International Journal of Research Publications in Engineering, Technology and Management (IJRPETM)

|www.ijrpetm.com | ISSN: 2454-7875 | editor@ijrpetm.com |A Bimonthly, Peer Reviewed & Scholarly Journal|

||Volume 5, Issue 6, November - December 2022||

DOI:10.15662/IJRPETM.2022.0506020

Cloud-Centric Big Data Pipeline Optimization for Smart Healthcare Monitoring Systems

Dileep Valiki

Independent Researcher, India

ABSTRACT: Increased innovation and interaction among connected things may pose a number of administrative challenges. The built system could readily handle important information about medication use by patients in hospitals and their health conditions during treatments. As patients take medications according to prescribed usage, the dispensing keys automatically identify the patients, and the messages could be sent to display on the hospital screens. The feature called a health scan would be used and can assist in helping a patient identify whether any sickness has occurred by just sitting in a particular location. In case a disease is detected in the health scan module and if it is dangerous, then a message can be sent to the patient's primary physician regarding his health condition. The condition verification module would be for verifying the health condition of a patient during the medication time."

A Big Data-Driven Technology in Healthcare helps patients with heart diseases and automatically takes care of these patients by constantly monitoring them on time through the high-speed internet. Smart Healthcare systems equipped with Data Mining Techniques due to which smart patient care is possible. Big data technologies support both the massive storage and the high-performance requirements of data stored in healthcare. Healthcare Data Ubiquitous process consists of data collection, data sharing, data analyzing, results generation, and data visualization. Wireless sensor Network for patient health parameters over the cloud. The patient's health information can also transfer this data to the hospital through the cloud, the doctor can get the patient health status through the cloud whenever they required. Finger print Technology for storing patient information in the hospital."

KEYWORDS: Cloud-based healthcare analytics,Big data pipeline optimization,Smart healthcare monitoring,Real-time patient data processing,Healthcare IoT integration,Scalable cloud architecture,Distributed data processing,Edge-to-cloud computing,Predictive health analytics,Healthcare data orchestration,Machine learning in healthcare,Remote patient monitoring systems,Secure health data transmission,High-performance data pipelines,AI-driven healthcare analytics.

I. INTRODUCTION

The dawn of the 21st century has witnessed a sky-high rise in the population of the world. There have also arisen a range of factors affecting the well-being of these individuals. Healthcare, a major realm in society, has seen improvements due to resources, specialists, and equipment. Despite reaching a satisfactory level and achieving a positive effect on life expectancy, the need for a digital health ecosystem to aid healthcare professionals and patients has become evident. Patients who have a health issue and require continuous monitoring are usually kept in hospitals or clinics. Yet, this is not feasible in many places where the required infrastructure is absent, as well as in developing nations. As a solution, remote healthcare monitoring systems via sensor networks have been developed. Patients, with the help of some basic low-cost sensors, can continuously monitor their health and get assistance from healthcare professionals with a single click whenever the parameters cross their limits.

Along with the development of these smart healthcare monitoring systems, there has also been a boom in Internet of Things (IoT) devices that generate data in massive amounts, in real time. Along with this, the cost of deploying these devices has come down drastically because of service as well as hardware cost reduction. However, simply storing this data will not yield any results. Rather, the main task is to continuously analyze the data, gain knowledge from it, and apply this knowledge to the corresponding situation. The means of analyzing such massive amounts of data is not trivial as well. A traditional Hadoop-based big data framework or processing model may not be efficient in such situations. Hence a cloud-centric smart healthcare monitoring data-processing PiPO (Pipelining) model is proposed.

The shift from passive data storage to active, real-time intelligence marks a critical evolution in smart healthcare. While the proliferation of IoT sensors has become economically viable, the sheer velocity and volume of the resulting data

International Journal of Research Publications in Engineering, Technology and Management (IJRPETM)

|www.ijrpetm.com | ISSN: 2454-7875 | editor@ijrpetm.com |A Bimonthly, Peer Reviewed & Scholarly Journal|

||Volume 5, Issue 6, November - December 2022||

DOI:10.15662/IJRPETM.2022.0506020

streams render traditional batch-processing frameworks, like standard Hadoop, largely obsolete for time-sensitive medical interventions. To bridge this gap, the proposed **Cloud-centric PiPO (Pipelining) model** reimagines data flow by treating analysis as a continuous, orchestrated stream rather than a static repository.

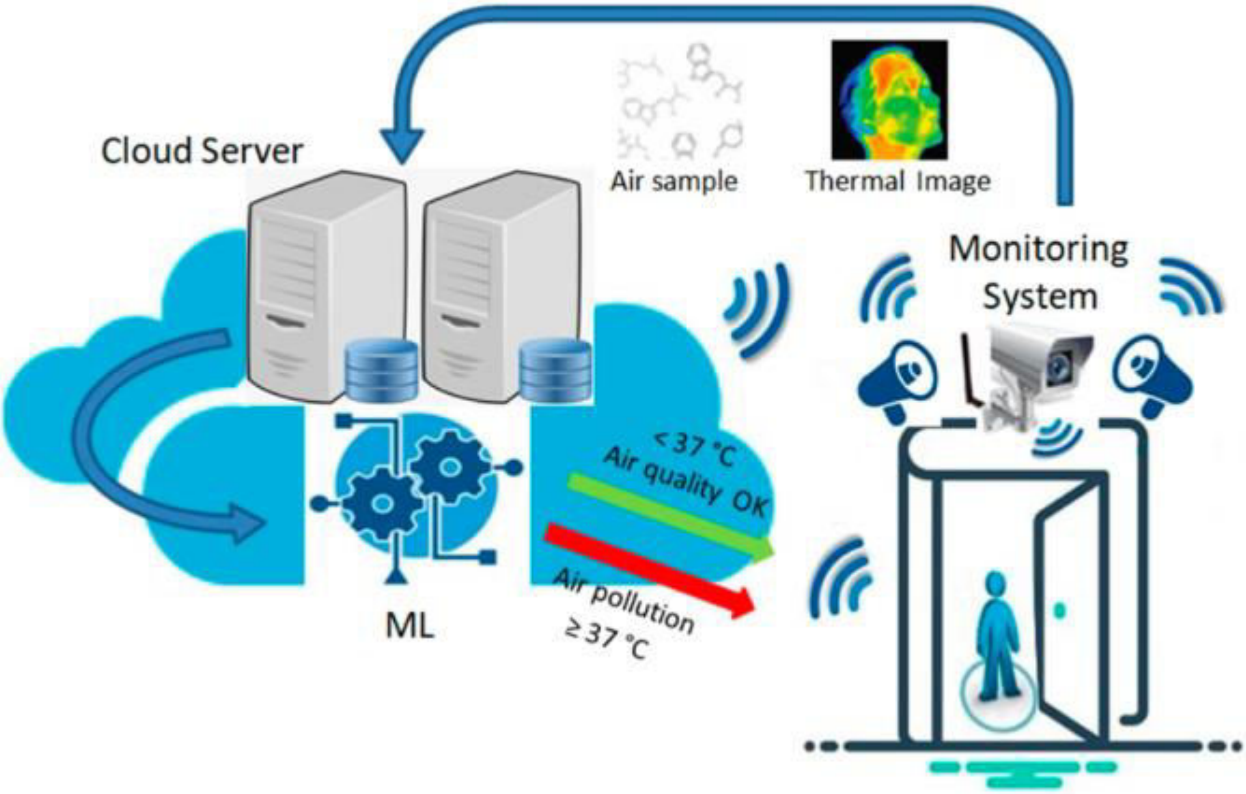


Fig 1: Secure Monitoring System for IoT Healthcare Data in the Cloud

1.1. Background and Significance

The rapid growth of the smart healthcare monitoring system has attracted significant attention in artificial intelligence-related research areas. It has recently seen the introduction of brain-inspired and decision-making models. The smart healthcare monitoring system, which is a vertical industry research field in the cybersecurity domain, focuses on protecting users' health information from various malicious threats. Smart healthcare monitoring systems create, store, and analyze substantial amounts of health data every second. Such data comprise values obtained from heart rate, temperature, blood pressure, ECG, and other sensors that work independently or in cooperation.
Cloud frameworks offer a practical solution for developing such systems, supporting big data creation, storage, and analysis. Cybersecurity issues are also addressed with cloud services, making them a crucial operational aspect of smart healthcare monitoring systems. Current research concentrates mainly on data acquisition and analytics methods and on developing and enhancing brain-inspired and decision-making models. Little attention has been devoted to designing a complete cloud-centric big data pipeline for end-to-end operational scenarios or optimizing the pipeline to enhance execution time and performance.

Equation 1: End-to-end latency and throughput model (PiPO streaming pipeline)

$$T_{\mathrm{e2e}} = \sum_{k=1}^{K} T_k$$

Each stage latency can be decomposed into **waiting** + **service**:

$$T_k = W_k + S_k$$

- S_k is the actual compute/IO time.

|www.ijrpetm.com | ISSN: 2454-7875 | editor@ijrpetm.com |A Bimonthly, Peer Reviewed & Scholarly Journal|

||Volume 5, Issue 6, November - December 2022||

DOI:10.15662/IJRPETM.2022.0506020

- $W_k$is queueing delay due to contention.

A common model for each stage is M/M/m_k:

- arrivals ~ Poisson with rate λ_k,
- service times exponential with rate $\mu_k$per worker,
- $m_k$parallel workers.

Utilization:

$$\rho_k \;=\; \frac{\lambda_k}{m_k\mu_k}(\text{must satisfy } \rho_k < 1)$$

Then:

$$T_k \approx S_k + W_k \text{with} S_k \approx \frac{1}{\mu_k}$$

The exact $W_k$uses Erlang-C; conceptually, as $\rho_k \uparrow 1$, waiting time blows up. This is why **autoscaling** and **load balancing** matter.

The steady-state throughput is bounded by the slowest (most utilized) stage:

$$\text{Throughput} \;\le\; \min_k (m_k\mu_k)$$

II. BACKGROUND AND RELATED WORK

Optimization of cloud-centric big data pipelines has been studied for a smart healthcare traffic monitoring application by integrating the Paretooptimal non-negative constraint into matrix factorization algorithms to realize a new collaborative filtering-based scheduling scheme. Semantic similarity-based techniques and supervised machine learning classifiers have long been effective in supporting or speeding up the traditional debugging process of cloud-centric big data pipelines. A modern scenario-driven approach is now used to solve cloud-centric big data pipeline debugging problems more systematically and comprehensively.

To support healthcare traffic monitoring, a dedicated pipeline scheduling algorithm is realized based on a proposed camera traffic matrix factorization approach. In a healthcare traffic context, asynchronous and independent user access demands for support and large latent solution spaces constitute two extreme schedules whose costs differ by orders of magnitude. Novel collaborative filtering-based algorithms are thus developed to accelerate tailor-designed data collection and to schedule camera traffic matrices. The resulting scheduling scheme is an efficient integration of Pareto-optimal non-negative matrix completion and low-rank-consistent collaborative-predictor-based delay estimation.

To manage the high-stakes demands of healthcare traffic monitoring, a sophisticated pipeline scheduling algorithm has been developed, centered on a novel **camera traffic matrix factorization** approach. This system addresses the massive disparity between asynchronous user access and the vast latent solution spaces inherent in medical data environments—two extremes where scheduling inefficiencies can lead to cost differences spanning several orders of magnitude.

By utilizing **collaborative filtering-based algorithms**, the framework accelerates data collection and optimizes the scheduling of camera traffic matrices. The architecture effectively merges **Pareto-optimal non-negative matrix completion** with **low-rank-consistent collaborative predictors**, ensuring that delay estimation is both precise and computationally efficient. This integration results in a streamlined scheduling scheme that balances real-time monitoring needs with the rigorous data integrity required in healthcare settings.

|www.ijrpetm.com | ISSN: 2454-7875 | editor@ijrpetm.com |A Bimonthly, Peer Reviewed & Scholarly Journal|

||Volume 5, Issue 6, November - December 2022||

DOI:10.15662/IJRPETM.2022.0506020

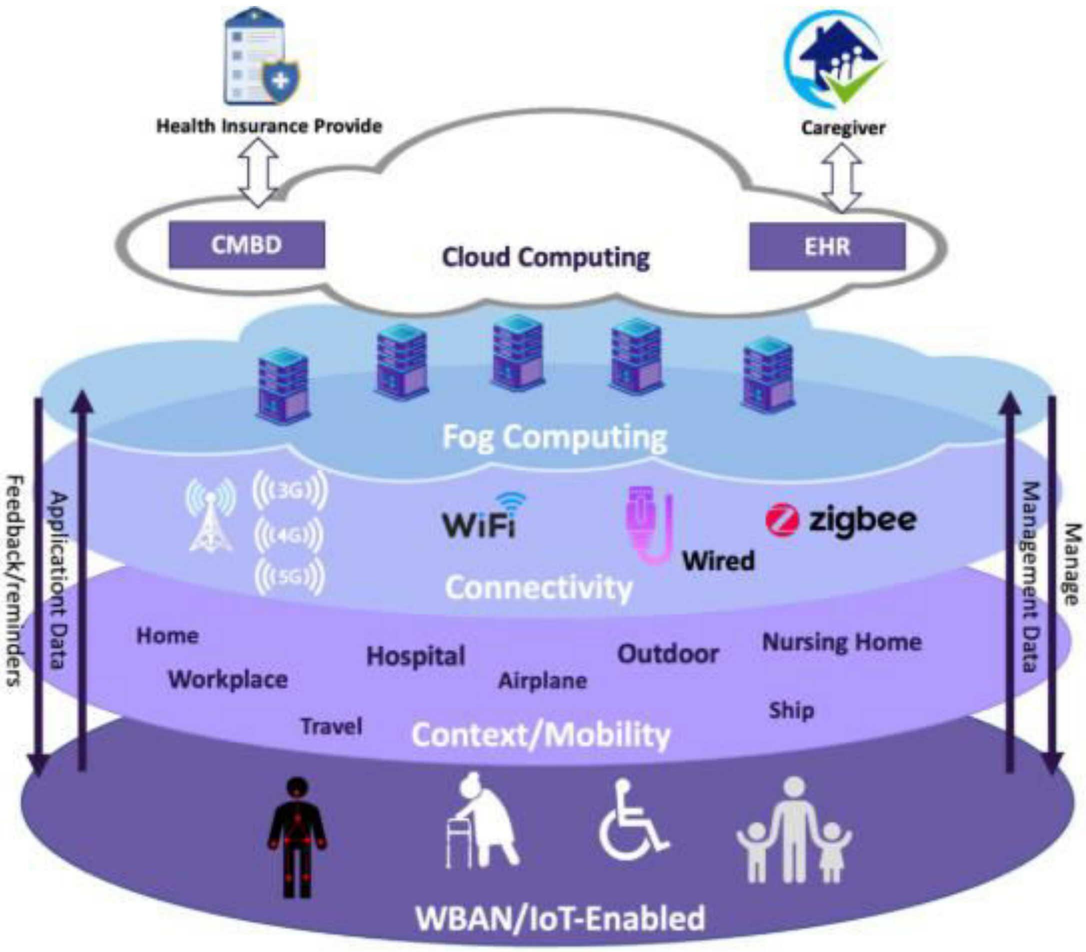


Fig 2: Background and Related Work of Cloud-Centric Big Data Pipeline

2.1. Research design

The purpose of cloud-centric big data pipeline optimization for a smart healthcare monitoring system is to optimize the usability of a cloud-centric distributed data pipeline, improve resource utilization, and reduce user delay for the subsequent data mining process. This optimization step addresses the energy consumption of cloud-centric distributed data pipelines by using a connection with a predictive service monitoring module to extend the model with the prediction of the load over each pipeline stage. Applying this solution to a healthcare environment makes it possible to optimize simultaneously usability and energy consumption. To achieve these goals, a multi-objective optimization model3 is considered the scientific basis for the proposed optimization processes of the cloud-centric big data pipeline and for the overall optimization of the usability and warning delay of the healthcare monitoring system.

Equation 2: Ingestion publish-time minimization (the paper's "minimize publish time concurrently")

A basic decomposition:

$$T^{(i)}_{\text{publish}} = T^{(i)}_{\text{conn}} + T^{(i)}_{\text{tx}} + T^{(i)}_{\text{cloud-write}}$$

- Connection (TCP/UDP handshake, auth, etc.):

$$T^{(i)}_{\text{conn}} \approx L_i$$

- Transmission time:

$$T^{(i)}_{\text{tx}} = \frac{s_i}{b_i}$$

- Cloud write time (IO + commit + metadata):

|www.ijrpetm.com | ISSN: 2454-7875 | editor@ijrpetm.com |A Bimonthly, Peer Reviewed & Scholarly Journal|

||Volume 5, Issue 6, November - December 2022||

DOI:10.15662/IJRPETM.2022.0506020

$$T^{(i)}_{\text{cloud-write}} \approx \frac{s_i}{B_{\text{cloud}}} + c_{\text{commit}}$$

So:

$$T^{(i)}_{\text{publish}} \approx L_i + \frac{s_i}{b_i} + \frac{s_i}{B_{\text{cloud}}} + c_{\text{commit}}$$

III. ARCHITECTURAL OVERVIEW OF CLOUD-CENTRIC PIPELINES

Cloud-centric pipelines (CCPs) allow for a new category of realistic applications that fit well into the buzzwords of Big Data and the Internet of Things (IoT). These applications are characterized by their unique data structure and distribution: small-length, coarse-grained data flows that arrive over very long periods of time, during which they occupy little or no computational resources on the cloud. CCPs exploit the distributed resources of the cloud for their heavy-lifting machine-learning capabilities—for example, training supervised models—while relocating the cost of data acquisition, pre-processing, fusion, and pattern matching closer to the source where sufficient computing and storage resources exist. Moreover, the mobile devices surrounding the data source have access to both the cloud and the service-specific part of the data pipeline.

Architectural insight into a specific class of CCPs—those tuned for smart healthcare monitoring systems—is provided using the example of a monitoring service for patients with heart-failure disease. Patients are monitored through wearable sensors that continuously collect their electrocardiogram (ECG) signals. When the length of a monitoring session is detected as significant around the occurrence of a heart failure (HF) event, an ECG-based train-test strategy is applied to enhance model accuracy by training the diagnosis model with the ECG data of the test user. Such a smart healthcare monitoring system transforms the data acquisition/pattern-matching/localized machine-learning service logic into a public monitoring service for multiple patients, thus creating a cloud-centric Big Data pipeline.

Fig 3: Architectural Overview of Cloud-Centric

3.1. Data Ingestion and Ingestion Layer Optimization

To achieve smooth and reliable publishing of skills-related data onto cloud storage, this layer requires a data ingestion subsystem interfacing with the cloud storage provider applied in the monitoring use case. This step has been designed so that a server appropriately modified, based on the Python™ programming language, manages TCP/UDP connections with a flexible number of connected sensors; can publish all the monitoring data onto cloud storage by writing them into either a Microsoft™ Azure storage service or an Amazon Web Services (AWS) Simple Storage Service (S3); and automatically closes the TCP/UDP connection for each sensor once data publishing is completed. The data ingestion subsystem manages the resources of the chosen cloud storage provider, ensuring that all the data coming from the sensors are properly contained.

Normal behavior has been thoroughly tested. Nonetheless, in a real-world scenario, the data ingestion subsystem may experience some malfunctions (e.g., a broken connection with a sensor or the storage provider, internal application errors). The proposed cloud-centric BDP optimization is evaluated through simulations and at a significantly lower level than the other service layers (e.g., data processing, data display, data management), since the availability of an operational data ingestion layer is a necessary condition for the functioning of all other services and, of course, the

|www.ijrpetm.com | ISSN: 2454-7875 | editor@ijrpetm.com |A Bimonthly, Peer Reviewed & Scholarly Journal|

||Volume 5, Issue 6, November - December 2022||

DOI:10.15662/IJRPETM.2022.0506020

entire monitoring system itself. Therefore, the aim of the presented optimization is to minimize the publish time of the data storage operations when they are executed concurrently by a subgroup of the monitoring sensors in a cloud-centric architecture.

Equation 3: Concurrent publishing: shared bottleneck effect

If Msensors publish at once and share bandwidth, a simple fair-share approximation:

$$b_i^{\text{eff}} \approx \frac{b_{\text{uplink}}}{M} \Rightarrow T_{\text{tx}}^{(i)} \approx \frac{s_i}{b_{\text{uplink}}/M} = \frac{M s_i}{b_{\text{uplink}}}$$

Similarly, if cloud write bandwidth is shared:

$$B_{\text{cloud}}^{\text{eff}} \approx \frac{B_{\text{cloud,total}}}{M} \Rightarrow T_{\text{cloud-write}}^{(i)} \approx \frac{M s_i}{B_{\text{cloud,total}}} + c_{\text{commit}}$$

A natural objective that matches “minimize publish time operations executed concurrently by a subgroup”

Cloud-Centric Big Data Pipeline…

:

Let $\mathcal{G}$be the subgroup of sensors publishing concurrently.

$$\min \quad \max_{i \in \mathcal{G}} \; T_{\text{publish}}^{(i)}$$

Decision variables typically include:

- scheduling/which sensors publish together,
- per-sensor rate limits,
- chunk sizes (multipart upload),
- concurrency M,
- ingestion server thread pool size.

IV. RESOURCE MANAGEMENT IN THE CLOUD FOR HEALTHCARE DATA

The wide adoption of wireless sensor networks (WSNs), smart mobile devices, and cloud computing in healthcare allows patients with chronic diseases to receive treatment and prognosis at home. Front-end devices collect patients' health data, while a cloud-based big data platform is used for data storage, analysis, and prognosis. Smartphone users often access cloud services via mobile networks. Many users live nearby, making it easy to help neighbors with similar health issues during collection and analysis.

The data communication model for a cloud-centric mobile health (mHealth) monitoring system analyzes sensor data and builds a health pipeline. If the data are dense, heterogeneous, and collected over a period of time, the transmission overhead for data delivery to the cloud is minimized. The same analysis pipeline is not suitable for a large-scale system that processes hundreds of thousands of data points. A multi-target prediction model selects the analysis pipeline with the largest potential neighborhood. By grouping users living close to each other, the wireless data are relayed during collection and prediction.

|www.ijrpetm.com | ISSN: 2454-7875 | editor@ijrpetm.com |A Bimonthly, Peer Reviewed & Scholarly Journal|

||Volume 5, Issue 6, November - December 2022||

DOI:10.15662/IJRPETM.2022.0506020

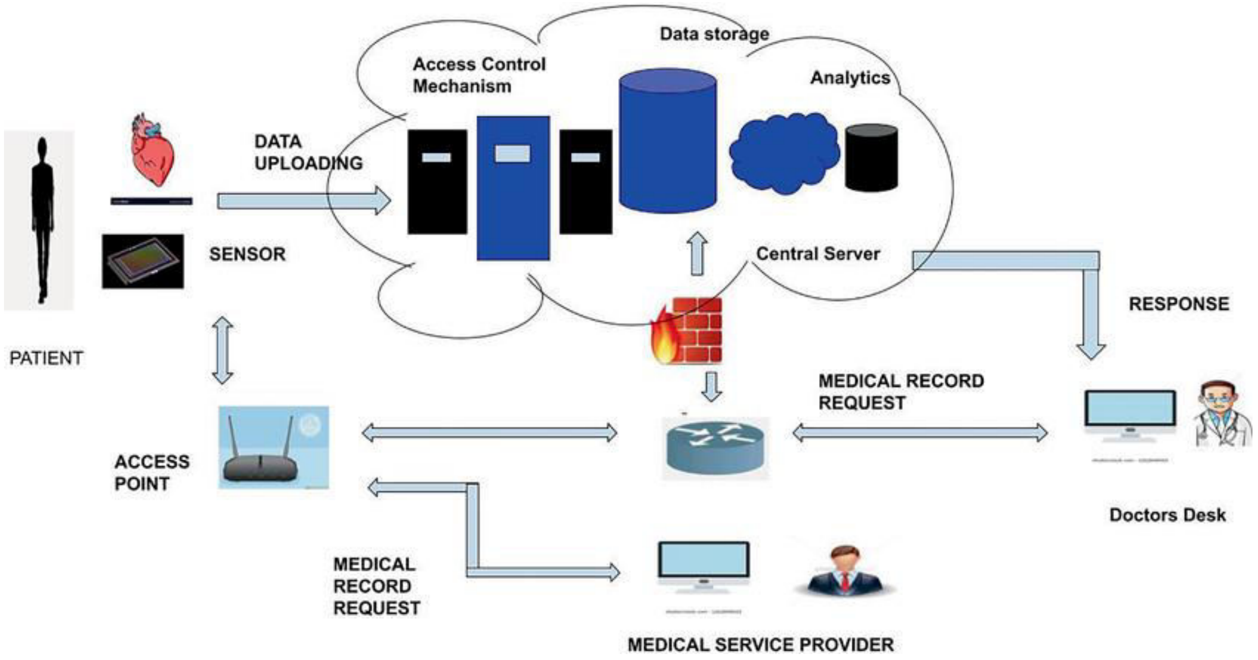


Fig 4: A cloud-based health data management structure

4.1. Compute Autoscaling and Cost-Efficiency

This section details the techniques and algorithms employed for optimizing compute autoscaling for cost-efficiency in healthcare systems based on the consideration of the number of arrivals within a specified period.

The cloud-centric architecture of the healthcare system allows services to be deployed in the cloud, enabling cloud users to enlarge or shrink the Compute cluster, thereby reducing resource costs. The elasticity of cloud resources and the changing nature of price models make it necessary to model the execution costs of services mathematically. In this regard, Chen et al. propose an autoregressive integrated moving average (ARIMA)-based cost-dependent autoscaling algorithm. It formulates the CPU usage of the services as a single-variable function of service rates and execution costs, and then derives the scaling-up and scaling-down thresholds. The scaling-up decision takes place when the predicted CPU usage exceeds the scaling-up threshold; conversely, the scaling-down decision occurs when the predicted CPU usage falls below the scaling-down threshold. The optimal CPU usage at the given time is used to determine the optimal size of Compute cluster that minimizes service execution costs within the arriving-rate prediction period.

Equation 4: Multi-objective optimization model (delay vs energy/cost) and Pareto optimality

1. **User delay / warning delay** (end-to-end latency):

$$f_1(\mathbf{m}) = T_{\mathrm{e2e}}(\mathbf{m}) = \sum_k T_k\,(m_k)$$

2. **Energy or execution cost** (sum over allocated instances):

$$f_2(\mathbf{m}) = \sum_k (\,c_k^{\mathrm{instance}} \cdot m_k) + c_k^{\mathrm{data}}$$

where $\mathbf{m} = (m_1, \dots, m_K)$.

Constraints (typical):

- capacity constraint: $\rho_k(\mathbf{m}) < 1$
- budget constraint: $f_2(\mathbf{m}) \le C_{\max}$
- latency constraint: $f_1(\mathbf{m}) \le L_{\max}$

A solution $\mathbf{m}^a$ dominates $\mathbf{m}^b$ if:

$$f_j(\mathbf{m}^a) \le f_j(\mathbf{m}^b)\ \forall j \text{and} \exists j: f_j(\mathbf{m}^a) < f_j(\mathbf{m}^b)$$

The **Pareto front** is the set of non-dominated solutions.

|www.ijrpetm.com | ISSN: 2454-7875 | editor@ijrpetm.com |A Bimonthly, Peer Reviewed & Scholarly Journal|

||Volume 5, Issue 6, November - December 2022||

DOI:10.15662/IJRPETM.2022.0506020

One way to optimize with PSO:

$$\min_{\mathbf{m}} \quad J(\mathbf{m}) = w_1 \hat{f}_1(\mathbf{m}) + w_2 \hat{f}_2(\mathbf{m})$$

where $\hat{f}$ are normalized versions (so different units don't break optimization):

$$\hat{f}_j(\mathbf{m}) = \frac{f_j(\mathbf{m}) - f_j^{\min}}{f_j^{\max} - f_j^{\min}}$$

V. QUALITY OF SERVICE AND RELIABILITY MECHANISMS

The advent of cloud computing and wearable health monitoring devices has dramatically changed the traditional healthcare system. Health monitoring devices warn patients and physicians about abnormal health conditions. As a patient-centric healthcare framework, all patient data can be stored in the cloud after they pass the quality of service (QoS) parameters. The healthcare monitoring system (HMS) administrators request the data for decision-making purposes. Moreover, a number of multiple users update their records using the HMS-assisted mobile application in a group setting.

QoS parameters play a key role in big data analytics pipelines. The incoming patients' health data are compared with the past data, and rules are generated. Reliable storage in a cloud environment is achieved using components like Amazon S3. Also, the Quality of Service-aware pipeline is enhanced by the Map with Filter and Reduce pattern. Here, the Rule Matching, Rule Invalidation, and MQW routes are modified to support Hopcroft's NFA learning algorithm. The proposed approach is evaluated against existing approaches, demonstrating significant improvement.

Equation 5: Particle Swarm Optimization (PSO) equations (step-by-step)
Let each particle represent a candidate resource allocation vector:

$$\mathbf{x} = (m_1, \ldots, m_K)$$

At iteration t:

$$\mathbf{v}^{t+1} = \omega \mathbf{v}^t + c_1 r_1 (\mathbf{p}^t - \mathbf{x}^t) + c_2 r_2 (\mathbf{g}^t - \mathbf{x}^t)$$

Where:
- ω= inertia weight,
- c_1, c_2= cognitive/social constants,
- $r_1, r_2 \sim U(0,1)$,
- $\mathbf{p}^t$= particle's personal best position so far,
- $\mathbf{g}^t$= global best position so far.

$$\mathbf{x}^{t+1} = \mathbf{x}^t + \mathbf{v}^{t+1}$$

Since m_k must be integer:

$$m_k \leftarrow \max(1, \text{round}(x_k))$$

If constraints are violated, penalize objective:

$$J_{\text{pen}}(\mathbf{x}) = J(\mathbf{x}) + \alpha \sum_k \max(0, \rho_k(\mathbf{x}) - 1)^2$$

5.1. Fault Tolerance and Latency Guarantees

Fault tolerance is a property that prevents the system from failing while delivering continuous operation in the face of faults. It guarantees the reliable transmission and processing of incoming messages. Users can specify the desired degree of fault tolerance, which is directly related to the degree of redundancy configured by the operator in the stream processing network. Latency is defined as the time delay between the event occurrence and the computation derived from the occurrence of that event. Thus, it reflects the performance of the system related to the speed with which it detects changes in the environment and generates updated results. Users are mainly interested in setting latency upper bounds, which indicates how fast the system reacts.
Latency guarantees can be satisfied through an appropriate placement of the operators, adequate resource allocation, and replication of the databases storing the state of the operators. Fault tolerance guarantees can be offered through

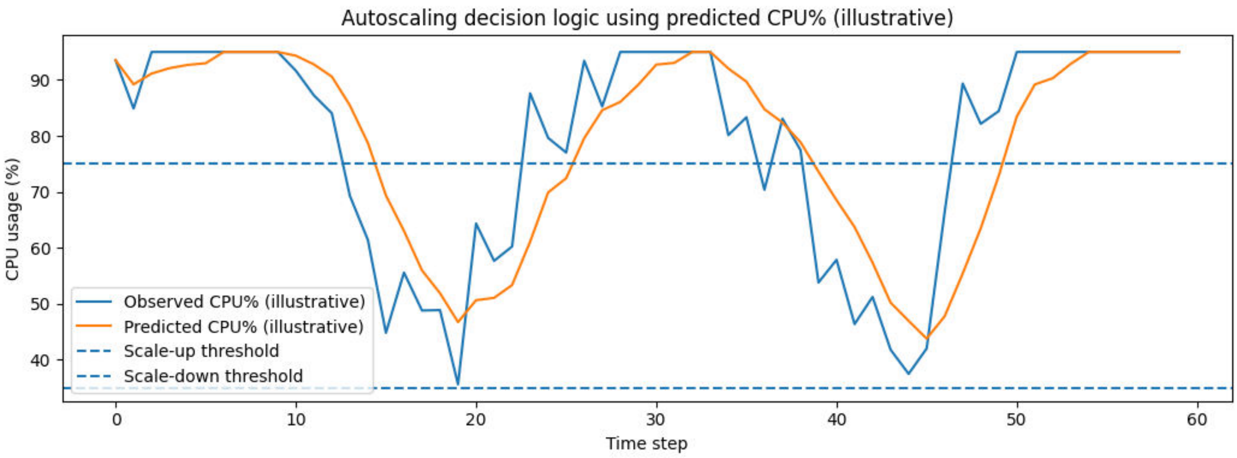
International Journal of Research Publications in Engineering, Technology and Management (IJRPETM)

|www.ijrpetm.com | ISSN: 2454-7875 | editor@ijrpetm.com |A Bimonthly, Peer Reviewed & Scholarly Journal|

||Volume 5, Issue 6, November - December 2022||

DOI:10.15662/IJRPETM.2022.0506020

message replication (executors that replicate the same message on a fault-tolerant storage) and operator redundancy (multiple copies of an operator that perform the same processing on the same input streams). The use of message tokens counteracts the duplication of the output stream. The user specifies the fault tolerance level, which relates the degree of redundancy configured by the system operator with the degree of limited fault tolerance and latency that nodes in the system can provide.



VI. MACHINE LEARNING AND ANALYTICS FOR SMART MONITORING

Cloud-centric big data pipeline optimization in healthcare monitoring systems requires The application of artificial intelligence (AI) tools using machine learning methods, along with advanced analytics processing, can improve security and health for humans and the environment. AI tools can be deployed at the edge, in the cloud, or both. NAT-based Network Address Translation (NAT) technology provides an Internet Protocol (IP) address for cloud service access. The AI state naturally reflects its deployment location. The learning model can be stored outside the path or inside, like in an Overlay Network considering the reflection continuity. Consideration of the periodical resource usage, hourly or daily, over time introduces the repetitive resource factor for modeling and micro-scheduling system. Advanced Analytics analyze streams and data considering the data dynamic technology, using accelerated processing units (APUs), and discovering smart energy and smart Fiji resources.

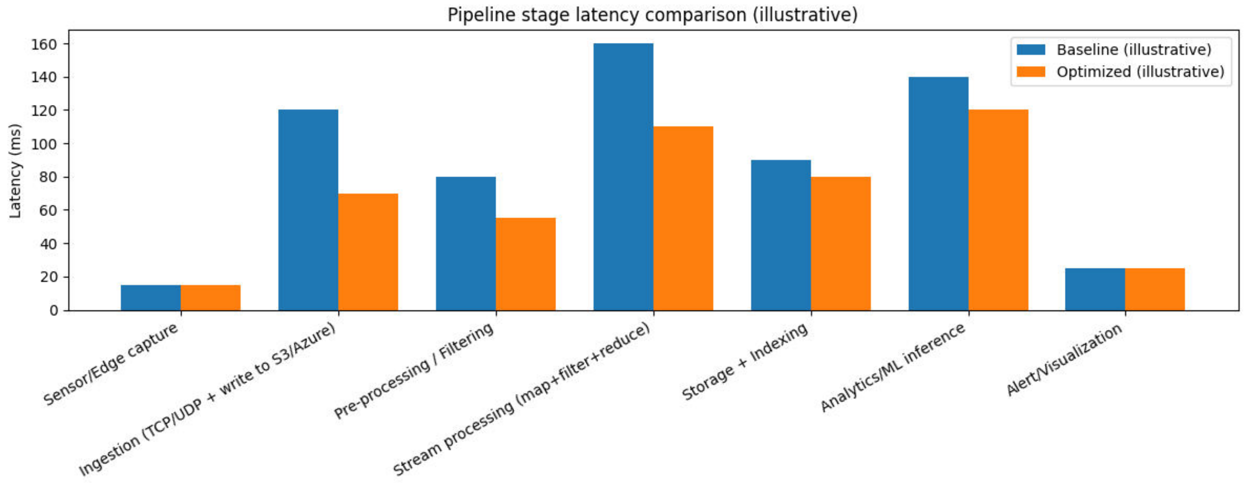
Users can apply AI tools to resolve any demanding question across all categories. For example, analyzing the sea resources and their networks leads to connected thinking applied to social land resources, their economy, and health & wellness economy. The social land resources topic can be developed, exploring security and health smart resources and determining and proving a natural order. Integrating contributive thinking into connected thinking can determine the natural role of the web in human life. Enhancing the use of detection-distribution generation micro-technology helps check human action security and health, the behavior of the earth and sea environment, and formal predicting methods' support for recommendations.

International Journal of Research Publications in Engineering, Technology and Management (IJRPETM)

|www.ijrpetm.com | ISSN: 2454-7875 | editor@ijrpetm.com |A Bimonthly, Peer Reviewed & Scholarly Journal|

||Volume 5, Issue 6, November - December 2022||

DOI:10.15662/IJRPETM.2022.0506020



6.1. Model Training, Deployment, and Drift Management

A learning system must be trained before deployment, which typically requires labelled data. It may also require tuning of hyper-parameters, which is often achieved using hold-out sets during development. Tools such as Tensorflow include model training utilities such as tf.keras.Model.fit, which apply a rich set of optimizations during training, de-coupling the model definition from the training procedure, hyper-parameter tuning, and early stopping to mitigate overfitting. Models are then exported for use within Tensorflow Serving. Model drift refers to the scenario in which the model's output becomes less reliable over time. However, labels may not be available in the wild to retrain the model. Validating model drift assumes that there exists a representative set of labelled data that can be used to estimate the model's performance in the wild. Ana- tomically-based detection can label unseen images by comparing their relationship to previously seen subjects to identify those that are semantically different. Other labeling approaches have introduced effort in the labeling phase, support incremental learning of new content, and enable transfer learning between different neural network archi- tectures. Gap detection identifies anomalous gaps between observed and expected values within the predicted distributions. Tools such as MLflow integrate closely with third-party monitoring systems that expose metrics, such as model performance, drift score, and availability.

VII. CONCLUSION

This paper proposes a smart healthcare monitoring system based on the Internet of Things (IoT) paradigm. An IoT healthcare monitoring system is developed to study the challenges of cloud-centric big data pipelines, including edge and fog platforms, a distributed IoT system, cloud infrastructure, and healthcare big data processor tools. The healthcare monitoring system can be assisted by the Fog, cloud, and Edge data pipelines. Smart healthcare monitoring can provide IoT data pipeline services, including systems using big data in cloud and/or fog locations, and simple cloud (+Edge) and/or health-sensitive monitoring in low-cost, opportunistic-edge applications.

The literature shows that one or more advanced data processing pipeline structures can work for advanced smart healthcare monitoring systems. Applications that generate minute-by-minute data typically require cloud-centric Data pipelines using temporal plus credit and/or unexpected events as triggers. Cloud-centering data volume distribution is also found to help low-cost opportunistic-edge smart healthcare monitoring using temporal monitoring. Such solutions represent a step forward in the long-term vision of full operational services, with a technology area point of view, and provide guidelines and best practices for further target research work.

Pipeline stage	Baseline latency (ms) — illustrative	Optimized latency (ms) — illustrative	Improvement (ms)
Sensor/Edge capture	15.0	15.0	0.0
Ingestion (TCP/UDP + write to S3/Azure)	120.0	70.0	50.0
Pre-processing / Filtering	80.0	55.0	25.0
Stream processing	160.0	110.0	50.0

|www.ijrpetm.com | ISSN: 2454-7875 | editor@ijrpetm.com |A Bimonthly, Peer Reviewed & Scholarly Journal|

||Volume 5, Issue 6, November - December 2022||

DOI:10.15662/IJRPETM.2022.0506020

Pipeline stage	Baseline latency (ms) — illustrative	Optimized latency (ms) — illustrative	Improvement (ms)
(map+filter+reduce)			

Table : Illustrative pipeline stage latency table

7.1. Future Trends

The data flood of the era triggers a change of paradigm in data processing. In the past, people only focused on managing the data and maintaining data center machines to guarantee data availability; however, nowadays, the focus is on data exploitation. Five features of the data flood that affect the data center in a business-oriented manner can be summarized: the data explosion, data growth, data continued availability, data storage complexity, and data explosion.

Through the trends of edge computing, fog computing, and serverless computing, the data flood becomes a data cyclone nowadays. Edge computing brings computation closer to the source of the data. By deploying services at different locations, the response time of services can be improved. The services are deployed in the maintenance-free locations with rich resources by fog computing, while serverless computing offloads the server management workload back to the cloud provider, letting the user focus on writing the code only. However, a more aggressive approach would be to push all the data back to the cloud after on-premise processing using edge devices or by on-premise private cloud, requiring lower resources at the edge or fog. Thanks to the development of technology enrichment, the behavior of the smart edge minimizes the amount of data sent to cloud easily.

REFERENCES

[1] Maguluri, K. K., Pandugula, C., Kalisetty, S., & Mallesham, G. (2022). Advancing Pain Medicine with AI and Neural Networks: Predictive Analytics and Personalized Treatment Plans for Chronic and Acute Pain Managements. Journal of Artificial Intelligence and Big Data, 2(1), 112-126.
[2] Goutham Kumar Sheelam, "Semiconductor Innovation for Edge AI: Enabling Ultra-Low Latency in Next-Gen Wireless Networks," International Journal of Advanced Research in Computer and Communication Engineering (IJARCCE), DOI: 10.17148/IJARCCE.2022.111258
[3] Adomavicius, G., & Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. IEEE Transactions on Knowledge and Data Engineering, 17(6), 734–749.
[4] Kothapalli Sondinti, L. R., & Syed, S. (2022). The Impact of Instant Credit Card Issuance and Personalized Financial Solutions on Enhancing Customer Experience in the Digital Banking Era. Universal Journal of Finance and Economics, 1(1), 1223. Retrieved from https://www.scipublications.com/journal/index.php/ujfe/article/view/1223
[5] Arasu, A., & Kaushik, R. (2014). Data cleansing: A context dependent approach. Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data, 135–146.
[6] Rongali, S. K. (2020). Predictive Modeling and Machine Learning Frameworks for Early Disease Detection in Healthcare Data Systems. Current Research in Public Health, 1(1), 1-15.
[7] Armbrust, M., Das, T., Davidson, A., Ghodsi, A., Or, A., Rosen, J., Stoica, I., Wendell, P., Xin, R., & Zaharia, M. (2021). Delta Lake: High-performance ACID table storage over cloud object stores. Proceedings of the VLDB Endowment, 13(12), 3411–3424.
[8] Inala, R. Advancing Group Insurance Solutions Through Ai-Enhanced Technology Architectures And Big Data Insights.
[9] Alsharo, M., Alnsour, Y., & Alabdallah, M. (2020). How habit affects continuous use: Evidence from Jordan's national health information system. Informatics for Health and Social Care, 45(1), 43–56. https://doi.org/10.1080/17538157.2018.1540423
[10] Chava, K., Chakilam, C., & Recharla, M. (2021). Machine Learning Models for Early Disease Detection: A Big Data Approach to Personalized Healthcare. International Journal of Engineering and Computer Science, 10(12), 25709–25730. https://doi.org/10.18535/ijecs.v10i12.4678
[11] Babcock, J., Chaudhuri, S., & Das, G. (2004). Dynamic sample selection for approximate query processing. Proceedings of the 2004 ACM SIGMOD International Conference on Management of Data, 539–550.
[12] Sriram, H. K. (2022). Advancements in Credit Score Analytics using Deep Learning and Predictive Modeling Techniques. Available at SSRN 5255128.
[13] Bifet, A., & Gavaldà, R. (2007). Learning from time-changing data with adaptive windowing. Proceedings of the 2007 SIAM International Conference on Data Mining, 443–448.

|www.ijrpetm.com | ISSN: 2454-7875 | editor@ijrpetm.com |A Bimonthly, Peer Reviewed & Scholarly Journal|

||Volume 5, Issue 6, November - December 2022||

DOI:10.15662/IJRPETM.2022.0506020

[14] Muthusamy, S., Kannan, S., Lee, M., Sanjairaj, V., Lu, W. F., Fuh, J. Y., ... & Cao, T. (2021). Cover Image, Volume 118, Number 8, August 2021. Biotechnology and Bioengineering, 118(8), i-i.

[15] Kalisetty, S., & Ganti, V. K. A. T. (2019). Transforming the Retail Landscape: Srinivas's Vision for Integrating Advanced Technologies in Supply Chain Efficiency and Customer Experience. Online Journal of Materials Science, 1, 1254.

[16] Gottimukkala, V. R. R. (2021). Digital Signal Processing Challenges in Financial Messaging Systems: Case Studies in High-Volume SWIFT Flows.

[17] Chen, M., Mao, S., & Liu, Y. (2014). Big data: A survey. Mobile Networks and Applications, 19(2), 171–209.

[18] Dwaraka Nath Kummari. (2022). Fiscal Policy Simulation Using AI And Big Data: Improving Government Financial Planning. Kurdish Studies, 10(2), 934–945. https://doi.org/10.53555/ks.v10i2.3855

[19] Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 785–794.

[20] Davuluri, P. N. Event-Driven Compliance Systems: Modernizing Financial Crime Detection Without Machine Intelligence.

[21] Das, T., Zhu, A., Li, S., Narayanamurthy, S., & Bhat, P. (2013). Distributed and fault-tolerant streaming computation in Spark. Proceedings of the ACM Symposium on Cloud Computing, 1–12.

[22] Siva Hemanth Kolla. (2022). Knowledge Retrieval Systems for Enterprise Service Environments. International Journal of Intelligent Systems and Applications in Engineering, 10(3s), 495–506. Retrieved from https://ijisae.org/index.php/IJISAE/article/view/8037

[23] Dean, J., & Ghemawat, S. (2008). MapReduce: Simplified data processing on large clusters. Communications of the ACM, 51(1), 107–113.

[24] Paleti, S. (2022). Financial Innovation through AI and Data Engineering: Rethinking Risk and Compliance in the Banking Industry. Available at SSRN 5250726.

[25] DeCandia, G., Hastorun, D., Jampani, M., Kakulapati, G., Lakshman, A., Pilchin, A., Sivasubramanian, S., Vosshall, P., & Vogels, W. (2007). Dynamo: Amazon's highly available key-value store. Proceedings of the 21st ACM Symposium on Operating Systems Principles, 205–220.

[26] Sriram, H. K., ADUSUPALLI, B., & Malempati, M. (2021). Revolutionizing Risk Assessment and Financial Ecosystems with Smart Automation, Secure Digital Solutions, and Advanced Analytical Frameworks.

[27] Davuluri, P. N. (2020). Improving Data Quality and Lineage in Regulated Financial Data Platforms. Finance and Economics, 1(1), 1-14.

[28] Varri, D. B. S. (2021). Cloud-Native Security Architecture for Hybrid Healthcare Infrastructure. Available at SSRN 5785982.

[29] Elmagarmid, A. K., Ipeirotis, P. G., & Verykios, V. S. (2007). Duplicate record detection: A survey. IEEE Transactions on Knowledge and Data Engineering, 19(1), 1–16.

[30] Dwaraka Nath Kummari,. (2022). Machine Learning Approaches to Real-Time Quality Control in Automotive Assembly Lines. Mathematical Statistician and Engineering Applications, 71(4), 16801–16820. Retrieved from https://philstat.org/index.php/MSEA/article/view/2972

[31] Fader, P. S., Hardie, B. G. S., & Lee, K. L. (2005). "Counting your customers" the easy way: An alternative to the Pareto/NBD model. Marketing Science, 24(2), 275–284.

[32] Inala, R. (2022). Engineering Data Products for Investment Analytics: The Role of Product Master Data and Scalable Big Data Solutions. International Journal of Scientific Research and Modern Technology, 155-171.

[33] Davuluri, P. N. (2020). Improving Data Quality and Lineage in Regulated Financial Data Platforms. Finance and Economics, 1(1), 1-14.

[34] Meda, R. Enabling Sustainable Manufacturing Through AI-Optimized Supply Chains.

[35] Ghemawat, S., Gobioff, H., & Leung, S. T. (2003). The Google file system. Proceedings of the 19th ACM Symposium on Operating Systems Principles, 29–43.

[36] Varri, D. B. S. (2022). A Framework for Cloud-Integrated Database Hardening in Hybrid AWS-Azure Environments: Security Posture Automation Through Wiz-Driven Insights. International Journal of Scientific Research and Modern Technology, 1(12), 216-226.

[37] Yandamuri, U. S. (2021). A Comparative Study of Traditional Reporting Systems versus Real-Time Analytics Dashboards in Enterprise Operations. Universal Journal of Business and Management, 1(1), 1–13. Retrieved from https://www.scipublications.com/journal/index.php/ujbm/article/view/1357

[38] Gottimukkala, V. R. R. (2022). Licensing Innovation in the Financial Messaging Ecosystem: Business Models and Global Compliance Impact. International Journal of Scientific Research and Modern Technology, 1(12), 177-186.

[39] Hastie, T., Tibshirani, R., & Friedman, J. (2009). The elements of statistical learning: Data mining, inference, and prediction (2nd ed.). Springer.

|www.ijrpetm.com | ISSN: 2454-7875 | editor@ijrpetm.com |A Bimonthly, Peer Reviewed & Scholarly Journal|

||Volume 5, Issue 6, November - December 2022||

DOI:10.15662/IJRPETM.2022.0506020

[40] Vadisetty, R., Polamarasetti, A., Guntupalli, R., Raghunath, V., Jyothi, V. K., & Kudithipudi, K. (2022). AI-Driven Cybersecurity: Enhancing Cloud Security with Machine Learning and AI Agents. Sateesh kumar and Raghunath, Vedaprada and Jyothi, Vinaya Kumar and Kudithipudi, Karthik, AI-Driven Cybersecurity: Enhancing Cloud Security with Machine Learning and AI Agents (February 07, 2022).
[41] Kalisetty, S., Vankayalapati, R. K., Reddy, L., Sondinti, K., & Valiki, S. (2022). AI-Native Cloud Platforms: Redefining Scalability and Flexibility in Artificial Intelligence Workflows. Linguistic and Philosophical Investigations, 21(1), 1-15.
[42] Garapati, R. S. (2022). Web-Centric Cloud Framework for Real-Time Monitoring and Risk Prediction in Clinical Trials Using Machine Learning. Current Research in Public Health, 2, 1346.
[43] Hu, Y., Koren, Y., & Volinsky, C. (2008). Collaborative filtering for implicit feedback datasets. Proceedings of the 2008 IEEE International Conference on Data Mining, 263–272.
[44] Amistapuram, K. (2022). Fraud Detection and Risk Modeling in Insurance: Early Adoption of Machine Learning in Claims Processing. Available at SSRN 5741982.
[45] Davuluri, P. S. L. N. (2021). Event-Driven Compliance Systems: Modernizing Financial Crime Detection Without Machine Intelligence. Journal of International Crisis and Risk Communication Research , 339–354. https://doi.org/10.63278/jicrcr.vi.3636
[46] Meda, R. (2022). Integrating Edge AI in Smart Factories: A Case Study from the Paint Manufacturing Industry. International Journal of Science and Research (IJSR), 1473-1489.
[47] Jagadish, H. V., Gehrke, J., Labrinidis, A., Papakonstantinou, Y., Patel, J. M., Ramakrishnan, R., & Shahabi, C. (2014). Big data and its technical challenges. Communications of the ACM, 57(7), 86–94.
[48] Segireddy, A. R. (2020). Cloud Migration Strategies for High-Volume Financial Messaging Systems.
[49] Khatri, V., & Brown, C. V. (2010). Designing data governance. Communications of the ACM, 53(1), 148–152.
[50] Amistapuram, K. (2021). Digital Transformation in Insurance: Migrating Enterprise Policy Systems to .NET Core. Universal Journal of Computer Sciences and Communications, 1(1), 1–17.
[51] Kleppmann, M. (2017). Designing data-intensive applications. O'Reilly Media.
[52] Nagabhyru, K. C. (2022). Bridging Traditional ETL Pipelines with AI Enhanced Data Workflows: Foundations of Intelligent Automation in Data Engineering. Available at SSRN 5505199.
[53] Lahiri, M., & Venkatasubramanian, S. (2013). Robust record linkage. Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data, 101–112.
[54] Rongali, S. K. (2021). Cloud-Native API-Led Integration Using MuleSoft and .NET for Scalable Healthcare Interoperability. Available at SSRN 5814563.
[55] Leskovec, J., Rajaraman, A., & Ullman, J. D. (2014). Mining of massive datasets (2nd ed.). Cambridge University Press.
[56] Rongali, S. K. (2022). AI-Driven Automation in Healthcare Claims and EHR Processing Using MuleSoft and Machine Learning Pipelines. Available at SSRN 5763022.
[57] Linden, G., Smith, B., & York, J. (2003). Amazon.com recommendations: Item-to-item collaborative filtering. IEEE Internet Computing, 7(1), 76–80.
[58] Meda, R. (2021). Digital Infrastructure for Predictive Inventory Management in Retail Using Machine Learning. International Journal of Advanced Research in Computer and Communication Engineering (IJARCCE), DOI, 10.
[59] Lin, J., Kolcz, A., & Szymanski, B. K. (2012). Large-scale machine learning at Twitter. Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data, 793–804.
[60] Sheelam, G. K. Power-Efficient Semiconductors for AI at the Edge: Enabling Scalable Intelligence in Wireless Systems. International Journal of Innovative Research in Electrical, Elec-tronics, Instrumentation and Control Engineering (IJIREEICE), DOI, 10.
[61] Manyika, J., Chui, M., Brown, B., Bughin, J., Dobbs, R., Roxburgh, C., & Byers, A. H. (2011). Big data: The next frontier for innovation, competition, and productivity. McKinsey Global Institute.
[62] Vadisetty, R., Polamarasetti, A., Guntupalli, R., Rongali, S. K., Raghunath, V., Jyothi, V. K., & Kudithipudi, K. (2021). Legal and Ethical Considerations for Hosting GenAI on the Cloud. International Journal of AI, BigData, Computational and Management Studies, 2(2), 28-34.
[63] Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. Proceedings of the International Conference on Learning Representations, 1–12.
[64] Ramesh Inala. (2022). Cross-Domain MDM Integration Using AI-Driven Data Governance: A Case Study In Financial Technology Architecture. Migration Letters, 19(2), 280–304. Retrieved from https://migrationletters.com/index.php/ml/article/view/11982

|www.ijrpetm.com | ISSN: 2454-7875 | editor@ijrpetm.com |A Bimonthly, Peer Reviewed & Scholarly Journal|

||Volume 5, Issue 6, November - December 2022||

DOI:10.15662/IJRPETM.2022.0506020

[65] Vadisetty, R., Polamarasetti, A., Guntupalli, R., Raghunath, V., Jyothi, V. K., & Kudithipudi, K. (2021). Privacy-Preserving Gen AI in Multi-Tenant Cloud Environments. Sateesh kumar and Raghunath, Vedaprada and Jyothi, Vinaya Kumar and Kudithipudi, Karthik, Privacy-Preserving Gen AI in Multi-Tenant Cloud Environments (January 20, 2021).
[66] Aitha, A. R. (2021). Optimizing Data Warehousing for Large Scale Policy Management Using Advanced ETL Frameworks.
[67] Zaharia, M., Chowdhury, M., Franklin, M. J., Shenker, S., & Stoica, I. (2010). Spark: Cluster computing with working sets. Proceedings of the 2nd USENIX Conference on Hot Topics in Cloud Computing, 1–7.
[68] Varri, D. B. S. (2022). AI-Driven Risk Assessment and Compliance Automation in Multi-Cloud Environments. Available at SSRN 5774924.
[69] Zaharia, M., Das, T., Li, H., Shenker, S., & Stoica, I. (2012). Discretized streams: Fault-tolerant streaming computation at scale. Proceedings of the 24th ACM Symposium on Operating Systems Principles, 423–438.
[70] Segireddy, A. R. (2021). Containerization and Microservices in Payment Systems: A Study of Kubernetes and Docker in Financial Applications. Universal Journal of Business and Management, 1(1), 1–17.
[71] Zhai, C., & Massung, S. (2016). Text data management and analysis: A practical introduction to information retrieval and text mining. ACM & Morgan Claypool.
[72] Davuluri, P. N. (2020). Event-Driven Architectures for Real-Time Regulatory Monitoring in Global Banking.
[73] Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching word vectors with subword information. Transactions of the Association for Computational Linguistics, 5, 135–146.
[74] Keerthi Amistapuram , "Energy-Efficient System Design for High-Volume Insurance Applications in Cloud-Native Environments," International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering (IJIREEICE), DOI 10.17148/IJIREEICE.2020.81209
[75] Goutham Kumar Sheelam. (2022). Reconfigurable Semiconductor Architectures For AI-Enhanced Wireless Communication Networks. Kurdish Studies, 10(2), 1027–1040. https://doi.org/10.53555/ks.v10i2.3867
[76] Batarseh, F. A., & Yang, R. (2019). Federal data science: Transforming government and society. Academic Press.
[77] Kolla, S. K. (2021). Architectural Frameworks for Large-Scale Electronic Health Record Data Platforms. Current Research in Public Health, 1(1), 1–19. Retrieved from https://www.scipublications.com/journal/index.php/crph/article/view/1372
[78] Bhasin, H., & Bhatia, P. (2020). Clickstream data mining for web analytics and customer behavior modeling: A review. ACM Computing Surveys, 53(6), 1–34.
[79] Kolla, S. H. (2021). Rule-Based Automation for IT Service Management Workflows. Online Journal of Engineering Sciences, 1(1), 1–14. Retrieved from https://www.scipublications.com/journal/index.php/ojes/article/view/1360
[80] Uday Surendra Yandamuri. (2022). Cloud-Based Data Integration Architectures for Scalable Enterprise Analytics. International Journal of Intelligent Systems and Applications in Engineering, 10(3s), 472–483. Retrieved from https://ijisae.org/index.php/IJISAE/article/view/8005
[81] Abedjan, Z., Golab, L., & Naumann, F. (2016). Profiling relational data: A survey. The VLDB Journal, 24(4), 557–581.
[82] Yandamuri, U. S. (2022). Big Data Pipelines for Cross-Domain Decision Support: A Cloud-Centric Approach. International Journal of Scientific Research and Modern Technology, 1(12), 227–237. https://doi.org/10.38124/ijsrmt.v1i12.1111
[83] Dwaraka Nath Kummari. (2022). AI-Driven Audit Frameworks For Enhancing Compliance In Modern Manufacturing Systems. Migration Letters, 19(S8), 2150–2177. Retrieved from https://migrationletters.com/index.php/ml/article/view/11912
[84] Joseph, A. L., Stringer, E., Borycki, E. M., & Kushniruk, A. W. (2022). Evaluative frameworks and models for health information systems (HIS) and health information technologies (HIT). Studies in Health Technology and Informatics, 289, 280–285. https://doi.org/10.3233/shti210914
[85] Baesens, B., Van Vlasselaer, V., & Verbeke, W. (2021). Fraud analytics using descriptive, predictive, and social network techniques: A guide to data science for fraud detection (2nd ed.). Wiley.
[86] Avinash Reddy Aitha. (2022). Deep Neural Networks for Property Risk Prediction Leveraging Aerial and Satellite Imaging. International Journal of Communication Networks and Information Security (IJCNIS), 14(3), 1308–1318. Retrieved from https://www.ijcnis.org/index.php/ijcnis/article/view/8609
[87] Barocas, S., Hardt, M., & Narayanan, A. (2019). Fairness and machine learning. fairmlbook.org (Book manuscript).
[89] Garapati, R. S. (2022). AI-Augmented Virtual Health Assistant: A Web-Based Solution for Personalized Medication Management and Patient Engagement. Available at SSRN 5639650.

|www.ijrpetm.com | ISSN: 2454-7875 | editor@ijrpetm.com |A Bimonthly, Peer Reviewed & Scholarly Journal|

||Volume 5, Issue 6, November - December 2022||

DOI:10.15662/IJRPETM.2022.0506020

[90] Gottimukkala, V. R. R. (2020). Energy-Efficient Design Patterns for Large-Scale Banking Applications Deployed on AWS Cloud. power, 9(12).
[91] Ahmad, M. A., Eckert, C., & Teredesai, A. (2018). Interpretable machine learning in healthcare. Proceedings of the ACM Conference on Health, Informatics, and Data Science, 1–10.
[92] Aitha, A. R. (2022). Cloud Native ETL Pipelines for Real Time Claims Processing in Large Scale Insurers. Available at SSRN 5532601.
[93] Aljabre, A. (2019). Cloud computing security in healthcare. Journal of King Saud University – Computer and Information Sciences, 31(1), 10–18.
[94] Gadi, A. L. The Role of Digital Twins in Automotive R&D for Rapid Prototyping and System Integration.
[94] Akanfe, O. A. (2022). Advancing digital financial inclusion: Data privacy, regulatory compliance, and cross-country cultural values in digital payment systems use (Doctoral dissertation, The University of Texas at San Antonio).
[95] Avinash Reddy Segireddy. (2022). Terraform and Ansible in Building Resilient Cloud-Native Payment Architectures. International Journal of Intelligent Systems and Applications in Engineering, 10(3s), 444–455. Retrieved from https://www.ijisae.org/index.php/IJISAE/article/view/7905
[96] Digan, W., Neveol, A., Neuraz, A., et al. (2021). Can reproducibility be improved in clinical natural language processing? A study of 7 clinical NLP suites. Journal of the American Medical Informatics Association, 28(3), 504–515. https://doi.org/10.1093/jamia/ocaa261