JAHIDUL HOQUE

APPROXIMATE COMPUTING IN COMMUNICATION SYSTEMS AND NETWORKS

Master's Thesis
Information Technology and
Communication Science (ITC)
Jari Nurmi
Hans Damsgaard
May 2024

ABSTRACT

Jahidul Hoque: APPROXIMATE COMPUTING IN COMMUNICATION
SYSTEMS AND NETWORKS
Master’s thesis
Tampere University
Master’s Degree Programme in Information Technology

May 2024

Approximate Computing (AxC) is a significant technological development that has been demonstrated to be an effective way to increase the performance of computer systems. In recent years, there has been an exponential surge in revolutionary research across both industry and academia, igniting innovation and propelling progress in diverse fields. In this study, several approximate commuting strategies, designs, and implementations in communication systems are discussed. Additionally, this thesis also outlines the challenges and obstacles that might arise when AxC is used in communication systems in the future. The analysis also investigates the way AxC impacts communication networks and systems' availability and stability. An overview of AxC techniques is provided, highlighting key methodologies and their respective impacts on network performance. Through comprehensive analysis, this thesis aims to elucidate the integral role of AxC in advancing communication systems and networks. Additionally, it seeks to propose contemporary applications of AxC techniques within these systems.

Keywords: Communication Networks, AxC, Communication Systems, AxC techniques.

THE AI TOOL USED IN MY THESIS AND THE PURPOSE OF THEIR USE HAVE BEEN DESCRIBED BELOW:

ChatGPT (3.5)

Purpose of use and the part in which it was used:

Language revision and to produce structurally fluent text for case study and examples chapter.

SciSpace

Purpose of use and the part in which it was used:

In background Theory writing by extracting information within the context of communication system overview.

I am aware that I am totally responsible for the entire content of the thesis, including the parts generated by AI, and accept the responsibility for any violations of the ethical standards of publications.

CONTENTS

Table of Contents

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

ADAS	Automated Driving Assistance Systems
ASICs	Application-Specific Integrated Circuits
AxC	Approximate computing
CPU	Central Processing Unit
DBMS	Database Management System
DCT	Discrete Cosine Transform
DSP	Digital Signal Processing
E/E	Electrical and Electronic
FEC	Forward Error Correction
FeRAM	Ferroelectric random-access memory
FFT	Fast Fourier Transform
FPGAs.	Field-Programmable Gate Arrays
GP-GPUs.	General-Purpose Graphics Processing Units.
GPU	Graphics Processing Unit
ICN	Information-centric networking
IoT	Internet Of Things
IP	Internet Protocol
ISP	Inter System Protocol
LNS	Logarithmic Number Systems

MLC	Multi-level cell
MRAM	Magnetic random-access memory,
PCM	Phase-change memory
QoI	Quality-of-Information
QoS	Quality of Service
RAM	Random Access Memory
RRAM	Resistive random-access memory
SNN	Semantic neural network
SNN	Spiking Neural Network
SNR	Signal to Noise Ratio
SVM	Support Vector Machines
WSNs	Wireless Sensor Networks

1.INTRODUCTION

AxC is being advocated as a viable choice for high-performance computing by alleviating the accuracy constraints typically imposed on applications. This trend is driven by the emergence of data-intensive disciplines such as image and video processing, machine learning, and big data analytics, where marginally flawed results are deemed acceptable within a specified variation.(Ling Wang, 2022). Communication systems and networks have successfully used AxC techniques (Raha & Raghunathan, 2019). These methods seek to loosen exactness restrictions while keeping acceptable results quality in order to increase computational efficiency (Raha & Raghunathan, 2019). A hybrid systems approach has been developed to simulate traffic flow in communication networks, employing averaging to approximate discrete variables like the congestion window and queue size (Yellu et al., 2019). This approach allows flexibility in simulating different congestion control systems and queueing policies while capturing the dynamics of transitory occurrences (Sen et al., 2017). Overall, by sacrificing precision for computational efficiency, AxC offers a viable strategy for increasing the effectiveness of communication systems and networks.

Since its inception, information technology has developed into a field that is now vital to modern society. A huge volume of information is always in motion and needs to be processed and stored. It is possible that the necessary tools to manage that amount of data, are not readily available, which could pose problems.

The idea of "Big Data" has grown during the past ten years. It describes a set of data or sets of data whose size, complexity, and rate of increase make it impossible for them to be effectively managed, processed, or analyzed with the use of customary technology and techniques, such as relational databases and conventional statistics (PowerData, 2020). One compelling rationale for the adoption of AxC lies in its capacity to judiciously accommodate a degree of imprecision in data processing. This strategic acceptance of occasional inaccuracies empowers the adjustment of computational dimensions such as size, complexity, and speed to align with the available resources and capabilities.

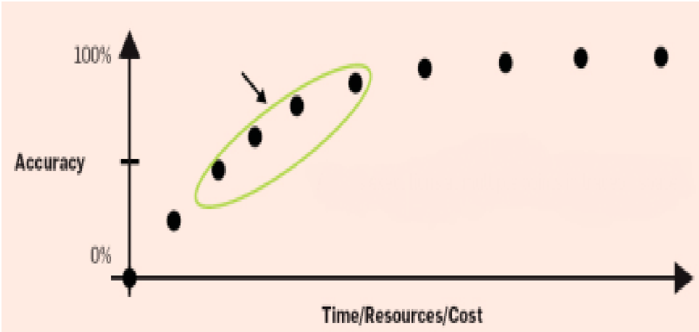


Fig. 1.1: Plot of the AxC accuracy (Kugler, 2015)

Figure 1.1 shows the trade-off space of quality vs computational cost/resource utilization/execution time. Reduced quality constraints lead to reductions in the other metrics.

To reduce compute costs and power consumption, approximation computing aims to accommodate minor errors or inaccuracies in calculations. AxC approaches can have a big impact on speed, power consumption, and cost in communication systems and networks where enormous volumes of data are processed and transferred. This strategy may make it possible to construct high-performance communication networks and systems that can handle the rising needs of contemporary applications and services. Understanding AxC's guiding principles and investigating its potential applications in communication system and network design is crucial in this situation.

1.2 Objectives of the study

The objective of this thesis is to review comprehensively the following:

- The design and implementation methodologies pertaining to AxC within communication systems and networks.
- The challenges and limitations inherent in the adoption of AxC strategies within communication systems and networks.
- The design optimization strategies employed in communication systems and networks leveraging AxC techniques.

- The impact of approximation on the responsiveness and reliability of communication networks and systems.

1.3 Thesis Organization

Chapter 2 delves into related work on AxC within communication systems. Following this, Chapter 3 provides an overview of communication systems. Chapter 4 presents a comprehensive overview of AxC techniques. In Chapter 5, we explore the applications of AxC in communication systems. Chapter 6 examines both the advantages and challenges associated with AxC. Chapter 7 offers case studies and examples to illustrate the practical implementation of AxC. Chapter 8 outlines future directions and research challenges in the field. Finally, Chapter 9 concludes the thesis by summarizing key findings and insights gleaned from the preceding chapters.

2. AxC FOR NETWORKS AND SYSTEMS FOR COMMUNICATION

AxC enhances memory and processing by allowing systems to tolerate errors and provide relaxed fault tolerance, thereby improving performance and durability. Key applications include optimizing memory storage with techniques like multi-level cells and approximate data copies, addressing high raw bit error rates in flash memory. Non-volatile memories such as PCM, RRAM, MRAM, and FeRAM benefit from application-level error tolerance, mitigating their limitations. DRAM also leverages AxC for better performance and energy efficiency. Overall, AxC reduces energy usage and costs while boosting efficiency and capacity, making it essential for advanced networks and communication systems. This chapter explores AxC's impact on these areas, highlighting its benefits and trade-offs.

AxC seeks to more closely match the precision of system abstractions with the demands of approximate applications (Sampson, 2015). The main challenge in AxC is developing abstractions that foresee and manage uncertainty without sacrificing performance advantages. Designing hardware and software with approximation in mind is the goal.

This research combines insights from numerous disciplines, including hardware engineering, architecture, system design, programming languages, etc. Among them are:

- **Studies on tolerance:** This category demonstrates how various components of an application can affect fidelity and reliability in various ways. Certain software elements, especially those engaged in control flow, must be protected against all of approximation's impacts.
- **Put architecture resilience to use:** Techniques for hardware approximation can lead to gains in verification complexity, manufacturing yield, or energy

efficiency. Computational units, memory, and the complete system architecture are the different categories for hardware-based approximation methodologies.

- **AxC for increasing the durability of memory:** AxC can help prolong the system's lifetime by tolerating errors in memory cells, thereby allowing the system to continue functioning despite deteriorating storage cells. However, it does not improve the actual lifespan of the memory cells themselves. Additionally, memory like flash can benefit from its probabilistic characteristics while being unnoticed by software. These memory approximation techniques often work by exposing soft errors and other comparable phenomena.
- **Relaxed fault tolerance:** Several circuit design techniques can be used to reduce the cost of redundancy by providing redundancy specifically for specific CPU instructions, DSP blocks, or GPU components. After carefully selecting the information, another use is to allot resources for software-level error detection and correction.

2.1 AxC in memory storage and optimization techniques:

In several academic disciplines, system approximations are a typical methodology. suggest building stochastic gradients and carrying out gradient search in decision spaces using approximations with increasing precision in optimization techniques. Boche and Pohl investigate the use of basis expansions to approximate stable causal transfer functions, emphasizing the shortcomings of the Fourier basis and the pursuit of more effective computing bases (Boche & Pohl, 2018). The idea of metric-based adaptation is expanded by Van Langenhove et al. to incorporate stochastic components, enabling the optimization of error control in both deterministic and stochastic approximation spaces (Van Langenhove et al., 2018). Hajek offers a thorough system of approximate solutions for frequency filters, highlighting the significance of selecting the best approximation type in accordance with real-world requirements (Hajek, 2018)

Most of the research on approximate system architectures is computational. Error tolerance in transient and persistent data is present in a wide range of application areas, from server software to mobile applications.

Performance, energy, space, and complexity are all key expenses associated with memories. It is because they must always guarantee absolute data integrity.

The following techniques are offered to boost performance, energy, and capacity while providing approximate storage:

1. Use multi-level cells to boost performance or density at the risk of occasionally inaccurate data retrieval.
2. Keep approximate copies of the data in blocks that have missing bits. Correction of higher-order bits is prioritized based on total value precision to lessen the impact of failed bits.

Transient data saved in main memory as well as the storage of files and databases are all subject to approximate storage.

The issue of increased errors in flash memory due to high raw bit error rates (RBER) has been addressed using interfaces for approximate data storage (Anžel et al., 2021). In order to increase the dependability of regular data while keeping approximate data unsecured, these interfaces take advantage of the abundance of approximate data that is readily available in flash storage and use a novel data organization strategy (Li et al., 2019). To enhance read performance, new data allocation algorithms and updated garbage collection schemes are also used. Compared to existing methods, the proposed approach has demonstrated an average 30% improvement in read performance.

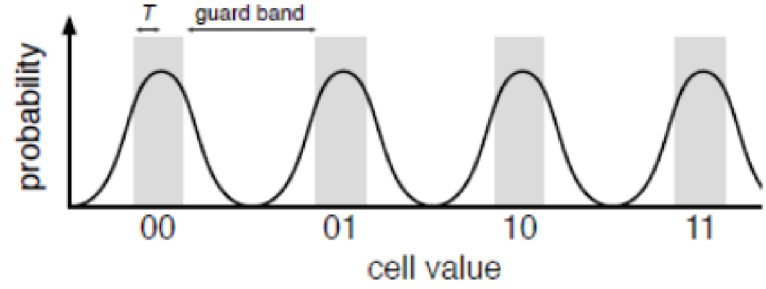
Technologies for non-volatile memory, such as Phase-change memory PCM, RRAM, MRAM, and FeRAM, are well-suited for storing approximate data. By combining these technologies with application-level error tolerance, we can overcome some of their disadvantages, such as low density and brief device lifetime. Additionally, DRAM is also a key area of research in AxC. Techniques such as refresh rate scaling are employed to trade off data integrity for improvements in performance and energy efficiency. By reducing the frequency of refresh operations, DRAM can save power and increase speed, albeit at the cost of occasionally losing data integrity. An application employs conventional exact storage when it requires strict data fidelity, ensuring a low error rate during data recovery. Conversely, the software uses the memory's approximation mode when it can tolerate sporadic faults in some data, accepting that data recovery problems may occur with a non-zero frequency.

2.1.1 *Main Memory in approximation*

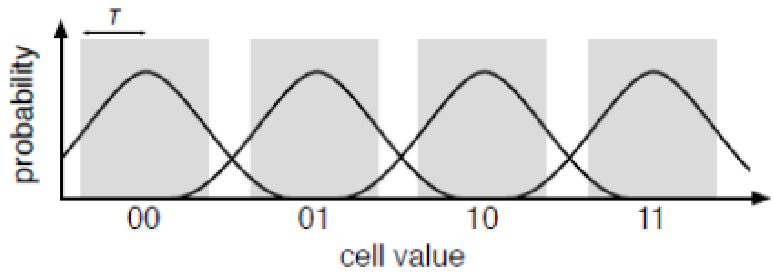
In the scientific community, main memory approximation techniques are being investigated. Through approaches that accommodate read/write errors, energy usage is reduced. Significant energy savings can be gained by adopting approximation memories, especially when taking into account applications with adaptable quality criteria. With up to 30% energy savings at a 10E-8 error rate and a 20% quality deterioration allowed, several applications have demonstrated a favorable energy-quality tradeoff (Teimoori et al., 2018). Additionally, power consumption can be further optimized without sacrificing application quality criteria by coordinating customizable approximation knobs across all layers of the memory structure. Up to 37% energy reductions in the memory subsystem have been reported using self-optimizing runtime managers, such as AXES, to continuously update and optimize approximation management strategies throughout runtime (Maity et al., 2020).

PCM, and other fast resistive storage technologies can be used as approximate main memory. In this context, approximate persistent storage refers to file systems, DBMSs, or flat address spaces. Large-scale image or video search databases require a significant amount of quick permanent storage. When minor data imperfections are tolerable, approximation storage can reduce expenses by boosting each storage module's performance and energy efficiency while increasing its lifespan and capacity. The interface to approximate memory consists of read and write operations with a precision flag. In the context of main memory, these operations are referred to as load and store instructions. For persistent storage, these are read and write requests that are processed block-by-block. The memory interface specifies the level of controlled approximation.

In PCM and other solid-state memories, an analog value is quantized and stored to expose digital storage. In MLC configurations, each cell can store multiple bits. For exact storage in MLC memory, there is a tradeoff between access cost and density: accessing several levels per cell consumes more time and energy. Guardbands are safety margins that make sure a system can continue to function properly even when there is uncertainty. Guardbands are employed in the context of memories to take temperature, voltage, and other variables into consideration that may have an impact on the dependability of memory cells (Leng et al., 2015). Guardbands are frequently lowered while estimating memories to enhance performance or save energy. But there is a chance that this will make mistakes more likely. Thus, while constructing approximate memory, it is crucial to carefully weigh the trade-offs between performance, energy efficiency, and dependability.



(a) Exact MLC **Source:** Sampson, 2015



(b) MLC approximation

Fig. 2.1: Exact MLC & MLC approximation **Source:** Sampson, 2015

Figure 2.1 shows the analog value range of an exact and approximate four-level cell. The shaded areas indicate the voltage levels targeted when writing to each cell. The curves represent the probability of reading a specific analog value after writing a level.

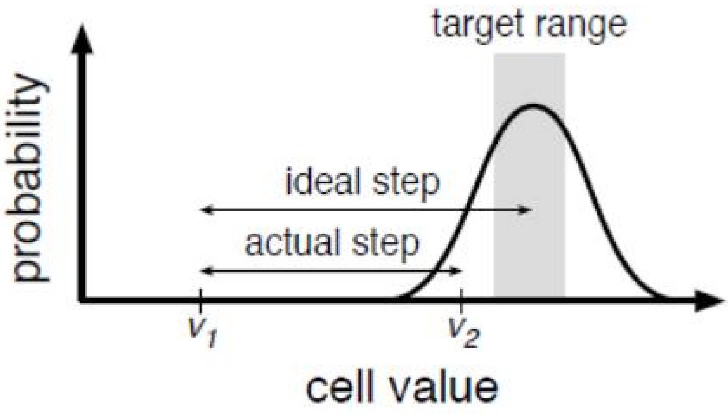


Fig. 2.2: Iterative program-and-verify writing in one step **Source:** Sampson, 2015

Figure 2.2 shows the value advancing from the starting point v1. The probability distribution shown by the curve indicates where the final value v2 might be located. If v2 is not within the target voltage range, further action is necessary.

To enhance performance and energy economy, an approximate MLC design loosens the rigorous precision limitations on iterative MLC writes.

3.COMMUNICATION SYSTEM OVERVIEW

3.1 Communication Systems

A communication system is a way to send and receive data over a long distance. In order to facilitate efficient communication, it entails the use of a variety of technologies and strategies. From telegraphy in 1837 to the present-day 5G and 6G systems, communication technologies have advanced over time (Goswami et al., 2023). These systems have improved in terms of data throughput, speed, and security because to features like encryption and very low air latency (Bentley et. al 2018). The components of communication systems include network interfaces, transmitters, and receivers. (Tamizhelakkiya et al., 2021). Both digital and analog signals can be processed by them, with digital systems having advantages including increased precision and efficiency. Furthermore, sound sources, audio outputs, and controllers may be included in communication systems to support multimedia communication. The ability to communicate effectively and reliably between various devices and networks is made possible in large part by communication systems.

Information exchange is the process of communication. A communication system's fundamental parts are its source, transmitter, channel, receiver, and destination (the person receiving the information). The schematic of an electronic communication system is shown in the image below.

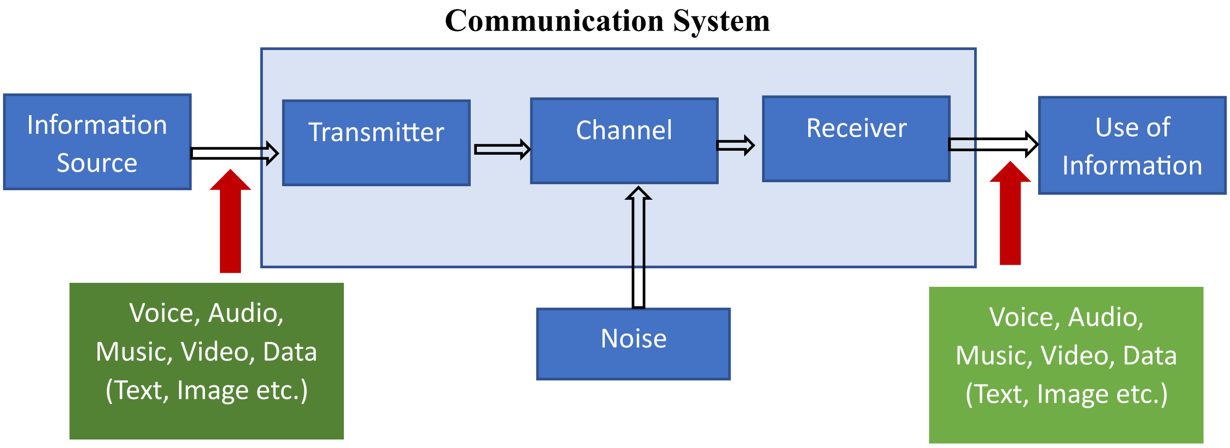


Fig 3.1: Block diagram of a generalized communication system

The following is a description of a communication system:

- The person or machine who creates the message to be conveyed by the sender (information source)
- A vehicle for conveying the message
- The recipient of the message (the information user)

3.2 Source of the Data

A source functions as a device designed to originate information or messages intended for transmission, pivotal in communication networks. Common examples include computer keyboards and microphones, capable of generating both analog and digital messages. Analog information sources encompass entities such as TV signals, cameras, and microphones, while digital information sources include teletypes and computer outputs, characterized by discrete symbols or letters. Cellphones also serve as intuitive examples, representing digital information sources in the context of communication networks.

3.2.1 Transmitter

The primary function of transmitters is to convert input messages from sources such as voice signals into electrical forms suitable for transmission. For example, a microphone acts as a transducer, converting sound waves into electrical audio signals, while a camera converts light data into a TV video signal. Similarly, a speaker functions as a transducer by converting electrical signals back into audible voice signals at the receiving end.

A key function of the transmitter is modulation, which combines a message signal with a high-frequency carrier signal. This process enhances the efficiency of signal transmission and ensures optimal conveyance of information across the communication channel. Modulation also preserves signal integrity and minimizes interference, facilitating coherent reception at the receiving end.

3.2.2 Channel

In a communication system, the transmitter and receiver are each located in different places and are connected by a communication channel. This channel can take various forms, such as wires, fiber optics, radio waves, or satellite links, through which the message signals are transmitted. The channel serves as the medium that physically carries the signals from the transmitter to the receiver.

If one were to call someone in the USA, the call might travel through the air, the space, etc. Thus, the channel in this instance consists of a number of wires, radio connections, satellites, and undersea cables. SNR and Bandwidth are the channel's crucial metrics. SNR is the measure of signal power to noise power.

3.2.3 Noise

Noise disrupts, interferes with, and has an impact on the sent signal, which is an unwanted signal. We are unable to stop it, but we can decrease its effects. It is impossible to forecast the behavior of noise because it is an unpredictable signal. A typical unit of measurement for noise is the SNR. Decibel (dB) units are used to express SNR values. It is imperative to acknowledge that achieving a high SNR is paramount for optimizing performance.

3.2.4 Receiver

Receiver’s primary duty is reconstructing the transmitted signal and sending it to the location known as the use of information. It receives the message being transmitted over the channel. Amplifiers, oscillators, mixers, tuned circuits, filters, and a demodulator are all components of a receiver. Demodulation is the removal of the carrier from the transmitted signal (Jiménez, 2022). Decoding, decompressing, error detection, and demodulation are the four categories under which the receiver's function falls. A receiver's output could be an audio signal, a visual signal, or computer data.

3.2.5 Transceiver

Digital communications are typically two-way. As a result, both end users need to be able to send and receive messages. As a result, the transmitter and receiver are typically combined into a single unit in communication equipment. All of the transmitter and receiver circuitry is contained in a single circuit called a transceiver. Devices like phones, fax machines, cell phones, and computer modem are examples of transceivers. (Krishna, n.d.)

3.3 Types of communication protocols

Communication protocols are essential components of systems architecture, governing the interaction between different modules. More than just individual modules adhering to protocols, architecture often involves the specification of these protocols, which dictate the rules of interaction. These protocols establish a set of guidelines that enable electronic devices to communicate and share data effectively, defining the formats and rules for digital messages. They are indispensable for both digital and analog communications.

Communication protocols are of two types. They are:

I. Inter System Protocols
II. Intra System Protocols

3.3.1 Inter System Protocol

ISP functions as a sophisticated communication protocol facilitating seamless data exchange between disparate electronic devices. Specifically, it orchestrates the interaction between intricate systems, exemplified by the communication interface between a computer and a microcontroller circuit.

Fig 3.2: Inter System Protocol

3.3.2 Intra System Protocol

The Intra-System Protocol makes it easier for components on the same or separate boards to communicate, ultimately reducing system design effort and possibly driving down manufacturing costs as well as power consumption. This simplifies the overall circuit generated by controlling communication paths so that it is possible to carry out regular data exchange. It also enforces more data access control to the system, boosting security and creating a stricter delineation within the architecture. This scalability is important in the case of hardware level protocols as these the communication back-and-forth between various components need to be done in an efficient and synchronized way, something that improves overall system performance and reliability.

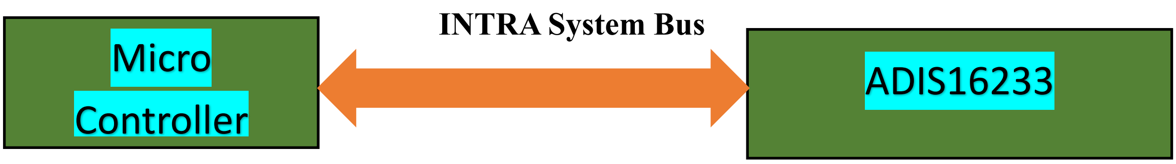


Fig 3.3: Intra System Protocol

3.4 Communication Architectures

In a number of fields, including as communication networks, (ADAS), (ICN), (E/E) designs, and fiber positioner systems, communication structures are essential. Graph neural networks (GNNs) have demonstrated superior convergence rates and enhanced generalization capabilities within communication networks compared to fully connected multi-layer perceptrons (MLPs) (Shen et al., 2022). Different communication architectures are examined for their usefulness in delivering alert signals to drivers by ADAS, which uses communication infrastructures to warn users of potentially hazardous circumstances (Giuliano et al., 2021). Since 2009, many ICN architectures have been proposed, with discussions of the functionality and characteristics of architectures like TRIAD, DONA, PURSUIT, and NetInf (Dutta et al., 2021). In order to efficiently distribute functionality, vehicular E/E designs are evaluated based on communication patterns (Lisova et al., 2022).

3.5 Signal processing and compression techniques in communication systems

An enabling technology, signal processing includes the fundamental theory, uses, algorithms, and applications of information processing (Roohum Jegan, 2023).

By facilitating effective digital signal transmission, storage, and analysis, signal processing and compression techniques play a significant role in communication systems. These methods aid in streamlining bandwidth usage, lowering storage needs, boosting signal quality, and enabling real-time processing in a variety of communication applications (Zhang et al., 2019). Signal processing techniques are employed in communication systems for a number of functions, including filtering, modulation, demodulation, equalization, synchronization, and error correction. These methods make sure that the received signals are precisely encoded, rebuilt, and received at the receiver end. Fast Fourier Transform (FFT), digital filters, and adaptive equalizers are examples of signal processing methods that are frequently used to enhance signal quality, reduce noise, and counteract channel deficiencies.

On the other side, compression techniques are used to minimize the size of digital data for effective storage and transmission (Al-Fuqaha et al., 2015). In order to achieve high compression ratios while maintaining passable signal quality, data compression algorithms like Huffman coding, arithmetic coding, and transform-based techniques like DCT and Wavelet Transform eliminate redundancy and exploit statistical properties of the data. In multimedia applications, such as audio and video compression, where vast volumes of data need to be efficiently conveyed or stored, compression techniques are commonly utilized. Modern communication systems like wireless networks, satellite communication, IP networks, and multimedia streaming depend on these signal processing and compression technique (Tataria et al., 2021) s. They improve the general caliber of communication services, allow for effective use of network resources, and cut down on transmission delays. Additionally, improvements in signal processing and compression algorithms continue to be the driving force behind developments in new communication technologies like 5G, the IoT, and virtual reality. These innovations enable faster data speeds, improved spectrum efficiency, and improved user experiences.

4. OVERVIEW OF AxC TECHNIQUES

AxC techniques represent a transformative approach in computing, addressing the growing need for high efficiency and reduced energy consumption in various applications. By intentionally introducing controlled approximations in computational processes, AxC optimizes the balance between accuracy and resource utilization. This chapter delves into the fundamental principles of AxC, highlighting its importance in enhancing system performance, fault tolerance, and error resilience. We explore key methodologies, including reduced precision computing, probabilistic computing, and stochastic computing, demonstrating their impact on diverse fields such as machine learning, signal processing, and communication systems. The integration of these techniques not only fosters innovation in computational architectures but also addresses the challenges posed by traditional computing paradigms.

4.1 Error-tolerant computing

The ability of a computing system to continue executing its duties correctly even in the face of defects is referred to as error-tolerant computing, with the aim of enhancing system dependability (Hin,,Liu., 2022). This is particularly significant for quantum computers, which are constructed using brittle and prone to error qubits (Atis et al., 2020). For example, mapping approaches that take advantage of high-quality qubits to enhance computation outputs have been developed as ways to accept errors in quantum computing. To guarantee system dependability and availability in conventional computing systems, fault-tolerant designs and redundancy are used. Tolerating single event errors in electronic circuits is also possible with the help of fault detection processors and programmable error filters. To improve system performance, error-tolerant computing also entails modifying operating settings depending on error detection.

The design and implementation of dependable and effective communication systems is notably impacted by error-tolerant computing and approximation computing. Error-tolerant computing methods are used in communications to guarantee accurate data transmission and reception. To find and fix faults created during transmission, error detection and correction codes are utilized, such as FEC. By enabling the recovery of faulty data, these strategies make sure that the information obtained is as correct as possible. On the other hand, AxC in communications can be used to optimize the use of network resources and enhance system performance (Barone et al., 2021). Data compression, signal processing, and data analysis are common components of communication networks, and these processes can all benefit from approximation computations. AxC can lower the computational complexity, power consumption, and bandwidth needs while still maintaining an acceptable level of service by reducing the accuracy criteria in some areas of communication systems (Tseliou, 2023). For instance, approximation computing techniques in video and picture compression can be used to reduce the amount of data broadcast by tolerating minute fluctuations or loss of information that might not be visible to the human eye. This compromise between compression effectiveness and precision enables more effective use of network bandwidth. AxC can be used to reduce power consumption in wireless communication systems, where it is essential to consider energy efficiency. Communication devices can save energy without significantly impacting the performance of the system by approximating complicated signal processing techniques, such as channel equalization or interference cancellation.

Additionally, approximation computing can be employed in contexts with restricted resources, such IoT devices, where there is a shortage of processing power and energy. IoT devices can save energy and increase their battery life by decreasing the precision of data representation and approximation computations (Mao et al., 2017).

Fault tolerance is the capability of a system—whether it be a computer, network, cloud cluster, or any other hardware component—to continue delivering uninterrupted service despite the failure of one or more of its components.

The primary objective of creating a fault-tolerant system is to safeguard critical applications and systems from failures due to single points of failure. Additionally, these systems should be designed to ensure that if one component fails, others can seamlessly take over without any loss of performance or lag time, thereby ensuring continuous operation. Fault-tolerant systems are engineered to withstand multiple failures and maintain operational integrity.

The failure of the computer's processing unit, I/O subsystem, memory cards, motherboard, power supply, or network components is automatically detected by such systems. The failure site is located, and without affecting service, a backup component or method is implemented right away (Kranz, 2008).

Each hardware component is duplicated to achieve fault tolerance at the hardware level. A disk is mirrored. The outputs of several processors are lock stepped together and checked for accuracy. When an anomaly happens, the problematic part is identified and removed from service, but the machine keeps working as usual.

Backup components are used in fault-tolerant systems to replace failing components automatically and prevent service interruptions. These consist of:

- Hardware systems that are supported by comparable or identical systems can enhance fault resilience. However, in fault situations with only two devices, such as running an identical server in parallel with the primary server and mirroring all operations to the secondary server, determining which device is "right" can be challenging (Sherman et al., 2021). Both devices are designed to provide fault resilience, but they may have different mechanisms for handling faults. To address this, it's important to consider the specific requirements and constraints of the system in question. For example, some systems may prioritize ensuring continuous operation by automatically switching to the secondary server in the event of a fault with the primary server. In contrast, others may prioritize data integrity and consistency by employing more complex fault-tolerance mechanisms, such as quorum-based protocols or data replication techniques.
- Sources of power that are fault-tolerant provide alternative generating sources utilizing different fuels or energy types. For instance, some businesses maintain backup power generators to ensure continuity in the event of an electricity supply

disruption (Hallegatte et al., 2019). Similarly, implementing redundancy allows any single point of failure component or system to achieve fault tolerance.

4.2 Failover and load balancing: fault tolerance for web applications

Fault tolerance in the delivery of web applications refers to the use of load balancing and failover solutions to guarantee availability through redundancy and quick disaster recovery.

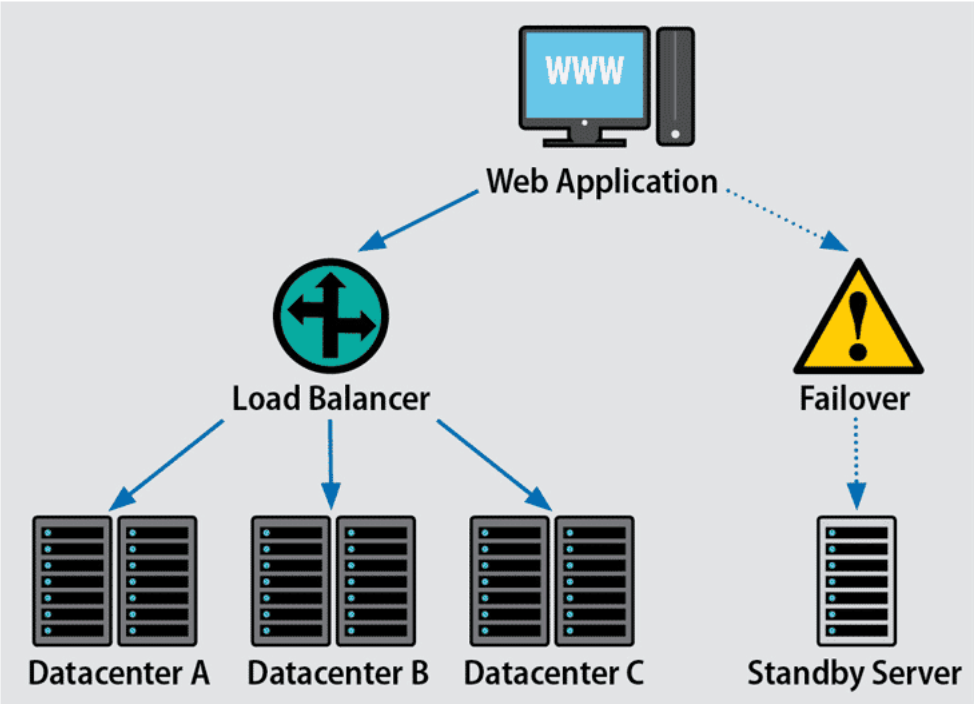


Fig. 4.1: Web applications' fault tolerance through failover and load balancing

Source: Paul Rubens, 2019

Fault tolerance in network systems relies on two key components: load balancing and failover.

Load balancing distributes workloads among multiple network nodes, reducing the risk of a single point of failure and managing activity surges. For instance, if one server in a system with two production servers fails, a load balancer can automatically reassign tasks to the other server.

Failover solutions address total network failures by activating a backup platform to maintain application functionality while the primary network is restored. "Hot" failover provides zero downtime by instantaneously shifting workloads to a standby system. If continuous standby is not feasible, "warm" or "cold" failover can be used, where the backup system takes some time to become operational.

Overall, the design of more resilient, effective, and dependable communication systems is made possible by the integration of AxC and error-tolerant computing in communications. While approximation computing approaches maximize resource use, energy efficiency, and system performance, error-tolerant algorithms maintain data quality and integrity (Alzoubi et al., 2022). Communication systems can be specifically designed to satisfy objectives and limits by carefully balancing the trade-off between accuracy and efficiency, resulting in increased communication capabilities and user experiences.

4.3 Probabilistic Computing

A new paradigm of computing called probabilistic computing makes use of random aspects to carry out tasks that classic von-Neumann computers find difficult. It is based on the idea that arithmetic operations are carried out using digital logic gates and that data is represented as random bitstreams. The fact that incomplete and uncertain data are frequently present in real-world problems serves as the inspiration for probabilistic computing. It is frequently more suitable to portray knowledge in these situations as probabilities rather than as absolute facts. Computers can learn from data, make predictions, and resolve issues in uncertain settings by employing probabilistic models.

- Indeed, while Bayesian networks, Markov models, Monte Carlo techniques, and probabilistic programming languages can be implemented in von Neumann computers, there are tasks that are inherently challenging for such architectures. (Del Ser et al., 2019). These tasks often involve complex mathematical equations and computations that may not align well with the sequential and deterministic nature of von Neumann processors.

  For example, tasks such as uncertainty quantification, probabilistic inference, and decision-making under uncertainty require extensive computations involving probability distributions, conditional probabilities, and iterative algorithms. These computations can be computationally intensive and may not be efficiently executed on traditional von Neumann architectures due to their limitations in parallel processing and handling probabilistic operations.

Furthermore, tasks involving large-scale probabilistic simulations or real-time decision-making may strain von Neumann processors, as they may struggle to process vast amounts of probabilistic data and perform complex calculations within strict time constraints.

Therefore, while von Neumann computers can indeed implement probabilistic models and algorithms, the inherent limitations of their architecture may pose challenges for certain tasks that involve intensive computations or real-time processing of probabilistic data. In such cases, alternative computing architectures, such as parallel or distributed systems, may offer better scalability and performance for probabilistic computing tasks (Arridge et al., 2019).

- Numerous fields, such as machine learning, robotics, computer vision, natural language processing, and cognitive computing, use probabilistic computing. The area has made considerable strides recently thanks to the development of strong computational tools and the increased accessibility of vast volumes of data.

These identifiable issues all appear to be distinct from one another, however they are related. All of these call for the application of inductive inference, which involves generalizing from observations of the outside world to their underlying causes.

Probabilistic computing has several benefits. It permits easy calculation while still allowing for the modeling of uncertainty and decisions in complicated contexts (Buot, 2006). Additionally, it offers a method for calculating or estimating reliability indices for networks with trustworthy nodes and poor communication lines. Probabilistic computing can also be used to algorithmically address probability problems, giving important insight into the computational elements of such problems. Compared to deterministic computing machines, such as traditional Von Neumann architectures, probabilistic computing introduces a broader class of computational models. These models include probabilistic Turing machines and other frameworks that inherently handle randomness and uncertainty, enabling more flexible and robust solutions to complex problems.

4.4 Stochastic computing

An alternative computing method called stochastic computing encodes values using probabilistic bit streams. Compared to binary-encoded processing, it delivers a higher computational density and uses less energy (Wang et al., 2020). At different layers of the computing stack, including design optimizations, architectural frameworks, compiler optimizations, and application transformations, stochastic computing has been proposed. It has been demonstrated that stochastic implementations are more suited for digital image processing algorithms than their conventional equivalents because they can tolerate more noise. It is frequently used in machine learning and artificial intelligence, as well as applications that need for high-speed signal processing, like the processing of images and audio (Mcquillan et al., 2019).

4.5 Reduced precision computing

Reduced precision computing is a technique used to reduce power usage and increase calculation latency, among other performance parameters. Instead of the typical 32-bit single-precision floating-point format, it calls for the use of smaller floating-point or fixed-point formats. Numerous fields, including embedded systems and high-performance computing (HPC), have examined this strategy. Recent research has concentrated on automating the precise customization process, even if human tuning and code transformation are frequently utilized in most areas. Although several tools have been created to automate this process, tools based on static analysis and integration with common compiler frameworks are still required (Cherubin & Agosta, 2020). In order to increase the speed, energy efficiency, and memory footprint of neural network implementations, lower precision has shown promising outcomes (Wang et al., 2019). As a holistic strategy to enhance performance and reliability, the idea of minimal-precision computing with precision-tuning, which encompasses both the hardware and software stack, has been put out.

Reduced precision number systems are used to represent and process data in reduced precision computing approaches. Accuracy, range, and computing/storage costs can all be traded off with these methods. Here are some instances of specific tradeoffs between various number systems:

Fixed-Point Number Systems: In fixed-point number systems, both the integer and fractional components of a number are represented using a fixed number of bits.

Reducing the number of fractional bits sacrifices precision, but it can increase the range and decrease storage requirements (Liu et al., 2022). Unlike floating-point numbers, fixed-point numbers can be operated on as if they were integers, simplifying the arithmetic operations. For instance, switching from a 32-bit floating-point representation to a 16-bit fixed-point representation reduces storage needs but may introduce rounding errors.

Logarithmic Number Systems (LNS): Instead of using fixed-point or floating-point representations, LNS uses logarithms to represent numbers. LNS can provide greater precision for a given bit count by more effectively dispersing the dynamic range. They may also have limited hardware support and need additional computational operations for arithmetic calculations.

Quaternary Number Systems: Instead of the usual base-2 binary representation, quaternary number systems use a base-4 representation (using four digits: 0, 1, 2, and 3). Accuracy, range, and storage expenses are all balanced by quaternary systems. They may be useful for particular applications, such as DNA sequence analysis or signal processing methods, when the operations are inherently compatible with the base-4 representation.

The examples provided show the trade-offs between accuracy, range, and computing/storage costs in several reduced precision number systems. The particular application needs and the intended balance between these aspects determine the number system to use. Developers can enhance the speed and effectiveness of their computations while still adhering to the limitations of the target platform by carefully choosing the right number system (Dwivedi et al., 2022).

As the need for high-performance computing in areas like machine learning and scientific simulations grows, reduced precision computing has grown in popularity over the past few years. The use of reduced precision arithmetic can considerably speed up the execution of these applications, which frequently demand enormous amounts of calculation. The use of decreased precision arithmetic does, however, present some difficulties, particularly in preserving numerical stability and accuracy. The development of methods for enhancing the accuracy and stability of calculations utilizing decreased precision arithmetic has so received a lot of scientific attention.

5.APPLICATIONS OF AxC IN COMMUNICATION SYSTEMS AND NETWORKS

In communication systems and networks, AxC has several uses. It can be used to create distributed software applications for networks or the Internet that are shared by network computing systems (Mason, 1990). Additionally, distributed averaging over unstable networks is a use case for approximation computation in networked control systems. As a result, the effects of inaccurate information flow are lessened and it is possible to create better ad hoc networked dispersed algorithms (Elia et al., 2014). Additionally, in network-connected computing systems, such as switch-based computing systems used for content distribution, approximation computing can optimize bandwidth use and speed up system performance. In general, approximation computing presents prospects for improving the effectiveness and performance of networks and communication systems in a variety of settings.

5.1 Application in Communication System

Due to its potential to boost throughput while maintaining acceptable levels of QoS, decrease latency, and increase energy economy, AxC has drawn substantial attention in the field of communication systems and networks. A few examples of approximation computation in communication systems and networks are as follows:

Error-resistant video coding: Video coding, in which video data is compressed and transferred through communication channels, is a crucial part of multimedia communication systems. Designing error-resistant video coding algorithms that are more energy-efficient and have lower latency than conventional methods is possible using AxC techniques. Examples include low-complexity transform coding methods that compromise video quality for computational complexity.

Network routing is the process of deciding the fastest route for data packets to take when moving from source to destination inside a network. By lowering the demands for routing accuracy, AxC approaches can be utilized to build routing algorithms that save energy and increase network throughput (Herrero, 2022).

Signal processing involves the analysis, filtration, and transformation of signals. It is a crucial component of many communication systems. Designing signal processing algorithms that utilize fewer resources and are more energy-efficient while retaining acceptable levels of signal quality is possible with AxC techniques.

WSNs are employed in a variety of settings, such as medical care, industrial automation, and environmental monitoring. WSNs are made up of a lot of small, low-power wireless sensors for data collection and transmission. The energy consumption of WSNs can be decreased using AxC approaches by creating effective data compression, aggregation, and fusion algorithms.

Cloud computing: A new paradigm in which computing resources are made available as a service over the internet. By creating effective resource allocation and task scheduling algorithms that consider the trade-off between computation precision and energy usage, AxC approaches can be utilized to increase the energy efficiency of cloud computing systems (Herrero, 2022).

5.2 Signal Processing

The performance of communication systems and networks is greatly influenced by signal processing, which is a critical component of these systems. Signal processing algorithms can use AxC approaches to become less computationally complex and energy-intensive while still providing adequate performance.

The use of low-precision arithmetic in DSP algorithms is an illustration of AxC in signal processing. The computations can be carried out faster and with less energy use if fewer bits are used to represent numbers. Several DSP techniques, including Fourier transforms, digital filters, and image and video compression, have been approximated with this method (Sze et al., 2017).

The employment of randomized algorithms for signal processing tasks including signal demising, image and video compression, and channel equalization is another illustration of approximation computing in signal processing. When compared to accurate methods, these algorithms can significantly reduce computing time and energy consumption by using random projections and sampling techniques to approximate the original signal.

Channel estimation is a crucial activity in wireless communication systems that uses a lot of energy and compute. Channel estimation methods can use approximate computation techniques to become simpler and use less energy while still providing a level of accuracy that is acceptable. In the case of channel estimation, for instance, the adoption of compressed sensing techniques can reduce the number of measurements needed, which can result in significant reductions in computing and energy consumption.

5.3 Machine Learning

ML technique comprises, but is not restricted to, methods like Bayes classifiers, neural networks, and support vector machines (SVM). With the help of these algorithms, predictions or classification may be made using the hidden patterns that can be found in a vast amount of data. In more detail, machine learning can be thought of as a set of algorithms for determining a function that maps inputs (sample data) to outputs (the intended class), but this function is too complex to be expressed in a practical manner.

5.3.1 Construction of a Machine Learning System

Different strategies and methodologies can be used to build a machine learning system. One strategy is to analyze and extract significant features using machine learning algorithms based on contextual data, which can produce a richer and more reliable feature representation. Another way to use a machine learning method is to save and alter training data sets, then create a machine learning algorithm utilizing those data sets.

To increase productivity and handle the trade-offs between accuracy and computational costs, machine learning techniques are being used increasingly in AxC for communication systems. Use of machine learning models to forecast channel quality in wireless communication networks is one example. In wireless communication, channel quality information is essential for selecting the best power allocation and modulation strategies, among other transmission factors (Ding et al., 2017). In the past, accurate but computationally expensive algorithms have been used to evaluate channel quality; however, these methods can be unworkable in situations when resources are limited. AxC methods can be used to predict channel quality with lower computational costs by utilizing machine learning. An AI model, such as a neural network, is trained using training data that includes different channel conditions and the accompanying quality measures (Wang et al., 2020). The model learns the patterns and correlations between channel quality metrics (such as signal-to-noise ratio, bit error rate, etc.) and input parameters (such as received signal strength, noise levels, etc.). Once trained, the machine learning model can be used to estimate channel quality instantly in real-time communication systems. As a result, decision-making processes can be quicker and more effective. One example is the dynamic adaptation of transmission settings based on expected channel quality. The prediction's approximate nature forces a trade-off between precision and processing expense. However, a balance that satisfies the unique needs of the communication system can be found by carefully training and optimizing the model.

5.3.2 Image and video processing

Filtering, compression, and object recognition are a few of the computationally demanding procedures that are frequently used in image and video processing applications. While retaining an acceptable degree of visual quality, AxC approaches can be used to lower the computational complexity and memory needs of these jobs. In order to make wise decisions on the level of approximation to apply at various stages of image and video processing pipelines, machine learning techniques can be used to examine the trade-off between accuracy and efficiency.

a) **Filtering:** AxC can be used for edge detection, demising, and other image and video filtering processes. Computational complexity can be decreased by sacrificing some degree of accuracy in these procedures, allowing real-time processing on devices with limited resources.
b) **Compression:** AxC can help image and video compression techniques like Joint Photographic Experts Group (JPEG) and Moving Picture Experts Group (MPEG). The size of the compressed data can be decreased by allowing some approximation in the compression methods, which lowers storage needs and boosts transmission effectiveness.
c) **Object identification:** Object identification tasks, such identifying items in pictures or movies, can be performed using AxC. The feature extraction and classification stages can be simplified while still retaining a suitable level of recognition accuracy by using approximation approaches.

5.2.3 Big Data Analytics

Big Data analytics is the practice of using sophisticated strategies and tactics to derive value from sizable and intricate databases. Massive amounts of data must be processed, analyzed, and translated in order to uncover new information and speed up discovery.

Big data has had a substantial impact on a number of industries, including public relations, supply chain management, computational research, and organizational decision-making. Data analysis, collection, curation, search, sharing, storage, transmission, visualization, and information privacy are among the difficulties in big data analytics (HaCohen et al., 2007). To overcome these issues and generate useful insights from big data, there are, however, also considerable opportunities in areas like machine learning, data mining, statistics, human-computer interfaces, and distributed systems (Matter, 2023); (Nereu et al., 2017).

Processing vast amounts of data in order to uncover patterns and insights is known as big data analytics. By lowering the computational complexity, AxC approaches can dramatically speed up data processing. Approximation can be employed without affecting the overall accuracy of the results by using machine learning techniques to determine the data subsets or features that are less important to the analysis. Big data analytics pipelines can be made faster and more scalable thanks to this combination.

a) **Data preprocessing:** By using approximation in the data cleaning, normalization, and feature extraction phases, AxC can be utilized to preprocess big datasets. The preprocessing pipeline can be optimized for scalability and efficiency by using machine learning algorithms that can be trained to discover the acceptable levels of approximation for various types of input (Zhou et al., 2017).

b) **Clustering and Classification:** In big data analytics, AxC helps speed up clustering and classification operations (Alam, 2018). The computing requirements can be drastically decreased while still retaining respectable clustering and classification accuracy by approximating the distance computations and feature comparisons.

c) **Dimensionality Reduction:** AxC is useful for dimensionality reduction methods like Principal Component Analysis (PCA). The computational cost of dimensionality reduction can be lowered by using approximation in the calculation of eigenvectors and eigenvalues, enabling quicker analysis of high-dimensional data.

5.2.4 Robotics & Autonomous Systems

By enhancing real-time decision-making capabilities, AxC can be useful in robotics and autonomous systems. The proper level of approximation in perception, planning, and control algorithms can be determined by analyzing sensor data, environmental factors, and task requirements using machine learning algorithms. As a result, decision-making can be done more quickly and effectively while yet maintaining acceptable error limitations for the system.

1. **IoT and Edge Computing:** A concept known as IoT enables digital assets and gadgets to exchange data over a network without the involvement of a human. The vast volume of data produced by IoT devices, however, presents problems for cloud computing in terms of processing speed and service level (Kaur & Batth, 2021). Edge computing has been proposed as a decentralized strategy that moves data processing closer to the network edge to overcome these difficulties. Response time, energy use, and resource usage for IoT applications are all improved via edge computing. Task scheduling, software-defined networks, network function virtualization, security, privacy, and blockchain integration are further features supported. To improve network performance and facilitate mobility, security, and privacy, edge computing designs like Multi-access Edge Computing, Fog Computing, Cloudlet Computing, and Mobile Cloud Computing have been proposed. Future research will examine replica placement and selection strategies, data provenance, and issues with data accessibility and access response times (Shao et al., 2021).

   AxC can help improve memory and energy use in contexts with limited resources, such as IoT devices and edge computing systems. The performance requirements and limitations of IoT applications can be analyzed by machine learning algorithms, which can then choose the right level of approximation to obtain the intended results. This makes it possible to do IoT-related operations like anomaly detection, predictive maintenance, and sensor data processing efficiently.

   a) **Processing of Sensor Data:** AxC can be utilized in IoT applications to handle sensor data gathered from multiple sources. Data filtering, feature extraction, and anomaly detection can all be done with fewer computational resources and more effectively in real-time by using approximation approaches.

b) **Predictive Maintenance:** By examining sensor data and finding patterns suggestive of possible failures or maintenance requirements, AxC can contribute to predictive maintenance. Predictive maintenance systems can be made faster and more effective by training machine learning algorithms to recognize important elements in the data and use approximation (Meng et al., 2020).

c) **Anomaly Detection:** Machine learning and AxC can be used to find anomalies in IoT systems. Approximate approaches can simplify anomaly detection algorithms and enable effective real-time detection of anomalies in sensor data streams by recognizing redundant or less important data elements.

6. ADVANTAGES AND CHALLENGES OF AXC IN COMMUNICATION SYSTEMS AND NETWORKS

In this chapter we will discuss advantages and challenges of AxC in communication system and networks. In communication systems and networks, AxC presents benefits as well as difficulties. On the one side, approximation computing can help with technological issues including high performance, circuit reliability, and power consumption. It enables better performance while still using the same amount of power, which is a big concern in the sector (Nong, Ye, 2008).

On the other hand, ensuring the accuracy of signals, logic values, devices, and interconnects presents difficulties. To ensure absolute accuracy, manufacturing and verification expenses will rise dramatically. Additionally, bridging the gap between thin-client devices and bandwidth-intensive applications and integrating edge-of-network devices into the network are issues presented by the convergence of computing and networking.

6.1 Advantages

There are various possible advantages to AxC in communication systems and networks. Small estimates made during system operation can result in significant energy savings. AxC has the ability to increase approximation quality while requiring less resources and less time to run programs in communication systems and networks (Magadum & Ghosh, 2021). A full-system perspective can be attained by simultaneously and coordinative approximating several computing platform subsystems, which opens up further chances for energy savings (Raha & Raghunathan, 2019). According to (Raha & Raghunathan, 2019) , employing approximation computing in communication systems and networks can result in energy savings with minimal influence on output quality. This method has shown the potential to conserve energy within the range of 1.8 to 5.5 percent with little application-level quality loss, compared to approximations made at the individual subsystem level. Additionally, by choosing skipping spike-triggered neuron updates with no effect on output quality, approximation computing can increase the computational effectiveness of analyzing large-scale SNNs (Xu et al., 2016). Reduced scalar operations for network evaluation have been demonstrated by this method, improving hardware and software energy without sacrificing application quality (Liu, Lombardi, et al., 2020).

In communication systems and networks, AxC techniques offer several benefits. They reduce energy consumption by minimizing computation needs, crucial for battery-powered devices. Additionally, they enhance performance by lowering latency and increasing throughput, particularly vital in high-speed networks. Moreover, AxC is cost-effective, demanding less expensive hardware and power, thereby improving affordability and adoption rates. Furthermore, these techniques enhance scalability by reducing the computation required for data analysis, enabling systems to manage higher volumes of data efficiently. Lastly, AxC improves reliability by mitigating mistakes caused by noise and interference, enabling fault tolerance and resource optimization, ultimately enhancing communication quality and reducing the need for error correction mechanisms. (Damsgaard et al., 2023)

6.2 Challenges of AxC in Communication Systems and Networks

There are various difficulties with approximate computation in communication systems and networks. The introduction of security flaws resulting from erratic and unpredictable errors during approximatively execution, which can be mistaken for malicious alterations, is a problem (Pande, 2023). To effectively utilize the advantages of approximation, it is necessary to rethink several layers of the system stack, including the architecture, programming model, and algorithms (Liu, Gu, et al., 2020). A framework for analyzing and contrasting security methods in approximation computing is also lacking (Agrawal et al., 2016). Additionally, additional study is required to address the issues this sector may face in the future (Xu et al., 2016).

Several key difficulties include:

- Accuracy trade-offs: Balancing the trade-offs between accuracy and performance is one of the fundamental difficulties in AxC. Approaches that put efficiency over accuracy run the risk of producing unacceptable levels of inaccuracy or deteriorating quality.

- Design complexity: Because precise and effective approximation requires specialized hardware, software, or algorithms, implementing AxC techniques in communication systems and networks can be difficult. Increased design complexity and expense may result from this.
- Robustness and reliability: The robustness and reliability of communication systems and networks may be impacted by AxC's tendency to raise the probability of errors. To ensure system reliability, strategies for error detection and repair must be used.
- QoS: Networks and communication systems' QoS may be impacted by AxC. A big problem is ensuring that the level of approximation is appropriate and does not dramatically affect the QoS.
- Compatibility may be impacted by the introduction of approximation computing techniques in communication systems and networks, which may necessitate alterations to current protocols and standards.

7. CASE STUDIES AND EXAMPLES

In this chapter, we will discuss some case studies and the application of AxC in communication systems. We will explore examples such as the implementation of AxC in WSN, data compression techniques for multimedia communication, and energy-efficient protocols for IoT devices. These case studies illustrate how AxC can enhance performance and reduce costs in different communication scenarios.

Compact and affordable wireless sensors are now possible due to recent advancements in microelectronic technology. These sensors benefit from AxC techniques, which allow for lower power consumption and extended operational life without significantly compromising data accuracy. Through these case studies, we aim to demonstrate the practical advantages and challenges of integrating AxC into communication systems, providing a comprehensive overview of its potential impact on the industry.

Energy conservation in sensor networks is crucial to extending network lifetime since sensors in sensor networks are provided with energy-limited batteries. The two fundamental problems of sensing coverage and sensor connectivity in sensor networks have been extensively discussed in the literature. The majority of the research on sensing coverage to date has concentrated on the (connected) full coverage problem, which aims to cover the entire monitored region with the fewest possible sensors. A partial coverage with a certain degree of certainty is appropriate in some application settings, whereas full coverage is sometimes either unnecessary or unattainable. In this thesis we examine the related coverage issue with a predetermined coverage guarantee.

7.1 AxC in WSN

In WSNs, AxC entails the use of algorithms and protocols that deliver effective and accurate results while preserving energy resources. To address this challenge, researchers have suggested a variety of strategies. One approach is utilizing in-network data aggregation methods, which combine partial results at intermediary nodes during message routing to reduce communication costs (Yavuz et al., 2010). Another method involves designing algorithms to optimize sensor and sink locations, activity schedules of deployed sensors, and data flow paths from sensors to sinks to enhance network longevity.

The development of algorithms for sensor deployment and coverage, routing, and sensor fusion in WSNs has also progressed. These methods aim to increase the performance and efficiency of WSNs while considering the limited energy and capabilities of the sensors (Sankardas et al., 2018). AxC in WSNs leverages the built-in redundancy and error tolerance of specific applications to trade off accuracy for efficiency and performance gains. WSNs are networks of numerous sparsely resourced sensor nodes deployed across an area to monitor physical occurrences and communicate collected data to a sink or centralized node.

Traditional WSNs often include sensor nodes doing exact computations on obtained data, which uses a lot of energy and shortens the lifetime of the network. Many WSN applications, however, can tolerate some degree of error without it materially hurting how well they work as a whole. The development of approximate computation techniques in WSNs is a result of this observation.

The following are some essential applications of approximation computing in WSNs:

Data aggregation: AxC methods allow sensor nodes to carry out local data aggregation rather than sending all raw sensor data to the sink. Redundant data can be removed by merging or summarizing data from nearby nodes, which lowers transmission costs and energy usage.

Approximation of computations: AxC in WSNs is a pivotal strategy to enhance efficiency and prolong the operational lifespan of sensor nodes. By intentionally relaxing the precision of certain computations, AxC techniques can significantly reduce the energy consumption and computational load. This trade-off is particularly advantageous in WSNs, where sensors are often deployed in resource-constrained environments and need to operate autonomously for extended periods. Applications such as environmental monitoring, target tracking, and health surveillance can tolerate minor inaccuracies, making them ideal candidates for AxC methods. Consequently, these techniques not only conserve energy but also improve response times and reduce communication overhead, thereby enhancing the overall performance and reliability of WSNs. Additionally, step-discontinuity approximation techniques have been utilized successfully to identify the polarization of low frequency echoes. Finally, applying higher-order approximations with the harmonic balancing approach can enhance the identification of limit cycles.

Routers in WSNs: It can use AxC to implement error-aware routing. Energy-efficient routing algorithms can take into account the approximate nature of the data and dynamically adjust the routing paths based on the desired level of precision to optimize energy consumption.

QoI conscious methods: QoI conscious strategies seek to guarantee that the data supplied by sensor networks satisfies specific standards for quality. These approaches evaluate the QoI using a variety of criteria, including accuracy, utility, uniqueness, relevancy, detection probability, false alarm probability, and average error probability. In order to control the QoI needs, they also involve optimizing sample rates and modifying the data sampling rates of sensors. QoI measurements may be based on the probability function, information gain, or the entropy of the covariance matrix. It is crucial to remember that covariance matrices alone cannot be the basis for trustworthy QoI measures since they might not appropriately reflect tracking performance. The smoothed residual likelihood, however, has demonstrated a strong correlation with tracking performance (Duc Van Le et al., 2017).

AxC techniques consider the required QoI for a specific application rather than concentrating only on reaching maximal accuracy. With the aid of QoI-aware approaches, the system is able to compromise accuracy in favor of latency, energy efficiency, or other application-specific needs. Extended network lifetime, lower energy usage, increased scalability, and higher performance are advantages of approximation computation in WSNs. To make sure that the introduced errors do not impair the overall functionality of the system, it is crucial to carefully assess how approximation may affect the requirements of the particular application. Approximate computation approaches in WSNs offer a viable way to balance energy efficiency and application accuracy while optimizing resource-constrained sensor networks.

7.2AxC driven green 6G downlink

For the downlink of 6G networks, AxC is a viable technique for creating energy-efficient processing hardware. AxC can result in significant reductions in dynamic power while preserving good arithmetic accuracy by swapping precise arithmetic units for their approximate counterparts. The suggested AC approaches provide up to 87% savings in dynamic power and extremely acceptable arithmetic accuracy in the setting of single-antenna and dual-antenna 6G downlink, with a structural similarity index (SSIM) above 93% and a correlation coefficient (CC) of 99% (Idrees et al., 2021). Additionally, the energy efficiency of 5G communication networks can be further improved by integrating with 5G and blockchain (BC) technology. The integration of smart technology in a distributed world of power consumption is made possible by the combination of 5G, green BC, which may offer quick, transparent, and safe data transmission (Mazen et al., 2021).

A new embedded computing paradigm called AxC substitutes approximate adders and multipliers for accurate ones in computing platforms like CPUs, FPGAs, ASICs, and others. The viability of the AxC for single-antenna and dual-antenna 6G downlink is being examined. AxC-powered transceiver design of a 6G downlink in which the decoders/equalizers and pulse shaping filters are implemented on the base station (BS) and user equipment (UE) sides, respectively, using cutting-edge approximate/inexact arithmetic units. Bit error rate (BER), (SSIM), and (CC) are used as performance metrics to assess the AxC impact on measurement accuracy, while dynamic power and on-chip power are used to assess the proposed AxC techniques' energy-efficiency advantages.

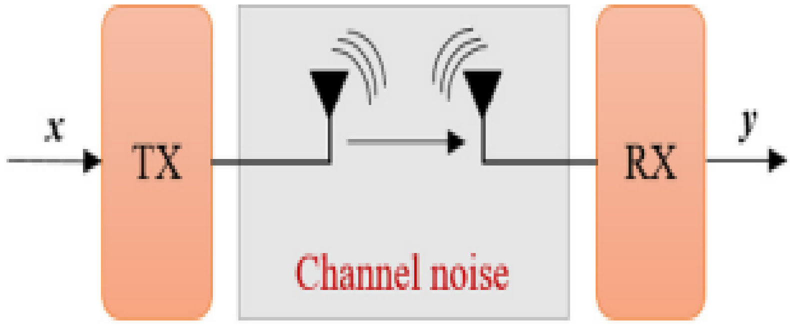


Fig. 7.1: Single antenna 6G downlink system model. Source: Maryam Idrees, 2021

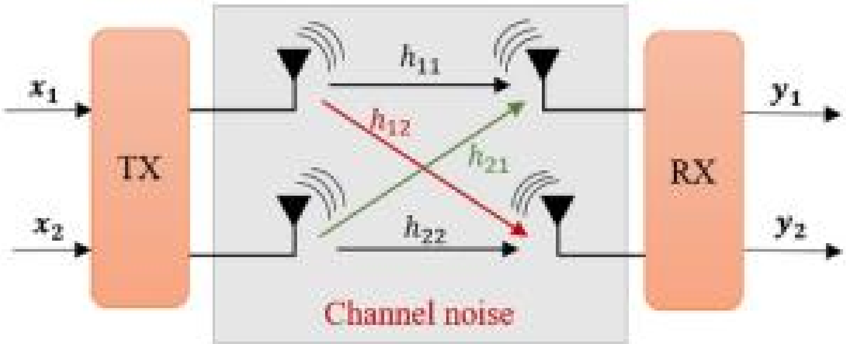


Fig 7.2: Network framework: Dual antenna 6G downlink. Source: Maryam Idrees, 2021

7.3 Challenges and Solutions in Integrating AxC with IoT Devices

The research community is interested in the problem of AxC in IoT devices. Implementing distributed databases on IoT devices is difficult due to their low processing power (Richardson et al., 2017). Fog computing, a kind of cloud computing, has been suggested as a solution for IoT-based applications that demand fault tolerance and scalability (Prabhu et al., 2018). Fog computing is one type of distributed computing that can assist in overcoming the difficulties associated with integrating IoT-based applications into centralized systems (Sihwa et al., 2017). The usage of distributed computing in the IoT can also offer benefits like scalability and fault tolerance (Burns et al., 2018). However, to handle massive data scenarios that emerge in the setting of fog computing, typical machine learning algorithms must be extended (Patel, 2019). In general, research in this field focuses on examining how distributed computing, particularly fog computing, might be used to develop IoT-based applications and handle the problems with scalability and fault tolerance.

The IoT today encompasses a vast array of assets, including wireless sensors, software, actuators, computing devices, and numerous other components. These elements are interconnected, enabling communication over the internet with other devices or individuals without requiring human intervention.

For instance, consider the IoT capabilities of a modern automobile. Its integrated IoT devices can detect traffic congestion and notify the individuals you are scheduled to meet of potential delays. Similarly, a pacemaker represents a simple IoT device that can communicate with other medical devices, enhancing the convenience and quality of life for users.

8. FUTURE DIRECTIONS AND RESEARCH CHALLENGES

Future research should concentrate on understanding how AxC techniques affect system security, creating reliable error correction systems, and investigating standardization initiatives to ease the adoption and interoperability of AxC techniques in communication systems.

8.1 Improving accuracy in AxC

In computing systems, AxC is a design choice that compromises accuracy for energy efficiency. By dynamically adjusting the level of approximation at runtime, one can increase accuracy in AxC (Ometov & Nurmi, 2022). This enables energy optimization while keeping the induced mistakes within bounds. Another method is to use a Bayesian network to describe the transmission of approximation across the data and assess the application-level correctness (Zervakis et al., 2020). This method forecasts the likelihood of outcomes falling into various accuracy classifications and offers precise estimates of the approximation error with little computational overhead. In order to guarantee user-specified accuracy at runtime, a heterogeneous platform can be constructed by integrating exact accelerators with approximation accelerators based on neural networks (Ono & Usami, 2019).

It is possible to apply neural networks in communications systems, taking use of their approximate propensity to improve the effectiveness and performance of communication activities (Gupta et al., 2020). A vital step in wireless communication systems, channel decoding, is one application where neural networks are used. The goal of channel decoding is to extract the delivered data from distorted and noisy signals. The computational complexity of conventional decoding algorithms can prevent their real-time implementation in devices with limited resources. AxC methods can be used to balance accuracy and computational complexity by using neural networks for channel decoding. In order to speed up processing while preserving a reasonable level of decoding accuracy, the neural network can be trained to approximate the decoding method (Damsgaard et al., 2023). Based on the value of the bits, the neural network can learn to make educated guesses and prioritize decoding some bits precisely over others. Additionally, neural network training itself can make use of approximate computation methods. Without sacrificing the network's overall performance, computational and storage expenses can be greatly decreased by quantizing the weights and activations of the neural network to lower precision. To make sure that the trade-off between accuracy and resource use satisfies the unique requirements of the communication system, this approximation process can be carefully managed. More hardware, which consumes more energy, is often used to increase a neural networks accuracy. Due to NNs' error tolerance and the applications, they are employed in, AxC methods can be used to reduce implementation costs.

8.2 Standardization of AxC techniques

Standardizing AxC methods involves creating uniform guidelines and practices for implementing approximate algorithms and techniques. This can lead to reduced execution time and energy consumption across various applications by optimizing the trade-off between accuracy and resource usage. By adhering to standardized methods, developers can ensure consistent output precision while benefiting from the efficiency gains of AxC. Data can be used alone, in conjunction with an external reference, or both to determine standard representation words (Malak et al., 2019). In a special edition, the technological advancements and contributions of approximation computing are examined, offering insight into potential future approaches (Weiqiang et al., 2020). Also covered are standardization efforts in the realm of cryptography, including those of international, European, and North American standard groups (Bart, Preneel, 1993).

Any technology or method, including approximation computing, needs to be standardized in order to be widely used and interoperable. While efforts to standardize have just started to develop, AxC is still a research topic that is active. The following are some elements of approximate computation that should be standardized:

Metrics and Evaluation: To compare and assess various AxC algorithms, it is essential to standardize metrics and evaluation methodologies. This includes developing standard criteria to assess precision, energy use, performance, dependability, and other pertinent aspects (Mourtzis et al., 2022). To enable fair comparisons and evaluations of approximation computing techniques, organizations and research communities are adopting standardized evaluation methodologies and standards.

Standards for Languages and Interfaces: Standardizing programming languages, application programming interfaces (APIs), and interfaces for approximation computing can facilitate implementation across many platforms and software tools. Guidelines, terminology, and libraries for implementing approximations into software and hardware designs can be defined by standardization efforts in this field. This can facilitate compatibility between various tools and frameworks and streamline the development process.

Algorithmic Standards: Designing and using approximation algorithms and methodologies is a common task in AxC. By making these algorithms more uniform, researchers and developers will be able to improve upon already-proven strategies and maintain consistency. Collaboration and knowledge sharing between the academic and business communities can be facilitated by disclosing and documenting best practices for approximation algorithms.

Hardware and Architecture Standards: To perform at their best, AxC approaches frequently need hardware support or architectural features. The smooth integration of approximation computing into many platforms and systems can be made possible by standardizing hardware interfaces, communication protocols, and architectural components (Wang et al., 2024). Attempts to standardize hardware elements can aid in compatibility promotion, design complexity reduction, and make approximation computing more widely adopted.

Industry Collaboration: The standardization initiatives in approximation computing must be driven by collaboration among industry participants, researchers, and standardization bodies. By bringing together stakeholders from academia, industry, and regulatory authorities, organizations like IEEE, ISO, and IEC can play a significant role in defining standards. Industry-wide standards that facilitate the use and integration of approximation computing techniques may be developed through cooperative initiatives (Ali et al., 2023).

It is important to keep in mind that the subject of approximation computing is still in the early phases of its rapid evolution. Standardization efforts are anticipated to pick up steam as the area develops and agreement is obtained on fundamental approaches and procedures, resulting in established standards that encourage the wider adoption and interoperability of approximation computing techniques.

8.3 Development of tools and methodologies for AxC

The creation of approaches and tools for approximation computing has attracted a lot of attention recently. To enable approximation computing and increase computer efficiency, researchers have proposed a variety of solutions at various levels of the computing stack, including circuits, architecture, and software (Ono & Usami, 2019). A different strategy is a combined multilevel optimization method, which improves on single-level methods by taking use of various optimizations that are evident at each level (Siyuan et al., 2017). In order to design, verify, test, and ensure the dependability of approximation computing systems, there is also a need for tools and procedures (Bosio et al., 2020). These advancements seek to make computing more energy-efficient and to promote a more environmentally friendly computing era (Hrishav et al., 2019).

The development of tools and methods for AxC has become a popular area of study in recent years. The objective is to give designers effective and dependable methods for spotting and taking advantage of approximation opportunities in their designs while maintaining the overall functionality and performance of the system.

Here are a few of the most important instruments and approaches for approximatively computing:

- Finding approximation possibilities and evaluating their effects on system performance are two of the main problems in AxC. To assist designers in quantifying the trade-off between accuracy and energy consumption, numerous modeling and analysis tools, including statistical analysis, machine learning, and information theory, have been created. These methods can also be used to examine how approximations affect a system's robustness, reliability, and other important metrics.
- Provide designers with tools that allow them to study and incorporate approximation techniques into their ideas is another crucial feature of AxC. Design professionals can quickly investigate and choose the optimal approximation solutions for their unique design goals and constraints with the use of approximation-aware design tools, which make use of techniques like optimization, synthesis, and automatic tuning.

- New programming languages and libraries that support approximation at different levels of the software stack are needed for AxC. For instance, developers can explicitly describe approximation and uncertainty in their programs when using probabilistic programming languages. Low-precision computations, which are crucial for effective deep learning on platforms with limited resources, are supported by libraries like Tensorflow and PyTorch.
- Techniques for evaluation and validation finally, accurate evaluation and validation mechanisms are needed for AxC to make sure that the approximations do not affect the system's overall functionality and performance. Fault injection, error propagation analysis, and benchmarking are a few examples of approaches that can be used to validate the approximation techniques employed in the system as well as to evaluate the accuracy, energy consumption, and other crucial metrics of the system.

8.4 Suggestions for additional study and development

The following ideas for more study and development in the area of approximation computation in communication systems are provided:

1. Investigate ways to include quality-aware approximation approaches into communication systems. For various communication activities, provide models and algorithms that may dynamically adjust the level of approximation based on desired QoS metrics including latency, dependability, and throughput.
2. Investigate hybrid approaches, which integrate approximation computing with conventional computing methods. To obtain the greatest balance between accuracy and energy efficiency in communication systems, examine the synergy between precise and approximation computations (Veers et al., 2023).
3. AxC in communication systems is made possible by hardware accelerators and specialized designs. Examine the ways in which the performance, energy efficiency, and scalability of approximation computing algorithms for communication tasks can be enhanced by bespoke hardware.
4. Consider the security and privacy implications of AxC approaches in communication systems. Investigate potential weaknesses brought on by approximation errors, and create reliable error correction systems and privacy-preserving techniques to reduce these risks.

5. Develop resource allocation and optimization methods that take into account the degree of approximation that various communication activities require. Learn how to dynamically distribute resources, like bandwidth and computing power, according to the particular needs of various applications and the degree of approximation used.
6. To assess the effectiveness and viability of approximation computing techniques in communication systems, real-world deployments and case studies must be conducted. In real-world applications, take into account elements like scalability, energy efficiency, dependability, and real-time restrictions.
7. Contribute to efforts at standardization in the area of approximation computation for communication systems. To ensure interoperability, accurate comparisons, and a wider usage of approximation computing approaches in communication systems, investigate ways to create standard frameworks, benchmarks, and assessment processes (Rasheed et al., 2020).
8. Energy- and spectrum-efficient techniques for approximation Create approximation methods that can jointly maximize communication systems' spectrum usage and energy efficiency (Khan et al., 2020). Examine how to accomplish energy- and spectrum-efficient communications by utilizing the redundancy in communication signals and the error resilience of approximation approaches.

8.5 Possible Future Work

Future research should concentrate on understanding how AxC techniques affect system security, creating reliable error correction systems, and investigating standardization initiatives to ease the adoption and interoperability of AxC techniques in communication systems.

There are numerous areas of potential future study and research in AxC in the context of communication networks. Here are some possible headings:

1. Energy-Efficient Communication Protocols: By using approximation computing techniques, communication protocols in wired and wireless networks can be optimized for energy efficiency. Future research can concentrate on creating routing and data transmission protocols that take advantage of the communication systems' tolerance for faults while maintaining respectable levels of dependability

and service quality. These protocols can dynamically change the level of approximation dependent on the state of the network and energy limitations.

2. Signal Processing and Compression Techniques: Signal processing and compression techniques are essential in communication systems because they enable functions including modulation, encoding, decoding, and data compression. Future studies could look into techniques for approximating these algorithms while taking use of communication networks' fault robustness. Without sacrificing the overall system performance, it could be able to save a large amount of energy by carefully controlling the level of approximation.
3. QoI-Conscious Approximation Techniques: Communication systems must consider the QoI, which has to do with how accurate and trustworthy the material being conveyed is. Future research can concentrate on creating QoI-aware approximation algorithms that take into account the unique needs of various applications and modify the level of approximation as necessary. To fulfill application-specific QoI limits, this can entail approaches for quantifying and optimizing the trade-off between accuracy and energy usage.
4. Hardware Acceleration for Enhanced AxC: AxC approaches can be considerably improved in terms of performance and energy economy by using hardware acceleration. Future research can examine the creation of specific hardware designs and accelerators that are geared for communication system approximation computing. These architectures can make use of methods like reduced-precision arithmetic, approximate memory structures, and specialized approximation units to make approximate computation in communication devices effective and high-performance.
5. Co-Design of Communication and Approximation Techniques: Co-design of communication protocols, algorithms, and approximation computing techniques can be the subject of future research. It is feasible to create integrated solutions that efficiently take use of approximation opportunities by considering the specific characteristics and requirements of communication systems from the very beginning of design. Through co-design, communication systems can be improved to save energy while still achieving performance and reliability goals.
6. Considerations for Security and Reliability: AxC creates flaws and inaccuracies that may influence the Security and Reliability of Communication Systems. Future research can investigate how approximation techniques affect attack

resilience, fault tolerance, error correction systems, and system security. This study may result in the creation of reliable approximation computing methods that preserve the desired levels of security and dependability in communication networks.

9.CONCLUSION

By tackling the issues with energy efficiency while maintaining acceptable levels of performance and reliability, AxC approaches have the potential to transform communication systems. AxC allows for large energy savings without sacrificing overall system functionality by purposefully introducing controlled errors and making use of the inherent redundancy in communication chores. AxC can reduce the energy consumption of communication systems while satisfying the unique QoI needs of various applications by using approximation techniques in communication protocols, signal processing, compression, and other crucial areas. This may lead to increased scalability, lower operational costs, and longer battery life for wireless devices. Additionally, the development of hardware designs, algorithms, and communication protocols together will shape the future of AxC in communication systems. Integrative solutions that effectively utilize the advantages of AxC can be produced by considering approximation opportunities from the very beginning of system design. It is possible to create dependable and energy-efficient communication solutions by ensuring that approximation computing techniques are smoothly integrated into communication systems. But there are still issues, particularly with security, dependability, and standardization. Future research should concentrate on understanding how AxC techniques affect system security, creating reliable error correction systems, and investigating standardization initiatives to ease the adoption and interoperability of AxC techniques in communication systems.

References

Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., & Ayyash, M. (2015b). Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications. IEEE Communications Surveys and Tutorials/IEEE Communications Surveys and Tutorials, 17(4), 2347–2376. https://doi.org/10.1109/comst.2015.2444095

Alam, M. (2018, October 4). Exact and Approximate Solutions for Energy Cost Optimization in Smart Homes. https://doi.org/10.22215/etd/2017-11794

Alberto, Bosio., S., Di, Carlo., Patrick, Girard., Ernesto, Sanchez., Alessandro, Savino., Lukas, Sekanina., Marcello, Traiola., Zdenek, Vasicek., Arnaud, Virazel. (2020). Design, Verification, Test and In-Field Implications of Approximate Computing Systems. doi: 10.1109/ETS48528.2020.9131557

Ali, S., Abuhmed, T., El-Sappagh, S., Muhammad, K., Alonso-Moral, J. M., Confalonieri, R., Guidotti, R., Del Ser, J., Díaz-Rodríguez, N., & Herrera, F. (2023). Explainable Artificial Intelligence (XAI): What we know and what is left to attain Trustworthy Artificial Intelligence. Information Fusion, 99, 101805. https://doi.org/10.1016/j.inffus.2023.101805

Alzoubi, Y. I., Gill, A., & Mishra, A. (2022). A systematic review of the purposes of Blockchain and fog computing integration: classification and open issues. Journal of Cloud Computing, 11(1). https://doi.org/10.1186/s13677-022-00353-y

Ansari, M. S., Mrazek, V., Cockburn, B. F., Sekanina, L., Vasicek, Z., & Han, J. (2020, February). Improving the Accuracy and Hardware Efficiency of Neural Networks Using Approximate Multipliers. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 28(2), 317–328. https://doi.org/10.1109/tvlsi.2019.2940943J. C. Anderson et al., “Environmentally controlled invasion of cancer cells by engineered bacteria,” Journal of Molecular Biology, vol. 355, no. 4, pp. 619–627, 2006.

Anžel, A., Heider, D., & Hattab, G. (2021). The visual story of data storage: From storage properties to user interfaces. *Computational and Structural Biotechnology Journal*, *19*, 4904–4918. https://doi.org/10.1016/j.csbj.2021.08.031

Arridge, S., Maass, P., Öktem, O., & Schönlieb, C. B. (2019). Solving inverse problems using data-driven models. Acta Numerica, 28, 1–174. https://doi.org/10.1017/s0962492919000059

Arsalan, Zahid, Piprani., Amjad, Ali., Adeel, A., Shah. (2023). Big Data Analytics. doi: 10.1201/9781003378822

Artiemjew, P. (2020, March 29). About Granular Rough Computing—Overview of Decision System Approximation Techniques and Future Perspectives. *Algorithms*, *13*(4), 79. https://doi.org/10.3390/a13040079

Atis, Degro., Rainald, Löhner. (2020). Simple Fault-tolerant Computing for Field Solvers. International Journal of Computational Fluid Dynamics, doi: 10.1080/10618562.2020.1773448

B. Aazhang et al., "Neural networks for multiuser detection in codedivision multiple-access communications," IEEE Transactions on Communications, vol. 40, no. 7, pp. 1212–1222, Jul 1992.

B. Atakan et al., "Body area nanonetworks with molecular communications in nanomedicine," IEEE Communications Magazine, vol. 50, no. 1, pp. 28–34, 2012.

B. Grzybowski, Chemistry in Motion: Reaction-Diffusion Systems for Micro- and Nanotechnology. Wiley, 2009.

Barone, S., Traiola, M., Barbareschi, M., & Bosio, A. (2021). Multi-Objective Application-Driven Approximate Design Method. IEEE Access, 9, 86975–86993. https://doi.org/10.1109/access.2021.3087858

Bart, Preneel. (1993). Standardization of cryptographic techniques. Lecture Notes in Computer Science,

Bentley, Jai, Dylan. (2018). A communications system.

Boche, H., & Pohl, V. (2018, April). On the Computability of System Approximations Under Causality Constraints. *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. https://doi.org/10.1109/icassp.2018.8461586

Brian, R., Gaines. (1967). Stochastic computing. doi: 10.1145/1465482.1465505

Burns, Gregory., Malsbary, Todd. (2018). Neworking internet of things (IoT) devices.

C. Gong and Z. Xu, "Channel estimation and signal detection for optical wireless scattering communication with inter-symbol interference," IEEE Transactions on Wireless Communications, vol. 14, no. 10, pp. 5326–5337, Oct 2015

Cao, Y., Yu, W., Ren, W., & Chen, G. (2012). An Overview of Recent Progress in the Study of Distributed Multi-Agent Coordination. *IEEE Transactions on Industrial Informatics*.

Choi, J., & Venkataramani, S. (2018, December 6). Approximate Computing Techniques for Deep Neural Networks. *Approximate Circuits*, 307–329. https://doi.org/10.1007/978-3-319-99322-5_15

D. Kingma and J. Ba, "Adam: A method for stochastic optimization," arXiv preprint arXiv: 1412.6980, 2014

Damsgaard, H. J., Ometov, A., Mowla, M. M., Flizikowski, A., & Nurmi, J. (2023). Approximate computing in B5G and 6G wireless systems: A survey and future outlook. Computer Networks, 233, 109872. https://doi.org/10.1016/j.comnet.2023.109872

Daniel, P., Richardson., Alan, C., Lin., John, M., Pecarina. (2017). Hosting distributed databases on internet of things-scale devices. doi: 10.1109/DESEC.2017.8073855

Daniele, Cesini., Elena, Corni., Antonio, Falabella., Andrea, Ferraro., Lucia, Morganti., Enrico, Calore., Sebastiano, Fabio, Schifano., Michele, Michelotto., R., Alfieri., Roberto, De, Pietri., Tommaso, Boccali., Andrea, Biagioni., Francesca, Lo, Cicero., Alessandro, Lonardo., Michele, Martinelli., Pierluigi, Paolucci., Elena, Pastorelli., Piero, Vicini. (2017). Power-Efficient Computing: Experiences from the COSA Project. Scientific Programming, doi: 10.1155/2017/7206595

David, Gell., Kenneth, L., Stanwood. (2013). Application quality management in a communication system.

Del Ser, J., Osaba, E., Molina, D., Yang, X. S., Salcedo-Sanz, S., Camacho, D., Das, S., Suganthan, P. N., Coello, C. a. C., & Herrera, F. (2019). Bio-inspired computation: Where we stand and what's next. Swarm and Evolutionary Computation, 48, 220–250. https://doi.org/10.1016/j.swevo.2019.04.008

Dharmendra, Patel. (2019). Distributed Computing for Internet of Things (IoT). doi: 10.4018/978-1-5225-7955-7.CH004

Ding, Z., Lei, X., Karagiannidis, G. K., Schober, R., Yuan, J., & Bhargava, V. K. (2017). A Survey on Non-Orthogonal Multiple Access for 5G Networks: Research Challenges and Future Trends. IEEE Journal on Selected Areas in Communications, 35(10), 2181–2195. https://doi.org/10.1109/jsac.2017.2725519

Duc, Van, Le., Chen-Khong, Tham., Yanmin, Zhu. (2017). Quality of Information (QoI)-aware cooperative sensing in vehicular sensor networks. doi: 10.1109/PERCOMW.2017.7917590

Dutta, N., Sarma, H. K. D., Jadeja, R., Delvadia, K., & Ghinea, G. (2021). Architectures for Content Communication. *Practical Networking*, 27–50. https://doi.org/10.1007/978-3-030-46736-4_2

Dwivedi, Y. K., Hughes, L., Baabdullah, A. M., Ribeiro-Navarrete, S., Giannakis, M., Al-Debei, M. M., Dennehy, D., Metri, B., Buhalis, D., Cheung, C. M., Conboy, K., Doyle, R., Dubey, R., Dutot, V., Felix, R., Goyal, D., Gustafsson, A., Hinsch, C., Jebabli, I., . . . Wamba, S. F. (2022). Metaverse beyond the hype: Multidisciplinary perspectives on emerging challenges, opportunities, and agenda for research, practice and policy. International Journal of Information Management, 66, 102542. https://doi.org/10.1016/j.ijinfomgt.2022.102542

E. Nachmani et al., "Learning to decode linear codes using deep learning," in 54th Annual Allerton Conference on Communication, Control, and Computing (Allerton), Sept 2016.

Giuliano, R., Mazzenga, F., Innocenti, E., Fallucchi, F., & Habib, I. (2021, October 16). Communication Network Architectures for Driver Assistance Systems. *Sensors*, *21*(20), 6867. https://doi.org/10.3390/s21206867

Gonçalo, Carvalho., Bruno, Cabral., Vasco, Pereira., Jorge, Bernardino. (2021). Edge computing: current trends, research challenges and future directions. Computing, doi: 10.1007/S00607-020-00896-5

Goswami, M., Panda, N., Mohanty, S., & Pattnaik, P. K. (2023, April 11). Machine Learning Techniques and Routing Protocols in 5G and 6G Mobile Network Communication System - An Overview. *2023 7th International Conference on Trends in Electronics and Informatics (ICOEI)*. https://doi.org/10.1109/icoei56765.2023.10125697

Gupta, S., Leszkiewicz, A., Kumar, V., Bijmolt, T., & Potapov, D. (2020). Digital Analytics: Modeling for Insights and New Methods. Journal of Interactive Marketing, 51, 26–43. https://doi.org/10.1016/j.intmar.2020.04.003

HaCohen-Kerner, Y., Stern, I., Korkus, D., & Fredj, E. (2007). AUTOMATIC MACHINE LEARNING OF KEYPHRASE EXTRACTION FROM SHORT HTML DOCUMENTS WRITTEN IN HEBREW. Cybernetics and Systems, 38(1), 1–21. https://doi.org/10.1080/01969720600998546

Hajek, K. (2018, July). System of Standard Approximations for Optimum Frequency Filter Design. *2018 15th International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design (SMACD)*. https://doi.org/10.1109/smacd.2018.8434909

Hallegatte, S., Rentschler, J., & Rozenberg, J. (2019). Lifelines: The Resilient Infrastructure Opportunity. https://doi.org/10.1596/978-1-4648-1430-3

Herrero, P. I. (2022, October 10). On-Chip Thermal Monitoring: Design, Placement and Interconnection of Temperature Sensors. https://doi.org/10.20868/upm.thesis.14709

Hideo, Yoshimi., Nobuyuki, Enomoto., Zhenlong, Cui., Kazuo, Takagi., Atsushi, Iwata. (2007). Communication system, communication method, and program.

Hin, , Liu. (2022). Error-Tolerant Mapping for Quantum Computing. doi: 10.1007/978-981-16-7487-7_12

Hrishav, Bakul, Barua., Kartick, Chandra, Mondal. (2019). Approximate Computing: A Survey of Recent Trends—Bringing Greenness to Computing and Communication. Journal of The Institution of Engineers : Series B, doi: 10.1007/S40031-019-00418-8

Idrees, M., Maqbool, M. M., Bhatti, M. K., Rahman, M. M. U., Hafiz, R., & Shafique, M. (2021). An approximate-computing empowered green 6G downlink. Physical Communication, 49, 101444. https://doi.org/10.1016/j.phycom.2021.101444

J. Vafaei, O. Akbari, M. Shafique and C. Hochberger, "X-Rel: Energy-Efficient and Low-Overhead Approximate Reliability Framework for Error-Tolerant Applications Deployed in Critical Systems," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 31, no. 7, pp. 1051-1064, July 2023, doi: 10.1109/TVLSI.2023.3272226.

Jiménez, F. Technologiesand Applications of Communicationsin Road Transport. Appl. Sci. 2022, 12,8997. https://doi.org/10.3390/app12188997

Jorge, Nereu., Ana, de, Almeida., Jorge, Bernardino. (2017). Big Data Analytics: A Preliminary Study of Open Source Platforms.. doi: 10.5220/0006470104350440

K., Koteswara, Rao., G., Svp, Raju. (2011). An Overview on Soft Computing Techniques. doi: 10.1007/978-3-642-22577-2_2

Kebin, Shi., Yonghui, Dai., Jing, Xu. (2020). Construction of a Security Vulnerability Identification System Based on Machine Learning. Journal of Sensors, doi: 10.1155/2020/7358692

Khan, R., Kumar, P., Jayakody, D. N. K., & Liyanage, M. (2020). A Survey on Security and Privacy of 5G Technologies: Potential Solutions, Recent Advancements, and Future Directions. IEEE Communications Surveys and Tutorials/IEEE Communications Surveys and Tutorials, 22(1), 196–248. https://doi.org/10.1109/comst.2019.2933899

Khan, S., Wilson, A., & Hussain, F. K. (2022). Contextual information aware optimal communication in radio networks in considering the pervasive computing -A literature review. Procedia Computer Science, 203, 127–134. https://doi.org/10.1016/j.procs.2022.07.018

Kranz, G. (2008). *Fault-tolerant*. Retrieved from https://www.techtarget.com/searchdisasterrecovery/definition/fault-tolerant

Krishna, G. (n.d.). *Introduction to Communication Systems*. Retrieved from Engineerstutor: https://engineerstutor.com/2018/07/31/introduction-to-communication-systems/

Kugler, L. (2015). Is" good enough" computing good enough?." Communications of the ACM. 12-14.

Kugler, L.: Is 'good enough' computing good enough? 2015. [Online; visited 11.02.2020]. Retrieved from: https://cacm.acm.org/news/is-good-enough-computing-good-enough/

Lempert, R., Popper, S., & Bankes, S. (2003). Shaping the Next One Hundred Years: New Methods for Quantitative, Long-Term Policy Analysis. In RAND Corporation eBooks. https://doi.org/10.7249/mr1626

Leng, Jingwen & Buyuktosunoglu, Alper & Bertran, Ramon & Bose, Pradip & Janapa Reddi, Vijay. (2015). Safe limits on voltage reduction efficiency in GPUs: a direct measurement approach. 294-307. 10.1145/2830772.2830811.

Li, Q., Shi, L., Yang, J., Zhang, Y., & Xue, C. J. (2019, June 2). Leveraging Approximate Data for Robust Flash Storage. *Proceedings of the 56th Annual Design Automation Conference 2019*. https://doi.org/10.1145/3316781.3317848

Ling Wang, X. W. (2022). Power-Efficient Network-on-Chips: Design and Evaluation. *Advances in Computers*.

Lisova, E., Broux, R., Denil, J., Bucaioni, A., & Mubeen, S. (2022, November 16). Communication Patterns for Evaluating Vehicular E/E Architectures. *2022 International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME)*. https://doi.org/10.1109/iceccme55909.2022.9987747

Liu, F., Cui, Y., Masouros, C., Xu, J., Han, T. X., Eldar, Y. C., & Buzzi, S. (2022). Integrated Sensing and Communications: Toward Dual-Functional Wireless Networks for 6G and Beyond. IEEE Journal on Selected Areas in Communications, 40(6), 1728–1767. https://doi.org/10.1109/jsac.2022.3156632

Malak, Michael., Rivas, Luis, E., Kreider, Mark, L., Philip, Ogren., Oberbreckling, Robert, James. (2019). Data standardization techniques.

Mao, Y., You, C., Zhang, J., Huang, K., & Letaief, K. B. (2017). A Survey on Mobile Edge Computing: The Communication Perspective. IEEE Communications Surveys and Tutorials/IEEE Communications Surveys and Tutorials, 19(4), 2322–2358. https://doi.org/10.1109/comst.2017.2745201

Marcello, Traiola., Alessandro, Savino., Stefano, Di, Carlo. (2019). Probabilistic estimation of the application-level impact of precision scaling in approximate computing applications. Microelectronics Reliability, doi: 10.1016/J.MICROREL.2019.06.002

Marcinkowska, H. (1987). On Galerkin approximations of parabolic equations in time dependent domains. *Applicationes Mathematicae*, *19*(2), 289–307. https://doi.org/10.4064/am-19-2-289-307

Mario, Barbareschi., Federico, Iannucci., Antonino, Mazzeo. (2016). Automatic Design Space Exploration of Approximate Algorithms for Big Data Applications. doi: 10.1109/WAINA.2016.172

Maryam, Idrees., Mohammed, Manzar, Maqbool., Muhammad, Khurram, Bhatti., Muhammad, Mahboob, Ur, Rahman., Rehan, Hafiz., Muhammad, Shafique. (2021). An approximate-computing empowered green 6G downlink. Physical Communication, doi: 10.1016/J.PHYCOM.2021.101444

Max, Buot. (2006). Probability and Computing: Randomized Algorithms and Probabilistic Analysis. Journal of the American Statistical Association, doi: 10.1198/JASA.2006.S74

Mazen, Al, Haddad., Magdy, Bayoumi. (2015). Green energy solution for femtocell power control in massive deployments. doi: 10.1109/ICEAC.2015.7352163

Mcquillan, D., Schieferdecker, I., Großmann, J., & Schneider, M. (2019). The Democratization of Artificial Intelligence. In AI critique. https://doi.org/10.14361/9783839447192

Meng, T., Jing, X., Yan, Z., & Pedrycz, W. (2020). A survey on machine learning for data fusion. Information Fusion, 57, 115–129. https://doi.org/10.1016/j.inffus.2019.12.001

Mike, Kong., Terence, H., Dineen., Paul, J., Leach., Elizabeth, A., Martin., Nathaniel, W., Mishkin., Joseph, N., Pato., Geoffrey, L., Wyant. (1990). Network Computing System Reference Manual.

Miomandre, H., Nezan, J. F., & Manard, D. (2022, July). Design Space Exploration for Memory-Oriented Approximate Computing Techniques. *2022 IEEE 33rd International Conference on Application-Specific Systems, Architectures and Processors (ASAP)*. https://doi.org/10.1109/asap54787.2022.00028

Mourtzis, D., Angelopoulos, J., & Panopoulos, N. (2022). A Literature Review of the Challenges and Opportunities of the Transition from Industry 4.0 to Society 5.0. Energies, 15(17), 6276. https://doi.org/10.3390/en15176276

N. Farsad et al., "Capacity of molecular channels with imperfect particleintensity modulation and detection," in IEEE International Symposium on Information Theory (ISIT), June 2017, pp. 2468–2472

N. Hayasaka and T. Ito, "Channel modeling of nondirected wireless infrared indoor diffuse link," Electronics and Communications in Japan (Part I: Communications), vol. 90, no. 6, pp. 9–19, 2007.

Naman, Saraf., Kia, Bazargan. (2016). Polynomial Arithmetic Using Sequential Stochastic Logic. doi: 10.1145/2902961.2902981

Nicola, Elia., Jing, Wang., Andalam, Satya, Mohan, Vamsi. (2014). Computing over Unreliable Communication Networks. doi: 10.1007/978-3-319-02150-8_8

Noel, Adam & Cheung, Karen & Schober, Robert. (2014). Diffusive Molecular Communication with Disruptive Flows. Proc. IEEE ICC. 10.1109/ICC.2014.6883880.

Nong, Ye. (2008). Secure Computer and Network Systems: Modeling, Analysis and Design.

Ometov, A., & Nurmi, J. (2022, March 14). Towards Approximate Computing for Achieving Energy vs. Accuracy Trade-offs. 2022 Design, Automation & Test in Europe Conference & Exhibition (DATE). https://doi.org/10.23919/date54114.2022.9774538

Othmar, Koch. (2008). Approximation of Meanfleld Terms in MCTDHF Computations by H-Matrices.

Overview of Communication Systems. (2021, January 19). *Communication Engineering Principles*, 1–56. https://doi.org/10.1002/9781119765448.ch1

Pooja Choudhary, L. B. (2022). Approximating Arithmetic Circuits for IoT Devices Data Processing . *Computers & Industrial Engineering, 174*, 108792.

Rasheed, A., San, O., & Kvamsdal, T. (2020). Digital Twin: Values, Challenges and Enablers From a Modeling Perspective. IEEE Access, 8, 21980–22012. https://doi.org/10.1109/access.2020.2970143

Roohum Jegan, R. J. (2023). Chapter 7 - Windowed modified discrete cosine transform based textural descriptor approach for voice disorder detection. In *Implementation of Smart Healthcare Systems using AI, IoT, and Blockchain* (pp. 147-167). doi:https://doi.org/10.1016/B978-0-323-91916-6.00012-6

Roohum Jegan, R. J. (2023). Chapter 7 - Windowed modified discrete cosine transform based textural descriptor approach for voice disorder detection. In Implementation of Smart Healthcare Systems using AI, IoT, and Blockchain (pp. 147-167). doi:https://doi.org/10.1016/B978-0-323-91916-6.00012-6

Russo, E., Palesi, M., Patti, D., Lahdhiri, H., Monteleone, S., Ascia, G., & Catania, V. (2022, May). Combined Application of Approximate Computing Techniques in DNN Hardware Accelerators. *2022 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*. https://doi.org/10.1109/ipdpsw55747.2022.00013

S. Slomovic et al., “Synthetic biology devices for in vitro and in vivo diagnostics,” Proceedings of the National Academy of Sciences, vol. 112, no. 47, pp. 14 429–14 435, Nov. 2015.

Sampson, A. (2015). Hardware and software for approximate computing. Retrieved from http://hdl.handle.net/1773/33693

Sampson, A., Nelson, J., Strauss, K., & Ceze, L. (2013). Approximate storage in solid-state memories. https://doi.org/10.1145/2540708.2540712

Sankardas, Roy., Mauro, Conti., Sanjeev, Setia., Sushil, Jajodia. (2008). Securely computing an approximate median in wireless sensor networks. doi: 10.1145/1460877.1460885

Shen, Y., Zhang, J., & Letaief, K. B. (2022, May 16). How Neural Architectures Affect Deep Learning for Communication Networks? *ICC 2022 - IEEE International Conference on Communications*. https://doi.org/10.1109/icc45855.2022.9839205

Sherman, J. A., Arissian, L., Brown, R. C., Deutch, M. J., Donley, E. A., Gerginov, V., Levine, J., Nelson, G. K., Novick, A. N., Patla, B. R., Parker, T. E., Stuhl, B. K., Sutton, D. D., Yao, J., Yates, W. C., Zhang, V., & Lombardi, M. A. (2021). A resilient architecture for the realization and distribution of coordinated universal time to critical infrastructure systems in the United States : https://doi.org/10.6028/nist.tn.2187

Shunsuke, Fukami., William, A., Borders., Ahmed, Zeeshan, Pervaiz., Kerem, Y., Camsari., Supriyo, Datta., Hideo, Ohno. (2020). Probabilistic computing based on spintronics technology. doi: 10.1109/SNW50361.2020.9131622

Siyuan, Xu., Benjamin, Carrion, Schafer. (2017). Exposing Approximate Computing Optimizations at Different Levels: From Behavioral to Gate-Level. IEEE Transactions on Very Large Scale Integration Systems, doi: 10.1109/TVLSI.2017.2735299

Stefano, Cherubin., Giovanni, Agosta. (2020). Tools for Reduced Precision Computation: A Survey. ACM Computing Surveys, doi: 10.1145/3381039

Swagath, Venkataramani., Srimat, Chakradhar., Kaushik, Roy., Anand, Raghunathan. (2015). Computing approximately, and efficiently. doi: 10.5555/2755753.2755924

Sze, V., Chen, Y. H., Yang, T. J., & Emer, J. S. (2017). Efficient Processing of Deep Neural Networks: A Tutorial and Survey. Proceedings of the IEEE, 105(12), 2295–2329. https://doi.org/10.1109/jproc.2017.2761740

T. Danino et al., “Programmable probiotics for detection of cancer in urine,” Science Translational Medicine, vol. 7, no. 289, pp. 289ra84– 289ra84, 2015.

Tamizhelakkiya, K., Shailesh, G., Singh, M., & Pushkar. (2021). An Overview of Communication System Model Using Deep Learning Techniques. *Advances in Power Systems and Energy Management*, 299–309. https://doi.org/10.1007/978-981-15-7504-4_29

Tataria, H., Shafi, M., Molisch, A. F., Dohler, M., Sjoland, H., & Tufvesson, F. (2021). 6G Wireless Systems: Vision, Requirements, Challenges, Insights, and Opportunities. Proceedings of the IEEE, 109(7), 1166–1199. https://doi.org/10.1109/jproc.2021.3061701

Teimoori, M. T., Hanif, M. A., Ejlali, A., & Shafique, M. (2018, March). AdAM: Adaptive approximation management for the non-volatile memory hierarchies. *2018 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. https://doi.org/10.23919/date.2018.8342113

Tseliou, G. (2023, October 11). Network virtualization in next generation cellular networks. https://doi.org/10.5821/dissertation-2117-166962

Van Langenhove, J., Lucor, D., Alauzet, F., & Belme, A. (2018, December). Goal-oriented error control of stochastic system approximations using metric-based anisotropic adaptations. *Journal of Computational Physics*, *374*, 384–412. https://doi.org/10.1016/j.jcp.2018.07.044

Veers, P., Bottasso, C. L., Manuel, L., Naughton, J., Pao, L., Paquette, J., Robertson, A., Robinson, M., Ananthan, S., Barlas, T., Bianchini, A., Bredmose, H., Horcas, S. G., Keller, J., Madsen, H. A., Manwell, J., Moriarty, P., Nolet, S., & Rinker, J. (2023). Grand challenges in the design, manufacture, and operation of future wind turbine systems. Wind Energy Science, 8(7), 1071–1131. https://doi.org/10.5194/wes-8-1071-2023

Venugopal, Govindaraju., Vijay, V., Raghavan., Calyampudi, Radhakrishna, Rao. (2015). Big data analytics.

Wang, S., Qureshi, M. A., Miralles-Pechuán, L., Huynh-The, T., Gadekallu, T. R., & Liyanage, M. (2024). Explainable AI for 6G Use Cases: Technical Aspects and Research Challenges. IEEE Open Journal of the Communications Society, 1. https://doi.org/10.1109/ojcoms.2024.3386872

Wang, X., Han, Y., Leung, V. C., Niyato, D., Yan, X., & Chen, X. (2020). Convergence of Edge Computing and Deep Learning: A Comprehensive Survey. IEEE Communications Surveys and Tutorials/IEEE Communications Surveys and Tutorials, 22(2), 869–904. https://doi.org/10.1109/comst.2020.2970550

Weiqiang, Liu., Fabrizio, Lombardi., Michael, Schulte. (2020). Approximate Computing: From Circuits to Applications [Scanning the Issue]. doi: 10.1109/JPROC.2020.3033361

Wikipedia. (2023, May 2). *Approximate computing*. Retrieved from https://en.wikipedia.org/wiki/Approximate_computing

Wong, W. F., Roy, P., Ray, R., & Ho, N. M. (2018, December 6). Compilation and Other Software Techniques Enabling Approximate Computing. *Approximate Circuits*, 443–463. https://doi.org/10.1007/978-3-319-99322-5_22

Xi Chen, J. T. (2020). Review of Unmanned Aerial Vehicle Swarm Communication Architectures and Routing Protocols. 3661. doi:https://doi.org/10.3390/app10103661

Xu, J., & Zheng, Z. (2023, May). Gradient-Based Simulation Optimization Algorithms via Multi-Resolution System Approximations. *INFORMS Journal on Computing*, *35*(3), 633–651. https://doi.org/10.1287/ijoc.2023.1279

Y. Is¸ık and N. Tas¸pınar, “Multiuser detection with neural network and pic in cdma systems for awgn and rayleigh fading asynchronous channels,” Wireless Personal Communications, vol. 43, no. 4, pp. 1185–1194, 2007.

Y. LeCun et al., “Deep learning,” Nature, vol. 521, no. 7553, pp. 436–444, May 2015. [Online]. Available: http://dx.doi.org/10.1038/nature14539.

Y. Moritani et al., “Molecular communication for health care applications,” in Proc. of 4th Annual IEEE International Conference on Pervasive Computing and Communications Workshops, Pisa, Italy, 2006, p. 5.

Yavuz, Bogaç, Türkogullari., Necati, Aras., I, K, Altınel., Cem, Ersoy. (2010). Optimal placement, scheduling, and routing to maximize lifetime in sensor networks. Journal of the Operational Research Society, doi: 10.1057/JORS.2008.187

Yi, Wang., Karim, Shahbazi., Hao, Zhang., Kwang-Il, Oh., Jae-Jin, Lee., Seok-Bum, Ko. (2019). Efficient spiking neural network training and inference with reduced precision memory and computing. Iet Computers and Digital Techniques, doi: 10.1049/IET-CDT.2019.0115

Yoshinori, Ono., Kimiyoshi, Usami. (2019). Approximate Computing Technique Using Memoization and Simplified Multiplication. doi: 10.1109/ITC-CSCC.2019.8793369

Yu, Wang. (2022). Research on the Construction of Human-Computer Interaction System Based on a Machine Learning Algorithm. Journal of Sensors, doi: 10.1155/2022/3817226

Yuko, Nagashige., Yoshikazu, Azuma., Takayuki, Kunieda., Tatsuo, Miyachi. (2016). Communications system and communications method.

Z. Li and Y. Yu, "Protein Secondary Structure Prediction Using Cascaded Convolutional and Recurrent Neural Networks," arXiv: 1604.07176, 2016.

Zervakis, G., Amrouch, H., & Henkel, J. (2020). Design Automation of Approximate Circuits With Runtime Reconfigurable Accuracy. IEEE Access, 8, 53522–53538. https://doi.org/10.1109/access.2020.2981395

Zhang, C., Patras, P., & Haddadi, H. (2019b). Deep Learning in Mobile and Wireless Networking: A Survey. IEEE Communications Surveys and Tutorials/IEEE Communications Surveys and Tutorials, 21(3), 2224–2287. https://doi.org/10.1109/comst.2019.2904897

Zhang, S., Cui, G., Long, Y., & Wang, W. (2021, July). Joint computing and communication resource allocation for satellite communication networks with edge computing. *China Communications*, *18*(7), 236–252. https://doi.org/10.23919/jcc.2021.07.019

Zhong-Liang, Shao., Cheng, Huang., Heng, Li. (2021). Replica selection and placement techniques on the IoT and edge computing: a deep study. Wireless Networks, doi: 10.1007/S11276-021-02793-X

Zhou, L., Pan, S., Wang, J., & Vasilakos, A. V. (2017). Machine learning on big data: Opportunities and challenges. Neurocomputing, 237, 350–361. https://doi.org/10.1016/j.neucom.2017.01.026