing isn't believing: QoE evaluation for privacy-aware users," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 7, pp. 1656–1665, 2019.

[130] M. Medvetskyi, M. Beshley, and M. Klymash, "A quality of experience management method for intent-based software-defined networks," in *2021 IEEE 16th international conference on the experience of designing and application of CAD systems (CADSM)*, pp. 59–62, IEEE, 2021.

[131] A. A. Barakabitze, N. Barman, A. Ahmad, S. Zadtootaghaj, L. Sun, M. G. Martini, and L. Atzori, "QoE management of multimedia streaming services in future networks: A tutorial and survey," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 1, pp. 526–565, 2019.

[132] P. Arcaini, E. Riccobene, and P. Scandurra, "Modeling and analyzing MAPE-K feedback loops for self-adaptation," in *2015 IEEE/ACM 10th International Symposium on Software Engineering for Adaptive and Self-Managing Systems*, pp. 13–23, IEEE, 2015.

[133] E. D. Canedo, I. N. Bandeira, A. T. S. Calazans, P. H. T. Costa, E. C. R. Cançado, and R. Bonifácio, "Privacy requirements elicitation: a systematic literature review and perception analysis of it practitioners," *Requirements Engineering*, vol. 28, no. 2, pp. 177–194, 2023.

[134] T. Plagemann, V. Goebel, M. Hollick, and B. Koldehofe, "Towards privacy engineering for real-time analytics in the human-centered internet of things," *arXiv preprint arXiv:2210.16352*, 2022.

[135] R. Dwarakanath, B. Koldehofe, Y. Bharadwaj, T. A. B. Nguyen, D. Eyers, and R. Steinmetz, "TrustCEP: Adopting a trust-based approach for distributed complex event processing," in *2017 18th IEEE International Conference on Mobile Data Management (MDM)*, pp. 30–39, IEEE, 2017.

[136] K. Tawsif, J. Hossen, J. E. Raja, M. Jesmeen, and E. Arif, "A review on complex event processing systems for big data," in *2018 Fourth International Conference on Information Retrieval and Knowledge Management (CAMP)*, pp. 1–6, IEEE, 2018.

[137] H. Gu, "Towards guaranteed privacy in stream processing: Differential privacy for private pattern protection," in *Proceedings of the 17th ACM International Conference on Distributed and Event-based Systems*, pp. 207–210, 2023.

[138] M. Hansen and H. Tschofenig, "Terminology for talking about privacy by data minimization: Anonymity, unlinkability, undetectability, unobservability, pseudonymity, and identity management," *draft-hansen-privacy-terminology-02 (work in progress)*, 2011.

[139] X. Liu, A. Doboli, and F. Ye, "Optimized local control strategy for voice-based interaction-tracking badges for social applications," in *2015 33rd IEEE International Conference on Computer Design (ICCD)*, pp. 688–695, IEEE, 2015.

[140] M. Rhahla, S. Allegue, and T. Abdellatif, "Guidelines for gdpr compliance in big data systems," *Journal of Information Security and Applications*, vol. 61, p. 102896, 2021.

[141] M. Florea, C. Potlog, P. Pollner, D. Abel, O. Garcia, S. Bar, S. Naqvi, and W. Asif, "Complex project to develop real tools for identifying and countering terrorism: real-time early detection and alert system for online terrorist content based on natural language processing, social network analysis, artificial intelligence and complex event processing," in *Challenges in cybersecurity and privacy-the European research landscape*, pp. 181–206, River Publishers, 2022.

[142] W. Asif, I. G. Ray, S. Tahir, and M. Rajarajan, "Privacy-preserving anonymization with restricted search (pars) on social network data for criminal investigations," in *2018 19th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)*, pp. 329–334, IEEE, 2018.

[143] R. Hirt, N. Kühl, D. Martin, and G. Satzger, "Enabling inter-organizational analytics in business networks through meta machine learning," *Information Technology and Management*, pp. 1–25, 2023.

[144] G. Liu, Z. Pang, J. Zeng, H. Hong, Y. Sun, M. Su, and N. Ma, "Iot lakehouse: A new data management paradigm for aiot," in *International Conference on Big Data*, pp. 34–47, Springer, 2023.

[145] Z. Abreu and L. Pereira, "Privacy protection in smart meters using homomorphic encryption: An overview," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 12, no. 4, p. e1469, 2022.

[146] A. Rodrigo, M. Dayarathna, and S. Jayasena, "Privacy preserving elastic stream processing with clouds using homomorphic encryption," in *Database Systems for Advanced Applications: 24th International Conference, DASFAA 2019, Chiang Mai, Thailand, April 22–25, 2019, Proceedings, Part II 24*, pp. 264–280, Springer, 2019.

[147] S. Bhatia and R. S. Virk, "Cloud computing security privacy and forensics: Issues and challenges ahead," *Int. J. Recent Trends Eng. Res.*, vol. 4, no. 3, pp. 10–13, 2018.

[148] L. Xu, E. Bertino, and Y. Mu, *Network and System Security: 6th International Conference, NSS 2012, Wuyishan, Fujian, China, November 21-23, Proceedings*, vol. 7645. Springer, 2012.

[149] L. Vegh and L. Miclea, "Complex event processing for attack detection in a cyber-physical system," in *2016 IEEE international conference on automation, quality and testing, robotics (AQTR)*, pp. 1–6, IEEE, 2016.

[150] B. Schilling, B. Koldehofe, K. Rothermel, and U. Ramachandran, "Access policy consolidation for event processing systems," in *2013 Conference on Networked Systems*, pp. 92–101, IEEE, 2013.

[151] H. Gu, T. Plagemann, M. Benndorf, V. Goebel, and B. Koldehofe, "Differential privacy for protecting private patterns in data streams," in *2023 IEEE 39th International Conference on Data Engineering Workshops (ICDEW)*, pp. 118–124, IEEE, 2023.

[152] G. Kellaris, S. Papadopoulos, X. Xiao, and D. Papadias, "Differentially private event sequences over infinite streams," 2014.

[153] X. Ren, S. Wang, X. Yao, C.-M. Yu, W. Yu, and X. Yang, "Differentially private event sequences over infinite streams with relaxed privacy guarantee," in *Wireless Algorithms, Systems, and Applications: 14th International Conference, WASA 2019, Honolulu, HI, USA, June 24–26, 2019, Proceedings 14*, pp. 272–284, Springer, 2019.

[154] C. C. Aggarwal and P. S. Yu, *A general survey of privacy-preserving data mining models and algorithms*. Springer, 2008.

[155] J. Roldán, J. Boubeta-Puig, J. L. Martínez, and G. Ortiz, "Integrating complex event processing and machine learning: An intelligent architecture for detecting IoT security attacks," *Expert Systems with Applications*, vol. 149, p. 113251, 2020.

[156] N. Giatrakos, E. Alevizos, A. Artikis, A. Deligiannakis, and M. Garofalakis, "Complex event recognition in the big data era: a survey," *The VLDB Journal*, vol. 29, pp. 313–352, 2020.

[157] A. M. Rahmani, Z. Babaei, and A. Souri, "Event-driven IoT architecture for data analysis of reliable healthcare application using complex event processing," *Cluster Computing*, vol. 24, no. 2, pp. 1347–1360, 2021.

[158] K. A. Alaghbari, M. H. M. Saad, A. Hussain, and M. R. Alam, "Complex event processing for physical and cyber security in datacentres-recent progress, challenges and recommendations," *Journal of Cloud Computing*, vol. 11, no. 1, p. 65, 2022.

[159] S. S. Kumar and S. Agarwal, "Rule based complex event processing for IoT applications: Review, classification and challenges," *Expert Systems*, p. e13597, 2024.

[160] J. Wanner, C. Wissuchek, and C. Janiesch, "Machine learning and complex event processing: A review of real-time data analytics for the industrial internet of things," *Enterprise Modelling and Information Systems Architectures (EMISAJ)*, vol. 15, pp. 1–1, 2020.

[161] Y. Liu, W. Yu, C. Gao, and M. Chen, "An auto-extraction framework for cep rules based on the two-layer lstm attention mechanism: A case study on city air pollution forecasting," *Energies*, vol. 15, no. 16, p. 5892, 2022.

[162] A. Alakari, K. F. Li, and F. Gebali, "A situation refinement model for complex event processing," *Knowledge-Based Systems*, vol. 198, p. 105881, 2020.

[163] W. Chen, A. El Majzoub, I. Al-Qudah, and F. A. Rabhi, "A CEP-driven framework for real-time news impact prediction on financial markets," *Service Oriented Computing and Applications*, vol. 17, no. 2, pp. 129–144, 2023.

[164] M. U. Şimsek, İ. Kök, and S. Özdemir, "Cepair: an AI-powered and fog-based predictive CEP system for air quality monitoring," *Cluster Computing*, pp. 1–15, 2024.

[165] S. Ilarri, I. Fumanal, and R. Trillo-Lado, "An experience with the implementation of a rule-based triggering recommendation approach for mobile devices," in *The 23rd International Conference on Information Integration and Web Intelligence*, pp. 562–570, 2021.

[166] M. U. Şimsek, İ. Kök, and S. Özdemir, "DeepFogAQ: A fog-assitted decentralized air quality prediction and event detection system," *Expert Systems with Applications*, p. 123920, 2024.

[167] S. R. Garzon and B. Louis, "Context flow graphs: Situation modeling for rule-based proactive context-aware systems," *IEEE Access*, vol. 8, pp. 212939–212960, 2020.

[168] C. Zhang, Y. Xie, H. Bai, B. Yu, W. Li, and Y. Gao, "A survey on federated learning," *Knowledge-Based Systems*, vol. 216, p. 106775, 2021.

[169] M. P.-L. Ooi, S. Sohail, V. G. Huang, N. Hudson, M. Baughman, O. Rana, A. Hinze, K. Chard, R. Chard, I. Foster, *et al.*, "Measurement and applications: Exploring the challenges and opportunities of hierarchical federated learning in sensor applications," *IEEE Instrumentation & Measurement Magazine*, vol. 26, no. 9, pp. 21–31, 2023.

[170] M. Xia, D. Jin, and J. Chen, "Short-term traffic flow prediction based on graph convolutional networks and federated learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 1, pp. 1191–1203, 2022.

[171] D. C. Nguyen, M. Ding, P. N. Pathirana, A. Seneviratne, J. Li, and H. V. Poor, "Federated learning for internet of things: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 3, pp. 1622–1658, 2021.

[172] J. Park, S. Samarakoon, A. Elgabli, J. Kim, M. Bennis, S.-L. Kim, and M. Debbah, "Communication-efficient and distributed learning over wireless networks: Principles and applications," *Proceedings of the IEEE*, vol. 109, no. 5, pp. 796–819, 2021.

[173] S. Feng and H. Yu, "Multi-participant multi-class vertical federated learning," *arXiv preprint arXiv:2001.11154*, 2020.

[174] S. Sharma, C. Xing, Y. Liu, and Y. Kang, "Secure and efficient federated transfer learning," in *2019 IEEE international conference on big data (Big Data)*, pp. 2569–2576, IEEE, 2019.

[175] V. Cardellini, F. Lo Presti, M. Nardelli, and G. Russo Russo, "Run-time adaptation of data stream processing systems: The state of the art," *ACM Computing Surveys (CSUR)*, 2022.

[176] M. Nardelli, V. Cardellini, V. Grassi, and F. L. Presti, "Efficient operator placement for distributed data stream processing applications," *IEEE Transactions on Parallel and Distributed Systems (TPDS)*, vol. 30, no. 8, pp. pp. 1753–1767, 2019.

[177] E. Volnes, T. Plagemann, B. Koldehofe, and V. Goebel, "Travel light: State shedding for efficient operator migration," in *Proceedings of the 16th International Conference on Distributed and Event-Based Systems (DEBS)*, pp. PP. 79–84, 2022.

[178] A. Slo, S. Bhowmik, and K. Rothermel, "State-aware load shedding from input event streams in complex event processing," *IEEE Transactions on Big Data*, vol. 8, no. 5, pp. p. 1340–1357, 2022.

[179] I. Kolchinsky and A. Schuster, "Efficient adaptive detection of complex event patterns," vol. 11, p. pp. 1346–1359, 2018.

[180] K. Aberer, M. Hauswirth, and A. Salehi, "A middleware for fast and flexible sensor network deployment," in *Proceedings of the 32nd International Conference on Very Large Data Bases (VLDB)*, p. p. 1199, 2006.

[181] M. Luthra, S. Hennig, K. Razavi, L. Wang, and B. Koldehofe, "Operator as a service: Stateful serverless complex event processing," in *Proceedings of International Conference on Big Data (Big Data)*, pp. pp. 1964–1973, IEEE, 2020.

[182] O. Corporation, "Oracle vm virtualbox manager," 2008.

[183] A. S. Foundation, "Apache kafka," 2011.

[184] Flink, "Flink-cep," 2017. https://nightlies.apache.org/flink/ flink-docs-master/docs/libs/cep/, Accessed on Oct 9th, 2023.

[185] S. Singh, S. Kumar, A. Nayyar, F. Al-Turjman, L. Mostarda, *et al.*, "Proficient QoS-based target coverage problem in wireless sensor networks," *IEEE Access*, vol. 8, pp. 74315–74325, 2020.

[186] A. I. Arafat, T. Akter, M. F. Ahammed, M. Y. Ali, and A.-A. Nahid, "A dataset for internet of things based fish farm monitoring and notification system," *Data in Brief*, vol. 33, p. 106457, 2020.

[187] M. N. Halgamuge, M. Zukerman, K. Ramamohanarao, and H. L. Vu, "An estimation of sensor energy consumption," *Progress In Electromagnetics Research B*, vol. 12, pp. pp. 259–295, 2009.

[188] R. Riesebos, V. Degeler, and A. Tello, "Smartphone-based real-time indoor positioning using ble beacons," in *Proceedings of 18th International Conference on Automation Science and Engineering (CASE)*, pp. pp. 1281–1288, IEEE, 2022.

[189] C. T. Li, J. C. Cheng, and K. Chen, "Top 10 technologies for indoor positioning on construction sites," *Automation in Construction*, vol. 118, p. p. 103309, 2020.

[190] Y. Wang, S. Han, Y. Tian, C. Xiu, and D. Yang, "Is centimeter accuracy achievable for lte-csi fingerprint-based indoor positioning?," *IEEE Access*, vol. 8, pp. pp. 75249–75255, 2020.

[191] L. Zou, A. Javed, and G.-M. Muntean, "Smart mobile device power consumption measurement for video streaming in wireless environments: Wifi vs. lte," in *Proceedings of International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*, pp. pp. 1–6, IEEE, 2017.

[192] L. Corral, A. B. Georgiev, A. Sillitti, and G. Succi, "A method for characterizing energy consumption in android smartphones," in *Proceedings of the 2nd International Workshop on Green and Sustainable Software (GREENS)*, pp. pp. 38–45, IEEE, 2013.

[193] C. Prud'homme and J.-G. Fages, "Choco-Solver: A java library for constraint programming," *Journal of Open Source Software (JOSS)*, vol. 7, no. 78, p. p. 4708, 2022.

[194] L. Li, B. Zhong, C. Hutmacher Jr, Y. Liang, W. J. Horrey, and X. Xu, "Detection of driver manual distraction via image-based hand and ear recognition," *Accident Analysis & Prevention*, vol. 137, p. p. 105432, 2020.

[195] Y. Gao, M. Diao, and T. Fujii, "Sensor selection based on dempster-shafer evidence theory under collaborative spectrum sensing in cognitive radio sensor networks," in *Proceedings of the 90th Vehicular Technology Conference (VTC2019-Fall)*, pp. pp. 1–7, IEEE, 2019.

[196] I. Younas and A. Naeem, "Optimization of sensor selection problem in IoT systems using opposition-based learning in many-objective evolutionary algorithms," *Computers & Electrical Engineering*, vol. 97, p. p. 107625, 2022.

[197] J. Zhang, J. Du, and L.-R. Dai, "Sensor selection for relative acoustic transfer function steered linearly-constrained beamformers," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. pp. 1220–1232, 2021.

[198] P. Asghari, A. M. Rahmani, and H. S. H. Javadi, "Service composition approaches in IoT: A systematic review," *Journal of Network and Computer Applications*, vol. 120, pp. pp. 61–77, 2018.

[199] S. K. Lee, S. Yoo, J. Jung, H. Kim, and J. Ryoo, "Link-aware reconfigurable point-to-point video streaming for mobile devices," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 12, no. 1, pp. pp. 1–25, 2015.

[200] C. Qin, H. Eichelberger, and K. Schmid, "Enactment of adaptation in data stream processing with latency implications — a systematic literature review," *Information and Software Technology*, vol. 111, pp. pp. 1–21, 2019.

[201] F. Gao and E. Curry, "Quality of service-aware complex event service composition in real-time linked dataspaces," in *Real-time Linked Dataspaces*, pp. pp. 169–190, Springer, 2020.

[202] A. N. Abosaif and H. S. Hamza, "Quality of service-aware service selection algorithms for the internet of things environment: A review paper," *Array*, vol. 8, p. p. 100041, 2020.

[203] S. Sefati and N. Jafari Navimipour, "A QoS-aware service composition mechanism in the internet of things using a hidden-markov-model-based optimization algorithm," *IEEE Internet of Things Journal*, vol. 8, no. 20, pp. pp. 15620–15627, 2021.

[204] A. Khalil, M. Lotfian Delouee, V. Degeler, T. Meuser, A. F. Anta, and B. Koldehofe, "Driving towards efficiency: Adaptive resource-aware clustered federated learning in vehicular networks," in *Proceedings of the 22nd Mediterranean Communication and Computer Networking Conference (MedComNet'24)*, IEEE, 2024.

[205] M. Lotfian Delouee., V. Degeler., P. Amthor., and B. Koldehofe., "APP-CEP: Adaptive pattern-level privacy protection in complex event processing systems," in *Proceedings of the 10th International Conference on Information Systems Security and Privacy - ICISSP*, pp. 486–497, INSTICC, SciTePress, 2024.

[206] M. Gundersen, "Statistics norway demands to know exactly what norwegians buy in the grocery store," 2022. https://nrkbeta.no/2022/05/28/ssb-krever-a-fa-vite-noyaktig-hva-nordmenn-kjoper-i-matbutikken/, Accessed on Oct 9th, 2023.

[207] B. Alt, M. Weckesser, C. Becker, M. Hollick, S. Kar, A. Klein, R. Klose, R. Kluge, H. Koeppl, B. Koldehofe, *et al.*, "Transitions: A protocol-independent view of the future internet," *In Proceedings of the IEEE*, vol. 107, no. 4, pp. 835–846, 2019.

[208] C. Stach and F. Steimle, "Recommender-based privacy requirements elicitation-epicurean: an approach to simplify privacy settings in IoT applications with respect to the gdpr," in *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*, pp. 1500–1507, 2019.

[209] D. Chen, S. L. Sain, and K. Guo, "Data mining for the online retail industry: A case study of rfm model-based customer segmentation using data mining," *Journal of Database Marketing & Customer Strategy Management*, vol. 19, pp. 197–208, 2012.

[210] J. Clore, K. Cios, J. DeShazo, and B. Strack, "Diabetes 130-US Hospitals for Years 1999-2008." UCI Machine Learning Repository, 2014. DOI: https://doi.org/10.24432/C5230J.

[211] T. Tran, P. Nguyen, and G. Erdogan, "A systematic review of secure IoT data sharing.," in *Proceedings of the 9th International Conference on Information Systems Security and Privacy (ICISSP'23)*, pp. 95–105, 2023.

[212] S. Tokas, G. Erdogan, and K. Stølen, "Privacy-aware IoT: State-of-the-art and challenges," in *Proceedings of the 9th International Conference on Information Systems Security and Privacy (ICISSP'23)*, pp. 450–461, 2023.

[213] J. Leicht and M. Heisel, "P2bac: Privacy policy based access control using p-lpl," in *Proceedings of the 9th International Conference on Information Systems Security and Privacy (ICISSP'23)*, pp. 686–697, 2023.

[214] F. Barber and S. Furnell, "Benchmarking consumer data and privacy knowledge in connected and autonomous vehicles.," in *Proceedings of the 8th International Conference on Information Systems Security and Privacy (ICISSP'22)*, pp. 426–434, 2022.

[215] P. Sahoo, A. K. Singh, S. Saha, V. Jain, S. Mondal, and A. Chadha, "A systematic survey of prompt engineering in large language models: Techniques and applications," *ArXiv preprint arXiv: 240207927*, 2024.

[216] A. Reiss, "PAMAP2 Physical Activity Monitoring." UCI Machine Learning Repository, 2012. DOI: https://doi.org/10.24432/C5NW2H.

[217] OpenAI, "GPT-4: Openai's language model." `https://openai.com/research/gpt-4`, 2023. Accessed: 2024-05-21.

[218] Google, "Google Gemini: Multimodal AI model." `https://ai.googleblog.com`, 2024. Accessed: 2024-05-21.

[219] Codestral, "Codestral: Empowering developers and democratising coding with mistral AI.." `https://mistral.ai/news/codestral/`, 2024. Accessed: 2024-08-19.

[220] H. Face, "Hugging face: State-of-the-art natural language processing." `https://huggingface.co`, 2024. Accessed: 2024-08-19.

[221] M. C, "A brief history of AI.." `https://medium.com/deno-the-complete-reference/a-brief-history-of-ai-0d495513f5c3`, 2024. Accessed: 2024-08-19.

[222] DeepSeek-AI, "Deepseek-v3 technical report," 2024.

[223] getgenie.ai, "Gpt 3 vs. GPT 4: What you need to know.." `https://getgenie.ai/`, 2024. Accessed: 2024-08-19.

[224] OpenAI, "Openai pricing," 2024. https://openai.com/api/pricing/, Accessed on Oct 9th, 2023.

[225] Z. Wang, X. Xiao, and S. Rajasekaran, "Novel and efficient randomized algorithms for feature selection," *Big Data Mining and Analytics*, vol. 3, no. 3, pp. 208–224, 2020.

[226] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, and G. Neubig, "Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing," *ACM Computing Surveys*, vol. 55, no. 9, pp. 1–35, 2023.

[227] S. Kirkpatrick, C. D. Gelatt Jr, and M. P. Vecchi, "Optimization by simulated annealing," *science*, vol. 220, no. 4598, pp. 671–680, 1983.

[228] K. A. Dowsland, "Simulated annealing," in *Modern heuristic techniques for combinatorial problems*, pp. 20–69, ACM, 1993.

[229] K. A. Dowsland, "Some experiments with simulated annealing techniques for packing problems," *European Journal of Operational Research*, vol. 68, no. 3, pp. 389–399, 1993.

[230] M. Rahman, R. R. Othman, R. B. Ahmad, and M. M. Rahman, "Event driven input sequence t-way test strategy using simulated annealing," in *2014 5th International Conference on Intelligent Systems, Modelling and Simulation*, pp. 663–667, IEEE, 2014.

[231] B. Chopard, M. Tomassini, B. Chopard, and M. Tomassini, "Simulated annealing," *An introduction to metaheuristics for optimization*, pp. 59–79, 2018.

[232] D. Delahaye, S. Chaimatanan, and M. Mongeau, "Simulated annealing: From basics to applications," *Handbook of metaheuristics*, pp. 1–35, 2019.

[233] M. Lotfian Delouee, D. G. Pernes, V. Degeler, and B. Koldehofe, "Towards federated LLM-powered CEP rule generation and refinement," in *Proceedings of the 18th ACM International Conference on Distributed and Event-Based Systems*, DEBS '24, (New York, NY, USA), p. 185–186, Association for Computing Machinery, 2024.

[234] O.-J. Lee and J. E. Jung, "Sequence clustering-based automated rule generation for adaptive complex event processing," *Future Generation Computer Systems*, vol. 66, pp. 100–109, 2017.

[235] Y. Liu, W. Yu, X. Zhai, B. Zhang, K. D. McDonald-Maier, and M. Fasli, "Multi-level CEP rules automatic extraction approach for air quality detection and energy conservation decision based on AI technologies," *Applied Energy*, vol. 372, p. 123724, 2024.

[236] R. Bruns and J. Dunkel, "Bat4CEP: A bat algorithm for mining of complex event processing rules," *Applied intelligence*, vol. 52, no. 13, pp. 15143–15163, 2022.

[237] C. O. Kumar, N. Gowtham, M. Zakariah, and A. Almazyad, "Multimodal emotion recognition using feature fusion: An LLM-based approach," *IEEE Access*, 2024.

[238] Q. Delooz, A. Willecke, K. Garlichs, A.-C. Hagau, L. Wolf, A. Vinel, and A. Festag, "Analysis and evaluation of information redundancy mitigation for v2x collective perception," *IEEE Access*, vol. 10, pp. 47076–47093, 2022.

[239] L. You, S. Liu, Y. Chang, and C. Yuen, "A triple-step asynchronous federated learning mechanism for client activation, interaction optimization, and aggregation enhancement," *IEEE Internet of Things Journal*, vol. 9, no. 23, pp. 24199–24211, 2022.

[240] Z. He, L. Wang, and Z. Cai, "Clustered federated learning with adaptive local differential privacy on heterogeneous IoT data," *IEEE Internet of Things Journal*, 2023.

[241] D. Jallepalli, N. C. Ravikumar, P. V. Badarinath, S. Uchil, and M. A. Suresh, "Federated learning for object detection in autonomous vehicles," in *2021 IEEE Seventh International Conference on Big Data Computing Service and Applications (BigDataService)*, pp. 107–114, IEEE, 2021.

[242] G. Rjoub, O. A. Wahab, J. Bentahar, and A. S. Bataineh, "Improving autonomous vehicles safety in snow weather using federated yolo CNN learning," in *International Conference on Mobile Web and Intelligent Information Systems*, pp. 121–134, Springer, 2021.

[243] T. Zheng, A. Li, Z. Chen, H. Wang, and J. Luo, "Autofed: Heterogeneity-aware federated multimodal learning for robust autonomous driving," *arXiv preprint arXiv:2302.08646*, 2023.

[244] Y. Chen, C. Wang, and B. Kim, "Federated learning with infrastructure resource limitations in vehicular object detection," in *2021 IEEE/ACM Symposium on Edge Computing (SEC)*, pp. 366–370, IEEE, 2021.

[245] S. Wang, C. Li, D. W. K. Ng, Y. C. Eldar, H. V. Poor, Q. Hao, and C. Xu, "Federated deep learning meets autonomous vehicle perception: Design and verification," *IEEE network*, 2022.

[246] A. M. Elbir, S. Coleri, A. K. Papazafeiropoulos, P. Kourtessis, and S. Chatzinotas, "A hybrid architecture for federated and centralized learning," *IEEE Transactions on Cognitive Communications and Networking*, vol. 8, no. 3, pp. 1529–1542, 2022.

[247] X. Zhou, W. Liang, J. She, Z. Yan, I. Kevin, and K. Wang, "Two-layer federated learning with heterogeneous model aggregation for 6g supported internet of vehicles," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 6, pp. 5308–5317, 2021.

[248] B. Li, Y. Jiang, W. Sun, W. Niu, and P. Wang, "Fedvanet: Efficient federated learning with non-iid data for vehicular ad hoc networks," in *2021 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, IEEE, 2021.

[249] M. Duan, D. Liu, X. Ji, Y. Wu, L. Liang, X. Chen, Y. Tan, and A. Ren, "Flexible clustered federated learning for client-level data distribution shift," *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, no. 11, pp. 2661–2674, 2021.

[250] R. Lu, W. Zhang, Y. Wang, Q. Li, X. Zhong, H. Yang, and D. Wang, "Auction-based cluster federated learning in mobile edge computing systems," *IEEE Transactions on Parallel and Distributed Systems*, vol. 34, no. 4, pp. 1145–1158, 2023.

[251] C. Quadri, V. Mancuso, V. Cislaghi, M. A. Marsan, and G. P. Rossi, "From plato to platoons," in *MedComNet'21*, pp. 1–8, 2021.

[252] A. Taik, Z. Mlika, and S. Cherkaoui, "Clustered vehicular federated learning: Process and optimization," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 12, pp. 25371–25383, 2022.

[253] L. Crosara, M. Brocco, C. Cavalagli, X. Wu, E. Gindullina, and L. Badia, "Data injection in a vehicular network framed within a game theoretic analysis," in *MedComNet'23*, pp. 25–28, 2023.

[254] ETSI, "ETSI TS 103 324 V2.1.1 (2023-06)," 2023. Accessed: February 29, 2024.

[255] Y. Sunyoto and I. Rubin, "Millimeter wave data networking for autonomous vehicle systems," in *MedComNet'20*, pp. 1–8, 2020.

[256] F. Busacca, C. Cirino, G. Faraci, C. Grasso, S. Palazzo, and G. Schembra, "Multi-layer offloading at the edge for vehicular networks," in *MedComNet'20*, pp. 1–8, 2020.

[257] A. Y. Ding, E. Peltonen, T. Meuser, A. Aral, C. Becker, S. Dustdar, T. Hiessl, D. Kranzlmüller, M. Liyanage, S. Maghsudi, N. Mohan, J. Ott, J. S. Rellermeyer, S. Schulte, H. Schulzrinne, G. Solmaz, S. Tarkoma, B. Varghese, and L. Wolf, "Roadmap for edge AI: a dagstuhl perspective," *SIGCOMM Comput. Commun. Rev.*, vol. 52, p. 28–33, mar 2022.

[258] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistic, PMLR 54*, pp. 1273–1282, 2017.

[259] A. Nilsson, S. Smith, G. Ulm, E. Gustavsson, and M. Jirstrand, "A performance evaluation of federated learning algorithms," in *Proceedings of the Second Workshop on Distributed Infrastructures for Deep Learning (DIDL'18)*, pp. 1—-8, ACM, 2018.

[260] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Proceedings of the 1st Annual Conference on Robot Learning*, pp. 1–16, 2017.

[261] G. Jocher, A. Chaurasia, and J. Qiu, "YOLO by Ultralytics," 2023.

[262] Ultralytics, "Ultralytics documentation: Train mode arguments," 2023.

[263] R. Padilla, S. L. Netto, and E. A. Da Silva, "A survey on performance metrics for object-detection algorithms," in *2020 international conference on systems, signals and image processing (IWSSIP)*, pp. 237–242, IEEE, 2020.

[264] AveesLab, "Carfree: Automatic ground truth generation in carla," 2021. GitHub repository.

[265] CARLA Simulator, "CARLA documentation - Map Town04," 2022. Online documentation.

[266] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings, *et al.*, "Advances and open problems in federated learning," *Foundations and Trends® in Machine Learning*, vol. 14, no. 1–2, pp. 1–210, 2021.

[267] A. Khalil, A. Wainakh, E. Zimmer, J. Parra-Arnau, A. Fernández Anta, T. Meuser, R. Steinmetz, *et al.*, "Label-aware aggregation for improved federated learning," in *IEEE International Conference on Fog and Mobile Edge Computing*, 2023.

Samenvatting

Deze thesis presenteert concepten en algoritmes om een kwaliteit- en privacybewust Complex Event Processing (CEP) systeem te realiseren, dat ondersteund wordt door autonome regelgeneratie en generaliseerbaar is naar verschillende IoT-toepassingen in dynamische en heterogene omgevingen. In het bijzonder worden vier bijdragen voorgesteld: (i) een mechanisme voor kwaliteitbewuste selectie en omschakeling van sensorconfiguraties als input voor CEP-systemen, (ii) een grafgebaseerd mechanisme voor privacybescherming op patroon-niveau, (iii) een AI-gebaseerde aanpak voor autonome regelgeneratie en -verbetering, en (iv) een toepassing van het privacy-utility trade-off (PUT) concept in een voertuignetwerk voor online objectdetectie.

In hoofdstuk 3 wordt AQuA-CEP geïntroduceerd, waarin we kwaliteitsbeleid door de consument definieerbaar maken om de detectie van complexe events te begeleiden. Deze beleidsregels worden geëvalueerd om automatisch geschikte gegevensbronnen binnen de sensorinfrastructuur te selecteren of te configureren. We hebben verschillende manieren onderzocht om kwaliteitsbeleid uit te drukken en geanalyseerd hoe deze van invloed zijn op het kwaliteitsbewakingsproces. Daarnaast hebben we verschillende methoden voor het evalueren en toepassen van kwaliteit-gerelateerde aanpassingen bekeken en hun effecten op de correlatie-efficiëntie geanalyseerd.

We hebben AQuA-CEP geëvalueerd in IoT-scenario's door de prestaties te beoordelen op basis van kwaliteitsbeleid en query-verwerkingsaanpassing, waarbij kennis uit kwaliteitsbewaking is gebruikt. Onze resultaten toonden aan dat AQuA-CEP de prestaties van DCEP-systemen kan verbeteren op

het gebied van resultaatkwaliteit, terwijl tegelijkertijd voldaan wordt aan de kwaliteitsvereisten van de consument (QR). Daarnaast bleek dat kwaliteitsgebaseerde aanpassing de levensduur van het netwerk kan verlengen door het energieverbruik van sensoren te optimaliseren via een efficiënte selectie van gegevensbronnen.

Ten tweede hebben we binnen het APP-CEP-framework de uitdaging aangepakt om privacy op patroon-niveau te integreren in event-gebaseerde systemen in hoofdstuk 4. APP-CEP maakt gebruik van selectieve obfuscatie technieken om privé-informatie te verbergen. In tegenstelling tot bestaande methoden, hebben we privacy afgedwongen zonder afhankelijk te zijn van daadwerkelijke events in de datastromen. Hiervoor hebben we CEP-achtige patronen gebruikt om zowel gebruikersvragen als privacyvereisten vast te leggen.

We beschermden de privacy door het genereren van patroon- afhankelijkheidsdiagrammen en het dynamisch selecteren van obfuscatie technieken die de interferentie met zowel gevoelige als niet-gevoelige patronen, essentieel voor het behouden van een acceptabele kwaliteit van service, minimaliseren. Daarnaast hebben we de kennis van potentiële aanvallers gemodelleerd om hun vermogen om de privacy te compromitteren te beoordelen en onze obfuscatie procedures dienovereenkomstig verfijnd. Door middel van evaluaties met een dataset van online transacties en een medisch dataset hebben we de effectiviteit van APP-CEP aangetoond in het balanceren van privacy en bruikbaarheid. Onze aanpak voor het modelleren van achtergrondkennis voorkomt effectief dat aanvallers de aanpassingen in de inputstreams detecteren.

In hoofdstuk 5 hebben we in de context van GPT-CEP gebruikgemaakt van grote taalmodellen en prompt engineering om nauwkeurige en efficiënte CEP-regels te genereren. Hiervoor werd de AGES-index geïntroduceerd, een uitgebreide maatstaf die de modelgrootte, efficiëntie van regelgeneratie en nauwkeurigheid van regels evalueert. De prestaties van GPT-CEP werden met deze index beoordeeld en vergeleken met basisstrategieën. GPT-CEP presteerde consistent beter dan de baselines, waarbij de AGES-index een cruciale rol speelde bij het selecteren van het optimale taalmodel en de beste prompt engineering-technieken.

GPT-CEP bevat ook een cluster-gebaseerde benadering voor het verfijnen van de initiële door LLM gegenereerde regels. Simulated annealing werd toegepast om de drempels van regels verder te optimaliseren, wat de detectienauwkeurigheid verder verbeterde. Over het geheel genomen vormt GPT-CEP een belangrijke stap vooruit in autonome regelgeneratie voor CEP-systemen. Door grote taalmodellen, prompt engineering en regelverfijning te combineren, biedt het een veelbelovende oplossing voor het automatiseren van regelcreatie en het verbeteren van de prestaties van CEP-toepassingen.

Ten slotte onderzoeken we in hoofdstuk 6 de toepassing van privacy-utility trade-offs in streamverwerkende systemen, met een speciale focus op autonome voertuignetwerken, in de context van AR-CFL. AR-CFL biedt een overtuigende oplossing door modeltraining mogelijk te maken tussen gedistribueerde voertuigen zonder dat ruwe sensorgegevens hoeven te worden gedeeld, waardoor privacyproblemen worden aangepakt. In voertuignetwerken zijn continue aanpassing en real-time modelupdates essentieel vanwege dynamische rijomstandigheden en constante datageneratie. AR-CFL faciliteert deze aanpassing door gebruik te maken van clustered federated learning, wat de communicatieoverhead vermindert en zorgt voor efficiënte benutting van bronnen door deelnemers te clusteren op basis van beschikbare bandbreedte, opslag en rekenkracht — belangrijke overwegingen in streamverwerkende systemen.

AR-CFL fungeert als een trade-off tussen privacy en bruikbaarheid, cruciaal voor het beheren van continue datastromen die door voertuigen worden gegenereerd, terwijl gevoelige informatie wordt beschermd. Ons framework optimaliseert dynamisch het aantal clusters en selecteert deelnemers op basis van beschikbare middelen, wat zorgt voor efficiënte eventverwerking. Evaluatieresultaten tonen aan dat AR-CFL robuuste detectieprestaties behaalt voor online perceptiemodeltraining, zelfs met non-IID-gegevens over verschillende verkeersdichtheden. Bovendien verbeterde de trainingsefficiëntie van deelnemende nodes met maximaal 25%, terwijl de communicatie-overhead met 33% verminderde in vergelijking met conventionele federated learning-methoden, terwijl privacy werd gewaarborgd door lokale data-analyse. Dit toont aan dat AR-CFL effectief kan voldoen aan de vereisten van streamverwerking in voertuignetwerken, met een evenwichtige trade-off tussen privacybescherming en modelprestaties.