Paper 1**: ML4IoT: A Framework to Orchestrate Machine Learning Workflows on Internet of Things Data**

The ML4IoT framework is designed to manage machine learning (ML) workflows specifically tailored for Internet of Things (IoT) data. It encompasses several key components, including data ingestion, which gathers IoT data from various sources, and workflow orchestration, which defines tasks such as data preprocessing, model selection, and evaluation. The framework supports a variety of ML models and is built to scale for large IoT deployments, optimizing resources efficiently. During workflow execution, the framework handles data preprocessing by cleaning, normalizing, and engineering features, followed by model training and evaluation to assess performance. Ultimately, ML4IoT facilitates real-time predictions upon deployment. Its use cases include predictive maintenance, anomaly detection, and energy optimization, while its primary research contribution lies in formalizing ML workflows for IoT data.

Paper 2: **Big Data Workflows: Locality-Aware Orchestration Using Software Containers**

In Internet of Things (IoT), significant amounts of data are generated at edge networks. With the data processing happening on geographically distributed systems across edge and cloud resources, reducing the delay and cost of transferring data over the network becomes crucial. Transferring massive amounts of data to the cloud is expensive and may incur latency, making low-latency scenarios unfeasible. To address these issues, the edge computing paradigm aims to complement cloud computing by leveraging hardware resources situated closer to the edge of the network to offload processing, reduce transfer cost, and satisfy the latency requirements.

The authors have proposed architecture for managing and executing big data workflows is structured around three main layers, each with distinct responsibilities

(Control layer, Data layer, and Compute layer).

1. Control Layer - This layer Manages the execution of workflows based on their definitions. Ensures correct step sequencing, processes data correctly, and coordinates the execution of workflows.
2. Data Layer- This layer Handles all aspects of data storage, retrieval, and movement between hosts.
3. Compute Layer- This layer Contains the processing logic for workflow steps. like Multiple compute steps, each performing specific processing tasks within the workflow.

In Data Processing They have used the parallel data processing

Parallel Processing: Each data unit is processed independently, allowing multiple units to be processed simultaneously across different computer steps.

Also, they have used Centralized Architecture which helped them in Data -Locality and Simplified Management.

Paper 3: **BDPS: An Efficient Spark-Based Big Data Processing Scheme for Cloud Fog-IoT Orchestration**

This paper introduces a novel framework BDPS (Big Data Processing Scheme), designed to enhance data processing efficiency in Cloud-Fog-IoT ecosystems using Apache Spark's which is a Resilient Distributed Datasets (RDDs). The authors address the challenges of high latency and inefficient data management in IoT networks by optimizing data routing and processing through in-memory computing and advanced algorithms.

An unprecedented amount of data is being generated, putting a tremendous strain on the Internet. That is where the cloud comes in, by connecting the IoT device user to the cloud - be it for device registration, device identity, storing data, or accessing enterprise data. Data collected through IoT devices is stored and processed on the cloud since IoT devices can be in a state of motion, the cloud serves as a collection point in closest proximity, minimizing the latency in reporting up the data points and providing a response back to the IoT application. So, from IOT platforms running entirely on the cloud to the interfaces used by customers to interact with these devices, to the backend analytics platforms - cloud computing supports and enables IoT.

The authors present BDPS, a Spark-based framework tailored for decentralized data processing across cloud and fog networks, significantly improving data delivery times and reducing network overhead.

The use of depth-first search-based shortest path algorithms in BDPS for efficient data routing is an innovative approach that outperforms traditional algorithms like Bellman-Ford, Floyd-Marshall, Dijkstra, and Hadoop’s map-reduce in terms of latency and resource management.

The paper provides a detailed performance analysis, demonstrating BDPS’s superiority in handling real-time data processing needs within IoT networks compared to existing methods.

Paper 4**: Cloud-Edge Orchestration for the Internet-of-Things: Architecture and AI-Powered Data Processing**

IoT applications generate vast amounts of data. Traditional cloud computing introduces latency due to physical distance from IoT devices. Edge cloud computing brings the cloud closer to devices, reducing delays. Cloud-edge orchestration combines cloud and edge resources.

The data processing optimization using AI.

1. Offloading: Offloading is critical for cloud-edge orchestration, determining which computation tasks are transferred to mobile edge devices, edge servers, or cloud servers. This decision depends on factors like computing capability, communication delay, power consumption, and IoT application requirements. Ai plays a crucial role in optimizing these decisions.
2. Edge server placement: It is vital due to the decentralized nature of edge computing. The placement affects IoT application performance and must consider scalability, device heterogeneity, and service fairness.
3. Network economy: examines economic factors in cloud-edge ecosystems, especially where multiple providers are involved.
4. Microservices, a service-oriented architecture variant, enable modular service deployment, aiding in configuration and fault management.
5. Efficient resource management in cloud-edge architectures is essential for optimizing the distribution of IoT computations.

**Paper 5:**