EVOLUTION OF BIG DATA & INTERNET OF THINGS (IOT) SECURITY

Dr. S. Sasikala, Associate Professor, Department of Data Analytics (PG)

Dr. S.Poongodi, Associate Professor, Department of Data Science

PSGR Krishnammal College for Women, Coimbatore.

PSGR Krishnammal College for Women, Coimbatore.

ABSTRACT

The rise of Big Data and the Internet of Things (IoT) has revolutionized data collection, processing, and usage across industries like manufacturing, healthcare, smart cities, and autonomous vehicles. While offering numerous benefits, these technologies introduce significant security challenges, such as ensuring data availability, confidentiality, and integrity. This overview highlights key security concerns, including data privacy, the massive data generated, and the resource limitations of IoT devices. Traditional security measures often fall short, requiring innovative approaches tailored to this evolving landscape. A proactive, holistic approach to security, from devices to the cloud, is essential for protecting sensitive data and ensuring the longevity of these transformative technologies. Addressing these issues allows businesses and consumers to safely maximize the potential of Big Data.

KEYWORDS: IoT, Big Data, Security, Ecosystem

I. INTRODUCTION TO IOT AND BIG DATA

1.1 DEFINE IOT

The "Internet of Things" refers to a network of connected devices, machinery, and structures equipped with sensors, software, and connectivity. These devices can collect and exchange data over the internet, enabling them to communicate and make decisions independently without human involvement.

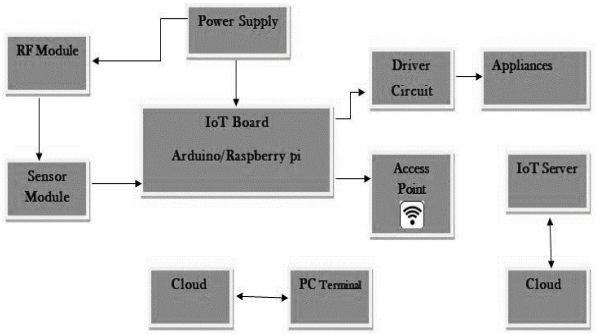


Fig 1.1 : Internet of Things

Note:

The IoT Board serves as the core of the application, where programs are written and uploaded. Depending on the application, the board may be an Arduino or Raspberry Pi. The Arduino Suite platform is used to upload programs (Python/C++) via I2C or Serial Communication Protocols. The Sensor Module measures physical parameters and provides output in voltage, which is fed into an analog input on the Arduino/Raspberry board (A0 to A5).

(link: Block Diagram of Internet of Things (IoT) The Figure 1 shows the block...Download Scientific Diagram (researchgate.net))

1.2 HISTROY

The Internet of Things (IoT) is transforming daily life, from smart thermostats to sensors enhancing farming. Billions of connected devices are collecting and sharing data worldwide, giving them digital intelligence. By 2019, 25% of enterprises used IoT, up from 13% in 2014. The World Economic Forum reports there are now more connected devices than people, with 41.6 billion devices expected by 2025.

IoT, alongside AI and robotics, is driving the Fourth Industrial Revolution, accelerated by COVID-19.

1.3 APPLICATIONS

let's see all these applications of IoT in different facets and industries of the world.

1. **Smart Farming:** IoT technology enhances farming by optimizing processes like irrigation, pest management, and crop/livestock monitoring. Key components include soil sensors, weather stations, and automated irrigation systems, which adjust based on real-time data.
2. **Smart Vehicles:** IoT in vehicles improves safety and efficiency through systems like telematics, vehicle-to-vehicle communication, and predictive maintenance. It also enables autonomous driving and fleet management.
3. **Smart Homes:** IoT connects devices for improved comfort, security, and energy efficiency. Smart thermostats, lighting, and security systems are controlled remotely via hubs like Amazon Echo.
4. **Smart Pollution Control:** IoT monitors air quality with sensors, GIS mapping, and predictive analytics. These systems provide real-time data and alerts to reduce pollution.
5. **Smart Healthcare:** IoT improves healthcare with wearable devices, remote patient monitoring, and smart health records. It enhances chronic disease management and ensures timely intervention.

1.4 DEFINE BIG DATA

Big data refers to extremely large and complex datasets that exceed the capabilities of traditional data processing tools. It is characterized by the "3 Vs":

1. **Volume:** Huge amounts of data, ranging from terabytes to exabytes.
2. **Velocity:** The rapid speed at which data is generated and must be processed, often in real-time.
3. **Variety:** Diverse types of data, including structured (databases), unstructured (text, images), and semi-structured (JSON, XML), which add complexity to analysis and management.

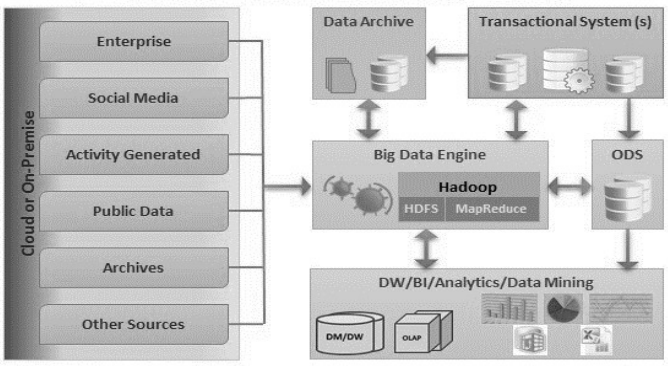


Fig 1.2: Big Data Architecture

Note: Big Data architecture involves multiple sources that arrive at different intervals, formats, and volumes. Here's a high-level overview of an enterprise data management system with a Big Data engine (link: Big Data Basics - Part 2 - Overview of Big Data Architecture (mssqltips.com).

1.5 HISTROY

Data gains value from processing programs. Big Data analytics began in the 1960s with IBM's systems for large data volumes, followed by database development in the 1970s and 1980s. Cloud computing accelerated Big Data adoption, enhancing decision-making. The concept of large datasets gained traction around 2005 with social media, coinciding with the launch of Hadoop for managing massive datasets and the rise of NoSQL databases. Open-source tools like Hadoop and Spark have made Big Data more manageable and cost-effective, leading to significant data generation from multiple sources.

1.6 APPLICATIONS

1. BANKING

1. Banking

- **Customer Insights:** Tailors products and services based on behavior analysis.
- **Fraud Detection:** Identifies unusual transaction patterns to prevent fraud.
- **Credit Scoring:** Improves assessments using diverse data sources.

2. Education

- **Learning Analytics:** Analyzes performance to provide personalized support.
- **Personalized Learning:** Develops adaptive systems for individual learning needs.
- **Curriculum Development:** Uses performance data for effective curriculum design.

3. Media

- **Audience Analytics:** Tracks viewer habits for content recommendations.
- **Content Optimization:** Personalizes content and assesses performance.
- **Production Insights:** Predicts success of scripts and movies.

4. Healthcare

- **Clinical Support:** Assists in diagnosis and treatment planning.
- **Patient Monitoring:** Enables remote monitoring for timely interventions.
- **Population Health:** Identifies high-risk groups for proactive care.

5. Agriculture

- **Precision Farming:** Optimizes input usage with data-driven insights.
- **Crop Monitoring:** Uses sensors for real-time health assessments.

THE CONVERGENCE OF IOT AND BIG DATA

2.1 THE CONVERGENCE OF IOT:

2.1 Convergence in the Internet of Things (IoT)

Convergence in IoT involves integrating various platforms, systems, and technologies to enhance interaction and information exchange among devices. Key aspects include:

1. **Interoperability**: Ensuring devices communicate regardless of manufacturer or technology through standardized protocols.
2. **Unified Platforms**: Developing platforms to manage diverse IoT devices and data streams.
3. **Data Integration and Analytics**: Combining and analyzing data from multiple sources for actionable insights.
4. **Edge Computing**: Processing data closer to its source to reduce latency and enable real-time responses.

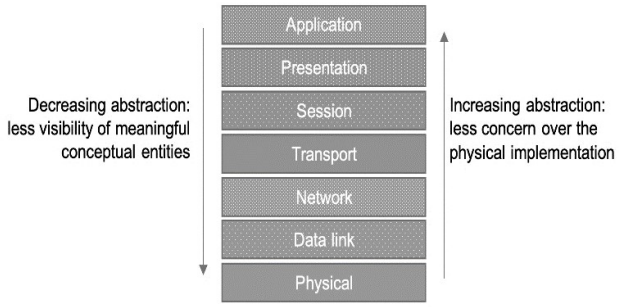
2.2 The Convergence of Big Data

Big data convergence integrates various processes for managing and analyzing large volumes of data, driven by its increasing significance across industries. Key aspects include:

1. **Data Variety**: Combining structured, semi-structured, and unstructured data formats.
2. **Data Volume**: Managing massive datasets that exceed traditional systems' capabilities.
3. **Data Velocity**: Processing real-time data streams for timely insights.
4. **Data Veracity**: Ensuring data quality through validation and cleansing.

2.3 IoT Architecture

The IoT architecture adapts to various applications without a universally accepted standard. It is built on fundamental processes that define its operational flow across different sectors.



Note: The image shows seven layers of IoT architecture: Physical Layer, Data Link Layer, Network Layer, Transport Layer, Session Layer, Presentation Layer, and Application Layer.

1. **Physical Layer:** The sensing layer collects data from the environment using sensors and

actuators for variables like temperature and humidity, connecting to the network layer via wired or wireless methods.
2. **Data Link Layer:** This layer manages communication protocols for device connectivity, including cellular networks, Wi-Fi, Bluetooth, and Zigbee, ensuring devices can communicate within a local network.
3. **Network Layer:** Responsible for enabling connectivity among devices, this layer uses technologies like 4G, 5G, and routers to facilitate communication both within and between networks, incorporating security measures like encryption.
4. **Transport Layer:** Ensures reliable and efficient data transmission between devices, managing data integrity and flow control, and addressing issues like retransmissions and packet loss.
5. **Session Layer:** Manages communication protocols for data transmission between IoT components, often using TCP or UDP to facilitate messaging.
6. **Presentation Layer:** Prepares and formats data for the application layer, handling translation, encryption, and compression to ensure proper data display.
7. **Application Layer:** The top layer interfaces directly with users, providing functionality through web portals and mobile apps, and includes middleware for data exchange and analytics tools for insight generation.

2.4 IOT DEVICE LAYERS (SENSORS, ACTUATORS, GATEWAYS):

The "IoT device layer" is a crucial level in an IoT system, housing the hardware necessary for data collection and user interaction.

1. **Sensors**: Core components that convert analog to digital signals, linking the physical and digital worlds. They gather data on parameters like temperature and humidity.
    - **Examples**: Temperature sensors, motion sensors, environmental sensors, GPS modules, cameras.
2. **Actuators**: Devices that convert electrical signals into movement, controlling physical objects based on sensor data.
    - **Examples**: Motors, valves, relays, pumps, locks.
3. **Gateways**: Data aggregators that connect sensors to the internet, transferring collected data to backend systems. Some sensors can function as built-in gateways, eliminating the need for separate devices.

2.5 COMMUNICATION PROTOCOLS IN IOT:

Communication Protocols in IoT: Essential for data sharing among devices, these protocols vary based on device type, range, power, and application.

Common IoT Protocols:

1. **MQTT**: Lightweight, ideal for low-bandwidth networks, using a publish-subscribe model.
2. **HTTP/HTTPS**: Flexible for web-based applications, though not IoT-specific.
3. **CoAP**: Designed for devices with limited resources, using UDP for efficiency.
4. **AMQP**: A standardized, secure messaging protocol for various IoT applications.
5. **WebSocket**: Supports real-time communication through persistent connections.

III. IOT DATABASES AND DATA WAREHOUSES

3.1 IoT Databases

IoT databases manage vast amounts of data from IoT devices, focusing on high volume, velocity, and varied information types.

1. **Time-Series Databases**: Store time-indexed data, ideal for sensor readings and optimized for retrieval.
2. **NoSQL Databases**: Handle unstructured/semi-structured data, highly scalable for large data volumes.
3. **In-Memory Databases**: Store data in RAM for rapid retrieval, suitable for real-time applications.
4. **Graph Databases**: Manage interconnected data, useful for complex relationships in IoT entities.
5. **Spatial Databases**: Optimize for location-based data, tracking movements or monitoring areas.

IoT Data Warehouses

IoT data warehouses store and analyze large volumes of IoT data, supporting business intelligence applications.

1. **Data Aggregation and Analysis**: Consolidate data for comprehensive insights.
2. **Data Processing and Transformation**: Preprocess IoT data for long-term analysis.
3. **Historical Data Storage**: Store large historical datasets for trend analysis.
4. **Integration with BI Tools**: Facilitate dashboard creation and reporting.
5. **Data Security and Compliance**: Ensure protection and regulatory compliance.

3.2 Challenges in IoT Data Handling

The explosion of IoT data necessitates reevaluation of data management strategies, focusing on scalability, integration, and security.

1. **Infrastructure Scaling**: Need for flexible, global scaling to manage data volumes.
2. **Data Gravity**: Increased data volume enhances value, necessitating efficient management.
3. **Secure Integration**: Challenges in securely connecting and managing data across devices.
4. **Data Security**: Emphasis on encrypting data at rest and during transfer.
5. **Scalability**: Systems must adapt to growing workloads without compromising performance.

3.3 Data Processing for IoT and Big Data

Processing IoT and big data involves various key factors:

1. **Data Collection**: Gathering sensor data via protocols like MQTT or HTTP.
2. **Data Ingestion**: Using streaming systems for real-time data input.
3. **Data Storage**: Employing distributed systems or cloud-based solutions.
4. **Data Transformation**: Cleaning and formatting raw data for analysis.

3.4 Stream Processing in IoT and Big Data

Stream processing allows real-time data analysis and decision-making.

1. **Data Ingestion for IoT**: Continuous data streams require efficient ingestion.
2. **Real-time Analytics**: Monitoring data for immediate insights.
3. **Immediate Responses**: Quick actions based on analyzed data.
4. **Event Processing**: Identifying trends through complex event processing.
5. **Scalability**: Systems must handle large data volumes efficiently.

3.5 Analytics in Real-Time for IoT and Big Data

Real-time analytics provide instant insights critical for various applications.

1. **Instantaneous Insights**: Quick decision-making based on live data.
2. **Predictive Maintenance**: Anticipating equipment issues to reduce downtime.
3. **Supply Chain Monitoring**: Optimizing logistics and inventory.
4. **Financial Fraud Detection**: Identifying fraudulent activity immediately.
5. **Customer Engagement**: Tailoring promotions in real-time.
6. **Healthcare Monitoring**: Spotting anomalies in patient data.
7. **Edge Analytics**: Processing data locally for faster responses.

Stream processing and real-time analytics are vital for leveraging IoT and big data effectively, enhancing operational efficiency and responsiveness.

IV. BIG DATA TECHNOLOGIES

4.1 Hadoop

Hadoop is an open-source, distributed computing framework for storing and processing large data across clusters of servers. Developed by Apache, it's key in

big data, with applications in data warehousing, machine learning, and analytics.

Key components:

- **Distributed Storage (HDFS)**: Divides data into blocks (128/256MB) and replicates them for reliability.
  - **Name Node**: Manages metadata.
  - **Data Nodes**: Store data blocks.
  - **Data Locality**: Processes data where it is stored.
  - **Write-Once, Read-Many**: Optimized for batch processing.
- **Cluster Architecture**: Master (Name Node) manages metadata; Worker (Data Nodes) store data blocks. Features like rack awareness improve fault tolerance.
- **MapReduce**: Parallel processing framework using key-value pairs.
  - **Map**: Breaks data into chunks, processes them, and groups results.
  - **Reduce**: Aggregates the results.
- **Fault Tolerance**: Data replication, task retries, and redundancy ensure reliability. HDFS uses checksums to verify data integrity.
- **Ecosystem**: Includes Hive (SQL-like queries), HBase (NoSQL), Spark (in-memory processing), Pig (data transformation), Sqoop (data transfer), and more.

4.2 MAPREDUCE

MapReduce is a programming model for efficiently processing large datasets in distributed computing environments. Initially developed by Google and later adopted by Apache Hadoop, it allows parallel data processing across multiple machines, ensuring scalability and fault tolerance.

1. **Parallel Processing**: Breaks down data into smaller chunks for simultaneous processing across nodes.
2. **Fault Tolerance**: Ensures redundancy and rescheduling of failed tasks.
3. **Data Localization**: Moves computation to where the data is stored to reduce network congestion.
4. **Data Flow Control**: Manages efficient data movement between processing stages.
5. **Distributed File System (DFS)**: Stores large datasets across multiple machines.
6. **User-Defined Functions (UDFs)**: Custom map and reduce functions defined by the user for data transformation.

Parallel Processing

- Input data is split into "input splits" for parallel processing by map tasks.
- Map tasks process these splits, producing key-value pairs.
- Intermediate data is shuffled and sorted in parallel.
- Reduce tasks finalize the output.

Fault Tolerance

- Data replication ensures availability during failures.
- Tasks are rescheduled on different nodes if needed.
- Intermediate states are periodically saved for faster recovery.
- Slow tasks are duplicated for faster completion.
- Heartbeat monitoring detects node failures promptly.

Data Localization

- Computation happens on nodes where data is stored, reducing data transfer across the network.

Data Flow Control

- Data is split into chunks for even distribution.
- Intermediate data is shuffled and sorted to group related data by key.
- Reduce tasks process the grouped data to produce final output.

Distributed File System (DFS)

- Distributes data blocks across nodes and replicates them for fault tolerance.
- Stores data close to computation for efficiency.

User-Defined Functions (UDFs)

- Custom map functions generate intermediate key-value pairs.

- Reduce functions process these pairs to produce final results.

Components of MapReduce

- Input data is split into input splits.
- The map function generates intermediate key-value pairs.
- Intermediate data is shuffled and sorted.
- The reduce function processes grouped data.
- The job tracker manages job execution, task assignment, and resources.
- Final output data is stored, often in HDFS.

4.3 Apache Spark

Apache Spark is a multi-language engine for executing data engineering, data science, and machine learning tasks on single-node machines or clusters. It is built for speed, scalability, and programmability, making it ideal for Big Data applications. Spark supports Python, Scala, Java, and R, and integrates with frameworks such as Hadoop, SQL, TensorFlow, and PyTorch.

- Spark's main abstraction is a distributed collection called a **Dataset**, which can come from sources like files or databases. It offers operations such as **map, filter, reduce, join, groupBy,** and **aggregate** for transforming and analyzing data. Spark also supports **SQL queries**, **streaming data**, **graph processing**, and **machine learning**.
- With a large open-source community, Spark is widely used in many industries for scalable computing.

V BIG DATA AND IoT SECURITY

5.1 Big Data Security

Big data security protects large datasets from breaches, data loss, and unauthorized access. Key considerations include:

1. **Data Encryption:** Encrypt data at rest and in transit.
2. **Access Control:** Use Role-Based Access Control (RBAC) or Attribute-Based Access Control (ABAC).
3. **Authentication:** Implement multi-factor authentication (MFA).
4. **Authorization:** Define user roles and permissions.
5. **Data Masking/Anonymization:** Protect sensitive data like PII.

5.2 IoT Security

IoT security ensures the safety, privacy, and reliability of connected devices. Key considerations include:

1. **Device Authentication:** Ensure only authorized devices connect.
2. **Secure Boot/Updates:** Secure boot processes and firmware updates.
3. **Data Encryption:** Encrypt data in transit and at rest using TLS.
4. **Access Control:** Use RBAC for device permissions.
5. **Network Segmentation:** Separate IoT devices from critical networks.

Conclusion

Big Data and IoT security are critical in today's digital world due to their rapid expansion across industries like healthcare, manufacturing, and smart homes. This growth increases security risks, making data protection essential to prevent identity theft, financial loss, and reputational damage. Strong access control, authentication, data governance, and compliance with regulations like GDPR and CCPA are vital. Continuous monitoring and a clear incident response plan can mitigate breaches. Industry collaboration on threats and vulnerabilities enhances security practices. Ensuring Big Data and IoT security requires ongoing vigilance, technology adaptation, and education.

References

[1] A. Wesolowski, C. O. Buckee, L. Bengtsson, E. Wetter, X. Lu, and A. J. Tatem, "Commentary: Containing the Ebola Outbreak – the Potential and Challenge of Mobile Network Data," PLOS Curr. Outbreaks, Sep. 2014.

[2] "CDC Tracks Cell Phone Location Data to Halt Ebola," Nextgov.com. [Online]. Available: http://m.nextgov.com/it-modernization/2014/10/cdctracks-cell-phone-location-data-halt-ebola/96239/. [Accessed: 22-Jan-2018].

[3] A. Rajandekar and B. Sikdar, "A Survey of MAC Layer Issues and Protocols for Machine-to-Machine Communications," IEEE Internet Things J., vol. 2, no. 2, pp. 175–186, Apr. 2015.

[4] S. Amendola, R. Lodato, S. Manzari, C. Occhiuzzi, and G. Marrocco, "RFID Technology for IoT-Based Personal Healthcare in Smart Spaces," IEEE Internet Things J., vol. 1, no. 2, pp. 144–152, Apr. 2014.

[5] C. Hennebert and J. D. Santos, "Security Protocols and Privacy Issues into 6LoWPAN Stack: A Synthesis," IEEE Internet Things J., vol. 1, no. 5, pp. 384–398, Oct. 2014.

[6] Z. Shelby, K. Hartke, and C. Bormann, "The Constrained Application Protocol (CoAP)," Proposed Standard 7252, 2014.

[7] Open Connectivity Foundation, "OCF - Specifications," Open Connectivity Foundation (OCF).

[8] J. Gold, "What is the internet of things (IoT)," Network World, 14-Jul-2017. [Online]. Available: http://www.networkworld.com/article/3207535/internetof-things/what-is-iot.html. [Accessed: 18-Jul-2017].

[9] Amazon, "Alexa." [Online]. Available: https://developer.amazon.com/alexa. [Accessed: 14-Aug-2017].

[10] Alphabet Inc., "Actions on Google | Actions on Google," Google Developers. [Online]. Available: https://developers.google.com/actions/. [Accessed: 30-Oct-2017].