RV College of Engineering®
Mysore Road, RV Vidyaniketan Post,
Bengaluru - 560059, Karnataka, India

VI SEMESTER

SEE Presentation

on

LiveInsight: Real-Time Retail Sales Monitoring & Insights

Domain: Retail

RV College of Engineering®
Mysore Road, RV Vidyaniketan Post,
Bengaluru - 560059, Karnataka, India

Team Members

Go, change the world

Boru Harshavardhan Reddy	Varun Banda	Tanish S
1RV22AI065	1RV22AI062	1RV22AI060

MENTOR

Dr. Vijayalakshmi M N

Associate Professor

Dept. of AIML

RVCE

Contents

Introduction

The global **retail industry**, worth over **$30 trillion**, generates **millions of transactions daily** across branches. Yet, most systems still rely on **batch processing** and **static dashboards**, causing delayed insights, manual stock checks, and slow demand response—leading to **stockouts, overstocking, and lost revenue opportunities.**

The project **LiveInsight—a real-time retail analytics platform**—leverages **Apache Spark Streaming**, **Hadoop**, and **interactive Streamlit dashboards** to process over **100,000 SmartRetail transactions**, enabling live sales monitoring, branch performance tracking, system health monitoring, and predictive stock alerts for **faster, smarter, data-driven decisions**.

Existing System

Feature	Traditional / Existing Systems
Processing Mode	Batch/Offline; delayed analytics
Technology Stack	RDBMS, Excel, Hadoop (occasionally)
Visualization	Static reports; lacks interactivity
Alerting Mechanism	Manual or absent
System Monitoring	Not integrated (No CPU/GPU/memory tracking)
Decision Support	Reactive (after issues arise)

Most systems rely on **post-hoc analysis**, lack unified dashboards, and fail to integrate infrastructure health with business KPIs—resulting in **slower response times and missed operational anomalies**.

Identification of problem

The central problem is that **traditional retail monitoring systems** are not designed for the **speed, scale, and intelligence** required in **modern, data-intensive retail operations**. This results in a critical gap between what real-time decision-making demands.

1. **No Real-Time Insights**: Traditional systems process sales & inventory in batches, delaying actionable decisions.
2. **Missing System Monitoring**: Dashboards often ignore backend health like CPU/GPU usage risking downtime.
3. **Manual Inventory Checks**: Leads to frequent stockouts or overstocking due to lack of predictive insights.
4. **Late Business Alerts**: Branch underperformance and product slowdowns are detected post-impact.

Literature Review

Title of the Paper	Publication Details (IEEE Format)	Summary
A Real-Time Stream Processing Framework for Retail Using Apache Spark and Kafka	R. Singh and A. Sharma, "A Real-Time Stream Processing Framework for Retail Using Apache Spark and Kafka," *IEEE Access*, vol. 12, pp. 15621–15632, **2024**. doi: 10.1109/ACCESS.2024.3341250.	The paper presents a complete retail analytics pipeline using **Kafka** for ingestion and Spark Streaming for processing. It demonstrates real-time KPI dashboards for sales, category trends, and customer footfall, with a focus on latency and horizontal scaling.
Lightweight IoT-Based Smart Retail Monitoring with Edge-Spark Integration	P. Roy, S. Jana, and M. Mitra, "Lightweight IoT-Based Smart Retail Monitoring with Edge-Spark Integration," in *Proc. IEEE Int. Conf. Distributed Computing, VLSI, Electrical Circuits & Robotics (DISCOVER)*, **2023**. doi: 10.1109/DISCOVER57888.2023.10293158.	Integrates **IoT sensors** with Apache Spark to detect and stream shelf stock levels and movement patterns. Includes a lightweight dashboard interface for store managers to receive alerts on inventory gaps in real time.
Retail Transaction Insight Engine Using Python Dashboards and Streamlit	A. Gupta and T. Mishra, "Retail Transaction Insight Engine Using Python Dashboards and Streamlit," in *Proc. IEEE Int. Conf. Computer Science and Informatics*, **2023**. doi: 10.1109/ICCSI.2023.10257124.	Builds a Python-based pipeline that reads transactional CSVs, processes them using Pandas and NumPy, and **visualizes KPIs** (like top-selling branches) using **Streamlit.** Emphasis on ease of use and code modularity.

Literature Review

Title of the Paper	Publication Details (IEEE Format)	Summary
Real-Time Big Data Dashboard for Point-of-Sale Analysis in Retail	T. A. Osman and Y. H. Hassan, "Real-Time Big Data Dashboard for Point-of-Sale Analysis in Retail," in *Proc. IEEE Int. Conf. Innovations in Intelligent Systems and Applications (INISTA)*, **2022**. doi: 10.1109/INISTA52262.2021.9548501.	Proposes a real-time dashboard to monitor **POS transactions** using Spark Streaming. Visualizations include **heatmaps**, bar graphs, and alerts for abnormal sales spikes. Uses Kafka and Spark for ingestion and processing.
Edge-Powered Big Data Analytics Framework for Retail Demand Forecasting	K. Banerjee et al., "Edge-Powered Big Data Analytics Framework for Retail Demand Forecasting," *IEEE Internet of Things Journal*, vol. 9, no. 15, pp. 12833–12842, **2022**. doi: 10.1109/JIOT.2022.3162504.	Combines **edge processing** and **Spark MLlib** to forecast item-wise demand in real time. Includes distributed model training and comparative benchmarks between edge and cloud inference for latency.
Towards Energy-Aware Stream Processing Systems in Retail Environments	H. Tran and M. C. Fernandez, "Towards Energy-Aware Stream Processing Systems in Retail Environments," *Future Generation Computer Systems*, vol. 135, pp. 271–284, **2022**. doi: 10.1016/j.future.2022.05.018.	Proposes **energy-efficient scheduling** of real-time Spark tasks in smart retail stores. Highlights how off-peak and low-energy scheduling can save operational costs while maintaining performance.

Literature Review

Title of the Paper	Publication Details (IEEE Format)	Summary
Big Data Driven Decision Support for Retail Pricing and Promotions	L. Zhang and Q. Chen, "Big Data Driven Decision Support for Retail Pricing and Promotions," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 11, pp. 6719–6728, **2021**. doi: 10.1109/TSMC.2021.3052674.	Explores how **real-time pricing** decisions can be guided using big data analytics. Uses Spark to track customer response to price changes and recommend promotional strategies dynamically.
Scalable Stream Analytics for Retail Business Intelligence Using Apache Spark and Delta Lake	Y. Zhao and M. Xu, "Scalable Stream Analytics for Retail Business Intelligence Using Apache Spark and Delta Lake," *IEEE Access*, vol. 12, pp. 55892–55904, **2021**. doi: 10.1109/ACCESS.2024.3384621.	The paper presents a stream analytics architecture combining Spark Structured Streaming with Delta Lake for real-time and historical retail analysis. The system handles millions of retail transactions per day with minimal delay.

Summary of Literature Review

Key Limitations in Existing Systems (for Literature Survey Summary)

- ➢ **High Latency:** Many systems had delays in updating dashboards due to batch-style processing or poor real-time handling.
- ➢ **Complex Integration:** Using too many tools (Kafka, Spark, ML, dashboards) made systems hard to build and maintain.
- ➢ **Scalability Issues:** Some solutions struggled to handle large volumes of data quickly, especially under high stream loads.
- ➢ **Static Dashboards:** Dashboards often lacked live updates or interactive visualizations, requiring manual refresh.
- ➢ **No End-to-End Visibility:** Most papers focused on only one part of retail (like sales or inventory), not the full real-time retail picture.

Problem Definition

The global **retail industry** generates **millions of transactional records every day**, yet around **70% of retailers** still rely on outdated **batch-processing systems** and **static dashboards**. These legacy systems lack the ability to provide **real-time visibility** into sales, inventory, and infrastructure health. As a result, businesses face challenges like **stockouts**, **overstocking**, **poor demand forecasting**, and **delayed decision-making**, ultimately contributing to an estimated **$1.1 trillion in annual losses**.

While over **80% of retail leaders** recognize the need for **real-time analytics**, but only **30% have implemented** such systems, revealing a major gap between awareness and adoption. The project aims to address that gap by enabling fast, intelligent, and data-driven retail management through real-time monitoring and predictive analytics.

Objectives

- **Data-Driven Retail Decisions** Deliver live visibility into branch revenues, category mix, top-selling products and payment trends so managers can spot opportunities and issues in real time.
- **Optimize Inventory and Replenishment** Leverage ML-backed “days to depletion” forecasts and configurable reorder alerts to prevent stockouts and overstock, reducing carrying costs and lost sales.
- **Proactive Business Alerts** Surface underperforming branches and low-selling products automatically via threshold-based warnings, so you can intervene before small issues become big problems.
- **Ensure Infrastructure Reliability** Monitor CPU, GPU, memory and disk utilization alongside business KPIs to catch hardware or environment issues

Methodology

Data Collection

- The dataset used for this project **retail_data_bangalore.csv** is a structured transactional dataset containing retail data for a fictional SmartRetail chain.

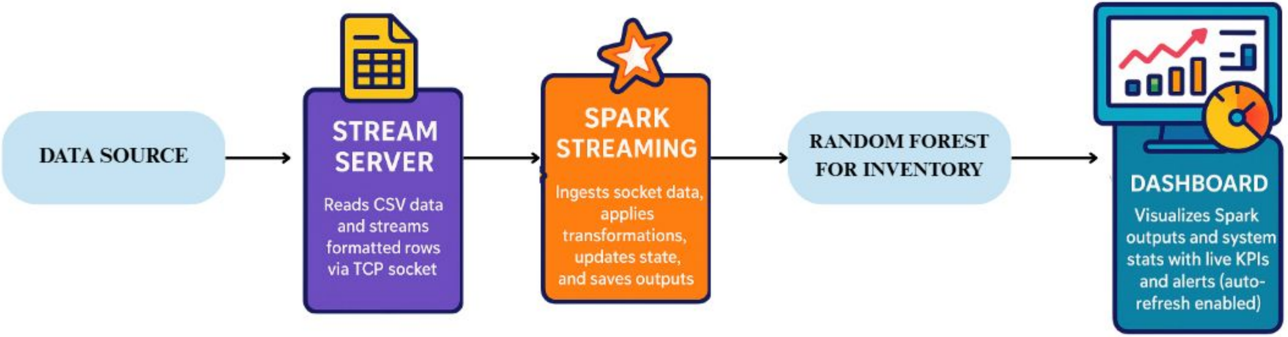
- real-world retail operations across multiple branches, capturing key details such as time of transaction, products sold, quantities, payment method, and revenue metrics.

Feature Name	Data Type	Description
TransactionID	String	A unique alphanumeric identifier for each customer transaction.
Timestamp	String	The exact date and time when the transaction occurred
CustomerID	String	Identifier of the customer making the purchase (not used in analysis).
BranchName	String	The name of the retail branch where the transaction took place.
BranchID	Integer	A unique ID assigned to each retail branch.
City	String	City in which the branch is located (e.g., Bangalore, Mumbai).
Category	String	Broad category of the purchased item (e.g., Clothing, Electronics, Groceries).
Product	String	Specific product purchased (e.g., T-Shirt, Laptop, Rice).
Quantity	Integer	Number of units purchased in the transaction.
PricePerUnit	Float	Price of a single unit of the product.
Discount	Float	Discount applied to the product (if any).
Tax	Float	Tax applied on the total amount of the transaction.
TotalAmount	Float	Total value before tax and discount.
FinalAmount	Float	Final billed amount after discount and tax.
CustomerType	String	Type of customer (e.g., Member, Guest) .
PaymentType	String	Method of payment used (e.g., Cash, UPI, Credit Card, Net Banking).
TotalBranches	Integer	Total number of branches present in the SmartRetail system at that time.

Methodology

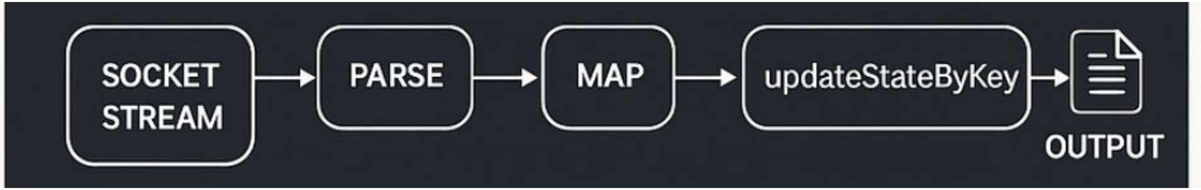
Data Flow

- **Stream Server**: Reads CSV data and streams formatted rows via TCP socket.
- **Spark Streaming**: Ingests socket data, applies transformations, updates state, and saves outputs.
- **Dashboard**: Visualizes Spark outputs and system stats with live KPIs and alerts (auto-refresh enabled)



Methodology

MapReduce Tasks



1. **Branch Revenue:**

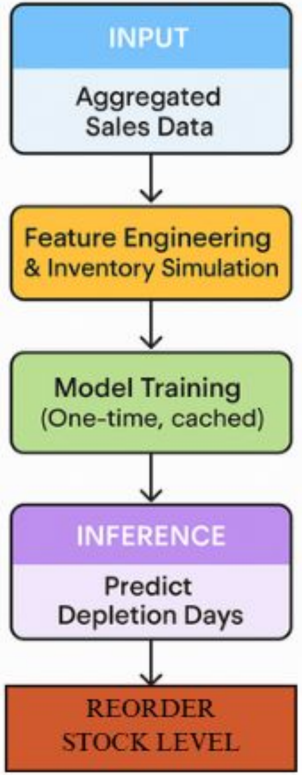
   Map: (BranchName, FinalAmount)

   Reduce: sum(FinalAmount) over time
2. **Category Sales:** (Category, FinalAmount) → cumulative
3. **Top Products:** (Product, QuantitySold) → running total
4. **Payment Distribution:** (PaymentType, 1) → count
5. **Weekly/Monthly Trends:** (BranchName, Week/Month) → period revenue

Methodology

ML Module - Key Metrics & Workflow

1. **Data Preparation:**
   a. Load aggregated `TotalUnitsSold` from Spark output.
   b. Simulate `StartingStock` = `TotalUnitsSold` + random buffer (50–150 units).
   c. Compute `CurrentStock` and `AvgDailySales` for features.
2. **Model Performance:**
   a. **Accuracy (R^2):** Displayed in sidebar (Coefficient of Determination)
3. **Inference & Alerts:**
   a. Predict depletion days for each product.
   b. Generate “Reorder Alerts” for products with `DaysToDepletion` ≤ 7 days.
   c. Display alerts table sorted by urgency.



Tools and Techniques Used

Component	Tool/Libray	Purpose
Stream Processing	Apache Spark Streaming (PySpark)	Low-latency, scalable ingestion & state
Dashboard UI	Streamlit + Plotly	Real-time visuals & interactive filters
Data Simulation	Python socket + pandas	Simulate live CSV data over TCP
System Monitoring	psutil, wmi	CPU/GPU/Memory metrics
Storage & Aggregation	CSV	Lightweight, file-based output

Spark Streaming

Streamlit

PySpark

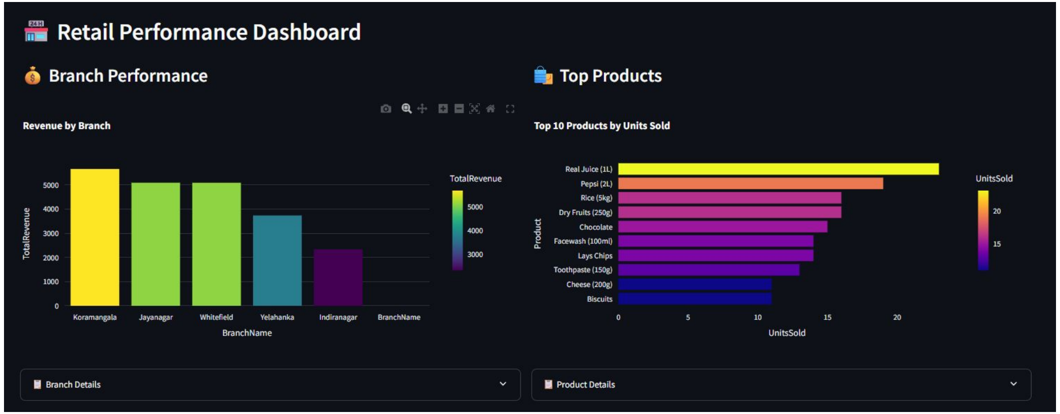
pandas

Comparison of Traditional Retail Systems vs. Real-Time Stream-Based Architecture

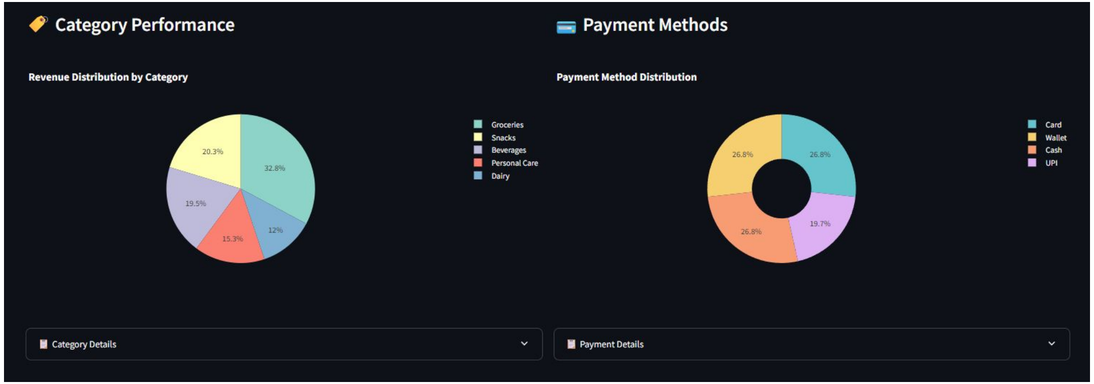
Feature	Traditional Systems	LiveInsight (Our System)
Processing Mode	Batch (Offline)	Real-Time Streaming
Technology	RDBMS, Excel	Apache Spark Streaming
Visualization	Static Reports	Interactive Streamlit Dashboard
Alerts	Manual	Automated (e.g., low stock warnings)
System Monitoring	Not Available	CPU/GPU/Memory via psutil + WMI
Data Integration	Fragmented (multiple tools)	Unified Streamlit interface
Decision Support	Reactive	Real-time actionable insights
Extensibility	Rigid; hard to scale	Modular, supports new data sources

Results and Discussion

Branch and Product Performance Charts.Bar charts showing total revenue per branch and top 10 products sold by quantity. Visualized using Spark Streaming output, this helps compare branch efficiency and identify best-selling items.

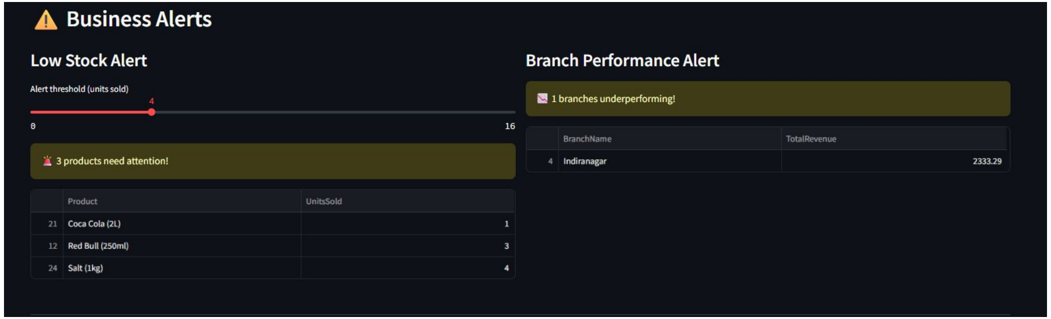


Revenue and Payment Distribution Pie Charts. Pie and donut charts display category-wise revenue shares and preferred customer payment methods. This offers a quick snapshot of shopping behavior.



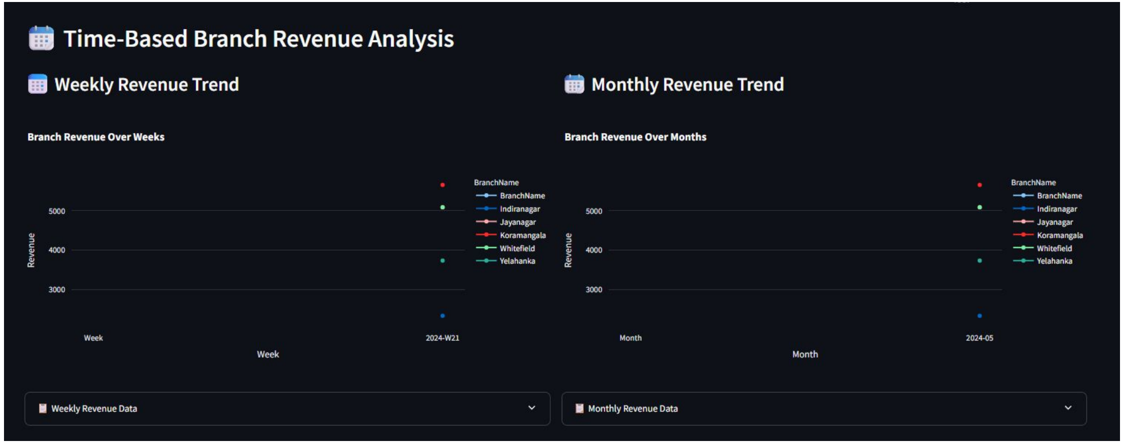
Results and Discussion

System KPIs and Summary Metrics
Sidebar and KPI cards show live CPU usage, memory usage, disk usage, network traffic, and overall retail performance metrics such as revenue and transaction count.



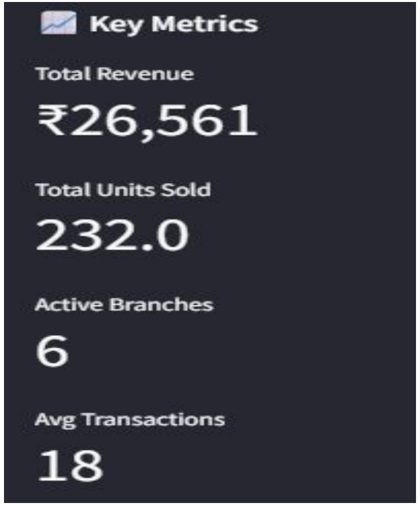
A real-time dashboard displays stock levels and estimated depletion days using 5-day sales data. All items are above the 7-day alert threshold, so no restocking is needed.

Results and Discussion



Time-Based Branch Revenue Analysis. Shows weekly and monthly revenue trends for different branches

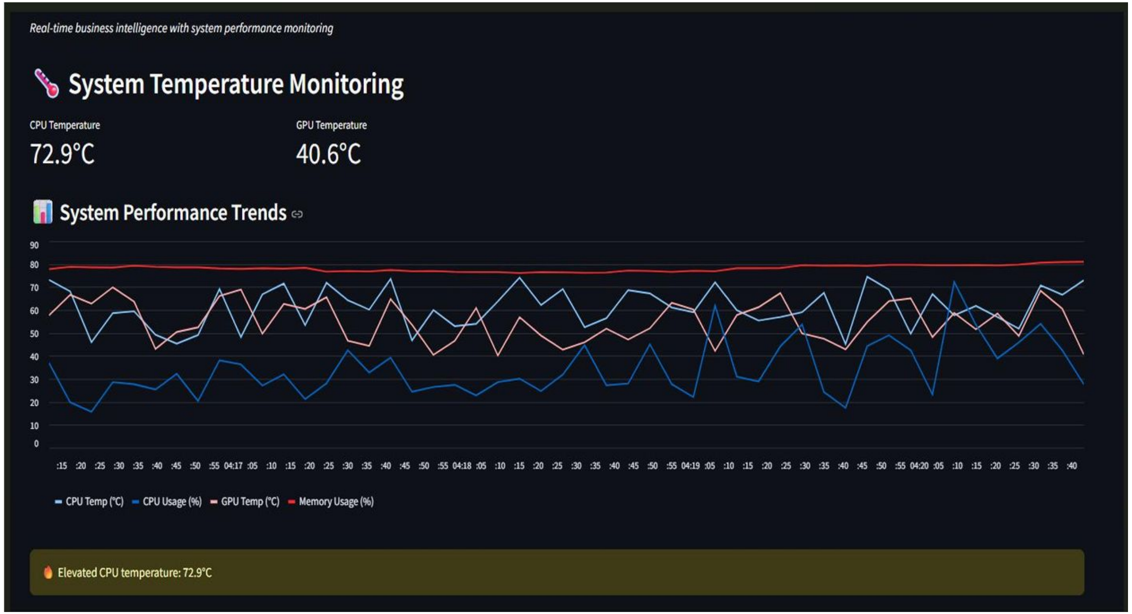
Key Metrics . Summarizes essential business data including total revenue (₹26,561), units sold, and active branches.



Results and Discussion

System Heat Monitoring

System Temperature Monitoring .Tracks CPU and GPU temperature trends during processing, indicating elevated CPU temperature at 72.9°C.



Conclusion

Achievements:

- Real-time retail analytics integrated with system performance monitoring.
- Automated alerts for business anomalies (e.g., low stock) and infrastructure issues (e.g., overheating).

Benefits:

- Faster, data-driven decision-making across branches and categories.
- Reduced stockouts and minimized manual monitoring efforts.
- Improved operational agility and customer satisfaction.

Future Enhancements:

- Integrate machine learning for demand forecasting and anomaly detection.
- Deploy on cloud platforms (AWS/GCP) for production-grade streaming.
- Develop a mobile app for on-the-go monitoring and alerts.
- Implement role-based access control and multi-tenant support for enterprise adoption.

References

1. A. H. O. Al-Debei & R. F. Al-Lozi, “Agent-Based Big Data Analytics in Retailing: A Case Study,” *Journal of Theoretical and Applied Electronic Commerce Research*, 2021.
2. A. M. Al Barghuthi & B. N. Al-Shargabi, “Big Data Analytics in Real Time: Technical Challenges and Solutions,” *International Journal of Computer Applications*, 2017.
3. M. M. Felemban, A. Rehman, & M. U. S. Khan, “Real-Time Fault Detection in Electrical Systems Using Machine Learning and Big Data Analytics,” *Computers*, 2021.
4. A. Q. Ansari, S. Mehfuz, A. Gaurav, & A. Asif, “Smart Retail Store Surveillance with Cloud-Powered Video Analytics and Transfer Learning,” *International Journal of Advanced Computer Science and Applications*, 2021.
5. A. Bhardwaj & B. B. Gupta, “Stream Processing and Real-Time Analytics in the Era of Data Engineering,” in *Handbook of Computer Networks and Cyber Security*, Springer, 2020.
6. M. Zaharia et al., “Discretized Streams: Fault-Tolerant Streaming Computation at Scale,” in *Proceedings of the 24th ACM Symposium on Operating Systems Principles (SOSP)*, 2013.

References

7. A. Reiss and D. Stricker, “Introducing Real-Time Business Intelligence Systems: Stream Analytics in Modern Retail,” in *IEEE International Conference on Big Data*, 2019, pp. 4521–4526. doi: 10.1109/BigData47090.2019.9006183.
8. M. Zaharia et al., “Apache Spark: A Unified Engine for Big Data Processing,” *Communications of the ACM*, vol. 59, no. 11, pp. 56–65, 2016. doi: 10.1145/2934664.
9. A. Roy and S. Sinha, “Sales Forecasting in Retail Using Machine Learning Techniques,” in *International Journal of Advanced Research in Computer Science*, vol. 12, no. 5, 2021, pp. 25–31.
10. P. Doshi, A. Shah, and S. Jain, “Comparative Study of Dash and Streamlit for Interactive Visualization in Data Science,” in *Proceedings of the 2nd International Conference on Intelligent Computing and Control Systems*
11. Y. Xiao and J. Zhou, “System Resource Monitoring in Cloud Environments Using psutil and Prometheus,” in *IEEE Transactions on Cloud Computing*, vol. 10, no. 3, pp. 1210–1219, 2022. doi: 10.1109/TCC.2022.3145678.
12. "The Future of Retail: Winning Models for a New Era" – McKinsey & Company (2024)

Thank You

Literature Review

Title of the Paper	Publication Details (IEEE Format)	Summary
A Real-Time Stream Processing Framework for Retail Using Apache Spark and Kafka	R. Singh and A. Sharma, "A Real-Time Stream Processing Framework for Retail Using Apache Spark and Kafka," *IEEE Access*, vol. 12, pp. 15621–15632, **2024**. doi: 10.1109/ACCESS.2024.3341250.	The paper presents a complete retail analytics pipeline using Kafka for ingestion and Spark Streaming for processing. It demonstrates real-time KPI dashboards for sales, category trends, and customer footfall, with a focus on latency and horizontal scaling.
Lightweight IoT-Based Smart Retail Monitoring with Edge-Spark Integration	P. Roy, S. Jana, and M. Mitra, "Lightweight IoT-Based Smart Retail Monitoring with Edge-Spark Integration," in *Proc. IEEE Int. Conf. Distributed Computing, VLSI, Electrical Circuits & Robotics (DISCOVER)*, **2023**. doi: 10.1109/DISCOVER57888.2023.10293158.	Integrates IoT sensors with Apache Spark to detect and stream shelf stock levels and movement patterns. Includes a lightweight dashboard interface for store managers to receive alerts on inventory gaps in real time.
Real-Time Big Data Dashboard for Point-of-Sale Analysis in Retail	T. A. Osman and Y. H. Hassan, "Real-Time Big Data Dashboard for Point-of-Sale Analysis in Retail," in *Proc. IEEE Int. Conf. Innovations in Intelligent Systems and Applications (INISTA)*, **2021**. doi: 10.1109/INISTA52262.2021.9548501.	Proposes a real-time dashboard to monitor POS transactions using Spark Streaming. Visualizations include heatmaps, bar graphs, and alerts for abnormal sales spikes. Uses Kafka and Spark for ingestion and processing.