FedGAD: Divergence-Regularized Federated GANs for Effective Cyber Attack Detection on Non-IID and Unlabeled Edge Activity Data

System Requirements

Hardware

- **Processor**: Intel Core i5-1235U (12th Gen) or equivalent
- **RAM**: 16 GB minimum
- **GPU**: CUDA-compatible GPU (optional but recommended)
- **Storage**: At least 10 GB free space for datasets and results

Software

- **OS**: Windows 10/11, or macOS, Linux (Ubuntu 20.04+)
- **Python**: 3.12.4 (recommended) or 3.8+
- **CUDA**: 11.8+ (if using GPU)

Installation

Step 1: Clone the Repository

```
git clone https://github.com/yourusername/fedgad.git

cd fedgad
```

Step 2: Create Python Virtual Environment

On Windows:

```
python -m venv venv

venv\Scripts\activate
```

Step 3: Upgrade pip

```
pip install --upgrade pip
```

Step 4: Install PyTorch

Choose the appropriate command based on your system:

For GPU (CUDA 12.1):

```
pip install torch torchvision torchaudio --index-url https://download.pytorch.org/whl/cu121
```

For CPU only:

```
pip install torch torchvision torchaudio
```

Step 5: Install Required Dependencies

```
pip install -r requirements.txt
```

Step 6: Verify Installation

```
python -c "import torch;
print(f'PyTorch Version: {torch.__version__}');
print(f'CUDA Available: {torch.cuda.is_available()}')"
```

Expected output:

```
PyTorch Version: 2.x.x
CUDA Available: True  # or False if CPU only
```

Project Structure

```
FedGAD/
── main.py               # Main training script
── dataloader.py         # Dataset loading and federated splitting
── model.py              # BiLSTMTCNGAN architecture
── regularizers.py       # Regularization components
── ReadMe.pdf            # This file
── datasets/             # Dataset directory (create this)
  ── ToN_IoT.csv
  ── CSE_CIC_IDS.csv
── results/              # Results directory (auto-created)
        ── results_*.txt
── checkpoints/          # Model checkpoints (optional)
```

Dataset Preparation

Step 1: Download Datasets

ToN-IoT Dataset

1. Download the processed CSV file
2. Place it in the datasets/ directory

CSE-CIC-IDS2018 Dataset

1. Download the dataset
2. Place it in the datasets/ directory

Step 2: Create Dataset Directory

```
mkdir -p datasets
```

Step 3: Verify Dataset Format

Ensure your CSV files have:

- Feature columns (network traffic features)
- A label column indicating normal/attack traffic
- Proper headers

Configuration

Hyperparameters

The following hyperparameters are used in the experiments (defined in main.py):

Parameter	Value	Description
Epochs (E)	100	Number of training epochs
Batch Size (B)	32	Training batch size
Clients (N)	100	Number of federated clients
Learning Rate (η)	0.001	Optimizer learning rate
Regularizer (λ)	0.01	Regularization weight
Base Regularization (λ_base)	0.01	Base regularization parameter
Scaling Parameter (α)	0.5	Scaling factor
Latent Dimension (d)	100	GAN latent space dimension
Bi-LSTM Hidden Size (h_lstm)	256	LSTM hidden units
TCN Hidden Channels (h_tcn)	128	TCN hidden channels
Dropout Rate	0.3	Dropout probability

Modifying Configuration

Edit main.py to change hyperparameters:

```
if __name__ == "__main__":
    file_path = "datasets/ToN_IoT.csv"  # Change dataset here
    train_federated_model(
        file_path,
        num_epochs=100,        # Modify epochs
        lr=0.001,              # Modify learning rate
        batch_size=32,         # Modify batch size
        iid=True,              # IID or non-IID distribution
        labelled=True,         # Labelled or unlabelled data
        ablation_mode="FedGAD" # Ablation study mode
    )
```

Running Experiments

Basic Execution

```
# Activate virtual environment first
venv\Scripts\activate     # Windows
# Run main experiment
python main.py
```

Running Different Scenarios

1. Labelled IID (Default)

```
python main.py
```

2. Labelled Non-IID

Edit main.py and uncomment:

```
print("\nLabelled non-IID")
```

```
train_federated_model(file_path, iid=False, labelled=True, ablation_mode="FedGAD")
```

3. Unlabelled IID

```
print("\nUnlabelled IID")

train_federated_model(file_path, iid=True, labelled=False, ablation_mode="FedGAD")
```

4. Unlabelled Non-IID

```
print("\nUnlabelled non-IID")

train_federated_model(file_path, iid=False, labelled=False, ablation_mode="FedGAD")
```

Ablation Studies

Run different ablation modes by changing the ablation_mode parameter:

```
# FedAvg baseline

train_federated_model(file_path, ablation_mode="FedAvg")

# FedProx baseline

train_federated_model(file_path, ablation_mode="FedProx")


# LeCam regularization

train_federated_model(file_path, ablation_mode="LeCam")


# Full FedGAD (proposed method)

train_federated_model(file_path, ablation_mode="FedGAD")
```

Running with GPU

```
# Use specific GPU

CUDA_VISIBLE_DEVICES=0 python main.py


# Use CPU only

CUDA_VISIBLE_DEVICES="" python main.py
```

Output and Results

Console Output

During training, you'll see:

```
Epochs=100, LR=0.001, Batch=32, IID=True, Labeled=True, Mode=FedGAD

Epoch [1/100] | Loss=0.5234 | Reg=0.0123 | D=0.3456 | G=0.2341 | SSD=0.1234

...

Accuracy=0.9567, Precision=0.9432, Recall=0.9523, F1=0.9477, ADS=0.9501
```

Result Files

Results are saved to results_{ablation_mode}.txt:

```
Mode=FedGAD, IID=True, Labeled=True, Acc=0.9567, Prec=0.9432, Recall=0.9523,
F1=0.9477, ADS=0.9501
```

Performance Metrics

- **Accuracy**: Overall classification accuracy
- **Precision**: True positive rate
- **Recall**: Sensitivity
- **F1-Score**: Harmonic mean of precision and recall
- **ADS**: Attack Detection Score = (Precision + Recall + (1 - SSD)) / 3

Troubleshooting

Common Issues

1. CUDA Out of Memory

```
# Reduce batch size in main.py

batch_size=16  # or 8
```

2. Module Not Found Error

```
# Ensure all required files exist

ls -la  # Check for dataloader.py, model.py, regularizers.py
```

3. Dataset Loading Error

```
# Verify dataset path
file_path = "datasets/ToN_IoT.csv"  # Use correct path
```

4. Import Errors

```
# Reinstall dependencies
pip install -r requirements.txt --force-reinstall
```

5. Variable ablation_mode Not Defined

Ensure you've added the parameter to the function definition:

```
def train_federated_model(
    file_path,
    num_epochs=1,
    lr=0.001,
    batch_size=32,
    iid=True,
    labelled=True,
    ablation_mode="FedGAD"  # Add this line
):
```

System-Specific Issues

Windows

```
# If execution policy error (PowerShell)
Set-ExecutionPolicy -ExecutionPolicy RemoteSigned -Scope CurrentUser

# Use Python directly
python main.py
```

Advanced Usage

Enabling System Profiling

Uncomment the following sections in main.py:

1. **Runtime & Memory Profiling**

```
# Uncomment import statements from main.py
import time
import psutil
import os
# Uncomment profiling code blocks
start_time = time.time()
# ... (in training function)
```

2. **Mode Collapse Detection**

```
# Uncomment import from main.py
from mode_collapse import compute_mode_collapse
# Uncomment mode collapse computation
missing_modes, mode_coverage = compute_mode_collapse(...)
```

3. **Communication Overhead**

```
# Uncomment function and calculation
comm_overhead = calculate_comm_overhead(...)
```

Running Baseline Comparisons

```
#Uncomment baseline imports and modify model initialization:
from baseline import FedTSRGNet, FedTrust, ADGAN, FedGANIDS
# Then select model based on MODEL_NAME variable
```

Acknowledgments

- ToN-IoT Dataset: UNSW Canberra
- CSE-CIC-IDS2018: Canadian Institute for Cybersecurity