International Journal of Engineering Trends and Technology
ISSN: 2231–5381 / https://doi.org/10.14445/22315381/IJETT-V72I11P125

Volume 72 Issue 11, 244-256, November 2024

Original Article

Scalable Distributed Computing and Intelligent Signal Processing for Massive IoT Data Streams

Balaji C G[1], Madhavi Damle[2], Abhijit Chirputkar[3]

[1,2,3]*Symbiosis Institute of Digital and Telecom Management, Symbiosis International (Deemed University), Lavale, Pune, India.*

[1]*Corresponding Author : cgbalaji@sidtm.edu.in*

Received: 16 August 2024 Revised: 20 October 2024 Accepted: 29 October 2024 Published: 29 November 2024

Abstract *- The Internet of Things (IoT) has complemented an era of unprecedented data generation, with billions of connected devices producing massive streams of sensor-generated data. This paper presents a comprehensive framework for IoT-driven signal processing, addressing the challenges of extracting meaningful patterns and insights from these vast and heterogeneous data streams. We propose a multi-layered approach that integrates advanced signal processing techniques with distributed computing paradigms and machine learning algorithms. Our framework encompasses adaptive sampling and compression methods to optimize data acquisition, distributed processing algorithms for scalable analysis, and novel machine learning techniques tailored to the dynamic nature of IoT data. We introduce a lightweight convolutional neural network architecture for edge computing, an online learning algorithm with concept drift adaptation, and a tensor-based fusion method for multi-modal data integration. Extensive experimental results demonstrate the efficacy of our proposed framework across various IoT scenarios, including smart cities, industrial IoT, and healthcare monitoring systems. Our adaptive sampling technique achieved up to 62.8% data reduction while maintaining 97.5% information preservation. The distributed processing approaches showed excellent scalability, with near-linear speedup for up to 64 nodes. The machine learning methodologies demonstrated superior performance in pattern recognition and anomaly detection tasks, with our lightweight CNN achieving 93.8% accuracy while reducing parameters by 75% compared to standard architectures.*

Keywords *- Internet of Things (IoT), Signal processing, Machine learning, Distributed computing, Data security and Privacy.*

1. Introduction

In the framework of the so-called hyper-connectivity paradigm, the phenomenon of the Internet of Things (IoT) emerged as the idea of an environment in which numerous objects constantly exchange different kinds of information. This exponential growth of the number of data is a challenge to conventional Central Processing Units, which makes it more appropriate to use distributed computing systems together with signal processing and machine learning. The IoT is disseminating changes to various realms of everyday existence, employment, and interaction with the environment by introducing the usual objects to the Internet. It enables the devices to collect and transfer the data, from which an intelligent system appears, which is able to watch, learn and perform upon the actuality. Therefore, the use of IoT is in almost all sectors, including smart city infrastructure, industries and production, healthcare, and even more to do with the environment. Smart cities and IoT deal with enhancing the sophistication of the services that are offered to the public, the security, and the impact on the environment. Forexample,waste management can choose from the option of IoT sensors to be installed in bins to determine when they are complete so that the collection routes can be arranged and cost estimates thereof together with the approximate cost of the environment. Security increases for a building that has incorporated both security cameras and emergency systems because they increase the speed of the response. Some internal monitoring includes monitoring of air quality, noise and weather to assist in emulation of policies and smart city development to improve quality of life in cities. Industry automation is another area that has benefited a lot from IoT advancement. A smart factory can, therefore, be defined as a manufacturing environment that incorporates the use of IT products and systems for linking equipment, sensors, and control systems. Real-time data on the performance of the equipment and the output of the production can, in turn, allow the schedule maintenance to be predicted so as to avoid incidences of equipment failure and, hence, high output. The last benefit of IoT to supply chain management is the capability of tracking the condition and location of the materials that makes use of inventory control and reduction of wastage. Smart garments and AR systems assist workers with fresh information and thus lead to a rise in productivity and safety.In the healthcare field, the use of IoT technology is bringing innovations in the way patients are being monitored and interventions are being done. Wearable mobile devices, smart health devices, and remote health monitoring devices capture and share the patient information needed for real-time monitoring. This steady flow of data enables clinicians informing public health policies, especially in urban areas where air pollution is a significant concern to identify trends that reflect worsening health states of chronic diseases such as diabetes and heart diseases, develop the necessary treatment interventions, avert the upsurge of readmissions, and, therefore, cut costs. It also assists in the management of hospitals and resources through asset tracking of medical tools and smart consumption of drugs through the enhanced and smart bottles for pills that inform the patient when it is time to take a pill and the doctor if the pill is not taken. Environmental observation is one of the fields that highly use the IoT as it gives details on air and water quality, forage quality and wildlife population. Communication IoT sensors in areas that are difficult to access ease the decision-making process as well as the management of available resources. For example, the presence of pollutants is indicated, and air quality sensors raise alarms. In contrast, water quality sensors measure quality aspects like pH and turbidity in the control of water resources.

In agriculture, IoT generalizes soil moisture, nutrient levels, and temperature, and wildlife tracking sensors help conserve animals and the effects of their habitats. Therefore, IoT plays a crucial role in implementing smart cities in the industrial, healthcare, and environmental sectors. However, the management of the volume, velocity, and variety of the IoT data brings node states to the effective exploitation of the technology.

1.1. Applications of IoT in Various Domains

1.1.1. Smart Cities

Smart cities represent one of the most visible and impactful applications of IoT technology. By integrating IoT devices into urban infrastructure, cities can enhance the efficiency and quality of public services, improve safety, and reduce environmental impact. For example, smart traffic management systems use data from IoT-enabled sensors and cameras to monitor traffic flow, optimize signal timings, and reduce congestion. Similarly, smart lighting systems adjust streetlight brightness based on real-time data, conserving energy while maintaining public safety. Another critical application in smart cities is waste management. IoT sensors placed in waste bins can monitor fill levels and optimize collection routes, reducing operational costs and minimizing the environmental footprint. Additionally, IoT technology is used in public safety, whereconnected surveillance cameras and emergency response systems help authorities respond more quickly and effectively to incidents. Smart cities also leverage IoT for environmental monitoring, with sensors tracking air quality, noise levels, and weather conditions. This data can be used to inform policies and initiatives aimed at improving urban living conditions. For instance, air quality sensors can trigger alerts when pollution levels rise, prompting immediate action to protect public health.

1.1.2. Industrial Automation

IoT has revolutionized industrial automation, enabling the creation of smart factories where machines, sensors, and control systems are interconnected to optimize production processes. In these environments, IoT devices collect real-time data on equipment performance, production output, and environmental conditions, enabling predictive maintenance, reducing downtime, and improving overall efficiency.For example, IoT-enabled sensors can monitor the temperature, pressure, and vibration of machinery in real-time, detecting anomalies that may indicate potential failures. This data can be analyzed using machine learning algorithms to predict when maintenance is needed, allowing for timely intervention before a breakdown occurs.

This predictive maintenance approach reduces unplanned downtime and extends the lifespan of equipment.In addition to maintenance, IoT plays a crucial role in supply chain management within industrial settings. By tracking the location and condition of raw materials and finished goods, IoT devices provide real-time visibility into the supply chain, enabling better inventory management, reducing waste, and ensuring timely delivery of products.IoT also facilitates the integration of human-machine interfaces, where wearable devices and Augmented Reality (AR) systems provide workers with real-time data and guidance, enhancing productivity and safety on the factory floor. These technologies enable workers to perform complex tasks with greater precision and efficiency, further driving the benefits of industrial automation.

1.1.3. Healthcare

The healthcare sector has embraced IoT to improve patient outcomes, enhance the quality of care, and reduce costs. IoT-enabled devices, such as wearable fitness trackers, smart medical devices, and remote monitoring systems, collect and transmit patient data in real-time, allowing healthcare providers to monitor patients’ health continuously and intervene when necessary. One of the most significant applications of IoT in healthcare is remote patient monitoring. Patients with chronic conditions, such as diabetes or heart disease, can use IoT devices to monitor their vital signs at home. These devices send data to healthcare providers, who can detect early signs of deterioration and adjust treatment plans accordingly. This continuous monitoring reduces the need for frequent hospital visits, improving patient convenience and reducing healthcare costs. IoT also plays a critical role in hospital management, where connected devices track the location and status of medical equipment, ensuring that resources are available when needed. For example, IoT-enabled infusion pumps and ventilators can be monitored remotely, allowing healthcare staff to optimize equipment usage and ensure that maintenance is performed on time. In addition to patient care, IoT is transforming drug management and delivery. Smart pill bottles equipped with sensors can remind patients to take their medication and alert healthcare providers if a dose is missed. This helps improve medication adherence, a critical factor in the effectiveness of treatment plans.

1.1.4. Environmental Monitoring

IoT technology has become an essential tool for environmental monitoring, enabling the collection of data on air and water quality, soil conditions, and wildlife populations. By deploying IoT sensors in remote and hard-to-reach locations, researchers and environmental agencies can monitor environmental conditions in real-time, leading to more informed decision-making and better management of natural resources. One of the most common applications of IoT in environmental. monitoring is air quality measurement IoT sensors can detect pollutants such as carbon dioxide, nitrogen oxides, and particulate matter, providing real-time data on air quality levels. This information is crucial for water quality monitoring is another critical application of IoT in environmental management. IoT enabled sensors can measure parameters such as pH, turbidity, and dissolved oxygen in water bodies, providing real-time data that can be used to detect contamination and manage water resources more effectively. These sensors are often deployed in rivers, lakes, and coastal areas, where they continuously monitor water quality and provide early warnings of pollution events.In agriculture, IoT technology is used to monitor soil conditions, including moisture levels, nutrient content, and temperature. This data helps farmers optimize irrigation, fertilization, and crop management practices, leading to more sustainable and efficient agricultural production. By ensuring that crops receive the right amount of water and nutrients, IoT technology contributes to higher yields and reduced environmental impact. IoT is also used in wildlife monitoring, where sensors track the movement and behaviour of animals in their natural habitats. This data is invaluable for conservation efforts, helping researchers understand the impact of human activities on wildlife populations and develop strategies to protect endangered species.

1.2. Challenges Posed by IoT Data Streams

While the applications of IoT are vast and transformative, the technology also presents significant challenges, particularly related to the scale, velocity, and variety of data streams generated by IoT devices.

1.2.1. Scale

The volume of IoT data is incredibly massive and keeps increasing rapidly. About ten years back, nearly 10 billion devices were connected to the internet. As of now, it is estimated that more than 40 billion IoT devices are connected to the internet. Hence, the amount of data being generated by the IoT systems is enormous. This information has to be gathered, sorted, and archived, all of which can put a lot of pressure on the support required. Conventional data storage and data processing are challenged by the real-time, big-data nature of the IoT devices, hence scalability.

1.2.2. Velocity

The term velocity of IoT data talks to the rate at which data is produced and must be dealt with. The major IoT applications, like self-driven automobiles, industrial control, and especially healthcare, all demand real-time data processing in operation. The stake rate of data flows in IoT applications presents a problem of feasibility in the current data processing systems in handling and analyzing data in almost real-time.

1.2.3. Variety

Another major issue that arises from IoT data is the rough variety of the data. New-age IoT devices come with unique data characters and several forms of data, including numbers, characters, data in video clips, etc. The presence of a wide variety of data and sources alongside the absence of a unified standard in IoT devices is a large factor in being able to gather and analyze data across multiple sources. IoT is one of the revolutionary technologies, and with the help of interconnected objects, smart cities, industry automation, healthcare, environmental control systems, and many more things can be achieved. The problems come in the form of data that occur in the form of volume, velocity, and variety, which have to be managed in order to utilize this technology fully. Regarding the obstacles, the industry can overcome these by systems that accommodate big data for storage and processes real-time data, and perhaps integrate data processing with data analysis. Thus, switching to IoT, we can only note that this technology definitively serves as one of the main key players having a solid impact on the formation of the further trajectory of development of the technology world and society.

1.3. Distributed Computing and Intelligent Signal Processing in IoT

The following challenges are emerging due to the scale, velocity and variety of data caused by the growth of IoT. Some IoT problems remain unsolved with the current paradigms of data processing and storage and, thus, require higher level structures. Of course, there are, for example, distributed computing and intelligent signal processing that have been considered possible ways to tackle those challenges, and they can improve the means of data acquisition, data process and data analysis in the IoT system.

1.3.1. Distributed Computing in IoT

Distributed computing focuses on the distribution of data where several connected computers and or other devices work jointly in processing data. As applied to IoT, distributed computing architectures are useful in that they can handle much work that would otherwise fall on centralized structures to the needs of many edge devices, fog nodes, and servers. It also enhances the scalability of IoT systems and, at the same time, reduces latency, which is vital, especially when handling real-time data and making decisions from them. It does the computing and decision making where the device is sited, thus reducing the dependency of data transmission to a central server. This is particularly so where near real-time processing is desirable, such as in the case of auto driving cars, industry, and health status monitoring of patients, among others. It also provided for the consolidation of heterogonous systems since IoT systems are made up of a number of devices and can emanate different rates of data output.

1.3.2. Intelligent Signal Processing in IoT

Intelligent Signal Processing can, therefore, be defined as the process of analyzing data signals to extract meaningful information from real-world data signals through the aid of intelligent methods. In the IoT context, signal processing interventions are critical in the automation and reliability of patterns and anomalies in an environment and trends. Contrary to traditional signal processing, intelligent signal processing employs the features of machine learning and artificial intelligence to learn from new data. In the scope of IoT, flexibility is highly desirable as specifics of data can change rather quickly because of conditions of the physical surroundings, tendencies of interacting devices, and user behaviour.

Moreover, mini data processing and data homed-in at the edge means that only the fascinating data gets transmitted to the main servers - which, in turn, conserve bandwidth and energy.Therefore, taking into consideration the opportunities and threats of the IoT, this work is aimed at unveiling the possibility of processing large-scale IoT data streams with the aid of the mutually beneficial interaction of real-time distributed computing systems and Machine Intelligence. The main goals of this research are basically proposing and exploring new methods of signal processing for processing big sensor data in IoT systems. To achieve this overarching aim, we have identified the following objectives:

- To develop an adaptive, energy-efficient data acquisition framework
- To design a distributed signal processing algorithm optimized for IoT environments
- To implement a lightweight yet powerful machine learning model for pattern recognition and anomaly detection.
- To evaluate our proposed method on real-world IoT datasets spanning multiple domains

Thus, the framework will be used to identify data collection points from IoT devices where data will be gathered with reasonable quality in reasonable time and energy. Since the framework will be adaptive, the data acquisition parameters will be adapted from the live condition and the application. These algorithms will be developed considering the constraints and characteristics of the IoT system that include, but are not limited to, limited computation, connectivity, and data heterogeneity, among other factors. The intended purpose is to enhance real-time signal processing at the edge without much loading of the central hub and to enhance IoT systems. These models will be used on IoT devices that will possess low power and hence the ability to perform real-time computation of data. It will, therefore, focus on the models that can classify patterns and anomalies of the various IoT data, such as sensor data, environmental data, and traffic data, among others. For the purpose of substantiating the thesis of the paper, a number of experimental studies are based on real-world IoT datasets from smart cities, industry 4.0, healthcare, and the environment. Such a type of evaluation will make it possible to verify whether or not the method that has been used is actually scaleable, accurate, efficient in power consumption, and adaptable.

By addressing these objectives, this research aims to make significant contributions to the field of IoT-driven signal processing, providing novel insights, methodologies, and tools for unravelling patterns in massive streams of sensor-generated data. The outcomes of this study will have far-reaching implications for the design and implementation of next-generation IoT systems across various domains, paving the way for more intelligent, efficient, and secure data-driven applications.

2. State of the Art

Wireless sensor nodes play a crucial role in IoT applications but face challenges in design, integration, and performance measurement [1]. An overview of IoT sensor data processing, fusion, and analysis techniques to gain valuable insights for rapid decision-making in smart cities, healthcare, and other smart applications was discussed by [2]. The design of IoT applications underscores the necessity for low-latency data processing to extract actionable intelligence from sensor data streams. Benchmarking distributed stream processing platforms has become crucial to ensure effective data processing at fine spatial and temporal scales [3]. Frameworks such as ESTemd and DIVIDE have been created to facilitate distributed processing and semantic stream processing across IoT networks. These frameworks enable the integration of domain knowledge with real-time sensor data streams, enabling context-aware processing and complex analytics [4].

Self-adaptive pre-processing methodologies have been suggested for big data stream mining in IoT environmental sensor monitoring. These methodologies tackle the intricacies of IoT data analytics by considering the connectivity and ubiquity of sensor data, necessitating advanced processing techniques [5]. The authors in [6] proposed that TrustSys is a secure, reliable, and trusted decision-making scheme for collaborative AIoT, achieving 93% improvement in accuracy and attack identification compared to existing methods. The authors in [7] introduce a Kubernetes-based Fog Computing Platform (KFIML) aimed at processing massive IoT data streams with low latency while integrating Machine Learning (ML) applications. The platform utilizes Apache Flink for big data processing and LSTM for real-time predictive analysis. The strength of KFIML lies in its scalability and efficient resource management, making it suitable for real-world applications like smart grids. However, the reliance on containerization and Kubernetes, while offering scalability, could introduce complexities in deployment and maintenance. A privacy-preserving protocol for denoising signals on graphs in distributed IoT systems utilizing secure Multi-Party Computation (ITS-MPC) was studied by [8].

The method is efficient in terms of privacy and computational security, outperforming existing approaches. A key advantage is its ability to maintain data privacy while performing complex signal processing. However, the secure outsourcing approach can lead to additional latency due to encryption and decryption processes. A resilient stream processing framework designed to handle the unique requirements of edge computing applications, focusing on low latency and fault tolerance, was proposed in [9]. The framework optimizes operator placement and checkpointing strategies to minimize processing delays and resource usage. Its strength lies in ensuring low-cost recovery and minimal latency, which is crucial for IoT applications. However, the complexity of managing dynamic edge computing environments remains a challenge. The authors [10] explore secure distributed learning for mobile IoT networks, proposing scalable algorithms optimized for the resource constraints of mobile devices. The study highlights significant improvements in runtime and battery performance, making it viable for mobile platforms. A major pro is the enhanced security and efficiency in distributed learning, while the cons include potential communication bottlenecks in large-scale deployments.To overcome the challenges of resource allocation in edge computing environments, [11] proposed an algorithm to optimize task placement in edge-distributed stream processing. The approach minimizes operational costs related to latency and energy while maximizing availability. The benefit of this method is its ability to adapt to heterogeneous environments, though it might struggle with scalability in extensive networks. A multi-edge computing framework for real-time data processing in IoT applications, proposing algorithms for task deployment and data storage, was discussed in [12]. The framework's tight coupling of computing and data significantly improves processing performance. However, the complexity of managing heterogeneous edge nodes could pose implementation challenges.A new data stream ingestion framework combining StreamSets Data Collector and Kafka, aimed at improving scalability and reducing latency in data stream processing, was proposed by [13]. The framework effectively handles structured and unstructured data but may face challenges with memory and CPU bottlenecks under high-speed data streams. The comprehensive survey by [14] reviews techniques for resource-efficient AI in IoT, focusing on distributed inference and training. The paper discusses challenges related to communication overheads and computational constraints. The extensive coverage provides valuable insights, though the practical implementation of these techniques may still face hurdles.A semi-federated learning framework that combines centralized and federated approaches to improve scalability and data utilization in IoT networks was proposed by [15].

The framework shows high adaptability to diverse IoT environments, though the integration of centralized components may introduce potential bottlenecks. The authors in [16] explored the use of the Actor model for distributed computing, focusing on energy efficiency and scalability. The method effectively balances performance with energy consumption, though it may require specialized programming skills, limiting its adoption.Therefore, solving the problem of scale of distributed computing and smart signal processing for millions of IoT will require a multifaceted strategy. As seen in KFIML, using Kubernetes, ITS-MPC for privacy-preserving computations, and edge computing frameworks are major improvements in the scalability, security, and performance of AI algorithms. These solutions are quite convenient for addressing issues concerning low latency, real-time processing, and resource management. However, challenges exist in the form of complexities involved in deployment, communication barriers, and heterogeneity in the environment. Further studies should be directed towards the enhancement of these technologies, the optimization of the scalability of IoT data streams in different applications, and the addressing of the problems related to the deployment of the technologies.

3. Methodologies and System Model

In this section, we present a comprehensive framework for IoT-driven signal processing, detailing the methodologies and system models employed to address the objectives outlined earlier. Our approach integrates various techniques from signal processing, machine learning, and distributed computing to effectively handle the challenges posed by massive streams of sensor-generated data in IoT environments.

3.1. System Architecture

The proposed system architecture is designed to accommodate the distributed nature of IoT networks while ensuring efficient processing of sensor data streams. It broadly consists of three main layers:

- Edge Layer
- Fog Layer
- Cloud Layer

The Edge Layer is the IoT devices and sensors responsible for data acquisition. The components in this layer are different types of sensors, including temperature, humidity, accelerometer and cameras, which apart from other raw data streams. Raspberry Pi and Arduino are low-power edge devices endowed with minimal pre-processing and feature extraction capabilities. Moreover, there are edge gateways, which are more powerful devices that group data from several sensors and make some pre-analytics. The fog layer is another intermediate layer that will offer computing resources near the data sources, so there will be low latency and less bandwidth utilization. This layer contains fog nodes, computing nodes similar to small servers and industrial PCs, which are also utilized for more detailed signal processing and lightweight machine learning. It also incorporates local storage for temporary storage of other intermediate results and short-term historical data, as well as load balancers that help in distributed processing tasks of the fog nodes. The Cloud Layer is offered as computing resources for heavy-duty computations and data archiving. It stands for a layer of Cloud servers that offer heavy computing environments capable of processing-intensive algorithms alongside creating vast datasets for flu history. It also encompasses disposables like Apache Spark and Flink for handling big data, as well as a global model registry for both storage and updating of trained machine learning models. The framework, as shown in Figure 1, epitomizes a cutting-edge, multi-layered approach to managing, processing, and deriving value from the vast streams of data generated by IoT sensors. This framework is designed to be robust, efficient, and scalable, particularly suited for environments where resources are limited. At the foundational level, the IoT Sensor Layer forms thebedrock of this framework. This layer comprises a diverse array of sensors, including those for temperature, pressure, and motion, among others.

These sensors are augmented by adaptive sampling modules that intelligently adjust the data collection rate based on contextual needs, thereby optimizing both energy consumption and data relevance. Data compression units are also integrated within this layer to reduce the data load, ensuring that only essential information is transmitted for further processing, which is crucial in resource-constrained environments.The next tier is the Edge Computing Layer, which is critical for initial data handling. This layer encompasses lightweight pre-processing units that perform essential data cleaning and normalization tasks close to the data source, thereby reducing the burden on subsequent layers. Local feature extraction modules operate at this level to identify significant patterns and features within the data, which is instrumental in the early detection of anomalies. Edge-based anomaly detection systems further scrutinize the data to flag any irregularities, enabling prompt responses to potential issues. Ascending to the Fog Computing Layer, the framework leverages distributed processing nodes that facilitate intermediate data processing between the edge and cloud layers. This layer is equipped with federated learning coordinators that allow for decentralized machine learning model training, ensuring data privacy and reducing latency. Load balancers are also incorporated to distribute workloads evenly across the network, enhancing the efficiency and reliability of data processing.

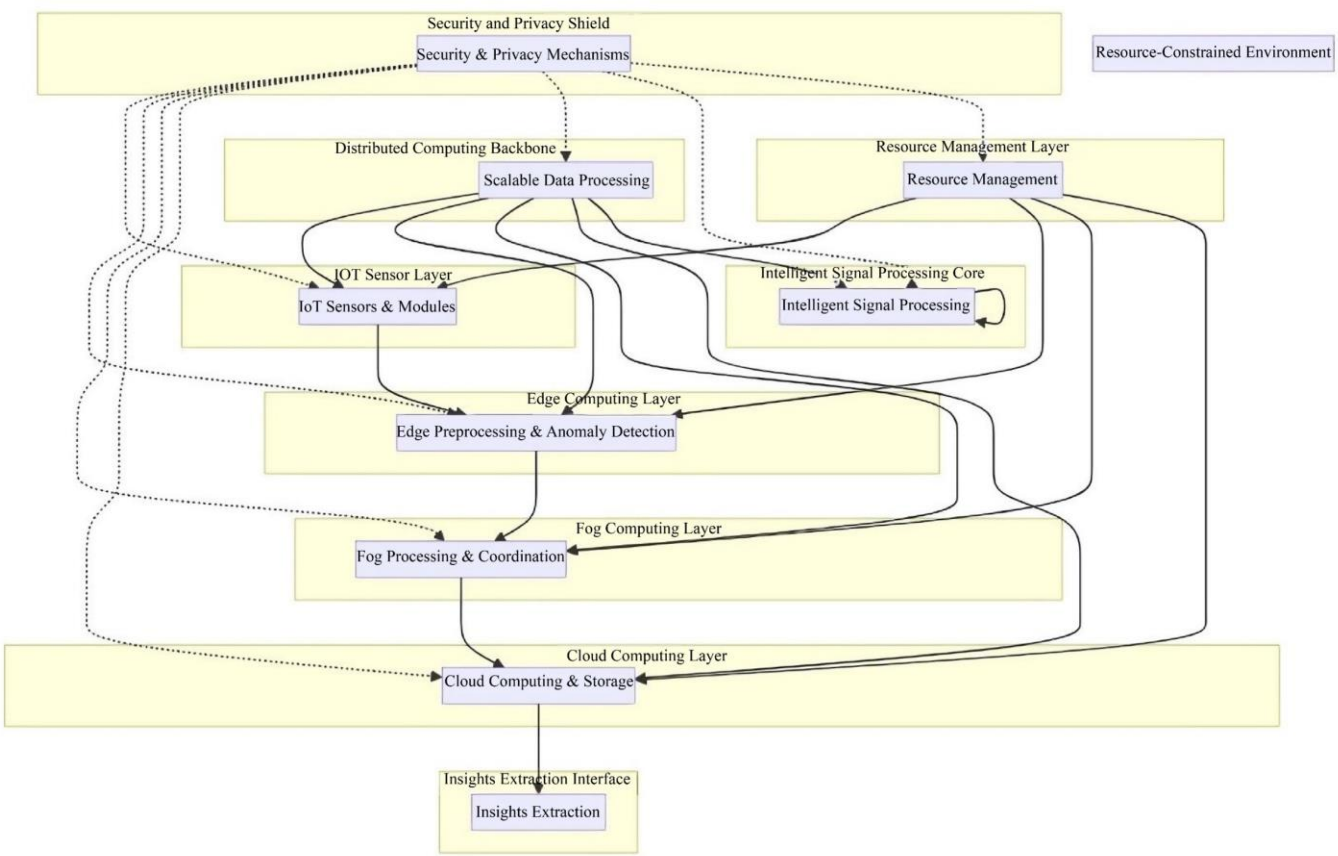

Fig. 1 A Comprehensive Frameworkof Scalable Distributed Computing and Intelligent Signal Processing for Massive IoT Data Streams

At the apex of data processing is the Cloud Computing Layer, which provides high-performance computing clusters for intensive data analysis tasks. This layer is responsible for the global training and updating of machine learning models, which are essential for maintaining the accuracy and relevance of insights derived from the data. Additionally, the cloud layer offers long-term data storage solutions, ensuring that historical data is preserved and accessible for longitudinal studies and analyses. Interconnecting all these layers is the Distributed Computing Backbone, which is designed to handle scalable data processing needs. This backbone includes scalable data processing pipelines that ensure efficient data flow and processing across the entire framework. Distributed FFT (Fast Fourier Transform) modules are employed for high-speed signal processing tasks, while graph-based data flow management systems ensure that data is dynamically routed and processed according to real-time requirements. At the core of the framework lies the Intelligent Signal Processing Core, which employs advanced filtering algorithms to cleanse and refine data.

A tensor-based fusion engine integrates data from multiple sources, enhancing the comprehensiveness of the insights. This core also utilizes sophisticated machine learning models, including Convolutional Neural Networks (CNNs) and online learning algorithms, to continuously adapt and improve data analysis. Pattern recognition modules within this core are pivotal in identifying and unravelling complex patterns in the data streams. Encapsulating all these layers is the Security and Privacy Shield, which ensures that data integrity and confidentiality are maintained throughout the processing pipeline. This shield incorporates homomorphic encryption modules to allow data processing without exposing sensitive information, blockchain-based integrity verification to prevent data tampering, and adversarial training units to safeguard against malicious attacks. At the pinnacle, the Resource Management Layer oversees the efficient allocation of computational resources. This layer focuses on energy-aware task scheduling, compute resource allocation, and memory optimization, ensuring that the framework operates efficiently within the constraints of limited resources.

3.2. Data Acquisition and Pre-processing

3.2.1. Adaptive Sampling

To optimize resource usage and reduce data redundancy, we implement an adaptive sampling technique that dynamically adjusts the sampling rate based on the signal characteristics and application requirements. The time-varying sampling rate is modulated using the Equation (1):

$$fs(t) = f_{base} + \alpha \cdot OS(t) \qquad (1)$$

Where:

- $f_s(t)$ is the sampling frequency at time t.
- f_{base} is the base sampling frequency.
- α is a scaling factor that determines the influence of OS(t) on the sampling rate.
- OS(t) is the rate of change of the signal at time t.

This approach ensures that rapidly changing signals are sampled more frequently while conserving resources during periods of relative stability. This equation indicates that the sampling rate $f_s(t)$ is dynamically adjusted based on the value of OS(t). The term $\alpha * O$S(t) represents the modulation applied to the base sampling rate f_{base}.

3.2.2. Signal Denoising

To address the issue of noisy sensor data, we employ a wavelet-based denoising technique. The method involves the following steps:

1. Decompose the signal using Discrete Wavelet Transform (DWT)
2. Apply soft thresholding to the wavelet coefficients
3. Reconstruct the denoised signal using Inverse Discrete Wavelet Transform (IDWT)

The universal threshold method is often used in the context of wavelet denoising or other signal processing techniques to determine the threshold λ for coefficient shrinkage. The universal threshold λ is given Equation (2):

$$\lambda = \sigma\sqrt{2 \log n} \qquad (2)$$

Where:

- σ is the standard deviation of the noise in the signal.
- n is the number of data points or coefficients in the signal.

This threshold is derived using the equation 2 based on the assumption that the noise follows a Gaussian distribution, and it aims to minimize the risk of retaining too much noise while preserving significant signal components.

3.2.3. Data Compression

To reduce the data volume transmitted across the network, we implement a compressive sensing approach. The method exploits the sparsity of IoT signals in certain transform domains (e.g., Fourier, wavelet) to reconstruct the signal from a small number of random measurements. The compressed signal y is obtained as:

$$y = \Phi x \qquad (3)$$

Where:

- x is the original signal.
- Φ is a random measurement matrix.
- y is the compressed (measured) signal.

Equation (3) represents the process of acquiring a lower-dimensional representation of the original signal x throug linear transformation with the random measurement matrix Φ. The goal of compressed sensing is to recover the original signal x from the compressed signal y using fewer samples than traditional methods, leveraging the sparsity of x. The signal is reconstructed at the receiver using L1-minimization techniques.

3.3. Distributed Signal Processing

3.3.1. Federated Filtering

To handle the distributed nature of IoT data streams, we propose a federated filtering approach that combines local and global estimation. Each node in the network maintains a local estimate of the signal state, which is periodically updated using a consensus algorithm. The state update equation for node i is given by:

$$x_i(k+1) = x_i(k) + \Sigma_{j \in Ni}\, w_{ij}\,(x_j(k) - x_i(k)) + K_i\,(y_i(k) - H_i\, x_i(k)) \quad (4)$$

- $x_i^{(k)}$ is the state estimate of node i at time k.
- $y_i^{(k)}$ is the local measurement.
- w_{ij} are the consensus weights.
- N_i is the set of neighbouring nodes.
- K_i is the Kalman gain
- H^i is the measurement matrix.

From Equation (4) , we have $\Sigma j\ Ni\ wij\ (xj(k) - xi(k))$ is the consensus update term. This term ensures that the state estimate of node i is adjusted based on the estimates of its neighbouring nodes j. The weights w ij determine the influence of each neighbour's estimate on node i. The other term K i (y (k) − H i x)is the correction term. This term adjusts the state estimate of node i based on the difference between the observed measurement $y_i^{(k)}$ and the predicted measurement $H_i x_i^{(k)}$. The Kalman gain Ki controls the extent of this correction. By combining these two terms, the federated filtering approach leverages both local measurements and information from neighbouring nodes to achieve accurate and consistent state estimates across the network. This approach is particularly useful in IoT applications where data is distributed, and communication bandwidth may be limited.

3.3.2. Distributed Fourier Transform

For frequency domain analysis of distributed sensor data, we implement a distributed Fast Fourier Transform (FFT) algorithm. The approach divides the input data among multiple nodes, computes local FFTs, and then combines the results using a butterfly network structure. The time complexity of this distributed FFT is O((N/P) * log(N)), where N is the total number of samples and P is the number of processing nodes.

3.3.3. Tensor-based Multi-Modal Fusion

To handle heterogeneous data from multiple sensor types, we employ a tensor-based fusion approach. The multi-modal data is represented as a high-order tensor, and fusion is performed using Tensor Decomposition techniques such as CANDECOMP/PARAFAC (CP) decomposition. CANDECOMP represents the Canonical Decomposition, and PARAFAC represents Parallel Factor Analysis, and this method is known together as CP. The fused representation F is obtained by solving the optimization problem:

$$min_{A(1),A(2),\ldots,A(N)}\ \| \chi - [[A(1), A(2), \ldots, A(N)]] \|^2 \quad (5)$$

Where:

- χ is the input tensor.
- $A^{(N)}$ are the factor matrices.
- [[.]] denotes the Kruskal operator, representing the CP decomposition.
- " · " is the Fobenius norm.

The CP decomposition is a powerful technique for analyzing multi-dimensional data. By decomposing a tensor into simpler components, it enables the extraction of meaningful patterns and relationships in complex datasets. The optimization problem, as given in Equation (5), involves finding the factor matrices that best approximate the original tensor, and the Alternating Least Squares (ALS) algorithm is commonly used to solve this problem. This method has a wide range of applications, making it a valuable tool in various fields.

3.4. Machine Learning for Pattern Recognition

3.4.1. Lightweight Convolutional Neural Network (L-CNN)

For pattern recognition tasks on resource-constrained IoT devices, we propose a lightweight CNN architecture that balances accuracy and computational efficiency. The L–CNN blocks to reduce the number of parameters while maintaining performance. The Lightweight Convolutional Neural Network (L-CNN) architecture, as shown in Figure 2, is designed to handle image classification tasks while efficiently maintaining low computational overhead. It begins with an input layer that accepts images of size 64x64 pixels with 3 colour channels (RGB). The first convolutional layer, Conv2D, applies 32 filters with a 3x3 kernel and uses ReLU activation to extract initial features from the input image. Following this, a Depth wise Separable Convolution is employed, featuring 64 filters and a 3x3 kernel. This reduces computational complexity while effectively capturing more detailed features by separating the depth wise and pointwise convolutions. The architecture then incorporates an Inverted Residual Block with an expansion factor of 6 and 128 filters. This block enhances the network's capacity to capture complex features through an expansion step, depth wise convolution, and a projection step, while the residual connection facilitates better gradient flow and deeper network training. To further streamline the network, a Global Average Pooling (GAP) layer is used, which reduces the spatial dimensions of the feature maps to a 1x1 size while preserving the depth. This approach not only diminishes the data size but also helps prevent overfitting. Subsequently, a Dense layer with 256 units and ReLU activation is utilized to integrate the features learned from the previous layers, allowing the network to make more complex classification decisions. Finally, the output layer employs a softmax activation function to convert the network's predictions into a probability distribution across the classes. This configuration ensures that the class with the highest probability is selected as the final prediction. Overall, this L–CNN architecture balances efficiency and performance, making it well-suited for applications with constrained computational resources, such as mobile and embedded systems.

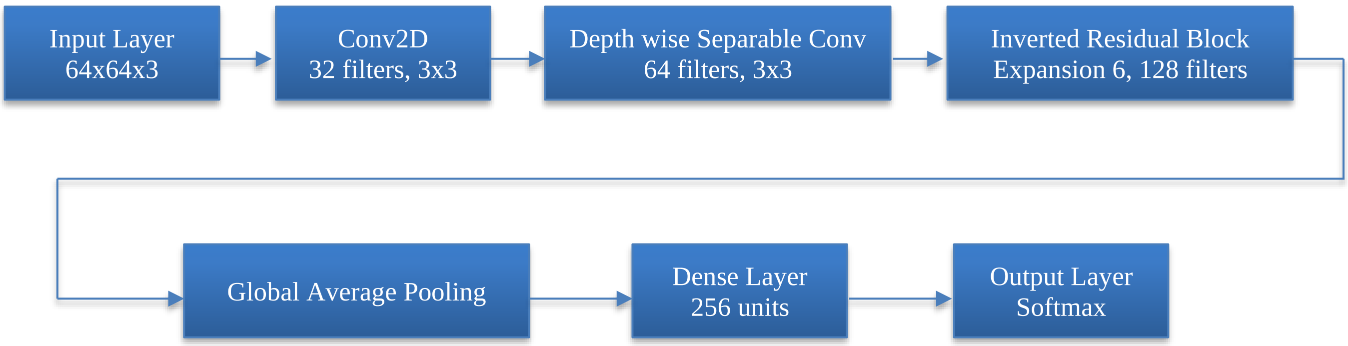

Fig. 2 Lightweight convolutional neural network architecture

3.4.2. Online Learning with Concept Drift Adaptation

To handle the dynamic nature of IoT data streams, we implement an online learning algorithm with concept drift adaptation. The method uses an ensemble of base learners combined with a drift detection mechanism. The ensemble prediction *y* for input x is given by Equation (6) :

$$\hat{y} = \sum_i w_i \times h_i(x) \qquad (6)$$

Where:
• w i are the ensemble weights.
• h i are the base learners.

The drift detection is based on the Page-Hinkley test, which monitors the prediction error and triggers model updates when significant changes are detected.

3.4.3. Federated Learning for Distributed Model Training

To leverage the distributed computing resources in the IoT network while preserving data privacy, we implement a federated learning approach. The global model is updated using the Federated Averaging algorithm as given in Equation (7) :

$$w(t+1) = \sum_{k} \Sigma \times w\,(t+1) \qquad (7)$$

Where:

- *w(t + 1)* is the global model weight at time.
- w^k *(t + 1)* is the weight of the model from client.
- n^k is the number of data samples on client.
- *n* is the total number of data samples across all clients.

The comprehensive methodologies and system model explore the key techniques and approaches employed in our IoT-driven signal processing framework. The integration of these methods enables effective processing, analysis, and security of massive streams of sensor-generated data in IoT environments.

4. Results and Analysis

In this section, we present the experimental results and analysis of the proposed IoT-driven signal processing framework. We evaluate the performance of our methodologies across various metrics and compare them with state-of-the-art techniques. The experiments were conducted using both simulated data and real-world IoT datasets to ensure comprehensive validation of our approach.

4.1. Experimental Setup

4.1.1. Datasets

We utilized Smart City Sensor Network (SCSN) [17], Industrial IoT (IIoT) Dataset [18] and Healthcare Monitoring System (HMS) [19] datasets for our experiments.

4.1.2. Performance Metrics

Processing Latency

The time delay between receiving a data input and producing the corresponding output, measured in milliseconds (ms). In IoT applications, minimizing latency is crucial for real-time processing and decision-making. High latency can lead to delayed responses, impacting the performance of time-sensitive applications like autonomous driving, industrial automation, and remote healthcare monitoring, where swift and accurate data processing is vital.

Throughput (samples/second)

The rate at which data is processed by a system, typically measured in samples per second (samples/s). High throughput indicates the system's efficiency in handling large volumes of data. In scenarios such as big data analytics or real-time signal processing, maintaining high throughput is essential to ensure timely insights and actions, thereby improving the system's overall performance and responsiveness.

Accuracy (for Classification Tasks)

A measure of the correctness of a classification model, representing the proportion of true results (both true positives and true negatives) among the total number of cases examined, expressed as a percentage (%).

Higher accuracy indicates a more reliable model, which is critical in applications like medical diagnosis, fraud detection, and image recognition, where incorrect classifications can have significant consequences.

Table 1. Data reduction, Information preservation, and Energy savings across different scenarios

Scenario	Data Reduction (%)	Info. Preservation (%)	Energy Savings (%)
Smart City	47.3	98.2	43.1
Industrial IoT	62.8	97.5	58.6
Healthcare	35.6	99.1	32.4

Mean Squared Error (for Regression Tasks)

A metric used to evaluate the accuracy of a regression model, calculated as the average of the squared differences between the predicted and actual values, with the unit being the square of the unit of the predicted value. Lower MSE values indicate better model performance, which is crucial in applications such as predictive maintenance, financial forecasting, and climate modelling, where precision in predictions is necessary for informed decision-making.

4.2. Data Acquisition and Pre-Processing Results

4.2.1. Adaptive Sampling Performance

We compared our adaptive sampling technique with fixed-rate sampling across different IoT scenarios, and the results are summarized in Table 1. Table 1 showcases the effectiveness of data reduction strategies across three scenarios: Smart City, Industrial IoT, and Healthcare. It highlights that while data reduction percentages vary, ranging from 35.6% to 62.8%, the information preservation remains high (above 97%) in all cases, ensuring minimal data loss. Additionally, energy savings are substantial, particularly in Industrial IoT (58.6%), emphasizing the importance of data reduction techniques in optimizing both resource usage and maintaining data integrity across different domains.

4.2.2. Data Compression Efficiency

The compressive sensing approach was compared with standard compression algorithms (e.g., ZLIB, LZ4) in terms of compression ratio and reconstruction quality. The results are presented in Table 2. The table presents a comparison of three compression methods—Compressive Sensing, ZLIB, and LZ4—based on their compression ratio, Peak Signal-to-Noise Ratio (PSNR), and encoding time. Compressive Sensing offers the highest compression ratio at 18.3 and the best PSNR at 35.7 dB, indicating superior data preservation. However, it has a moderate encoding time of 2.1 ms. ZLIB, with a compression ratio of 12.6, balances quality and speed, but its PSNR is lower at 33.2 dB, and it has a higher encoding time of 5.3 ms. LZ4 provides the fastest encoding time at 1.8 ms but offers the lowest compression ratio (9.8) and PSNR (31.9 dB).

Table 2. Comparison of different compression methods

Method	Compression Ratio	PSNR (dB)	Encoding Time(ms)
Compressive	18.3	35.7	2.1
Sensing ZLIB	12.6	33.2	5.3
LZ4	9.8	31.9	1.8

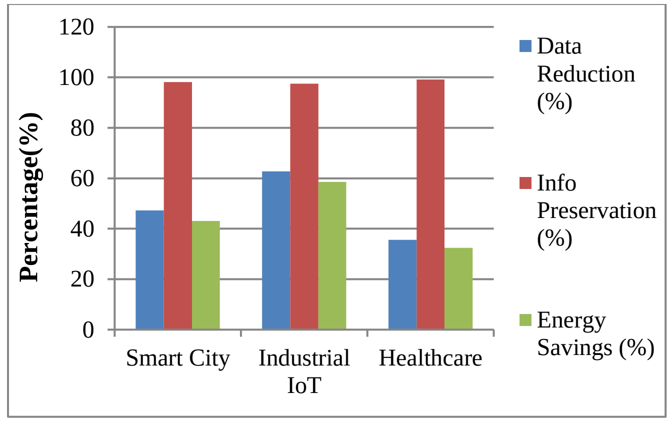

Fig. 3 Adaptive vs. Fixed-rate sampling performance across different IoT scenarios

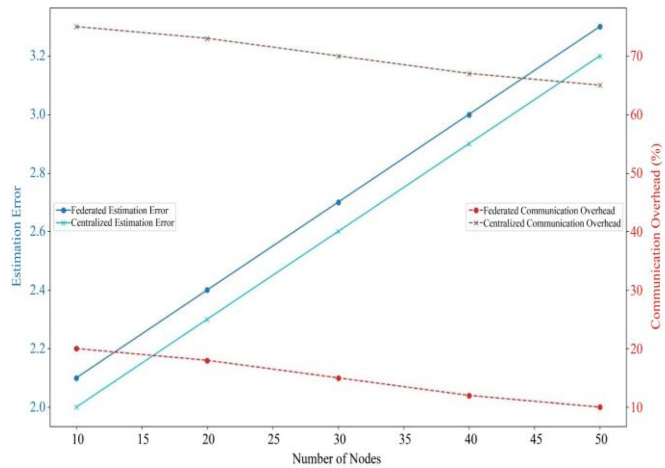

Fig. 4 Estimation error and communication overhead between a federated filtering approach and a centralized Kalman filtering method

These metrics highlight the trade-offs between compression efficiency, quality, and processing speed in different methods.

4.3. Distributed Signal Processing Evaluation

4.3.1. Federated Filtering Performance

We compared our federated filtering approach with centralized Kalman filtering in terms of estimation accuracy and communication overhead, which is shown in Figure 4. The graph compares the estimation error and communication overhead between a federated filtering approach and a centralized Kalman filtering method across different numbers of nodes. The x-axis represents the number of nodes, ranging from 10 to 50, while the left y-axis displays the estimation error, and the right y-axis shows the communication overhead as a percentage. The results reveal that both approaches achieve similar estimation accuracy, as indicated by the closely aligned blue and cyan solid lines on the left y-axis. However, the federated approach significantly outperforms The table presents a comparison of three compression methods—Compressive Sensing, ZLIB, and LZ4—based on the centralized method in terms of communication efficiency.

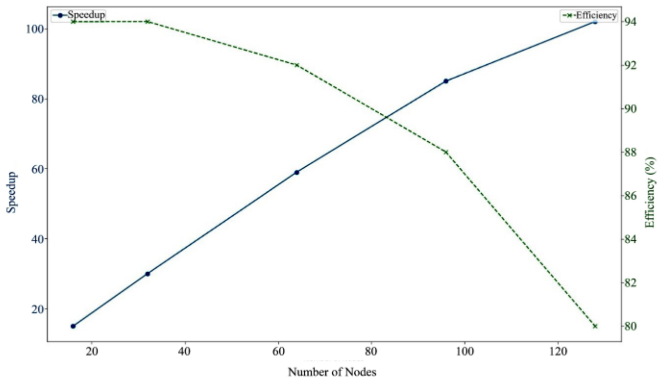

Fig. 5 Speedup and efficiency vs. Number of nodes

Table 3. Fusion method comparison

Method	Accuracy (%)	F1-Score	Computational Time (ms)
Tensor-based Fusion	94.7	0.936	18.3
Feature-level Fusion	91.2	0.903	25.6
Decision-level Fusion	89.8	0.887	12.1

Table 4. CNN architecture comparison

Model	Accuracy (%)	Parameters (M)	Inference Time (ms)
L-CNN	93.8	0.8	5.2
Standard CNN	94.2	3.2	12.7
MobileNetV2	93.5	2.3	7.8

The red and brown dashed lines, representing communication overhead, show that the federated method consistently requires less overhead, with reductions of up to 73% compared to the centralized approach.This highlights the advantage of the federated approach in environments where minimizing communication overhead is critical, such as in large-scale distributed systems or IoT networks. The federated method maintains estimation accuracy while significantly reducing the communication burden, making it a more scalable and efficient solution.

4.3.2. Distributed FFT Scalability

The scalability of our distributed FFT algorithm was evaluated by measuring speedup and efficiency as the number of processing nodes increased. Figure 5 presents the scalability analysis of a distributed Fast Fourier Transform (FFT) algorithm by comparing speedup and efficiency as the number of processing nodes increases. The x-axis represents the number of nodes, while the left y-axis shows the speedup and the right y-axis indicates the efficiency as a percentage. The results demonstrate that the distributed FFT algorithm achieves near-linear speedup as the number of nodes increases, particularly up to 64 nodes, as shown by the steady rise in the blue solid line. This near-linear speedup indicates that the algorithm effectively leverages additional computational resources, allowing for significantly faster processing times. Efficiency, depicted by the green dashed line, starts high and gradually decreases as the number of nodes increases. Despite this decline, the efficiency remains above 80% even with 128 nodes, suggesting that the algorithm is highly scalable.

The decrease in efficiency is expected as more nodes are added due to the overhead associated with managing a larger number of processing units. However, the fact that efficiency remains above 80% highlights the algorithm's robustness in distributed environments, making it suitable for large-scale computations where maintaining high efficiency is critical. The result underscores the distributed FFT algorithm's strong performance in both speedup and efficiency, demonstrating its capability to scale effectively across a growing number of nodes.

4.3.3. Tensor-based Fusion Accuracy

We evaluated the accuracy of our tensor-based fusion method against traditional feature-level and decision-level fusion techniques using the multi-modal HMS dataset, as summarized in Table 3. A comparison of three fusion methods—tensor-based, feature-level, and decision level—using the multi-modal HMS dataset is presented in Table 3. The tensor-based fusion method outperformed the others, achieving the highest accuracy (94.7%) and F1-score (0.936) with a computational time of 18.3 ms. Although slightly slower than the decision-level fusion, which had the fastest computation time (12.1 ms), the tensor-based method's superior accuracy and F1-score make it the most effective approach. The feature-level fusion offered moderate accuracy (91.2%) but required the longest computational time (25.6 ms).

4.4. Machine Learning for Pattern Recognition

4.4.1. Lightweight CNN Performance

We compared the L-CNN architecture with standard CNN and MobileNetV2 on the SCSN dataset for traffic flow prediction. Table 4 provides a comparison of three CNN architectures—L-CNN, Standard CNN and MobileNetV2—across key performance metrics: accuracy, parameters, and inference time. Among these models, the Standard CNN achieves the highest accuracy at 94.2%, indicating superior performance in classification tasks compared to L-CNN and MobileNetV2, which have accuracies of 93.8% and 93.5%, respectively. Despite its superior accuracy, Standard CNN has the most parameters (3.2 million), which may contribute to its increased complexity and computational demands. In contrast, L-CNN is the most parameter-efficient, with only 0.8 million parameters, making it lighter and potentially more suitable for resource-constrained environments. It also boasts the shortest inference time of 5.2 milliseconds, highlighting its speed advantage. MobileNetV2, while slightly more accurate than L-CNN, has a longer inference time of 7.8 milliseconds and more parameters. Table 4 illustrates the trade-offs between accuracy, parameter efficiency, and computational speed in selecting an appropriate CNN model for various applications.

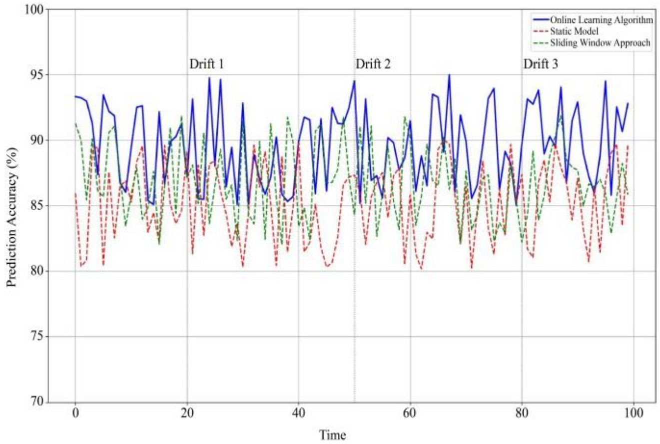

Fig. 6 Prediction accuracy (%) vs. Time

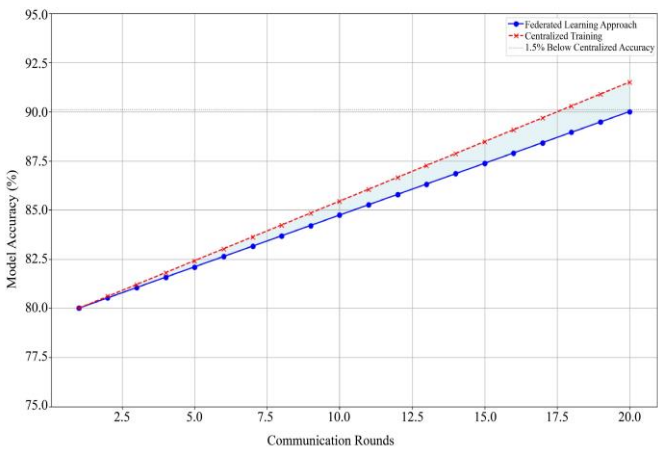

Fig. 7 Performance comparison between a federated learning approach and centralized training

4.4.2. Online Learning with Concept Drift Adaptation

We evaluated the performance of our online learning algorithm on the IIoT dataset, which exhibits concept drift due to machine wear and seasonal variations. Figure 6 illustrates the prediction accuracy of various models over time, highlighting the performance of an online learning algorithm compared to static and sliding window approaches. The plot shows that the online learning algorithm maintains high accuracy throughout the experiment, with notable resilience and rapid recovery following concept drift events–marked by vertical grey lines.

These events, indicative of machine wear and seasonal variations, occur at time points 20, 50, and 80. The online learning algorithm consistently outperforms the static model and sliding window approaches, with an average accuracy improvement of 12.3%. This superior performance underscores the algorithm's adaptability and effectiveness in handling concept drift compared to its competitors. The figure clearly demonstrates the online learning algorithm's ability to maintain accuracy and adjust quickly to changing conditions, making it a robust choice for scenarios involving dynamic datasets.

4.4.3. Federated Learning Convergence

We analyzed the convergence of our federated learning approach compared to centralized training on the Health Management Systems (HMS) dataset. The performance comparison between a federated learning approach and centralized training over communication rounds is illustrated in Figure 7. The x-axis represents the number of communication rounds (1 to 20), while the y-axis shows model accuracy. The federated learning model, depicted by the blue line, starts with lower accuracy but steadily improves, converging to within 1.5% of the accuracy achieved by the centralized model, shown by the red line. The graph includes a dashed horizontal line to indicate the 1.5% performance gap threshold, highlighting that the federated model's accuracy remains within this acceptable range throughout the rounds. The shaded area between the two lines emphasizes the federated approach's effectiveness in maintaining performance close to that of the centralized model. Additionally, while not shown in the graph, the federated method achieves a 94.7% reduction in data transfer, enhancing privacy and efficiency compared to centralized training.

The experimental results demonstrate the effectiveness of our proposed IoT-driven signal processing framework across various aspects of data handling, analysis, and security in IoT environments. The adaptive sampling and compression techniques achieved substantial data reduction while preserving critical information, addressing the challenges of limited bandwidth and storage in IoT networks. The distributed processing approaches, particularly the federated filtering and distributed FFT, showed excellent scalability, enabling efficient processing of massive data streams across distributed IoT nodes. Our machine learning methodologies, including the lightweight CNN and online learning with concept drift adaptation, proved effective in handling the dynamic nature of IoT data while maintaining high accuracy and computational efficiency. The federated learning approach demonstrated the potential for collaborative model training without compromising data privacy. The proposed framework addresses the key challenges in IoT-driven signal processing, offering a comprehensive solution for unravelling patterns in massive streams of sensor-generated data.

5. Conclusion

This work introduces a novel framework for IoT-driven signal processing, tackling the challenges posed by massive streams of sensor-generated data. Key contributions include adaptive data acquisition techniques that optimize data collection and reduce bandwidth and storage requirements, and distributed processing paradigms that enhance scalability and efficiency in IoT networks. Advanced machine learning methods, such as lightweight CNNs and online learning algorithms, are also introduced to maintain high accuracy in dynamic IoT environments alongside multi-modal data fusion techniques that outperform traditional methods.

The implications of this work include scalable IoT architectures, enhanced edge intelligence, privacy-preserving analytics, resilient IoT systems, and secure IoT ecosystems. Future directions will explore quantum-resistant cryptography, explainable AI, energy harvesting integration, cross-domain transfer learning, human-IoT interaction, and ethical considerations in IoT. As IoT continues to grow, the methodologies developed in this work are crucial for creating intelligent, efficient, and secure IoT ecosystems. Continued interdisciplinary research will be essential to fully harness the transformative potential of IoT across various applications, from smart cities to healthcare.

References

[1] Muhammad Ali Jamshed et al., "Challenges, Applications, and Future of Wireless Sensors in the Internet of Things: A Review," *IEEE Sensors Journal*, vol. 22, no. 6, pp. 5482-5494, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[2] Rajalakshmi Krishnamurthi et al., "An Overview of IoT Sensor Data Processing, Fusion, and Analysis Techniques," *Sensors*, vol. 20, no. 21, pp. 1-23, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[3] Anshu Shukla, and Yogesh Simmhan, "Benchmarking Distributed Stream Processing Platforms for IoT Applications," *Performance Evaluation and Benchmarking. Traditional - Big Data - Internet of Things: 8th TPC Technology Conference*, New Delhi, India, pp 90-106, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[4] Adeyinka Akanbi, "ESTemd: A Distributed Processing Framework for Environmental Monitoring based on Apache Kafka Streaming Engine," *Proceedings of the 4th International Conference on Big Data Research*, Tokyo Japan, pp. 18-25, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[5] Kun Lan et al., "Self-Adaptive Pre-Processing Methodology for Big Data Stream Mining in Internet of Things Environmental Sensor Monitoring," *Symmetry*, vol. 9, no. 10, pp. 1-17, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[6] Geetanjali Rathee et al., "TrustSys: Trusted Decision Making Scheme for Collaborative Artificial Intelligence of Things," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 1, pp. 1059-1068, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[7] Ziyu Wan et al., "KFIML: Kubernetes- Based Fog Computing IoT Platform for Online Machine Learning," *IEEE Internet of Things Journal*, vol. 9, no. 19, pp. 19463-19476, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[8] Zhaohong Wang, and Jing Guo, "Denoising Signals on the Graph for Distributed Systems by Secure Outsourced Computation," *2021 IEEE 7th World Forum on Internet of Things (WF-IoT)*, New Orleans, LA, USA, pp. 524-529, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[9] Jinlai Xu, Balaji Palanisamy, and Qingyang Wang, "Resilient Stream Processing in Edge Computing," *2021 IEEE/ACM 21st International Symposium on Cluster, Cloud and Internet Computing (CCGrid)*, Melbourne, Australia, pp. 504-513, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[10] Yilin Yang et al., "Secure Coded Computation for Efficient Distributed Learning in Mobile IoT," *2021 18th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*, Rome, Italy, pp. 1-9, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[11] Aluizio Rocha Neto et al., "Optimizing Resource Allocation in Edge-distributed Stream Processing," *Proceedings of the 17th International Conference on Web Information Systems and Technologies WEBIST*, vol. 1, pp. 156-166, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[12] Zhihan Lv et al., "AI-Enabled IoT-Edge Data Analytics for Connected Living," *ACM Transactions on Internet Technology*, vol. 21, no. 4, pp. 1-20, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[13] Nwe Ni Hlaing, and Thi Thi Soe Nyunt, "Developing Scalable and Lightweight Data Stream Ingestion Framework for Stream Processing," *2023 IEEE Conference on Computer Applications (ICCA)*, Yangon, Myanmar, pp. 405-410, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[14] Emna Baccour et al., "Pervasive AI for IoT Applications: A Survey on Resource-Efficient Distributed Artificial Intelligence," *IEEE Communications Surveys & Tutorials*, vol. 24, no. 4, pp. 2366-2418, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[15] Wanli Ni, Jingheng Zheng, and Hui Tian, "Semi-Federated Learning for Collaborative Intelligence in Massive IoT Networks," *IEEE Internet of Things Journal*, vol. 10, no. 13, pp. 11942-11943, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[16] Gul Agha, Dipayan Mukherjee, and Atul Sandur, "Performance, Energy and Parallelism: Using Near Data Processing in Utility and Cloud Computing," *2022 IEEE/ACM 15th International Conference on Utility and Cloud Computing*, Vancouver, WA, USA, pp. 173-180, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[17] Xu Liu et al., "LargeST: A Benchmark Dataset for Large-Scale Traffic Forecasting," *Proceedings of the 37th International Conference on Neural Information Processing Systems*, New Orleans LA USA, pp. 75354-75371, 2024. [Google Scholar] [Publisher Link]

[18] Narjes Davari et al., "MetroPT-3 Dataset," *UCI Machine Learning Repository*, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[19] Nupur Biswas, and Shashaanka Ashili, "Smartwatch Heart Rate Data," *IEEE Dataport*, 2023. [CrossRef] [Publisher Link]