

UNIT 4

Data and Analytics for IoT

(10 Hours) — Internet of Things (CACS460)

⚠ Please Note

These notes were prepared quickly and **may contain errors or omissions**. Please cross-check important definitions, formulas, code and diagrams with your textbook, class notes and the official TU syllabus before relying on them in exams.

4.1 An Introduction to Data Analytics for IoT

An IoT deployment can produce enormous streams of data. **Data analytics** is the process of examining that data to find patterns, detect anomalies, predict outcomes and support decisions. Raw numbers alone do not create value — analytics converts them into insight and action.

Structured vs Unstructured Data

Basis	Structured data	Unstructured data
Format	Fixed schema, rows and columns	No predefined model
Examples	Temperature readings table, device registry, meter readings	Images, video, audio, free text, logs
Storage	Relational DB (SQL)	NoSQL, object storage, data lake
Analysis	SQL queries, statistics	ML, computer vision, NLP

IoT Data vs Typical Business Data

- Mostly **time-series**, timestamped, often generated continuously.
- High **velocity** and often small messages; sources are many and heterogeneous.
- Data may be noisy, missing or out-of-order; sensors drift and fail.
- Value of data often decays quickly — late data may be useless for real-time control.

Types of Data Analysis

Type	Question answered	IoT example
Descriptive	What happened?	Average daily temperature of a cold room
Diagnostic	Why did it happen?	Why did the compressor overheat last night?
Predictive	What will happen?	Predicting when a motor bearing will fail
Prescriptive	What should we do?	Auto-adjust irrigation schedule for forecast rain

Data Pipeline for IoT Analytics

Typical stages: **Collect** (sensors) → **Ingest** (MQTT/Kafka) → **Clean & pre-process** (remove noise, fill gaps, normalise) → **Store** (time-series DB, data lake) → **Analyse** (statistics, ML) → **Visualise & act** (dashboards, alerts, actuators).

4.2 Machine Learning for IoT

Machine learning (ML) allows systems to learn patterns from data and make predictions without being explicitly programmed for every rule. IoT supplies the data; ML turns it into automated intelligence.

Types of Machine Learning

Type	Description	Common algorithms	IoT use
Supervised	Learns from labelled input–output examples	Linear/logistic regression, decision tree, SVM, k-NN, Random Forest, neural networks	Predict fault (yes/no), forecast energy use
Unsupervised	Finds structure in unlabelled data	k-means, hierarchical clustering, PCA, Isolation Forest	Group similar devices, anomaly detection
Reinforcement	Agent learns by trial and reward	Q-learning, policy gradient	Smart HVAC control, robot navigation

Common ML Tasks in IoT

- **Classification:** assigns a category (normal / faulty, activity recognition from accelerometer).
- **Regression:** predicts a number (tomorrow's temperature, remaining useful life of a machine).
- **Clustering:** groups similar data (customer or device behaviour).
- **Anomaly detection:** finds unusual readings (sensor failure, intrusion, leak).

- **Forecasting:** time-series prediction (ARIMA, LSTM) for demand, weather, traffic.

Typical ML Workflow

- Collect data → clean and label → split into training and test sets → select features → train model → evaluate (accuracy, precision, recall, RMSE) → deploy → monitor and retrain.

Example 4.1 — Simple linear regression for energy prediction

A smart meter records daily temperature (x , °C) and electricity use (y , kWh) for 5 days:

$x = 20, 22, 24, 26, 28$ $y = 12, 14, 16, 18, 20$. Predict y for $x = 30$ °C.

Solution

Mean $\bar{x} = 24$, mean $\bar{y} = 16$.

Deviations: $(x - \bar{x}) = -4, -2, 0, 2, 4$ and $(y - \bar{y}) = -4, -2, 0, 2, 4$.

Slope $b = \Sigma(x - \bar{x})(y - \bar{y}) / \Sigma(x - \bar{x})^2 = (16 + 4 + 0 + 4 + 16) / (16 + 4 + 0 + 4 + 16) = 40/40 = 1$.

Intercept $a = \bar{y} - b \cdot \bar{x} = 16 - 1 \times 24 = -8$. Model: $y = x - 8$.

Prediction for $x = 30$: $y = 30 - 8 = \mathbf{22 \text{ kWh}}$.

Challenges of ML in IoT

- Limited compute, memory and battery on devices (leads to **TinyML** and edge inference).
- Poor-quality, unlabelled or imbalanced data; concept drift as environment changes.
- Privacy and security of sensitive data; need for explainable models.

4.3 Big Data Analytics Tools and Technology

IoT data fits the **5 V's of Big Data**: **Volume** (huge amounts), **Velocity** (fast arrival), **Variety** (many formats), **Veracity** (data quality/trust) and **Value** (usefulness).

Hadoop Ecosystem

- **HDFS:** distributed file system that stores large files across many commodity machines with replication for fault tolerance.
- **MapReduce:** batch programming model — *Map* splits and processes data in parallel, *Reduce* aggregates results.
- **YARN:** resource manager and job scheduler for the cluster.

- **Hive:** SQL-like queries over data in Hadoop; **Pig:** data-flow scripting; **HBase:** column-oriented NoSQL database; **Sqoop:** moves data between RDBMS and Hadoop; **Flume:** collects log/stream data.

Apache Spark

In-memory cluster computing engine, much faster than MapReduce for iterative work. Includes Spark SQL, **Spark Streaming** (micro-batch streams), **MLlib** (machine learning) and GraphX (graph processing).

Streaming and Messaging Tools

- **Apache Kafka:** distributed publish–subscribe log for high-throughput, durable real-time data ingestion.
- **Apache Storm / Flink:** real-time stream processing frameworks for low-latency event-by-event computation.

NoSQL and Time-Series Storage

Type	Examples	Use in IoT
Key–value	Redis, DynamoDB	Latest device state, caching
Document	MongoDB, CouchDB	Flexible JSON device messages
Column family	Cassandra, HBase	Massive write-heavy sensor logs
Time-series	InfluxDB, TimescaleDB, OpenTSDB	Timestamped metrics with fast range queries

Basis	Batch processing	Stream processing
Data	Large stored datasets	Continuous flow of events
Latency	Minutes to hours	Milliseconds to seconds
Tools	Hadoop MapReduce, Hive	Kafka Streams, Storm, Flink, Spark Streaming
Use	Historical reports, model training	Real-time alerts, live dashboards

Example 4.2 — MapReduce word-count style idea for IoT

Count how many times each sensor ID reports a reading above 50 °C in a large log file.

Solution

Map: for each log line, if temperature > 50 emit (sensorID, 1).

Shuffle: group all pairs with the same sensorID.

Reduce: sum the 1s for each sensorID, giving (sensorID, count).

4.4 Edge Streaming Analytics

Edge analytics processes data at or near the source (on the device or a gateway) rather than sending everything to the cloud. **Streaming analytics** analyses each event or small window of events *as it arrives* instead of storing it first.

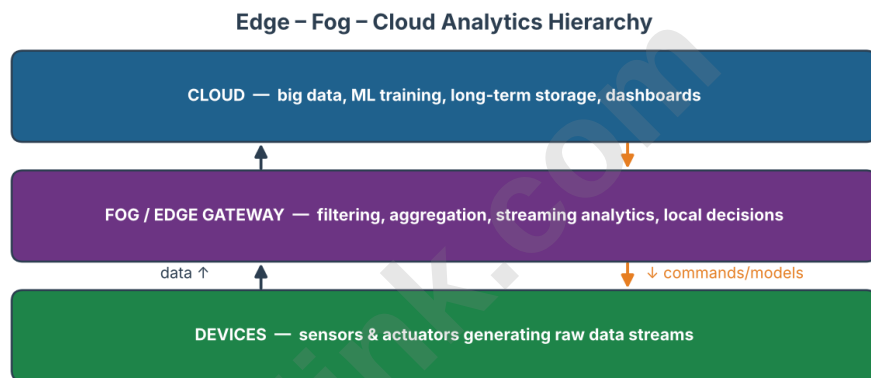


Figure 4.1 — Edge, fog and cloud analytics hierarchy

Why Edge Streaming Analytics?

- **Low latency:** instant local decisions (stop a machine, close a valve).
- **Bandwidth saving:** send only filtered or aggregated results (e.g., one average per minute).
- **Reliability:** keeps working when connectivity to the cloud is lost.
- **Privacy:** sensitive raw data (video, health) can stay local.
- **Cost:** less cloud storage and data transfer.

Common Streaming Operations

- **Filtering:** discard irrelevant or out-of-range data.
- **Windowing:** compute over time windows — tumbling (fixed, non-overlapping), sliding (overlapping), session windows.
- **Aggregation:** average, min, max, count per window.

- **Pattern/event detection (CEP):** detect a sequence like “temperature rising three times in 10 s”.
- **Inference:** run a trained ML model locally on streaming data.

Example 4.3 — Sliding-window average

Readings (°C) arrive every second: 30, 32, 31, 45, 33. Compute the average over a sliding window of size 3 at each position, and flag any window average above 36 °C.

Solution

Window 1 (30, 32, 31): $(30+32+31)/3 = 31.0$

Window 2 (32, 31, 45): $(32+31+45)/3 = 36.0$

Window 3 (31, 45, 33): $(31+45+33)/3 = 36.33 \rightarrow$ above 36, **raise alert**.

The window average smooths one-off spikes, so only sustained rises trigger alarms.

Fog Computing

Fog computing (Cisco) places compute, storage and networking between the edge devices and the cloud — in routers, gateways or local servers — to provide distributed, low-latency processing close to the data source.

4.5 Network Analytics

Network analytics studies the data produced by the network itself — traffic, topology, device behaviour — to keep an IoT network healthy, efficient and secure. With thousands of devices, manual monitoring is impossible.

Sources of Network Data

- **Flow data** from routers (NetFlow/IPFIX): who talks to whom, how much, when.
- **SNMP** counters and device logs; packet captures; protocol statistics.
- Device metadata: signal strength (RSSI), battery level, packet loss, latency, retransmissions.

Uses of Network Analytics

Use	What it does	Example
Performance monitoring	Tracks latency, throughput, packet loss	Detect congested gateway
Traffic profiling	Learns normal traffic of each device type	A smart bulb should only talk to its cloud
Anomaly / security detection	Flags unusual flows	Camera suddenly sends data to unknown IP (possible botnet)
Capacity planning	Predicts growth in devices and traffic	Upgrade bandwidth before deployment expands
Fault and battery management	Finds failing nodes or weak links	Replace sensor with falling RSSI/battery
Flow-based billing/QoS	Prioritises critical traffic	Alarm messages over bulk telemetry

Note: Network analytics also benefits from ML: baseline normal behaviour, then detect deviations.

Key Takeaways

- ✓ Analytics types: descriptive, diagnostic, predictive, prescriptive.
- ✓ Supervised, unsupervised and reinforcement learning; know examples of each and common IoT tasks.
- ✓ Big data = 5 V's; Hadoop (HDFS, MapReduce) for batch, Spark/Kafka/Storm/Flink for streams.
- ✓ Edge streaming analytics gives low latency, bandwidth savings and privacy; know windowing and filtering.
- ✓ Network analytics uses flow and device data for performance, planning and security.

— End of Unit 4 —