Federated Learning with Cognitive Defense Mechanisms

A modular federated learning framework implementing cognitive defense strategies based on OODA loop and MAPE-K frameworks, with support for various attacks and defences.

Features

- **Modular Architecture**: Separation of concerns with pluggable attacks and defences
- **Multi-Client Orchestration**: Automated management of 10+ client processes with resource monitoring
- **Cognitive Defense**: OODA loop and MAPE-K framework implementation
- **Attack Simulation**: Label flipping, gradient noise, model replacement, and more
- **Explainable AI**: Decision logging with reasoning and evidence
- **Deterministic Experiments**: Reproducible results with proper seeding

Quick Start

1. Setup

```
# Clone and setup
git clone https://github.com/self1am/FL_CognitiveDefence.git
cd FL_CognitiveDefence
make setup
```

2. Run Basic Experiment

```
make run-basic
```

3. Run Custom Experiment

```
python -m src.orchestration.experiment_runner --config
experiments/configs/your_config.yaml
```

Project Structure

```
federated-cognitive-defense/
├── src/
│   ├── attacks/          # Attack implementations
│   ├── defences/         # Defense strategies
```

```
│   ├── clients/          # Client implementations
│   ├── server/           # Server implementations
│   ├── models/           # Neural network models
│   ├── datasets/         # Dataset handlers
│   ├── orchestration/    # Multi-client orchestration
│   └── utils/            # Utilities and configuration
├── experiments/
│   ├── configs/          # Experiment configurations
│   ├── scripts/          # Helper scripts
│   └── results/          # Experiment results
└── tests/                # Unit and integration tests
```

Configuration

Experiments are configured using YAML files. Example:

```yaml
experiment:
  experiment_name: "cognitive_defense_test"
  seed: 42
  num_rounds: 10
  server_address: "0.0.0.0:8080"

defense:
  strategy: "cognitive_defense"
  anomaly_threshold: 0.7

attacks:
  - enabled: true
    attack_type: "label_flip"
    intensity: 0.1
    target_clients: [0, 1, 2]

orchestration:
  num_clients: 10
  batch_size: 3
```

Hardware Requirements

- **MacBook M1 (8GB)**: Orchestrator + 2-3 lightweight clients
- **Azure VMs (4GB each)**: Server + 3-4 clients each
- **Total**: Support for 10+ concurrent clients

Distributed Setup

For multi-machine experiments:

```
# Setup distributed environment
./scripts/run_distributed.sh

# Monitor progress
tail -f logs/experiment_name.log
```

Development

```
# Install development dependencies
make dev-install

# Run tests
make test

# Format code
make format

# Lint code
make lint
```

Experiment Results

Results are automatically saved to:

- `experiments/results/`: JSON experiment summaries
- `logs/`: Detailed execution logs
- Individual client logs with training history

Next Steps

1. **FEMNIST Integration**: More realistic FL dataset
2. **Quantum Neural Networks**: PennyLane integration
3. **Advanced defences**: Krum, Trimmed Mean, FreqFed
4. **Adaptive Attacks**: Learning-based adversarial strategies

Contributing

1. Create feature branches for new functionality
2. Add tests for new components
3. Update documentation
4. Submit pull requests

License

MIT License - see LICENSE file for details.