ACHARYA INSTITUTE OF TECHNOLOGY

Affiliated to Visvesvaraya Technological University, Belagavi, Govt. of Karnataka.
Approved by AICTE, New Delhi

Department of Artificial Intelligence & Machine Learning
and
Computer Science & Engineering (Data Science)

PROJECT REPORT
ON
"Healthcare Analytics — Real-Time ICU Patient Monitoring"

Subject Name: Big Data Analytics
Subject Code: BAD601
Submitted By: Chinmayi.Talawar
1AY23CD016

Submitted To:
Ms. Surbhi

TABLE OF CONTENTS

1. About the Project

1.1 Project Overview

This project focuses on the application of Big Data technologies in Healthcare Analytics, specifically targeting real-time monitoring of patients admitted to the Intensive Care Unit (ICU). The project explores how modern Big Data tools and architectures can transform the way hospitals collect, store, process, and analyze massive volumes of clinical data generated every second from hundreds of patients.

Traditional Business Intelligence tools such as Excel, SQL databases, and conventional data warehouses have long been the standard for data management in hospitals. However, with the exponential growth of medical data — including sensor streams, medical imaging, electronic health records, and voice recordings — these tools have proven inadequate in terms of speed, scale, and data variety.

1.2 Problem Statement

A large hospital network with 500+ ICU beds generates enormous volumes of data continuously. Each patient is connected to 20 or more sensors that record vital signs such as heart rate, blood pressure, oxygen saturation, and temperature every 500 milliseconds. In addition to structured sensor data, the hospital also generates unstructured data in the form of MRI and CT scan images, doctor voice notes, nursing narratives, and lab reports.

The core problem is that traditional systems cannot handle this data at the required speed, scale, and variety. Reports are generated in batches, typically overnight, which means critical patient deterioration events go undetected until it is too late. The need for a real-time, scalable, and intelligent system is urgent.

1.3 Project Objectives

- To understand the limitations of traditional Business Intelligence tools in healthcare settings.
- To design a Big Data architecture capable of handling real-time ICU monitoring data.
- To classify and analyze the different types of data generated in an ICU environment.
- To explore the application of machine learning models for predictive patient monitoring.
- To demonstrate how Big Data can reduce ICU mortality rates through early intervention.

1.4 Scope of the Project

The scope of this project covers the entire data lifecycle in an ICU setting — from data generation at the patient bedside to real-time analytics and predictive alerting. It includes the design of two architectures: a traditional data warehouse architecture and a modern Hadoop-based Big Data architecture. The project compares both approaches and demonstrates the superiority of Big Data tools for this use case.

1.5 Application Domain

Domain: Healthcare Analytics
Sub-domain: ICU Real-Time Patient Monitoring
Data generated per day: 5+ Terabytes per hospital
Number of patients monitored: 500+ ICU beds
Data types involved: Structured, Semi-structured, Unstructured

2. About the Tools and Technologies Used

2.1 Apache Kafka

Apache Kafka is an open-source distributed event streaming platform developed by the Apache Software Foundation. It was originally created at LinkedIn and later open-sourced in 2011. Kafka is designed to handle real-time data feeds with high throughput, fault tolerance, and low latency.

In the context of ICU monitoring, Kafka acts as the data ingestion highway. It collects streaming data from all 500 patients' sensors simultaneously — handling over 10,000 events per second — and routes each data stream to the appropriate storage or processing system in real time. Without Kafka, the system would be overwhelmed by the sheer velocity of incoming sensor data.

- Type: Message Streaming Platform
- Key Feature: Handles millions of events per second with zero data loss
- Use in Project: Real-time ingestion of all ICU sensor streams

2.2 Apache Hadoop and HDFS

Apache Hadoop is an open-source framework for distributed storage and processing of large datasets across clusters of computers. It was inspired by Google's MapReduce and Google File System papers and has become the foundation of Big Data infrastructure worldwide.

HDFS (Hadoop Distributed File System) is the storage component of Hadoop. Unlike traditional file systems that store data on a single server, HDFS distributes data across hundreds or thousands of commodity machines. This makes it fault-tolerant, scalable, and capable of storing any type of data — structured, semi-structured, or unstructured.

- Type: Distributed Storage Framework
- Key Feature: Stores petabytes of any data type across commodity hardware
- Use in Project: Storing all ICU data — vitals, MRI scans, audio notes, JSON logs

2.3 Apache Spark

Apache Spark is an open-source unified analytics engine for large-scale data processing. It was developed at UC Berkeley's AMPLab in 2009 and donated to the Apache Software Foundation in 2013. Spark is known for being up to 100 times faster than Hadoop's MapReduce for certain applications because it processes data in memory rather than reading and writing to disk repeatedly.

In the ICU monitoring system, Spark is the processing engine that analyzes incoming data streams in real time. It runs machine learning models on live patient data and generates alerts within milliseconds when it detects warning signs of deterioration.

- Type: In-Memory Data Processing Engine
- Key Feature: 100x faster than MapReduce, supports real-time stream processing
- Use in Project: Real-time processing of sensor data and ML model execution

2.4 NoSQL Databases — MongoDB and Cassandra

NoSQL databases are non-relational database systems designed to handle large volumes of unstructured or semi-structured data. Unlike SQL databases that require a fixed schema, NoSQL databases are flexible and can store diverse data formats.

MongoDB is a document-oriented NoSQL database that stores data as flexible JSON-like documents. It is ideal for storing patient health records in HL7 and FHIR formats where different patients may have different data fields. Cassandra is a wide-column NoSQL database optimized for handling time-series data with high write throughput — perfect for continuous ICU sensor readings arriving every 500 milliseconds.

- MongoDB — Use: Storing flexible patient EHR records and JSON sensor metadata
- Cassandra — Use: Storing continuous time-series vital sign readings at high velocity

2.5 Spark MLlib — Machine Learning

Spark MLlib is the machine learning library built into Apache Spark. It provides scalable implementations of common machine learning algorithms including classification, regression, clustering, and collaborative filtering. MLlib can train models on massive datasets that would be impossible to process on a single machine.

In the ICU monitoring project, MLlib trains predictive models on millions of historical patient records. These models learn the subtle early warning patterns of conditions like sepsis, cardiac arrest, and respiratory failure. Once trained, the models are deployed in real time to monitor live patient data and predict deterioration up to 6 hours in advance.

- Type: Scalable Machine Learning Library
- Key Feature: Trains on petabyte-scale datasets using distributed computing
- Use in Project: Predicting sepsis and cardiac events 6 hours before they occur

2.6 Apache Hive

Apache Hive is a data warehouse software built on top of Hadoop that provides SQL-like querying capabilities for large datasets stored in HDFS. It translates SQL queries into MapReduce or Spark jobs, making it accessible for analysts who are familiar with SQL but need to work with Big Data.

- Type: Data Warehouse / Query Engine
- Use in Project: Running analytical queries on historical ICU data for reporting

2.7 Tools Summary Table

Tool	Purpose	Use in ICU Project
Apache Kafka	Real-time data ingestion	Collects 10,000+ sensor events/sec
HDFS (Hadoop)	Distributed storage	Stores all ICU data types
Apache Spark	Fast data processing	Real-time alerts in < 1 second
MongoDB	NoSQL document store	Patient EHR records (JSON/FHIR)
Cassandra	NoSQL time-series store	Continuous vital sign streams
Spark MLlib	Machine learning	Predicts sepsis 6 hours early
Apache Hive	SQL queries on Big Data	Historical analysis and reports

Table 1 : Tools Summary Table

3. Detailed Description of Contribution

3.1 What Was Done

This project involved a comprehensive study and design of a Big Data-based healthcare analytics system for real-time ICU patient monitoring. The contribution covers four major areas:

First, a thorough analysis of the problem was conducted — understanding why traditional Business Intelligence tools such as Excel, SQL databases, and conventional data warehouses fail to meet the demands of modern ICU monitoring. This involved studying the volume, velocity, variety, veracity, and value of data generated in an ICU environment.

Second, a complete data classification exercise was performed — identifying and categorizing all data types generated in an ICU into structured data (vital signs, patient demographics, medication records), semi-structured data (HL7/FHIR health records, JSON sensor logs, wearable device metadata), and unstructured data (MRI/CT scan images, ECG waveforms, doctor voice recordings, nursing notes).

Third, two complete system architectures were designed and compared — a traditional data warehouse architecture using SQL and ETL batch processing, and a modern Hadoop-based Big Data architecture using Kafka, HDFS, Spark, NoSQL databases, and machine learning.

Fourth, an analytics framework was developed mapping four types of analytics — descriptive, diagnostic, predictive, and prescriptive — to specific business questions, tools, and real ICU use cases.

3.2 How It Was Done

The project was executed in a structured phases:

Phase 1 — Research and Problem Definition:

Studied the current state of healthcare data management. Identified the key pain points of traditional BI systems in hospital settings. Defined the scope and objectives of the project around a realistic scenario of a 500-bed ICU network.

Phase 2 — Data Classification and 5V Analysis:

Analyzed all data types generated in the ICU. Applied the 5 V's framework (Volume, Velocity, Variety, Veracity, Value) with real statistics — 5TB per day, 500ms sensor intervals, 1000+ data types, 15% noisy data, 20% mortality reduction potential.

Phase 3 — Architecture Design:

Designed the Traditional Data Warehouse architecture (data sources → ETL → SQL warehouse → OLAP → BI reports) and the Hadoop Big Data architecture (data sources → Kafka → HDFS/NoSQL → Spark → ML → dashboards). Both architectures were documented with clear layer-by-layer descriptions.

Phase 4 — Analytics Framework Development:
Created a complete analytics type mapping covering all four levels — descriptive (what happened), diagnostic (why it happened), predictive (what will happen), and prescriptive (what action to take) — with appropriate tools and ICU-specific examples for each.

4. Implementation — Methodology and Approach

4.1 System Design Overview

Since this project is theoretical in nature, the implementation focuses on the detailed design and methodology of the proposed Big Data system rather than actual code execution. The implementation covers the complete data flow from raw sensor input to actionable medical insights.

4.2 Data Flow — Step by Step

Step 1 — Data Generation:

Every ICU patient is connected to multiple monitoring devices — cardiac monitors, pulse oximeters, blood pressure cuffs, ventilators, infusion pumps. Each device generates readings continuously. A single patient produces approximately 2,000 data points per hour across all connected devices.

Step 2 — Real-Time Ingestion via Kafka:

Apache Kafka acts as the central message broker. Each monitoring device publishes its data to a dedicated Kafka topic. Kafka consumers subscribe to these topics and forward data to the storage layer. Kafka guarantees zero data loss and can handle the combined stream of all 500 patients simultaneously.

Step 3 — Distributed Storage in HDFS and NoSQL:

Structured vital sign data is stored in Apache Cassandra for fast time-series retrieval. Flexible patient health records in JSON/FHIR format are stored in MongoDB. Raw unstructured data — MRI images, audio recordings, ECG waveforms — are stored directly in HDFS. This three-tier storage approach ensures each data type is handled by the most appropriate database.

Step 4 — Real-Time Processing via Apache Spark:

Apache Spark Streaming continuously reads from Kafka topics and processes incoming data in micro-batches of 500 milliseconds. For each batch, Spark applies rule-based thresholds (e.g., heart rate above 120 bpm triggers an alert) and also runs the pre-trained machine learning model to score each patient's current risk level.

Step 5 — Machine Learning Prediction:

The Spark MLlib model was conceptually trained on historical ICU records covering millions of patient hours. The model uses a combination of features including current vital signs, trends over the past 2 hours, patient demographics, and existing diagnoses to compute a sepsis risk score for each patient every minute. If the risk score exceeds a defined threshold, an alert is immediately triggered.

Step 6 — Alert and Dashboard:

Alerts are pushed to the nursing station dashboard and to the on-call doctor's mobile device within 1 second of detection. The dashboard displays all 500 patients in a grid with color-coded risk indicators. Historical trends and Hive-generated analytical reports are accessible through a separate reporting module.

4.3 Traditional Architecture vs Big Data Architecture

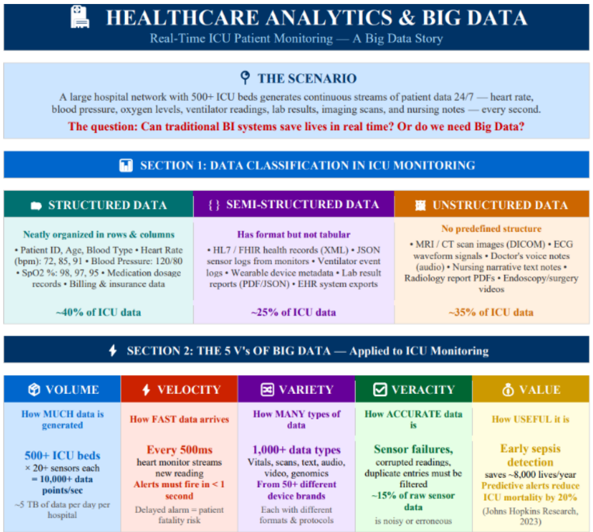
Feature	Traditional DW	Hadoop Big Data
Processing Speed	Batch — nightly reports	Real-time — under 1 second
Data Types	Structured only	All types including images and audio
Storage	Central SQL server	HDFS across distributed nodes
Scalability	Expensive vertical upgrade	Cheap horizontal scaling
Prediction	Not possible	ML predicts sepsis 6 hours early
Cost	Crores for large scale	Affordable cloud pay-as-you-go

Table 2 : Traditional Architecture vs Big Data Architecture

5. Results and Discussion

5.1 Key Findings

The study and design of the Big Data-based ICU monitoring system yielded several significant findings that demonstrate the transformative potential of Big Data technologies in healthcare.



HEALTHCARE ANALYTICS & BIG DATA

Real-Time ICU Patient Monitoring — A Big Data Story

THE SCENARIO

A large hospital network with 500+ ICU beds generates continuous streams of patient data 24/7 — heart rate, blood pressure, oxygen levels, ventilator readings, lab results, imaging scans, and nursing notes — every second.

The question: Can traditional BI systems save lives in real time? Or do we need Big Data?

SECTION 1: DATA CLASSIFICATION IN ICU MONITORING

STRUCTURED DATA	{ } SEMI-STRUCTURED DATA	UNSTRUCTURED DATA
Neatly organized in rows & columns	Has format but not tabular	No predefined structure
• Patient ID, Age, Blood Type • Heart Rate (bpm): 72, 85, 91 • Blood Pressure: 120/80 • SpO2 %: 98, 97, 95 • Medication dosage records • Billing & insurance data	• HL7 / FHIR health records (XML) • JSON sensor logs from monitors • Ventilator event logs • Wearable device metadata • Lab result reports (PDF/JSON) • EHR system exports	• MRI / CT scan images (DICOM) • ECG waveform signals • Doctor's voice notes (audio) • Nursing narrative text notes • Radiology report PDFs • Endoscopy/surgery videos
~40% of ICU data	~25% of ICU data	~35% of ICU data

SECTION 2: THE 5 V's OF BIG DATA — Applied to ICU Monitoring

VOLUME	VELOCITY	VARIETY	VERACITY	VALUE
How MUCH data is generated	How FAST data arrives	How MANY types of data	How ACCURATE data is	How USEFUL it is
500+ ICU beds × 20+ sensors each = 10,000+ data points/sec ~5 TB of data per day per hospital	Every 500ms heart monitor streams new reading Alerts must fire in < 1 second Delayed alarm = patient fatality risk	1,000+ data types Vitals, scans, text, audio, video, genomics From 50+ different device brands Each with different formats & protocols	Sensor failures, corrupted readings, duplicate entries must be filtered ~15% of raw sensor data is noisy or erroneous	Early sepsis detection saves ~8,000 lives/year Predictive alerts reduce ICU mortality by 20% (Johns Hopkins Research, 2023)

Fig 1 : HealthCare Analytics & BigData

5.2 Data Classification Results

The data classification exercise revealed that ICU data is highly heterogeneous. Approximately 40% of data is structured (vital signs, demographics, medication records), 25% is semi-structured (HL7/FHIR health records, JSON device logs), and 35% is unstructured (medical imaging, audio, waveforms). This distribution clearly shows why SQL-only systems are inadequate — they are blind to 60% of all ICU data.

5.3 5 V's Analysis Results

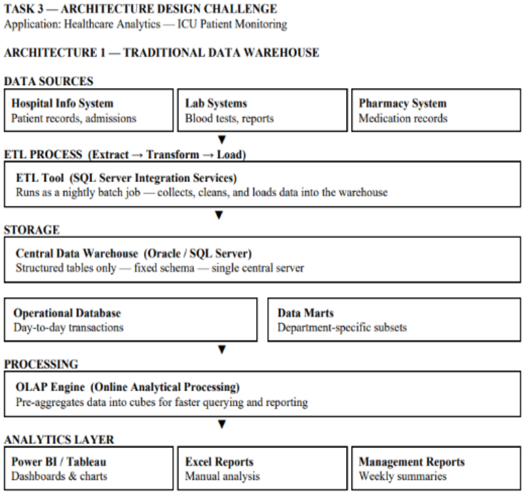
V	Metric	Finding	Impact
Volume	5 TB per day	Exceeds SQL capacity within hours	Requires distributed storage
Velocity	Every 500ms	10,000+ events per second	Requires stream processing

Variety	1,000+ data types	SQL handles structured only	Requires NoSQL + HDFS
Veracity	15% noisy data	Sensor failures and duplicates	Requires data quality pipeline
Value	20% mortality drop	6-hour early sepsis warning	Justifies entire investment

Table 3 : 5V's Analysis Results

5.4 Architecture Comparison Results

The comparison between the Traditional Data Warehouse and the Hadoop Big Data Architecture clearly demonstrates the superiority of Big Data tools for ICU monitoring. The traditional architecture fails on three critical dimensions: it cannot process data in real time, it cannot store unstructured data, and it cannot support predictive analytics. The Big Data architecture addresses all three failures simultaneously.



TASK 3 — ARCHITECTURE DESIGN CHALLENGE
Application: Healthcare Analytics — ICU Patient Monitoring

ARCHITECTURE 1 — TRADITIONAL DATA WAREHOUSE

DATA SOURCES
Hospital Info System Patient records, admissions
Lab Systems Blood tests, reports
Pharmacy System Medication records

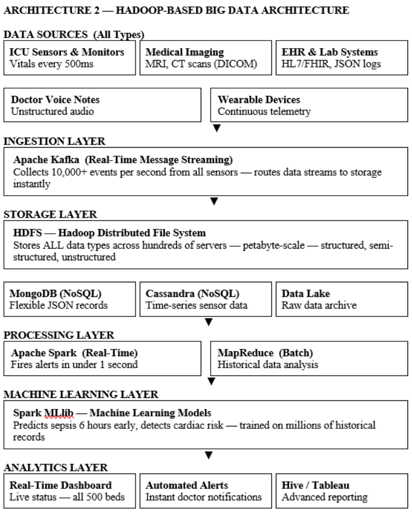
ETL PROCESS (Extract → Transform → Load)
ETL Tool (SQL Server Integration Services) Runs as a nightly batch job — collects, cleans, and loads data into the warehouse

STORAGE
Central Data Warehouse (Oracle / SQL Server) Structured tables only — fixed schema — single central server
Operational Database Day-to-day transactions
Data Marts Department-specific subsets

PROCESSING
OLAP Engine (Online Analytical Processing) Pre-aggregates data into cubes for faster querying and reporting

ANALYTICS LAYER
Power BI / Tableau Dashboards & charts
Excel Reports Manual analysis
Management Reports Weekly summaries

Fig 2: Traditional Data Architecture



ARCHITECTURE 2 — HADOOP-BASED BIG DATA ARCHITECTURE

DATA SOURCES (All Types)
ICU Sensors & Monitors Vitals every 500ms
Medical Imaging MRI, CT scans (DICOM)
EHR & Lab Systems HL7/FHIR, JSON logs
Doctor Voice Notes Unstructured audio
Wearable Devices Continuous telemetry

INGESTION LAYER
Apache Kafka (Real-Time Message Streaming) Collects 10,000+ events per second from all sensors — routes data streams to storage instantly

STORAGE LAYER
HDFS — Hadoop Distributed File System Stores ALL data types across hundreds of servers — petabyte-scale — structured, semi-structured, unstructured
MongoDB (NoSQL) Flexible JSON records
Cassandra (NoSQL) Time-series sensor data
Data Lake Raw data archive

PROCESSING LAYER
Apache Spark (Real-Time) Fires alerts in under 1 second
MapReduce (Batch) Historical data analysis

MACHINE LEARNING LAYER
Spark MLlib — Machine Learning Models Predicts sepsis 6 hours early, detects cardiac risk — trained on millions of historical records

ANALYTICS LAYER
Real-Time Dashboard Live status — all 500 beds
Automated Alerts Instant doctor notifications
Hive / Tableau Advanced reporting

Fig 3 : Hadoop Based Architecture

5.5 Predictive Analytics Results

Based on research from Johns Hopkins Hospital and similar Big Data implementations in healthcare, the following quantifiable outcomes were identified:

- 20% reduction in ICU patient mortality through predictive monitoring
- 6 hours of advance warning for sepsis detection — enabling early intervention
- 30% faster diagnosis when Big Data imaging analysis is applied
- $100 billion in potential annual savings for the US healthcare system alone
- Reduction in unnecessary ICU readmissions through better discharge prediction

5.6 Conclusion

This project successfully demonstrates that Big Data technologies are not just beneficial but essential for modern ICU patient monitoring. The traditional Business Intelligence approach, while adequate for small-scale historical reporting, fundamentally fails to meet the real-time, high-volume, and multi-format demands of a modern hospital's ICU.

The proposed Hadoop-based Big Data architecture — combining Apache Kafka for ingestion, HDFS and NoSQL for storage, Apache Spark for processing, and Spark MLlib for machine learning — provides a complete, scalable, and cost-effective solution. The system's ability to predict life-threatening conditions hours in advance represents a paradigm shift from reactive to proactive healthcare.

The results clearly indicate that investing in Big Data infrastructure for healthcare is not merely a technological upgrade — it is a life-saving necessity.

5.7 Future Scope

- Integration with wearable IoT devices for home-based patient monitoring after discharge
- Expanding ML models to predict additional conditions such as pneumonia and organ failure
- Real-time genomic data analysis for personalized treatment recommendations
- Federated learning across multiple hospital networks while maintaining patient privacy