DML Group No: 18

Group Members Names:

1. ANIRUDDHA DILIP MADURWAR - 2023AA05982
2. PEYALA SAMARASIMHA REDDY - 2023AA05072
3. PEGALLAPATI SAI MAHARSHI - 2023AA05924
4. TUSHAR DEEP - 2023AA05885

Part 1: Theory and Concepts (10 Points)

1. Define **vertical partitioning** and **horizontal partitioning** in the context of AI models.
2. Compare the advantages and disadvantages of these partitioning methods, focusing on computational efficiency, scalability, and real-world application in wireless networks.

---

1. Define **vertical partitioning** and **horizontal partitioning** in the context of AI models.

- **Vertical Partitioning**:
    - **In Data**: Vertical partitioning involves splitting a dataset by its features (columns). Each partition contains different attributes but represents the same set of data instances. This is commonly used when different organizations or systems store only part of the total feature set for security or efficiency reasons.
        - **Example**: In a federated learning setup, a hospital may store a patient's medical history, while an insurance company holds their financial records. Both institutions collaborate on a machine learning model without sharing raw data.
    - **In Models**: Vertical partitioning in AI models refers to dividing the model into different layers or functional components. Different computational units handle different aspects of processing.

        - **Example**: In a neural network, one server might handle feature extraction (lower layers), while another server manages classification (higher layers).
- **Horizontal Partitioning**:
    - **In Data**: Horizontal partitioning divides data by instances (rows). Each partition contains a subset of records but retains all features. This is useful for scaling large datasets across multiple servers or locations.
        - **Example**: A mobile network provider collects call data from different regions. Each region stores user records separately, but every partition has the same attributes (e.g., call duration, location, signal strength).
    - **In Models**: Horizontal partitioning in AI models means distributing different subsets of training data across multiple machines, while keeping the model structure the same.
        - **Example**: In federated learning for wireless networks, different edge devices train the same model on their local user data before sharing updates with a central server.

---

2. Compare the advantages and disadvantages of these partitioning methods, focusing on computational efficiency, scalability, and real-world application in wireless networks.

Comparison of Vertical and Horizontal Partitioning

Aspect	Vertical Partitioning	Horizontal Partitioning
Memory Usage	Reduces memory load by processing only a subset of features at a time.	Reduces memory demand per node by splitting large datasets across machines.
Parallelization	Limited, as features may be interdependent and require synchronization.	Highly parallelizable, since different nodes train on different subsets of data independently.

Training Time	Can be slower due to dependencies across feature partitions.	Faster, as multiple nodes train in parallel.
Computational Load	Each node processes a different part of the model, which may lead to imbalanced workloads.	Load is evenly distributed since each node processes the same model on different data.
Scalability	Scales well when features are independent and can be processed separately.	Scales efficiently for large datasets, as new nodes can be added to handle more data.
Communication Overhead	High, as feature alignment and merging are required before training.	Lower, as each node processes complete records independently.
Privacy Protection	Stronger, as different institutions can train models together without sharing raw data.	Weaker, since entire records are stored on each node, increasing exposure risk.
Security Risks	Higher risk during feature merging, as combining datasets can reveal sensitive information.	Risk of data breaches, as complete user records may be transferred.
Federated Learning	Used when different parties hold different feature sets and collaborate without sharing raw data (e.g., hospital vs. insurance company).	Used when different network nodes hold different user records and need to train a common model (e.g., mobile towers in different cities).
Edge AI	Useful when different edge devices specialize in processing different types of features (e.g., image processing vs. text analysis).	Useful for training AI models at different network locations by distributing user data across multiple devices.
Wireless Network Optimization	Helps optimize AI models for feature-specific tasks (e.g., predicting network congestion based on signal characteristics).	Useful for real-time data processing across multiple network nodes (e.g., traffic load balancing in cellular networks).
Synchronization	Requires coordination between different feature sets before training.	Requires periodic updates across devices to merge model updates.

Result Integration	More complex, as different feature sets need to be combined correctly.	Easier, since all data samples follow the same structure.

Comparison of Vertical and Horizontal Partitioning in Real-World Wireless Network Applications

In wireless networks, AI-driven applications require efficient data partitioning to handle large-scale data processing and model training across distributed environments. Both **vertical partitioning** and **horizontal partitioning** play crucial roles in optimizing performance, scalability, and privacy.

1. Federated Learning in Wireless Networks

- **Vertical Partitioning** is useful when different telecom providers, network operators, or service providers hold **different feature sets** for the same users but need to train a shared AI model.
    - **Example**: A telecom provider might store network usage statistics (e.g., call duration, data consumption), while a cloud service provider holds application-specific metrics (e.g., video buffering rate). By using vertical partitioning, these entities can collaborate on a machine learning model without exchanging raw data, ensuring **data privacy** and regulatory compliance (e.g., GDPR).
- **Horizontal Partitioning** is beneficial when different **base stations, edge devices, or regional servers** hold **different user records** but need to train the same AI model.
    - **Example**: A federated learning system for mobile networks may assign **local AI models to different cell towers**, each trained on its own subset of user data. This allows **distributed training** without transferring massive amounts of data to a central server, improving **efficiency and latency**.

2. Edge AI for Wireless Networks

- **Vertical Partitioning** can be applied when different **network layers or hardware units** specialize in different types of processing.
    - **Example**: In a 5G network, one AI model might focus on **radio signal analysis**, while another processes **traffic management data**. These models can be partitioned across different hardware accelerators (e.g., GPUs for deep learning, FPGAs for signal processing).
- **Horizontal Partitioning** is useful for **scaling AI inference** across multiple edge devices.
    - **Example**: In a smart city, different **IoT gateways** process network traffic data from different geographic regions. Each gateway runs the same AI model on **local data**, reducing the need for central computation and enabling **real-time decision-making** (e.g., dynamic bandwidth allocation for autonomous vehicles).

3. Network Optimization and Load Balancing

- **Vertical Partitioning** can help optimize AI-driven network resource management by **dividing responsibilities** among different entities.
    - **Example**: A telecom provider might use **one AI model** for spectrum allocation and another for **user behavior analysis**. These models can be processed separately on different network nodes, improving modularity.
- **Horizontal Partitioning** is more efficient when distributing **real-time network monitoring** across multiple locations.
    - **Example**: Each cell tower can analyze **local congestion patterns** and apply traffic shaping policies **independently**, reducing the need for a centralized control system.

4. Privacy and Security Considerations in Wireless AI

- **Vertical Partitioning** is advantageous when different companies or regulatory bodies enforce strict **data privacy laws**. By ensuring that sensitive attributes remain separated across different parties, it minimizes privacy risks.
    - **Example**: A user's **biometric data** (stored by a telecom provider) and **online activity logs** (stored by a cloud service) can be used together to enhance fraud detection without sharing raw data.
- **Horizontal Partitioning** reduces the risk of **large-scale data breaches** by **keeping user records decentralized** across multiple edge devices.
    - **Example**: Instead of storing millions of user profiles in a central server, user data can be processed locally on **edge devices** to enhance **security and reduce latency**.

Conclusion

In real-world wireless networks, **vertical partitioning** is often used when different organizations or devices handle **specific types of data** (e.g., signal data vs. user behavior), ensuring privacy and modularity. On the other hand, **horizontal partitioning** is ideal for **distributed AI training** where data is split across multiple locations, enabling scalability and efficiency in edge AI applications.

In practice, a **hybrid approach** is often used, combining vertical partitioning for privacy-sensitive data handling and horizontal partitioning for large-scale distributed AI training across wireless networks.