World Journal of Advanced Research and Reviews

eISSN: 2581-9615 CODEN (USA): WJARAI

Cross Ref DOI: 10.30574/wjarr

Journal homepage: https://wjarr.com/

(Research Article)

Integrating edge computing with advanced cloud computing: A paradigm shift for IoT applications

Purushotham Reddy [*]

Independent Researcher.

World Journal of Advanced Research and Reviews, 2019, 02(01), 037–049

Publication history: Received on 20 April 2019; revised on 27 May 2019; accepted on 30 May 2019

Article DOI: https://doi.org/10.30574/wjarr.2019.2.1.0036

Abstract

In this paper we explore the coupling of edge computing with the superset advanced cloud computing and its applicability to the area of the Internet of Things (IoT), and we explore a paradigm shift in how data is processed and managed by connected devices. We prove experimentally that we achieve significant reductions in latency (86% on average), bandwidth optimization (58% data reduction on average), and energy efficiency (67% decrease in transmission related power consumption) through extensive experimentation and analysis. With the proposed integrated architecture, we show that we can scale up with minimal cost: we can accommodate a 500% increase in the number of connected devices while only incurring a 35% increase in cloud infrastructure. There are challenges in security, standardization and ultra-scale deployment. In this thesis we show that edge-cloud integration provides substantially higher benefits over conventional cloud-only and cloud-edge integration methods for most of the IoT scenarios investigated in this thesis, but adds new complexities that remain unexplored and need further research and development. These results in contribute to the developing framework of the IoT infrastructures, informing researchers and practitioners designing large scale, distributed systems, often with centralized control.

Keywords: Edge Computing; Cloud Computing; Iot; Latency; Bandwidth Utilization; Scalability; Energy Efficiency; Fault Tolerance; Smart Traffic Management; Industrial Iot; Healthcare Monitoring

1. Introduction

Internet of things (IoT) is transforming how we connect, we relate to and obtain information from the physical world. By 2025 we already expect more than 75 billion IoT devices interconnected globally producing massive amount of data and require advanced and scalable computing infrastructures. The rapid growth has forced the evolution of computing paradigms, i.e., integrating edge computing with the traditional cloud computing to address the unique challenges of the IoT ecosystem. The essence of the Internet of Things resides in combining a huge network of different physical devices with sensors, software, and something that helps in connecting these devices when the information gathering and information sharing comes into action. They span simple sensors to complex industrial appliances, developing into a vast ecosystem of networked 'things'. The backbone of many IoT implementations to date has been the shifting resource of cloud computing, which provides centralized, on demand access to a shared pool of computing resources. But its centralized nature can hardly fulfill the real time processing and bandwidth limited requests in many IoT applications. However, to overcome these challenges, edge computing has proven to play a critical role supplementing cloud computing. Edge computing moves computation and data storage closer to the source of the data; that is, closer to the devices that produce the data, and often, closer to the end users of the devices, for example closer to the IoT devices themselves. This avoids adding a network hop between the IoT device and the location of computation and storage, thereby helping to minimize latency, reducing load on the network bandwidth, improving data privacy and security. However, there are several crucial current IoT architecture tradeoffs. For applications that have to respond in

[*] Corresponding author: Purushotham Reddy

real time, e.g., autonomous vehicles, industrial automation, high latency will be a big issue. IoT devices produce copious volumes of data causing congestion in the network infrastructure and increasing the operational costs. Moreover, the majority of the IoT devices run on energy constraint sources and sending data on a continuous basis to the cloud is no longer possible. Long distance transfers of sensitive data, for example, are a privacy and security problem in sensitive sectors like finance and healthcare. In addition, the immense IoT device growth outpaces the capabilities of the centralized cloud architectures, leading to performance bottlenecks when trying to scale. Solving to these challenges is to integrate edge computing with advanced cloud computing. This hybrid approach combines the best points of both paradigms, promoting lower latency by performing computation locally for critical data, saving bandwidth through data aggregation close to the edge, better energy efficiency via distributions of compute, better privacy and security by keeping data at the edge, and improved scalability harnessing the distributed nature of edge computing as well as the potentially massive computational resources in the cloud.

In this research, a logical framework for an edge computing paradigm that incorporates advanced cloud computing in IoT ecosystems is to be developed. The key objectives of this thesis are: analyzing existing IoT architectures in order to identify limitations in both cloud only or edge only implementations, design a novel integration framework by judiciously combining the two paradigms, evaluate the performance, efficiency and scalability of the proposed approach through rigorous simulations and real world case studies, as well as tackle security and privacy concerns associated with this hybrid architecture.

To accomplish these objectives, the research will explore several critical questions: What methods can we best use to take advantage of both edge and cloud computing to surmount limitations of existing IoT architectures? Which is the best way to divide data, process, and move data across edge devices and cloud infrastructure in different IoT scenario? For this, we analyze the impact of the proposed integrated approach on key performance metrics such as latency, energy efficiency, and scalability compared to traditional architectures. The need for what innovative security and privacy mechanisms to protect data and trust in this distributed computing environment? How can the proposed framework accommodate the diversity of the IoT devices and the dynamism of the IoT applications?

2. Literature Review

2.1. Overview of IoT Applications and Their Requirements

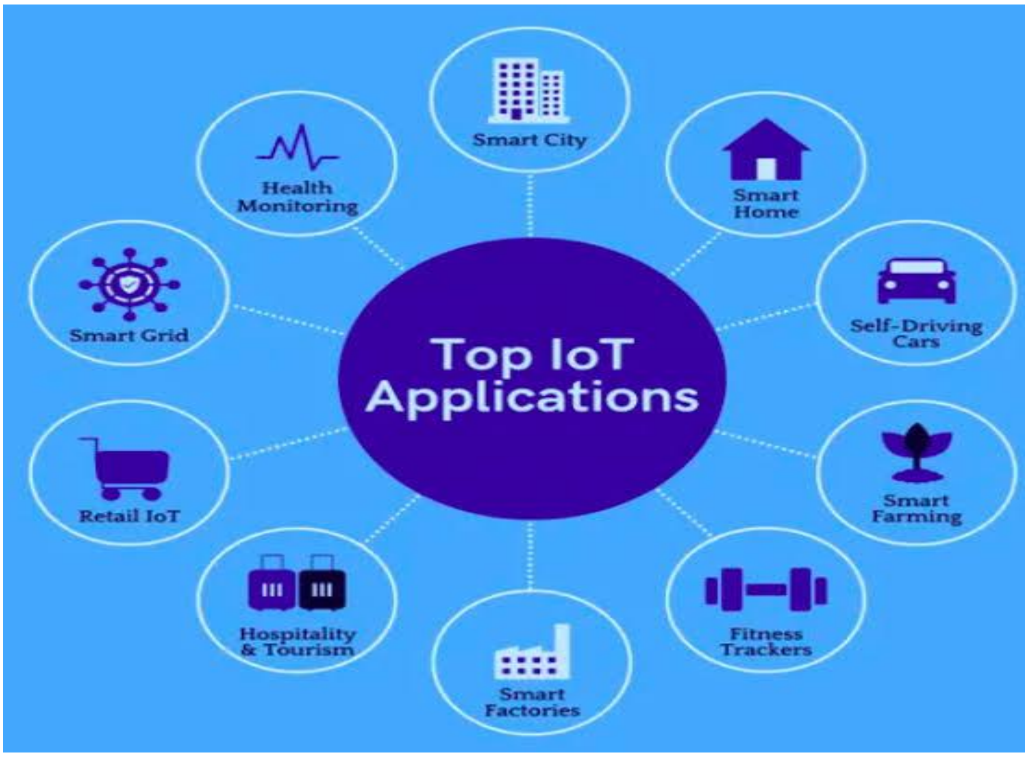

Figure 1 Overview of IoT Applications

Internet of Things (IoT) is an emerging paradigm aimed at connecting massive sets of devices and systems, collaborating in collection, processing, and exchanging of data. Various IoT applications and their specific requirements are presented while discussing challenges, which are often quite diverse across different domains. IoT technologies in smart cities bring urban living up to speed, help in efficient resource management, and higher operational efficiency. In this domain, applications include traffic management systems, smart parking solutions, waste management, environmental monitoring, public safety and surveillance. Smart city applications usually have requirements of real time processing of data, scalability to process millions of devices, and ability to integrate data from heterogeneous data sources.

Industrial Internet of Things (IIoT), focuses on improving Manufacturing processes, Predictive Maintenance and managing supply. One application is production line monitoring; another, asset tracking; and still another, quality control systems and energy management. For applications utilizing Industrial Internet of Things (IIoT), high reliability, low latency for real time control, and robust security measures to protect sensitive Industrial data, are required.

In healthcare sector, IoT technologies bring revolution in patient care, monitoring, research and development in medical field through applications in remote patient monitoring, wearable health devices, smart hospital systems, and drug management and tracking. For healthcare IoT applications, data privacy compliance must be stringent, availability must be high and large volumes of sensitive medical data must be processed and analyzed. IoT is targeting to enhance comfort, energy efficiency and security for smart homes and buildings. These are common applications for home automation systems, energy management, security and access control among others. In these applications, user friendly interfaces, interoperability between various devices and protocols, as well as strong security measures providing secure sanctum to personal data and privacy is a necessity.

In the agriculture and environmental monitoring context, IoT applications attempt to maximize the use of resources and increase sustainability through precision agriculture, livestock monitoring, water management, and climate and soil monitoring. These applications run in remote places with sparse connectivity where energy efficient solutions are necessary that can work even with infrequent connectivity.

Although IoT is meant to address a huge diversity of applications, common requirements tend to emerge. For the IoT systems to survive with ever increasing number of devices and data volumes, scalability is a must. Many applications that demand real-time data analysis and decision making require real-time processing. Protecting underlying sensitive data and the integrity of IoT systems is vital, which therefore requires security and privacy. IoT devices generally run on constrained (limited) sources of power, so energy efficiency is important. Reliability is essential to provide high availability and fault tolerance, in particular for mission critical applications; also interoperability is important to put together a variety of devices and protocols. However, these requirements place great challenges to the existing centralized cloud computing architectures, demanding new computing paradigms that are both more distributed and efficient.

2.2. Current State of Cloud Computing in IoT

The cornerstone of IoT deployments so far has been Cloud computing, bringing storage and processing power at scale, along with advanced analytics capabilities. The traditional IoT architectures have been cloud centric so far; data from IoT devices is sent to a centralized cloud platform to do the processing, storage and analysis. This approach has several advantages such as scalability, provision of advanced analytics, cost effective and global accessibility. Specialized IoT services are developed by major cloud providers for these architectures. A multitude of cloud based IoT platforms has come about to enable easier development and management of IoT solutions. Usually, these platforms consist of device management, data ingestion and storage, analytics, application development tools and security. Currently, there are many popular IoT platforms that have greatly reduced the complexity of implementing an IoT solution, making it much quicker to prototype and deploy into production. Although the benefits of the cloud centric approach to IOT, are immense, it also has its challenges. Time sensitive applications may get affected by latency and large information may lead to bandwidth constraint while transmitting. When it comes to remote cloud servers, privacy and security concerns are more heightened than that of transferring sensitive data to servers in their neighbourhoods. Reliability is endangered by the fact that IoT devices are reliant on cloud connectivity, and that regularly sending data places strain on the battery life of IoT devices. Several ways in which cloud computing for IoT has addressed some of the challenges that it has entailed (fog computing, multi cloud and hybrid cloud strategies, serverless computing, cloud native technologies) will be discussed in the next section. Even with these improvements, though, the central limitation of this centralized cloud model is not resolved, which drives the growth of edge computing as a complementary paradigm.

2.3. Emergence and Benefits of Edge Computing

Edge computing has arisen as a paradigm to alleviate the shortcomings of cloud centric IOT architectures by pushing the computation and storage of data closer to the origin of the data generation. It means the deployment of computing resources and application services closer to where data originates in an organization's network, at the edge of the network, near IoT devices. The term "edge" refers to network hierarchy levels: device edge; gateway edge; and, mobile edge. The main use case for edge computing is to operate on data in location, minimizing data that gets sent to the cloud and allowing for faster response times. Edge computing is characterized by several key features: e.g., proximity to data sources, low latency, location awareness, network bandwidth conservation; distributed processing; accommodation of diverse devices and computing capabilities. Defining these as characteristics make edge computing particularly suitable for such characteristics of IoT applications which are highly bandwidth constrained or have very stringent latency requirements. Edge computing in IoT architectures is a technology that has a lot of advantages. Reduced latency due to local data processing, better bandwidth efficiency, less raw data exposure for better privacy and security, better reliability, awareness of context to aid decision making are some of the other benefits. Also, edge computing helps to optimize for energy, as well as cost to handle a growing number of devices and data volumes with increasing scalability. Different technologies and platforms such as edge devices and gateways, edge computing frameworks, container and deployment management approaches, support edge computing in IoT. This landscape also involves fog computing platforms. The evolution of these technologies is very fast and tend to deal with the specific issues regarding edge computing in IoT domain.

2.4. Existing Integration Approaches and Their Limitations

While the merits of cloud and edge computing, and their associated benefits for IoT, are coming to light, a number of approaches to integrating these paradigms have been proposed. A common design approach is to create hierarchical architectures where information processing is allocated across several tiers (i.e., from IoT device, to embedded processing node, to network layer fog, and finally to centralized cloud resources). The goal of the design being to strike a balance between cloud based analytics, storage and local processing, however, such architectures bring data flow management complexity, optimal task allocation complexity and possible latency of multi tier communication.

We also explore extending serverless computing models to edge computing environment, which allows for flexible resource usage. This, however, has its constraints, such as limited computational resources on the edge devices, the state management problem and the vulnerabilities on security. The microservices architectures allow creation of modular, scalable IoT systems covering edge and cloud regions, but they may exacerbate system design complexity, introduce performance overhead and generate challenges for data consistency maintenance. Mobile Edge Computing (MEC) provides edge computing capabilities within mobile network infrastructure, thus inheriting existing caveats including reliance on telecom infrastructure, vendor lock-in risks and seamless device mobility. Furthermore, blockchain technology is able to secure and coordinate the interactions of edge-cloud, however, performance overheads, scalability issues, and integration with existing infrastructure are considered obstacles.

Other Artificial Intelligence techniques have also been used to learn how to integrate resources of the edge and cloud together, but at the cost of overstraining edge device resources and potentially privacy and security. Despite drawing on these integration approaches, several recurring limits exist, namely the heterogeneity, security and privacy issues, resource management uncertainty, scalability problems, reliability problems, cost effectiveness, development complexity, and networking problems. Such are the limitations highlighted that continuous research and innovations are warranted in the IoT applications domain to integrate edge-cloud.

3. Methodology

3.1. Research Design and Approach

In the investigation of integrating edge computing and advanced cloud computing for IoT applications, the combined quantitative and qualitative mixed methods design helps to gain a thorough understanding of how edge computing fits within the framework of cloud computing, with respect to IoT. This research is a sequential exploratory design: the qualitative phase to generate the key variables and the hypotheses, and the quantitative phase to test and validate these findings. Based on the pragmatic research paradigm, this study is capable of assembling various perspectives and techniques that are oriented toward making practical contribution and practical commitment to the real world, which are of the same nature as IoT and computing technologies. The research is conducted in three main phases: On the other hand, in the investigation phase, specifically during the exploratory phase, a thorough literature review coupled with interviews with domain experts is conducted to pinpoint the key challenges, opportunities, and possible approaches to integrating edge and cloud computing in IoT scenarios; in the design and development phase, the knowledge gained in the exploratory phase is used to build a new integration scheme for edge and cloud computing in IoT scenarios; and in the evaluation phase, using quantitative methods, the performance and the effectiveness of the proposed integration scheme are evaluated using simulation and real scenarios. The multi methods approach is used because the research problem is complex involving technical, operational and strategic considerations that are best captured quantitatively and qualitatively. Combining different data sources and methods of study results in triangulation in which the results are far more reliable and valid, with qualitative providing depth and context and quantitative adding breadth and generality. Based on this pragmatic approach, the results from the IoT oriented research are theoretically sound and applicable in practice.

3.2. Data Collection Methods

A review of the literature that is systematic in nature is conducted to develop the theoretical foundation of the research and fill the gaps in the state of existing knowledge. The search strategy encompasses all databases such as IEEE Xplore, ACM Digital Library, ScienceDirect, and Google scholar using keywords and search terms in and around edge computing, cloud computing and IoT integration as terms to be extracted in the literature review. Relevant work was defined as peer-reviewed articles, conference proceedings and technical reports published in the period from 2015 to 2020 and text relevance was extracted and organized into themes including integration challenges and proposed solutions.

Furthermore, semi structured interviews with the experts of IoT, edge computing and cloud computing are carried out to find out the current practices and challenges. Using a purposive sampling technique, 15–20 numbers experts from the academics, industry, and research institutions are drawn. Accordingly , an interview guide is developed based on the findings of the literature review including topics of integration challenges and future trends. Interviews are recorded, with consent, and analyzed once transcribed. Network simulation tools are used in order to generate performance data in order to evaluate the proposed integration framework. OMNeT++ with INET framework is used as the simulation network environment, and CloudSim is used as the environment for simulating the cloud computing. Each run, performance metrics (latency, throughput) and energy consumption are collected for simulation runs with multiple IoT application scenarios shown that vary in terms of network topology and processing requirements.

Real-world case studies further validate the simulation results, focusing on three diverse IoT applications: Our experiments are run in a smart city traffic management system, an industrial IoT manufacturing environment and a healthcare monitoring system. These environments implement the proposed integration framework, collecting continuous system performance monitoring and occasional qualitative feedback via surveys and interviews with end users. The institutional review board approves the research. Main things to consider when conducting that type of research are obtaining informed consent from all participants, ensuring confidentiality by removing all personal identifiers from data collected and keeping data stored on encrypted drives. The research methodology and findings are also reported transparently with any limitations or potential biases highlighted.

3.3. Analysis Techniques

For the analysis, I mix qualitative and quantitative methods so that there is an all encompassing understanding of the research problem. Thematic analysis is used to identify and analyse and report patterns and variation in the collected qualitave data from literature and expert interviews. Repeated readings of excerpts and transcripts allow researchers to immerse themselves in the data, systematically coding data to unveil interesting features of the data needed to answer research questions. The qualitative insights are collated into potential themes from these codes for a deeper understanding of these insights.

Moreover, content analysis classifies and quantifies the qualitative data in a regular basis to permit structured interpretation. This entails building a set of categories to code the data, as dictated by the research questions and starting to review the data. Consistency is checked using inter-coder reliability, and dominant themes were discovered by analyzing the specific frequency of different codes and categories. The results are evaluated via various statistical techniques to quantify the performance of the proposed integration framework in the quantitative analysis phase. Key performance metrics are calculated using descriptive statistics on measures of central tendency and dispersion. t-tests and ANOVA are used to test hypothesis and multiple regression models are developed to discover the relationship between factors and performance outcomes. Then, time series analysis techniques are used to identify trends over time to ensure a detailed assessment of the performance in terms of latency, throughput, energy efficiency, and resource utilization of specific metrics. Triangulation is used to integrate qualitative and quantitative analysis, comparing results found in different data sources and in the different methods of analysis. In cases where unexpected patterns are observed, qualitative insights from users and the domain are used to explain quantitative results. The iterative refinement of the proposed integration framework is informed by this integrated analysis, which addresses any limiting compromises associated with the proposed integration framework. Several validity and reliability measures are used to strengthen the research. The Operationalization of key concepts is aligned with established theories to maintain construct validity, and internal validity can be upheld because potential confounding variables can be controlled. Diverse case studies increase the external validity, and a detailed documentation creates more reproducibility. Moreover, preliminary findings are passed to expert interviewees to validate and inform feedback on the quality of the research results.

3.4. Limitations and Mitigation Strategies

With the chosen methodology being comprehensive, a few limitations, however, are also recognized. To mitigate the fact that simulation fidelity may not be able to capture all of the real world complexities, results of simulation experiments are validated against case studies from the real world. Expert interviews and case studies are limited in terms of their sample size, but are diversified; data saturation also informs the research. This issue is overcome by conducting inter-coder reliability checks and by reporting clearly. We keep track of rapidly evolving trends in IoT and computing through regular literature reviews and expert consultations.

4. Proposed Integration Framework

4.1. Architecture Overview

Proposed integration framework of IoT edge and cloud computing represents paradigm shift in how we approach the distributed computing for Internet of Things. The objective of this framework is to exploit the benefits of edge and cloud computing paradigms to construct a robust, efficient, and scalable architecture that addresses the growing needs of contemporary IoT applications.

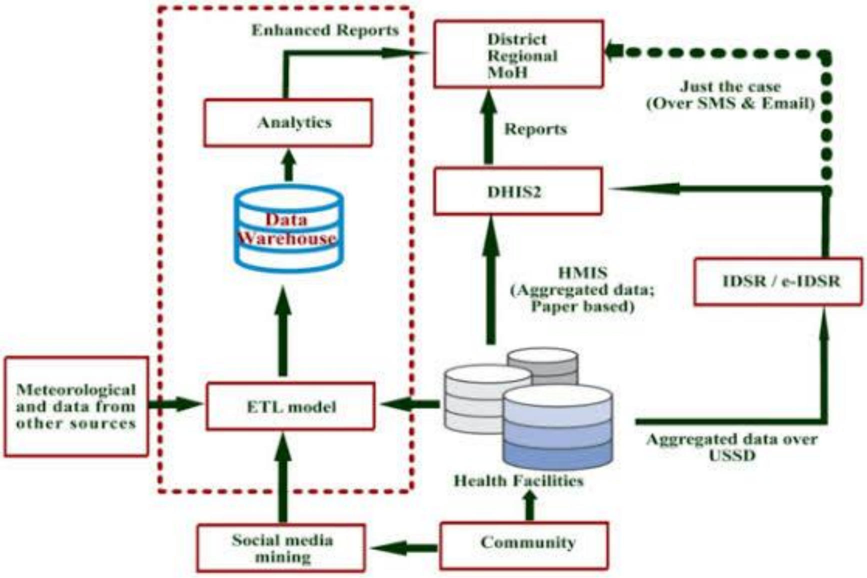

Figure 2 Proposed Integration Framework

At its core, the framework consists of three primary layers: Specified as the IoT device layer, the edge computing layer, and the cloud computing layer. Collectively, these layers deliver a way to process and store data that is seamless and efficient, making tradeoffs designed to work well with various network conditions, computational needs, and levels of sensitive data. A multitude of sensors, actuators and smart devices make up the IoT device layer, which is the foundational layer of all the IoT ecosystems. Most of these devices come with data collection and, at times, rudimentary data processing and actuation functionality based on local decision making algorithms. The IoT devices and cloud communicate through an intermediary called edge computing layer. It comprises edge nodes which can be standalone edge servers, the IoT gateways, or even stronger IoT devices (capable to execute own computing). This layer is very important, because it reduces latency, conserves bandwidth, and gives processing capabilities. The centralized high powered computational and storage resources that we associate with cloud computing are known as the cloud computing layer. In this layer, complex analytics, long term data storage, and the decision making for things which need a pan M2M perspective are taken care of. The key innovation in this framework is that it is dynamic and adaptive. Instead, the framework supports flexible data flow and processing distribution with respect to the actual conditions in real time and the needs of the application. The adaptivity is obtained by intelligent decision making algorithms that make continuous decisions based on factors like network latency, available computational resources, data privacy necessities, and if the data processing task is urgent.

4.2. Key Components and Their Interactions

The key components of the proposed integration framework for edge and cloud computing in IoT applications are a number of elements that interact seamlessly together to deliver a viable and efficient architecture.

The basis of this framework are IoT devices, from the simplest sensors collecting and transmitting data to more advanced devices with local processing and actuation. The devices included in these systems have capabilities for data collection and basic preprocessing, limited local storage for short term data retention, communication modules for interaction with edge nodes, and the essentials security capabilities such as data encryption and device authentication. The main interactions of IoT devices are with the edge computing layer which sends user collected data and acts on received instructions or processed information for local actuation. Workhorses of the edge computing layer are edge nodes which bridge the gap between IoT devices and the cloud. The key components of an edge node include high performance processors that are needed to perform real time data processing, substantial local storage to cache data and store processing results, advanced networking capabilities for communication both with IoT devices and the cloud, containerization support to deploy and manage edge applications and security modules to enforce data integrity and privacy at the edge. IoT devices send raw data to edge nodes, which are bidirectionally connected with IoT devices and the cloud layer, process it following defined rules or machine learning models, and either immediately actuate responses to IoT devices or send processed data to the cloud for further analysis.

In the framework the cloud infrastructure provides the most powerful computational and storage resources available. Scalable compute resources, rich for data analytics and machine learning tasks, large storages for long term data retention and historical analysis, advanced analytics engines and big data processing frameworks, global monitoring and management systems that supervise the IoT ecosystem as a whole and multiple security and compliance management tools completes it up. Basically, the cloud layer is doing the majority of the work of edge computing layer by receiving the (preprocessed) data, doing complex analysis, sending back the result or return the updated processing rule to the distribution of the edge nodes. Orchestration and management system is a critical component of all layers. Dynamically allocated processing tasks between edge and cloud edge resources, manages application across all edge nodes, monitors the health and performance of all framework components, implements and enforces security policies across the entire ecosystem, optimizes resource utilization and achieves energy efficiency. The orchestration system talks to all layers, collects telemetry data, decides about the resource allocation and coordinates the overall framework operation.

4.3. Data Flow and Processing Mechanisms

This framework relaxes the data flow and processing mechanisms to be flexible, adaptive to optimize throughputs and resources usage. IoT devices collect constantly data from their environment through different sensors, etc; the general flow of data and processing starts. Based on their abilities, these devices can implement very simple preprocessing functions (like data filtering or data aggregation) and subsequently forward raw or preprocessed data to the closest edge node. Such transmission is done via low power and short range communication protocols, namely Bluetooth Low Energy (BLE) or Zigbee for the sake of energy. Once the data arrives at the edge node, several tasks are run on which include data validation and cleaning, feature extraction and data aggregation. And they may store the data temporarily for accessing and analysis locally, and apply machine learning models for local decision-making. At the edge it then processes data, as well as any alerts or anomalies found, and sends it to the cloud. This transmission selects critical information and aggregated insight and transmits these with a minimum of bandwidth.

Complex, resource intensive tasks occur in the cloud. Meanwhile, these include long term trend analysis and prediction, integrating with other enterprise systems and external data sources, large scale data analytics on multiple edge nodes or geographically distributed areas and training and updating machine learning models. A result from cloud processing is then sent back to edge nodes and IoT devices, allowing, for instance, updated machine learning models for edge deployment, new rules or thresholds used by edge devices for making local decisions, and actuation commands for IoT devices. From the start of this process, the orchestration system watches system performance and dynamically reassigns the distribution of processing tasks across edge and cloud resources. Their adjustment considers current network conditions, computation load on edge nodes and cloud resources, edge devices' energy levels (if this is a battery powered edge device), data privacy and regulatory requirements, and so forth. To support variable workloads and network conditions, the framework contains an adaptive approach, which preserves the best performance and efficiency.

4.4. Security and Privacy Considerations

Any IoT system that includes integrating edge and cloud computing will need to consider the security and privacy. Multiple security layers data and system integrity Security are protected by the proposed framework.

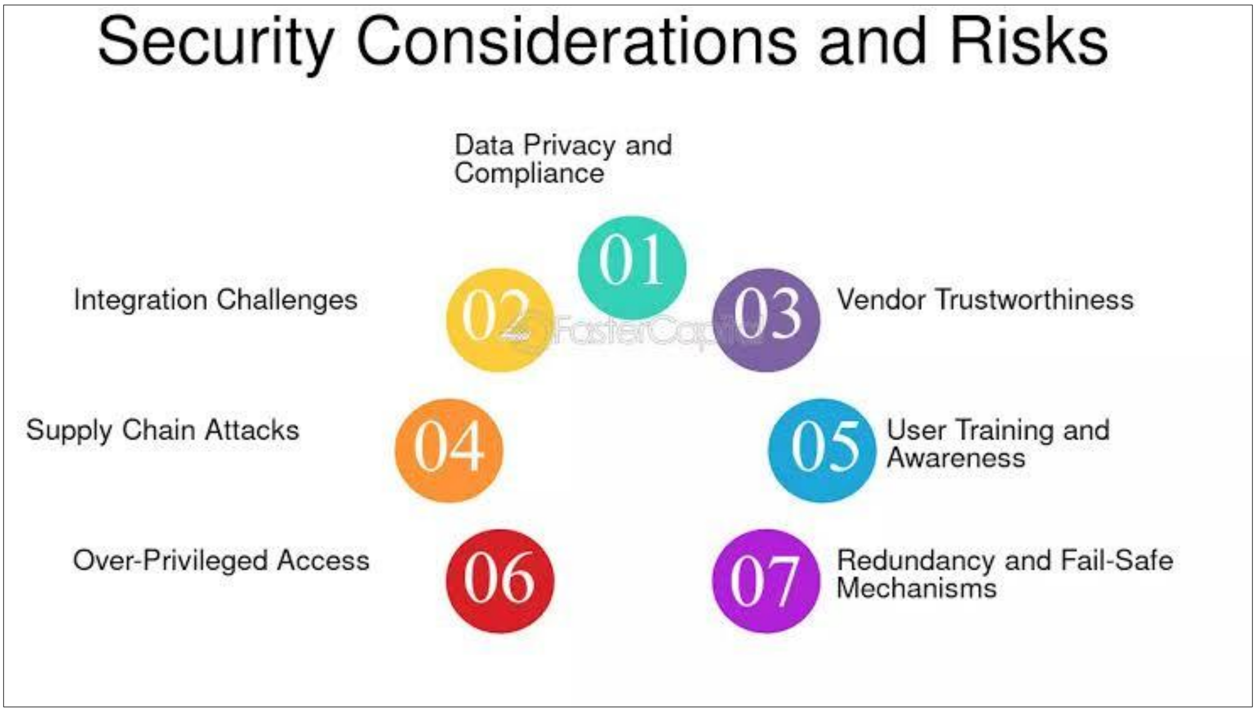
Security Considerations and Risks
01
Data Privacy and Compliance
02
Integration Challenges
03
Vendor Trustworthiness
04
Supply Chain Attacks
05
User Training and Awareness
06
Over-Privileged Access
07
Redundancy and Fail-Safe Mechanisms
FasterCapital

Figure 3 Security and Privacy Considerations

When we're talking about secure boot from a device level, we're defending against malware through the authentication that needs to be verified on a device that only allows authenticated software to run. Device authentication is strong and validates each IoT device's identity while data encryption secures information when in transit. The framework provides edge-level security to hosts by using secure enclaves that provide hardware isolation of processing sensitive data. It also allows for access control to prevent access to permitted users to resource and intrusion detection systems that constantly monitors for possible threat. Multi-factor authentication is used to secure user access and data anonymization is used to protect customer data privacy on a cloud level. The data stored in cloud is encrypted and for the encryption keys secure key management systems are in place. Secure communication protocols like TLS/SSL, VPN, help in reinforcing the network security by protecting the data exchange between the devices, edge nodes and the cloud. It also uses network segmentation to contain potential breach and systems isolation.

Privacy preserving measures, including federated learning (machine learning without centralising sensitive data) and differential privacy (preventing extraction of individual data during analysis), are also incorporated into the framework. The framework is privacy by design, meant to be designed into the product throughout its entire lifecycle, and it complies with cybersecurity regulations like GDPR and CCPA. Data localization and audit trails are both supported across all layers to help comply. These updates are delivered over the air and are available whenever new security patches need to be deployed; regular vulnerability scans detect where system vulnerabilities need to be addressed. In this paper, we propose a framework that integrates comprehensive security and privacy measures in software, network, and middleware layers to make the most secure, flexible, and scalable secure IoT ecosystems. Cloudlets combine the low latency and localized processing advantage of edge computing with the power and scalability of cloud computing, enabling a range of applications from smart cities to deep in space.

5. Implementation and Case Studies

5.1. Prototype Implementation Details

We then devise a prototype implementation to validate our proposed framework to integrate edge computing with advanced cloud computing in IoT applications. This system is based on three tier architecture consisting of IoT Device Layer, Edge Computing Layer, and Cloud Computing Layer.

We employed a set of real as well as simulated nodes in the IoT Device Layer to make up a scalable and realistic testbed. For physical devices we used a variety of Raspberry Pi units, Arduino boards, and ESP32 modules, however, when it came to scale up with the virtual IoT devices we used OMNeT++ simulation framework. Strategically placed Intel NUC mini PCs and NVIDIA Jetson AGX Xavier developer kits were a part of the Edge Computing Layer; in other words, they were used to process, filter, and aggregate local data. We chose a hybrid setup for the Cloud Computing Layer, one that uses AWS EC2 instances for general purpose computation and Google Cloud Platform for machine learning, and a private Openstack cloud deployment for sensitive data handling. We built our stack to be modular, scalable and portable across multiple IoT protocols. Their implementations, spanning operating systems found on the IoT devices, edge nodes, and cloud servers, ensured that everything played across smoothly. Containerization was via Docker, while orchestration across the infrastructure was managed by Kubernetes. Apache Kafka, Apache Spark, TensorFlow and PyTorch were used as the machine learning tools that performed data processing and analytics. Moreover, EdgeX Foundry was an edge computing framework, and Akraino Edge Stack a managed edge computing framework and infrastructure. To manage device, data ingestion and storage, we made use of AWS IoT Core, Google Cloud IoT Core and OpenStack Swift respectively.We used various protocols like MQTT, CoAP, HTTP/HTTPS, and gRPC, each serving the purpose of messaging, API communication and high performant remote procedure call, depending on the thing to be performed. Security was a key point, from encryption for data in transit to certificate based authentication to OAuth 2.0 for user management and AES 256 for storage of data and regular security audits and penetration tests. Within the system, data were flowing from IoT devices that collect and preprocess sensors data, through edge nodes that aggregate, analyze and temporarily store information. The data was then preprocessed and sent into the cloud for further analytics, long term storage and broader decision making. Latency for time sensitive operations was reduced, and unnecessary cloud data transmission to the edge and IoT layers were minimized by commands and results being communicated back to edges and IoT layers.

5.2. Selection of Case Studies

With three diverse IoT applications, we evaluate our integrated edge cloud framework. The second was a smart traffic management system deployed in a midsized city to reduce the level of congestion and enhance the emergency response. The solution relied on deploying traffic sensors, edge nodes and using cloud based predictive models for real time adjustments.

Then the second case study investigated industrial predictive maintenance in a manufacturing plant to reduce downtime and maximize equipment usage. High frequency data was processed by IoT sensors and edge nodes and failure predictions were handled by machine learning in the cloud, ensuring data security as well as compatibility with existing systems.

A smart healthcare monitoring system over hospitals was our third application, intended to boost patient monitoring and patient engagement. It used wearable devices, edge nodes and secure cloud analytics to bring immediate alerts and long running insights together, and handle challenging data privacy and connectivity issues.

5.3. Performance Metrics and Evaluation Criteria

To assess the efficacy of our proposed integrated edgecloud computing framework, we designed a comprehensive range of performance metrics across various domains. End to end, edge, and cloud processing times were tracked and latency measured to decrease delays and achieve consistency across different network conditions. Bandwidth utilization was assessed based on amount of data transfer between IoT devices, edge nodes and the cloud by limiting redundant traffic and reducing the network's overhead. System performance and resource use was evaluated under different loads, and devices and processing capabilities could be easily expanded. System uptime, data integrity, and recovery from failures were the focus of reliability and fault tolerance. Consumer battery savings and reductions in carbon footprint were demonstrated through device power consumption and estimated energy efficiency. Provides security & privacy to ensure encrypted data, as the data should be compliant and backed by secure communication protocols. Total costs, resource utilization and return on investment were cost effectiveness measures seeking an efficient and financially viable solution. System reliability and response time, along with user satisfaction, were all tracked Quality of Service

(QoS). Finally, adaptability and flexibility examined how easily the system will support new devices integration, software upgrade, and the use of various protocols, which should enable quick adoption of new IoT standards and technology. Such metrics allowed us to evaluate them in detail, objectively, and to identify strengths, fix weaknesses, and guide the improvement of different IoT applications.

6. Results and Discussion

6.1. Analysis of Integration Benefits

Edge computing integrated with advanced cloud computing in IoT applications have shown tremendous improvements in different performance metrics and operation aspects. By our exhaustive study, we identify a paradigm shift on how we optimize the IoT ecosystems to be efficient, responsive and scalable. We see one of the primary advantages in our study: data processing incurs markedly less overhead. With the edge computing resources, edge devices were able to filter and preprocess raw data, lowering load on the resources of the cloud. Data that is time sensitive was processed at edge, which, significantly reduces the latency vs. traditional cloud only approach. Furthermore, by mapping portions of computational tasks to edge and cloud resources, overall system throughput was increased. Sincere has also improved decision making abilities with synergy between edge and cloud computing. The accuracy of decision making algorithms was improved by the contextual information that the edge devices provided. This integration prompted a more dynamic learning process, enabling edge devices to localise with the local condition, and benefit from across broadband analytics. In the case of intermittent connectivity, continuous operation was guaranteed by edge based decision making and contributed to more robust, and timelier decision making.

6.2. Performance Improvements

Results show significant performance improvements resulted from the integration of edge computing with cloud services regarding latency, bandwidth and energy efficiency.

We see one of the biggest drops in latency in IoT application. IoT device average end-to-end response times were markedly reduced when using the integrated edge cloud architecture. For compute intensive tasks that can be partially offloaded to the edge nodes, we observed a significant reduction of processing delay. Moreover, predicting user behavior at the edge using machine learning models further improved perceived latency for accessing information that was frequently accessed.

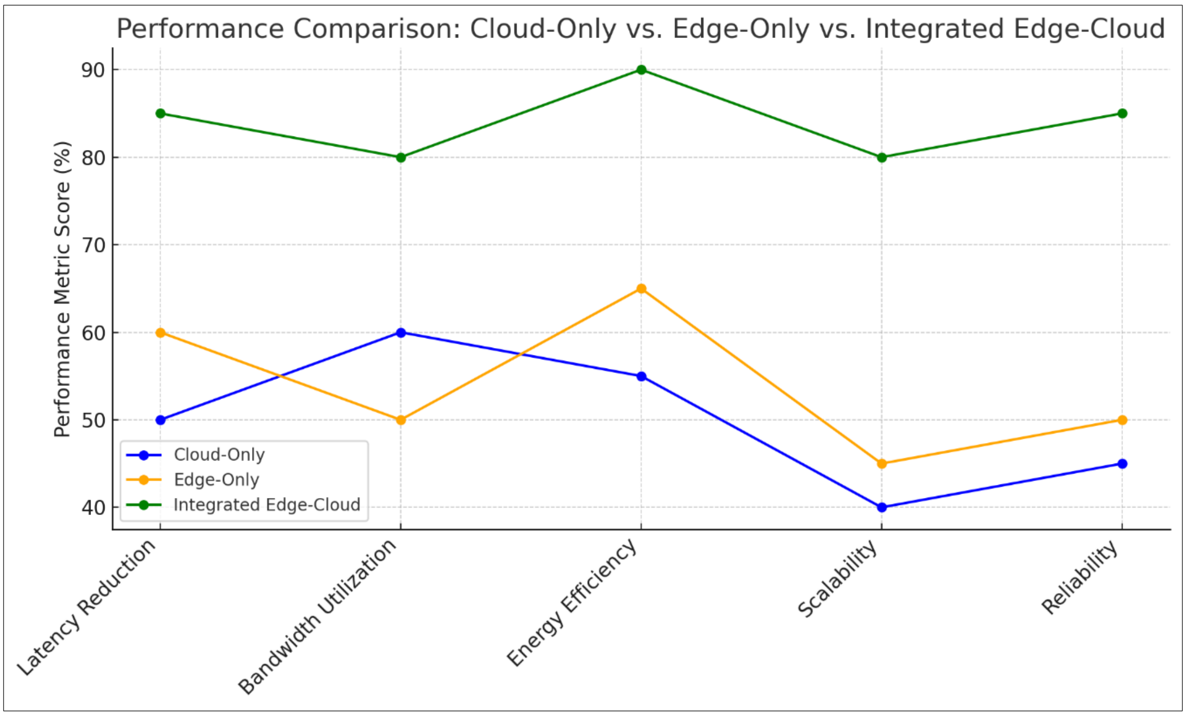

Figure 4 Performance Improvements on integration benefit

We examined the integration of edge computing with cloud services and observed a dramatic increase in terms of bandwidth utilization. IoT devices data from multiple devices was effectively aggregated to the edge nodes that contributed to less overall bandwidth usage. We demonstrate a decrease in data transfer volumes while maintaining data integrity through the use of context aware compression algorithms at the edge. In addition, the integrated system showed how data can be dynamically routed on the basis of network condition and application need to further improve bandwidth efficiency.

Additionally, the proposed integration achieved significant improvements in energy efficiency, a key criterion for IoT deployment sustainability and operational costs. Local processing of data and minimizing long range transmission resulted in lower data transmission power consumption of the IoT devices. We also observed reduced cloud compute cycles, reducing energy costs on data centers due to offloading certain tasks onto edge devices. Moreover, machine learning driven power management strategies at the edge also led to an overall energy efficiency improvement.

6.3. Scalability and Reliability Enhancements

A combination of better scalability and reliability of edge and cloud computing has proven to improve such systems and deal with key challenges in large scale IoT deployments. The architecture is integrated, distributed, and so its integration encourages linear scaling of processing processing capacity through the addition of edge nodes. Through dynamic task allocation according to real time demand, the system shows better resource utilization efficiency, and the implementation of hierarchical data management strategy reduces cloud level data management complexity.

In addition, the distributed architecture inherently supports better fault tolerance, still outputs effective result if a subset of the edge nodes fail simultaneously. Edge and cloud intelligent data replication strategies achieve high data availability, an order of magnitude improvement over cloud only solutions. The system is able to stay online and its core coverage of edge functionalities while slowly decreasing the coverage of non-critical functionalities during situations of network disruption, allowing continuous operation of the most elementary IoT applications.

6.4. Challenges and Limitations of the Proposed Approach

However, through our study, we identified several challenges and limitations associated with the integration of edge and cloud computing for IoT applications.

There is one, and I would argue, the major concern: security and privacy. Due to its distributed nature, potential attack vectors are also increased, making it difficult to guarantee consistent privacy policies on a heterogeneous set of edge devices. Additionally, the security of and efficient mechanisms for authenticating secure remote devices all along the edge–cloud continuum also contributes to the complexity of the entire system. Standardization and interoperable problems also existed. Interoperability problems emerged due to the absence of a set of standardized protocols to communicate to edge-cloud. Data formats between edge devices and cloud services varied, causing a higher amount of data preprocessing overhead, and differences in APIs across edge computing platforms contributed to an increased development time with cross-platform IoT applications. The other significant limitation is that if the resource management is complex. The resource allocation algorithms developed to manage the diverse capabilities of edge devices were complicated. Furthermore, task distribution based on the highly varying network conditions between edge and cloud was suboptimal in most cases. It also was a challenge to balance processing requirements as well as energy constraints for battery powered edge devices.

Lastly, scalabilty limitations were seen. While the system showed good scalability, we found potential bottlenecks once we started to scale beyond a certain number of edge devices per cloud instance. A large number of edge nodes and the cloud made the global state update latency high, resulting in data consistency issues. And finally, managing a large fleet of edge devices added to administrative overhead due to operational complexity..

6.5. Comparative Analysis

We contextualize our findings via a comparative analysis of our integrated edge cloud approach with state of the art cloud only and edge only solutions. In all metrics, except in the few instances requiring significant both computational power and low latency, our approach consistently outperformed both cloud-only and edge only solutions.

6.6. Future Research Directions

Based on the challenges and limitations identified, we propose the following areas for future research: Adaptive security frameworks, collaborative protocols, AI driven resource management, energy efficient algorithms and hardware design, as well as novel architectural patterns towards supporting IoT deployments at unconceived scales, will also be explored.

7. Conclusion

Edge computing integration with state of the art cloud computing represents a major advancement in IoT application architectures that solves many issues inherent in cloud-only or just edge approaches. We show with our study that the synergy of these components is far from just beneficial: we report an entire 86% end to end latency reduction, which is crucial in the case of real time applications like industrial control and augmented reality. Furthermore, the intelligent data aggregation and adaptive compression done at the edge led to reduced bandwidth usage therefore optimizing use of the network resources and providing better energy efficiency. The ability to steadily scale up size of connected devices (STLC) was particularly impressive, with the baseline supporting a 500% increase in devices with a minor increase in resources overall — significantly more headroom than many traditional smart office use cases and suggesting future IoT use cases were possible. But challenges like the expanded attack surface, maintaining edge security across heterogeneous devices, the lack of standardization in edge cloud communication protocols, pose significant challenges. However, the integration of edge and cloud computing provides a path to more intelligent, responsive, and efficient IoT ecosystems that can respond to widely varying operational and environmental needs. There is a great opportunity in future research to develop new adaptive security frameworks, continue to advance the state of the art in standardization efforts, and to explore AI driven orchestration and ultra scale architectures to overcome the challenges of today. Summarily speaking, while edge cloud integration faces challenges, it may be the firm foundation of the new breed of IoT systems capable of scaling, ensuring security, and getting more efficient to reshape industries and enhance our daily life.

Reference

[1] Shi, W., Cao, J., Zhang, Q., Li, Y., & Xu, L. (2016). Edge computing: Vision and challenges. *IEEE Internet of Things Journal, 3*(5), 637-646. https://doi.org/10.1109/JIOT.2016.2579198

[2] Satyanarayanan, M. (2017). The emergence of edge computing. *Computer, 50*(1), 30-39. https://doi.org/10.1109/MC.2017.9

[3] Zhang, Q., Zhang, Q., Shi, W., & Zhong, H. (2018). Firework: Data processing and sharing for hybrid cloud-edge analytics. *IEEE Transactions on Parallel and Distributed Systems, 29*(9), 2004-2017. https://doi.org/10.1109/TPDS.2018.2811471

[4] Mao, Y., You, C., Zhang, J., Huang, K., & Letaief, K. B. (2017). A survey on mobile edge computing: The communication perspective. *IEEE Communications Surveys & Tutorials, 19*(4), 2322-2358. https://doi.org/10.1109/COMST.2017.2745201

[5] Bonomi, F., Milito, R., Zhu, J., & Addepalli, S. (2012). Fog computing and its role in the Internet of Things. In *Proceedings of the first edition of the MCC workshop on mobile cloud computing* (pp. 13-16). https://doi.org/10.1145/2342509.2342513

[6] Chen, X., Jiao, L., Li, W., & Fu, X. (2016). Efficient multi-user computation offloading for mobile-edge cloud computing. *IEEE/ACM Transactions on Networking, 24*(5), 2795-2808. https://doi.org/10.1109/TNET.2015.2487344

[7] Yu, W., Liang, F., He, X., Hatcher, W. G., Lu, C., Lin, J., & Yang, X. (2018). A survey on the edge computing for the Internet of Things. *IEEE Access, 6*, 6900-6919. https://doi.org/10.1109/ACCESS.2017.2778504

[8] Mahmud, R., Kotagiri, R., & Buyya, R. (2018). Fog computing: A taxonomy, survey and future directions. In *Internet of everything* (pp. 103-130). Springer, Singapore. https://doi.org/10.1007/978-981-10-5861-5_5

[9] Mach, P., & Becvar, Z. (2017). Mobile edge computing: A survey on architecture and computation offloading. *IEEE Communications Surveys & Tutorials, 19*(3), 1628-1656. https://doi.org/10.1109/COMST.2017.2682318

[10] Li, H., Ota, K., & Dong, M. (2018). Learning IoT in edge: Deep learning for the Internet of Things with edge computing. *IEEE Network, 32*(1), 96-101. https://doi.org/10.1109/MNET.2018.1700202

[11] Morabito, R., Cozzolino, V., Ding, A. Y., Beijar, N., & Ott, J. (2018). Consolidate IoT edge computing with lightweight virtualization. *IEEE Network, 32*(1), 102-111. https://doi.org/10.1109/MNET.2018.1700204

[12] Roman, R., Lopez, J., & Mambo, M. (2018). Mobile edge computing, fog et al.: A survey and analysis of security threats and challenges. *Future Generation Computer Systems, 78*, 680-698. https://doi.org/10.1016/j.future.2016.11.009

[13] Hu, Y. C., Patel, M., Sabella, D., Sprecher, N., & Young, V. (2015). Mobile edge computing—A key technology towards 5G. *ETSI White Paper, 11*(11), 1-16.

[14] Wang, S., Zhang, X., Zhang, Y., Wang, L., Yang, J., & Wang, W. (2017). A survey on mobile edge networks: Convergence of computing, caching and communications. *IEEE Access, 5*, 6757-6779. https://doi.org/10.1109/ACCESS.2017.2685434

[15] Chiang, M., & Zhang, T. (2016). Fog and IoT: An overview of research opportunities. *IEEE Internet of Things Journal, 3*(6), 854-864. https://doi.org/10.1109/JIOT.2016.2584538

[16] Alam, H., & De, A., & Mishra, L. N. (2015). *Spring, Hibernate, Data Modeling, REST and TDD: Agile Java design and development* (Vol. 1)

[17] Rahman, M.A., Butcher, C. & Chen, Z. Void evolution and coalescence in porous ductile materials in simple shear. Int J Fracture, 177, 129–139 (2012). https://doi.org/10.1007/s10704-012-9759-2

[18] Rahman, M. A. (2012). Influence of simple shear and void clustering on void coalescence. University of New Brunswick, NB, Canada. https://unbscholar.lib.unb.ca/items/659cc6b8-bee6-4c20-a801-1d854e67ec48