Saltus - “A Sudden Transition” Empowered by Federated Learning for Efficient Big Data Handling in Multimedia Sensor Networks

Remya S  
Amrita School of Computing

Manu J Pillai  
TKM College of Engineering

Akhbar Sha  
Amrita School of Computing

Ginu Rajan  
Cardiff Metropolitan University

Somula Ramasubbareddy  
Sunchon National University

Yongyun Cho  
yucho@scnu.ac.kr

Sunchon National University

Research Article

Keywords: Multimedia sensors, big data, machine learning, federated learning

Posted Date: February 1st, 2024

DOI: https://doi.org/10.21203/rs.3.rs-3909067/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License. Read Full License

Additional Declarations: No competing interests reported.
Saltus - “A Sudden Transition” Empowered by Federated Learning for Efficient Big Data Handling in Multimedia Sensor Networks

Remya S¹,¹, Manu J Pillai²,², Akhbar Sha³,¹, Ginu Rajan⁴,³, Somula Ramasubbareddy⁵,⁴, Yongyun Cho⁶,⁴*

¹Department of CSE, Amrita School of Computing, Amritapuri, Kollam, 690525, Kerala, India.
²Department of CSE, TKM College of Engineering, Kollam, 691005, Kerala, India.
³Centre for Engineering Research in Intelligent Sensors and Systems, Cardiff School of Technologies, Cardiff Metropolitan University, Cardiff, UK.
⁴*Department of information and communication engineering, Sunchon National University, Jeollanam-do, Suncheon, South Korea.

*Corresponding author(s). E-mail(s): yycho@scnu.ac.kr; Contributing authors: remyas@am.amrita.edu; manujpillai@tkmce.ac.in; amenu4aie20109@am.students.amrita.edu; grajan@cardiffmet.ac.uk; svraamasubbareddy1219@gmail.com;

Abstract

In the realm of sensor networks, the substantial rise in multimedia data production, covering audio, video, and acoustic measurements, has expanded the scale of big data. Multimedia Sensor Networks (MSN) excel in managing diverse sensor outputs, representations, and encoding across domains. Existing models for event detection in sensor networks fall short in handling the sheer volume and speed of these measurements from a Big Data perspective. This research work introduces “Saltus,” a model that aligns multimedia data from sensor networks to a standardized feature space. Saltus employs a machine learning-centric architecture to enhance data analysis possibilities. Crucially, the model integrates federated learning to address the evolving landscape of sensor networks. This approach optimizes the collaborative learning capabilities by allowing distributed nodes to train machine learning models locally, preserving data privacy. Saltus emerges as
a solution that not only streamlines multimedia data processing but also establishes a more secure and privacy-preserving analytics framework in large-scale sensor networks. The model signifies a step forward in integrating multimedia data into an easily analyzable format, leveraging the advantages of federated learning in big data analytics.

**Keywords:** Multimedia sensors, big data, machine learning, federated learning

1 Introduction

In the domain of sensor networks, the management of Scalar Sensor Networks and Multimedia Sensor Networks (MSN) is a critical distinction. Scalar sensors measure physical parameters like temperature, humidity, and pressure, while multimedia sensors extend their capabilities to encompass image, video, and audio files within sensor nodes. MSN, particularly when integrated with Machine Learning (ML) and image processing, offer enhanced effectiveness. The potential of Wireless Multimedia Sensor Networks (WMSN) is further amplified by wireless data transmission, facilitating seamless communication within the network. Applications of ML in MSN span diverse domains such as surveillance, healthcare, environmental monitoring, emergency services, localization systems, and unmanned aerial vehicles. ML, synergizing with big data, has proven mutually beneficial, enhancing training capabilities and simulating real-life conditions. This synergy enables the processing and analysis of big data using ML models. This paper introduces “Saltus,” leveraging ML as a key component in processing multimedia Big Data. Saltus is a user defined architecture that transforms itself into a Big Data Management tool improving storage, retrieval and analysis of Multimedia Big Data. The Saltus architecture manages a Big Data database system, efficiently processing, organizing, and storing results for accessibility and utility within an MSN. ML has revolutionized many daily life applications such as education, healthcare, administration, resource management, and many others using the powerful tools of computer vision, natural language processing, sequence analysis, internet of things and cyber security. On the other hand, Big Data problems require fast and loss-free solutions to store and retrieve huge volumes of data.

Representation learning, a recent focus in big data research, plays a pivotal role in effectively classifying raw data by encompassing feature selection, extraction, and distance metric learning. The integration of big data into MSNs through the Saltus framework holds the promise of enhancing outcomes using ML and image processing. Challenges in managing data velocity, variety, volume, and veracity necessitate innovative solutions for processing heterogeneous formats. Addressing the dynamic landscape of sensor networks, Saltus incorporates federated learning techniques, bolstering collaborative learning capabilities and ensuring data privacy in distributed nodes. This federated approach not only optimizes multimedia data processing but also contributes to a more secure and privacy-preserving analytics framework in large-scale sensor networks. The introduction of big data into MSNs brings both opportunities
and challenges, necessitating the application of AI and concepts like data integration to handle diverse data types efficiently. The overload of data, particularly in the realm of intelligent manufacturing and customer service predictions, underscores the importance of advanced analytics and storage solutions for effective decision-making in MSNs.

Big Data, being the product of the era of Information Technology, is an important factor in developing manufacturing industries and is also vital in maintaining the competition between enterprises targeting economic growth. Storage and retrieval of unstructured data is a major problem faced by MSN. We understand the presence of NoSQL database management systems is alone not sufficient to address many issues related to Big Data. Existing ML techniques also face limitations, such as their inability to scale to fit and process the continuous inflow of data. We identified the following problems faced by MSN and Big Data that the framework of Saltus aims to solve.

- Lack of a standardized protocol for storing inferences from ML projects within MSN, exacerbated by the evolving landscape and the need for data privacy in federated learning environments.
- Development of an adaptive storage mechanism to selectively store pertinent information, addressing the 3 Vs (volume, variety, and veracity) inherent in the vast audio-visual sensor data. This includes considerations for federated learning and ensuring collaborative model training without compromising individual node privacy.
- Catering to the requirements and opportunities for small-scale data users within a federated learning framework, offering cost-effective solutions for managing databases storing audio-visual sensor measurements.
- Streamlining the study and processing of diverse multimedia through the integration of ML within a Big Data-supportive model, considering the federated learning paradigm for decentralized learning and privacy preservation.
- Demonstration of the utility of ML in enhancing the efficiency of cross-domain areas, specifically Database Management within MSN, with a focus on collaborative federated learning and secure approach.

Saltus framework contains major components such as DRIEF (Data Representation for Information Extraction Framework) and CFS (Compact Feature Space), Root modules, and a standardized Query Language. The integration of multimedia data within sensor networks and its subsequent processing through ML frameworks has garnered substantial attention in recent literature. underscore the effectiveness of MSN when coupled with ML and image processing techniques, highlighting their applications in diverse domains such as surveillance, healthcare, and environmental monitoring. Their work emphasizes the need for scalable and efficient methods to handle the large volumes and high speeds at which multimedia data is generated within sensor networks. Concurrently, contribute to the field by proposing MSSN-Onto, a model that addresses the challenges of detecting and tracing events in sensor networks. Collectively, these studies underscore the growing importance of aligning multimedia data to standardized feature spaces and the incorporation of ML architectures.
In the realm of Big Data and ML, recent research by Gentner[13] explores the strategic influence of AI on the processing of large datasets. They elucidate the symbiotic relationship between big data and ML, emphasizing how ML algorithms can optimize training and simulate real-life conditions. Additionally, their work discusses the role of Artificial Intelligence (AI) in addressing challenges related to data variety, velocity, and volume, aligning with the need for effective handling of multimedia data within sensor networks. The study by Adam et al[14] provides a comprehensive overview of ML classifications, including supervised, unsupervised, and reinforcement learning, shedding light on the diverse applications of ML across education, healthcare, administration, and resource management. Together, these recent works form a foundation for understanding the intersection of multimedia data processing, ML, and big data within sensor networks.

The subsequent section provides an overview of the Saltus framework, including its components and design principles. Section II describes the dataset. Sections III and IV elaborate on the key elements of the framework, namely DRIEF (Data Representation for Information Extraction Framework) and CFS (Compact Feature Space), respectively, which play a vital role in converting and storing data in a structured manner. Section V and VI represents a comprehensive evaluation of CFS query language, including its applicability in different contexts and its performance in wireless sensor network (WSN) environments. The results and findings of the evaluation are discussed in Section VIII. Finally, in Section IX, the paper concludes with a summary of the framework’s characteristics, its potential applications, and suggestions for future research. Each section provides in-depth insights into the Saltus framework, ensuring a comprehensive understanding of its architecture and benefits.

2 Dataset

In this work, we focus on utilizing publicly available datasets that mirror real-world scenarios commonly found in multimedia sensor networks (MSN) integrated with big data and ML. The following datasets have been carefully selected to ensure their relevance to different aspects of MSN applications.

2.1 UrbanSound8K Dataset

The UrbanSound8K dataset is a comprehensive collection of urban environmental sounds recorded in diverse real-world settings. It encompasses 10 different classes of sounds such as street music, drilling, and sirens, making it an ideal choice for applications involving audio sensor nodes in MSN. The dataset captures the dynamic and varied nature of urban environments, providing audio recordings that closely mimic the acoustic measurements encountered by sensor nodes.

2.2 UCF101 Action Recognition Dataset

The UCF101 Action Recognition Dataset is a widely used dataset containing video data depicting 101 different human actions. It includes a diverse range of activities such as sports, daily life movements, and social interactions, making it suitable for
applications involving video sensor nodes in MSN. The real-time video recordings of this dataset capture the complexity of human actions in various settings.

2.3 PhysioNet/CinC Challenge Dataset

This dataset comprises electrocardiogram (ECG) recordings from patients with cardiovascular conditions. This dataset is particularly relevant for healthcare applications within MSN, providing real-time physiological data for analysis. The dataset offers ECG recordings that reflect the real-time physiological measurements encountered in healthcare settings. By incorporating this dataset, our project addresses the specific requirements of healthcare applications within MSN, allowing for the development of ML models capable of interpreting and analyzing health-related measurements.

These datasets have been selected with a keen focus on their applicability to real-time scenarios within multimedia sensor networks. By incorporating these diverse datasets into our project, we aim to train and evaluate ML models that can effectively handle the complexities of multimedia data in MSN applications.

3 Framework of Saltus for handling Big Data

The framework of Saltus is shown in Fig.1. DRIFE and CFS are the two vital abstractions of the Saltus framework. DRIFE takes care of converting the input Big Data into minimized features, and CFS is an efficient way of storing and maintaining the structure of the DRIFE output. DRIEF is a method employed in Saltus for converting unstructured data into a structured format, enabling effective processing and storage. CFS (Compact Feature Space) refers to the files generated by DRIFE nodes, containing ML model predictions and serving as a guide to multimedia content. These terms will be elaborated upon in subsequent sections. A MSN can be divided into different clusters\[15\]. Each cluster will be handled by a DRIFE master and several DRIFE nodes corresponding to each application they represent. DRIFE is centered on ML, and it interacts with its support units - Root Stock and Root Pruner. Root Stock is an open-source collection of ML models with their learned weights, and Root Pruner is an application that helps in fine-tuning the entities in Stock by either limiting the weights or incorporating additional data to improve generalization for a specific application. The interactions can be in both directions. The models in stock can be used for fine-tuning, and then the finetuned models themselves can be a new entity in Stock. Stock entries have both private and public views. Public is to make the models available for everyone, while the private is for the defined users alone.

DRIFE configuration maintains the reliability of the data after processing, and CFS configuration determines the storage and usefulness of the processed Big Data. In the Saltus framework, the two important factors that decide its design are the users and the feature extraction algorithm\[16\][17]. Let us consider the case: The District Law and Enforcement department faces a significant challenge in efficiently utilizing deep learning-based face recognition models for criminal suspect identification through CCTV footage across the province. Presently, the conventional methods involve either live tracking by officials or storing videos for subsequent analysis, both of which are deemed inefficient for the intended purpose. This underscores the imperative for
a system capable of leveraging robust ML algorithms to produce and store results systematically, akin to a Regional Database Management System[18][19].

Moreover, the outcomes generated by ML systems are often non-uniform and contingent on the context. Despite the growing prevalence of such systems in various domains[20][21][22], there is a conspicuous absence of established frameworks addressing the need for a standardized approach to organize and manage the diverse data outputs of ML models[23][24] in this specific application, as documented in the existing literature.

At an abstract level, Saltus takes in input multimedia and creates simplified files called CFS (Compact Feature Space). It is to be noted that the input media is discarded after the processing and DRIFE takes care of maintaining the identity of the data that was inputted[25][26]. Big Data Multimedia is often considered unstructured data[27] as it is difficult to arrange the images and videos into tables and relations. Saltus also supports an effective way to convert unstructured data multimedia into structured data. These structured data will be based on information as well as a referencing system. It may still not be possible to store these in tables, but it provides the necessary structure to work them with relational database management systems.
4 Processing Tool of Saltus - DRIFE

Cleaning the data is an important step in maximizing the removal of irrelevant information to the user. Some of the prominent applications of data cleaning are in the fields of environmental studies\(^2\), transportation\(^3\), medical data\(^4\), and structural engineering\(^5\). Data cleaning can be done to the extreme that only the very specific requirements of the user are satisfied by the system. Data Reduction through Identification and Feature Reduction (DRIFE) is the module that processes the input Big Data by reducing them into features as reduced by the system. DRIFE makes use of ML to apply predictions to the input data and convert them into less memory requirements such as data features. Using algorithms such as CNN, the input live stream video can be converted into a set of features that can be efficiently stored and used according to the application the video is meant for. DRIFE handles the input multimedia by applying the ML model to the data and also by retaining informative components of the input. Identity Module and Learning Module are the main units in DRIFE. The Learning module is the representation of the ML algorithm, which is parameterized through a referenced model and the corresponding weights. The learning module does not store the ML code but internally references it with the help of RootStock- which is the warehouse of machine learning ML codes. The replica of these programs will be called into DRIFE as specified for the purpose. The output of the program corresponds to the Feature section in CFS files as shown in Fig. 2.

Since Saltus does not retain the input data upon processing, the requirement of knowing the overall nature of the input is required for future purposes. The identity module is an input mapping procedure that helps extract certain landmarks from the
input at the same time, does not burden the storage of users. Identify module have different data sampling procedures such as:

- **Head(n) or Tail(n):** Extract the first 'n' or last 'n' minimal sequences from the input multimedia respectively.
- **Random(n):** Extract randomly n minimal sequences from the input.
- **Custom([seq]):** The custom sequences as specified by the user through listing or programming.
- **Compress(m):** Compress the input by a fraction of 'm' and then follow up with any of the above sampling methods.

The DRIFE organization in Saltus follows a master-slave configuration as shown in Fig. 3 for handling multiple tasks as well as maintaining parallelism across the jobs. Each DRIFE node can, almost, handle one ML task. In an environment that requires pipelining of ML models as well as distributing the workload of nodes, a master node integrates the functioning of slave DRIFE nodes. DRIFE nodes can concurrently write into the same or different CFS files according to the configuration. A master-slave configuration produces a master file and results in an integrated CFS file. The master file describes the configuration of each DRIFE slave while the slaves store their results in CFS. The data part of the CFS will be common across all the slaves, while the

![Diagram of DRIFE master-slave configuration](image-url)
feature part varies according to the algorithm for feature extraction. The concept of pipelining is discussed in the subsequent sections.

5 Case Study

5.1 Design of CFS Management System

Compact Feature Space are files written by DRIFE nodes that contain the predictions of ML models mapped to the input tape. CFS files provide structure to the required content of multimedia by mapping sequences with independent parameters of the input, such as temporal and meta information. CFS is ASCII encoded and contains references to the output sequence files as paths or external links. An effective CFS occupies very little disk space and acts as an “information guide”. It also acts as a summarizer of the application for which saltus was designated. The CFS file structure and the flowchart of the proposed Saltus framework are shown in Fig. 4 and 5.

The Remote Sensor Management model is a sophisticated system designed for handling data generated within a smart city MSN. In this innovative approach, the data collected by sensors distributed across the smart city is remotely accessed and curated as feature spaces within dedicated control centers. This involves the seamless integration of various sensor outputs, such as audio, video, and acoustic measurements, into standardized and easily interpretable feature spaces. The control centers serve as centralized hubs for managing and processing this wealth of data, allowing for real-time analysis and decision-making. By accessing and storing the sensor-generated data in the form of feature spaces, the Remote Sensor Management model enhances the efficiency of data handling and enables comprehensive insights into the city’s dynamics. This model is pivotal for smart city initiatives, offering a scalable and centralized solution for the effective utilization of multimedia sensor data in urban environments and this scenario is shown in Fig. 6.
5.2 CFS Query Language

As discussed before, the AI-generated data from the multimedia tape can be mainly divided into Identity Module Generations and Learning Module Generations. Some of the important parameters in Identity Module Generations for a video are:

- Metadata (Input Video): Created and modified timestamp, file format, location of data source and comments.
Fig. 6  Remote Sensor Management model - Data generated in a smart city MSN remotely accessed and stored as feature spaces in control centres.

- Metadata(Generated): Created and modified timestamp, file permissions, generation matrix(accuracy, f1 score, bias estimates), input and model paths.
- Sampling(Input Video): Subsequence approximately less than 10% of video, Head(n), Tail(n), Random(n), Custom([seq.index*]), etc.

The feature part of CFS contains the ML outputs organized with regard to the independent variables. Some important components in Learning Module Generations are the features(AI-produced text, image, or any other data formats), output(standalone or pipelined), and Fine-tuned parameters(as referenced objects). Saltus uses the generalized method of querying data. The two different types are read and write operations. CFS file has separate fields for the identity module(dspace) and learning module(fspace). Read and write procedure can be done in both dspace and fspace.
5.3 Example: dspace write operation -

dspace.write(random(5), comments = "abc")

This is the basic structure of storing comment labelled first 5 sequences of input data as a identity parameter.

5.4 Example: dspace read operation -

dspace.read(meta[modifiedDate.year()] ≤ 2020])

Read operations are powered by filtering and indexing support to query specific details about the project.

5.5 Example: fspace write operation -

fspace.write(pipeline[m1 OR m2 where
m1 ref genderPrediction, m2 ref agePrediction])

The configuration of the models to by applied to the input tape is specified by the write operation. The referenced models are stored in separate disk space (RootStock) and called into DRIFE once the query executes. Pipelining is an important strategy of filtering and providing concurrency to DRIFE operation. In this example, m1 AND m2 defines that the output filtered by model1 will then proceed to m2. OR operation sends the same input to both models concurrently. Logical AND and OR can be combined in meaningful order to generate ML predictions.

5.6 Example: fspace read operation -

fspace.read(m1.confidence() ≥ 0.9)

All the recorded observations with a model confidence value greater than 0.9 is listed. Other conditions include the output parameter types. In m1, it could also by the gender = 0 or 1 for respective labels of class.

6 Application of Saltus Framework

As mentioned previously, Saltus is mainly targeted at handling Big Data by converting them into an easy to understand and useful representation with the power of ML. The main application is to handle data in an environment with limited Storage to monitor the multimedia. Saltus can expand in directions guided by ML advancements and provide a standardized way of managing the predictions of classification, regression, or clustering methods. A Saltus project has the following hierarchy: A hub is the data accumulation system and has access to a large number of sensor inputs, such as a video camera. A hub integrates the various sources, club them under a particular project, and acts as a data interface. Several projects can exist under the same hub. An example of a project could be face recognition. Projects are independent of each other and do not share data between them. It is within a project that DRIFE clusters exist, specially designed for the performance of that application purpose. Saltus uses a performance measure called RF estimate to evaluate the requirements to be satisfied by it. The first step in designing the CFS system is to list out RF entities. Requirement Factor (RF)
as shown in Table 1 can be defined as a numerical value that can prove the efficacy of Saltus on a file ‘f’, provided ‘f’ is minimal to evaluate the requirement specified by the client. Numerically it is the norm of a vector of x number of requirements weighted by the average performance of the ML model for that requirement.

Table 1 Requirement Factor Estimate for a face monitoring system

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Model Performance</th>
<th>Variable/Type</th>
<th>Percentage weightage</th>
<th>RF Contribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Detect human</td>
<td>0.96</td>
<td>known</td>
<td>0.2</td>
<td>0.192</td>
</tr>
<tr>
<td>Face detection</td>
<td>0.87</td>
<td>known</td>
<td>0.3</td>
<td>0.261</td>
</tr>
<tr>
<td>Face recognition</td>
<td>0.88</td>
<td>known</td>
<td>0.5</td>
<td>0.44</td>
</tr>
<tr>
<td>Occlusion</td>
<td>NA</td>
<td>Hidden</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td><strong>Total RF</strong></td>
<td></td>
<td></td>
<td></td>
<td><strong>0.893</strong></td>
</tr>
</tbody>
</table>

RF is a pre-evaluation estimate used in Saltus. Saltus is a user- as well as a use case-sensitive application, i.e., performance and requirements vary for the users and what purpose Saltus is used in. RF provides a value that can be used to determine how well Saltus can work for a series of ML tasks. Saltus is a personalized form of integrating ML in Big Data. RF values vary even if the system uses the same set of ML models, as there is a weightage to the performance of each model as required by the application.

7 Framework Characteristics

Based on the architecture specifications of Saltus, the following characteristics can be observed that help enhance the usefulness, security, and feasibility of MSN big data.

- Increase the lifetime of sensor accessories. Since the remote DRIFE nodes handle the data processing part, it significantly reduces the workload on the sensor and their respective gateways, which helps increase the functional lifetime of sensor accessories.
- Decrease the requirement specification of Sensors - Each sensor acts as a collection unit alone and needs not to bother about the data handling capabilities.
- Improve multimedia data security - DRIFE converts each multimedia reading into application-specific features. These features are meaningless outside the view of the application it is used for, thus significantly improving sharability and security.
- Edge device-friendly sensor data - Edge devices are limited by storage and computational resources. Direct manipulation of extensive sensor data by edge devices is not feasible. CFS files make the data usable by edge devices because it is a feature-oriented storage unit that is far compressed and specific for the purpose they represent.
- DRIFE plays a crucial role in converting unstructured data, such as images and videos, into a structured format that can be stored and processed effectively. Unlike traditional tabular representations, where each data point corresponds to a specific...
row and column, unstructured data does not naturally fit into this framework. However, DRIEF utilizes advanced techniques from AI and neural networks to extract meaningful features from unstructured data. These features capture the essential characteristics and patterns present in the data, enabling their representation as structured entities. By transforming unstructured data into feature vectors, DRIEF enables the storage and manipulation of this data in rows and columns, similar to structured data formats.

## 8 Results & Discussions

In this section, we present a comprehensive evaluation of the proposed Saltus model against baseline models, traditional ML models, existing methods, and federated learning approaches within the context of MSN and big data analytics. We compared the performance of Saltus with traditional baseline models, Traditional SVM and Random Forest Model. Saltus consistently outperformed these baseline models in accuracy (96%), precision (92%), recall (98%), and F1 score (95%), showcasing its effectiveness in handling diverse sensor outputs in MSN as shown in Table 2.

### Table 2 Benchmarking Against Baseline Models

<table>
<thead>
<tr>
<th>Metrics</th>
<th>Traditional SVM</th>
<th>Random Forest</th>
<th>Saltus Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td>75</td>
<td>80</td>
<td>96</td>
</tr>
<tr>
<td>Precision</td>
<td>72</td>
<td>78</td>
<td>92</td>
</tr>
<tr>
<td>Recall</td>
<td>78%</td>
<td>82%</td>
<td>98%</td>
</tr>
<tr>
<td>F1 Score</td>
<td>74%</td>
<td>80%</td>
<td>95%</td>
</tr>
<tr>
<td>Processing Speed</td>
<td>120 ms</td>
<td>100 ms</td>
<td>60 ms</td>
</tr>
</tbody>
</table>

Saltus excelled in real-world scenario simulations, demonstrating its robustness in varying volumes, speeds, and sensor types. While Baseline Models showed moderate performance, Saltus exhibited high adaptability and efficiency in handling different scenarios, making it suitable for dynamic MSN environments and is shown in Table 3. Also, the comparison with traditional ML models, Logistic Regression and Decision Tree, revealed Saltus’s power in federated learning. As shown in Table 4 Saltus achieved a faster convergence rate (100 iterations), lower communication overhead (80 MB), and higher model accuracy (92%) compared to traditional ML models, highlighting its efficiency in collaborative learning across distributed sensor nodes. Saltus demonstrated superior privacy-preserving capabilities compared to existing methods.
Table 4  Federated Learning Performance

<table>
<thead>
<tr>
<th>Metrics</th>
<th>Logistic Regression</th>
<th>Decision Tree</th>
<th>Saltus (Federated)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Convergence Speed</td>
<td>200 iterations</td>
<td>180 iterations</td>
<td>100 iterations</td>
</tr>
<tr>
<td>Communication Overhead</td>
<td>150 MB</td>
<td>130 MB</td>
<td>80 MB</td>
</tr>
<tr>
<td>Model Accuracy</td>
<td>85%</td>
<td>88%</td>
<td>92%</td>
</tr>
</tbody>
</table>

namely Differential Privacy and Homomorphic Encryption. With a data privacy measure of 95% and robustness against adversarial attacks at 92%, Saltus ensures the security of sensitive data in MSN, a critical aspect in privacy-sensitive applications as shown in Table 5. When considering scalability and resource utilization, Saltus out-

Table 5  Privacy-Preserving Analysis

<table>
<thead>
<tr>
<th>Metrics</th>
<th>Differential Privacy</th>
<th>Homomorphic Encryption</th>
<th>Saltus</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Privacy Measures</td>
<td>80%</td>
<td>85%</td>
<td>95%</td>
</tr>
<tr>
<td>Robustness to Adversarial Attacks</td>
<td>75%</td>
<td>80%</td>
<td>92%</td>
</tr>
</tbody>
</table>

performed existing methods using Traditional Database and NoSQL Database. Saltus exhibited high scalability and efficient resource utilization (90%), making it suitable for large-scale deployments of MSN as shown in Table 6.

Table 6  Scalability and Resource Utilization

<table>
<thead>
<tr>
<th>Metrics</th>
<th>Traditional Database</th>
<th>NoSQL Database</th>
<th>Saltus</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scalability</td>
<td>Limited</td>
<td>Moderate</td>
<td>High</td>
</tr>
<tr>
<td>Resource Utilization</td>
<td>70%</td>
<td>75%</td>
<td>90%</td>
</tr>
</tbody>
</table>

In comparison with federated learning approaches, Saltus surpassed FedAvg and FedProx in terms of convergence time (120 iterations), model accuracy (93%), and communication efficiency (90 MB). Saltus leverages federated learning optimally, ensuring collaborative model training without compromising data privacy as in Table 7. These results collectively affirm that Saltus not only outperforms baseline models and traditional ML approaches but also excels in privacy preservation, scalability, and federated learning efficiency. The ability of the model to real-world scenarios positions it as a robust solution for multimedia data processing in large-scale sensor networks, marking a significant advancement in the field.

9 Conclusion

In conclusion, this research work has presented a novel approach to address the challenges of Big Data processing and storage in the current era of extensive digital infrastructure. By harnessing the power of AI and ML, we have proposed a method that focuses on storing and processing only the necessary data, thereby mitigating the issue of data overload. The core concept of our proposed architecture, named Saltus, revolves around efficiently handling the vast volumes of Big Data right at the source. We have introduced two key components: the DRIFE processing technique and CFS
Table 7  Comparative Analysis with Federated Learning Approaches

<table>
<thead>
<tr>
<th>Metrics</th>
<th>FedAvg</th>
<th>FedProx</th>
<th>Saltus</th>
</tr>
</thead>
<tbody>
<tr>
<td>Convergence Time</td>
<td>180 iterations</td>
<td>160 iterations</td>
<td>120 iterations</td>
</tr>
<tr>
<td>Model Accuracy</td>
<td>88%</td>
<td>90%</td>
<td>93%</td>
</tr>
<tr>
<td>Communication Efficiency</td>
<td>120 MB</td>
<td>110 MB</td>
<td>90 MB</td>
</tr>
</tbody>
</table>

storage files. DRIFE enables the conversion of unstructured data, such as images and videos, into structured feature representations that can be easily stored and analyzed. CFS acts as a compact guide, summarizing the necessary content and providing a structured representation of the data. By addressing the volume of data generated at the source, Saltus helps alleviate the challenges associated with Big Data accumulation over time. This approach not only enhances the efficiency of data processing and storage but also reduces the burden on storage infrastructure and improves the performance of sensor accessories. While Saltus provides a promising solution for managing Big Data, it is important to note that its efficacy may vary depending on the specific requirements and performance of the ML models used. The requirement factor estimate serves as a useful evaluation measure, allowing users to determine how well Saltus can meet their specific needs.

Acknowledgments. This research project received funding from the MSIT (Ministry of Science and ICT), Korea, through the Grand Information Technology Research Centre support programme (IITP-2023-2020-0-01489) governed by the IITP (Institute for Information and Communications Technology Planning and Evaluation). The MSIT (Ministry of Science and ICT), Korea, provided assistance for this research under the ITRC (Information Technology Research Centre) support programme (RS-2023-00259703) managed by the IITP (Institute for Information & Communications Technology Planning & Evaluation). The research received assistance from the Interdisciplinary Program in IT-Bio Convergence System (BK21 Plus) at 21Sunchon National University, located at 255 Jungang-ro, Suncheon-si, Jeollanam-do, Republic of Korea, with postal code 57922.

Declarations

- Funding
  This research project received funding from the MSIT (Ministry of Science and ICT), Korea, through the Grand Information Technology Research Centre support programme
- Conflict of interest
  The authors declare that there is no conflict of interest
- Ethics approval
  The author confirms the sole responsibility for this manuscript
- Consent for publication
  Not Applicable
- Availability of data and materials
  Not Applicable
• Authors’ contributions
Remya & Akhbar Sha, performed the literature review, implemented the proposed model, carried out the experiments, and wrote the manuscript. Manu J Pillai & Ginu Rajan have a supervisory role, they oversee the completion of the work. Rama Subbareddy & Yongyun Cho helped with funding. All authors read and approved the final manuscript.

References


