

```

"""
                                few_shot_examples.py

Few-shot学習用のサンプルデータ定義
異常検知のための正常・異常の例を提供（タイムスタンプ付き）
"""

# ネットワークトラフィック用Few-shotサンプル（タイムスタンプ付き）
NETWORK_FEW_SHOT = """
# 正常なトラフィックの例：
- [2025-10-17 09:15:23] 送信元IP: 192.168.1.10, 宛先IP: 8.8.8.8, ポート: 443, プロトコル: HTTPS, パケット数: 150, バイト数: 45000
  → 正常: 一般的なDNSサーバーへのHTTPS通信

- [2025-10-17 09:20:45] 送信元IP: 192.168.1.25, 宛先IP: 10.0.0.5, ポート: 22, プロトコル: SSH, パケット数: 50, バイト数: 12000
  → 正常: 社内サーバーへのSSH接続

- [2025-10-17 09:35:12] 送信元IP: 192.168.1.30, 宛先IP: 172.217.175.46, ポート: 443, プロトコル: HTTPS, パケット数: 200, バイト数: 75000
  → 正常: Googleサービスへの通常アクセス

# 異常なトラフィックの例：
- [2025-10-17 02:45:18] 送信元IP: 192.168.1.50, 宛先IP: 185.220.101.5, ポート: 9050, プロトコル: TCP, パケット数: 5000, バイト数: 2500000
  → 異常: Torネットワークへの大量通信（疑わしい匿名化トラフィック）

- [2025-10-17 03:22:30] 送信元IP: 192.168.1.100, 宛先IP: 複数, ポート: 445, プロトコル: SMB, パケット数: 10000, バイト数: 500000
  → 異常: ポートスキャンの可能性（SMBポートへの大量接続試行）

- [2025-10-17 14:55:40] 送信元IP: 192.168.1.75, 宛先IP: 203.0.113.10, ポート: 80, プロトコル: HTTP, パケット数: 80000, バイト数: 15000000
  → 異常: DDoS攻撃の可能性（異常な大量トラフィック）
"""

# システムログ用Few-shotサンプル
LOG_FEW_SHOT = """
# 正常なログの例：
- [2025-10-08 10:30:15] INFO: User 'admin' logged in from 192.168.1.10
  → 正常: 通常の管理者ログイン

- [2025-10-08 10:35:22] INFO: Database backup completed successfully
  → 正常: 定期バックアップの正常完了

- [2025-10-08 14:20:00] INFO: Service 'nginx' restarted successfully
  → 正常: サービスの正常な再起動

- [2025-10-08 16:45:30] WARNING: Disk usage at 75% on /dev/sda1
  → 正常: ディスク使用量の警告（閾値内）

# 異常なログの例：
- [2025-10-08 03:15:45] WARNING: Failed login attempt for user 'admin' from 185.220.101.5 (attempt 15)
  → 異常: 深夜の海外IPからの複数回ログイン失敗（ブルートフォース攻撃の可能性）

- [2025-10-08 11:20:10] ERROR: SQL injection attempt detected in parameter 'user_id': ' OR '1'='1
  → 異常: SQLインジェクション攻撃の検知

- [2025-10-08 15:30:00] CRITICAL: Unauthorized file access attempt: /etc/passwd by process 'unknown_proc'
  → 異常: 不正なファイルアクセス試行

- [2025-10-08 18:00:00] ERROR: Buffer overflow detected in application 'web_app'
  → 異常: バッファオーバーフロー攻撃の可能性
"""

# センサーデータ用Few-shotサンプル
SENSOR_FEW_SHOT = """
# 正常なセンサーデータの例：
- 時刻: 10:00, 温度: 22.5°C, 振動: 0.05mm/s
  → 正常: 標準的な動作範囲内

- 時刻: 10:05, 温度: 23.1°C, 振動: 0.06mm/s
  → 正常: わずかな変動は正常範囲

- 時刻: 10:10, 温度: 22.8°C, 振動: 0.04mm/s
  → 正常: 安定した動作状態

- 時刻: 10:15, 温度: 23.5°C, 振動: 0.07mm/s
  → 正常: 正常な範囲内の変動

# 異常なセンサーデータの例：
- 時刻: 10:20, 温度: 35.8°C, 振動: 2.5mm/s
  → 異常: 温度・振動ともに急激に上昇（軸受の異常発熱・振動の可能性）

- 時刻: 10:25, 温度: 45.2°C, 振動: 4.1mm/s
  → 異常: 危険レベルへの進行（早急な点検が必要）

- 時刻: 10:30, 温度: 52.3°C, 振動: 5.2mm/s
  → 異常: 緊急レベルの温度・振動（即座に機器停止を推奨）
"""

```

```
- 時刻: 10:35, 温度: 60.5°C, 振動: 7.8mm/s
→ 異常: 故障直前の危険な状態 (機器損傷のリスク)
"""
```

```
def get_few_shot_examples(data_type: str) -> str:
```

```
    """
    データタイプに応じたFew-shotサンプルを取得
    """
```

```
    Args:
```

```
        data_type: 'network', 'log', 'sensor' のいずれか
```

```
    Returns:
```

```
        Few-shotサンプルの文字列
    """
```

```
    examples = {
        'network': NETWORK_FEW_SHOT,
        'log': LOG_FEW_SHOT,
        'sensor': SENSOR_FEW_SHOT
    }
```

```
    return examples.get(data_type, "")
```

```
def load_custom_few_shot(filepath: str) -> str:
```

```
    """
    カスタムFew-shotサンプルをファイルから読み込み
    """
```

```
    Args:
```

```
        filepath: サンプルファイルのパス
```

```
    Returns:
```

```
        読み込んだサンプルの文字列
    """
```

```
    try:
```

```
        with open(filepath, 'r', encoding='utf-8') as f:
            return f.read()
```

```
    except FileNotFoundError:
```

```
        print(f"⚠ カスタムFew-shotファイルが見つかりません: {filepath}")
```

```
        return ""
```

```
    except Exception as e:
```

```
        print(f"⚠ Few-shotファイル読み込みエラー: {e}")
```

```
        return ""
```

```
def save_few_shot_template(data_type: str, filepath: str):
```

```
    """
    Few-shotサンプルのテンプレートをファイルに保存
    """
```

```
    Args:
```

```
        data_type: 'network', 'log', 'sensor' のいずれか
```

```
        filepath: 保存先のファイルパス
    """
```

```
    examples = get_few_shot_examples(data_type)
```

```
    if not examples:
```

```
        print(f"✗ 不正なデータタイプ: {data_type}")
```

```
        return
```

```
    try:
```

```
        with open(filepath, 'w', encoding='utf-8') as f:
            f.write(examples)
```

```
        print(f"✓ Few-shotテンプレートを保存しました: {filepath}")
```

```
    except Exception as e:
```

```
        print(f"✗ 保存エラー: {e}")
```

```
# テンプレート生成用のメイン関数
```

```
def main():
```

```
    """Few-shotテンプレートファイルを生成"""
```

```
    import os
```

```
    # few_shotディレクトリを作成
```

```
    os.makedirs('few_shot_examples', exist_ok=True)
```

```
    print("=" * 70)
```

```
    print("Few-shot学習用サンプルファイル生成")
```

```
    print("=" * 70)
```

```
    print()
```

```
    # 各データタイプのテンプレートを保存
```

```
    templates = [
        ('network', 'few_shot_examples/network_examples.txt'),
        ('log', 'few_shot_examples/log_examples.txt'),
        ('sensor', 'few_shot_examples/sensor_examples.txt')
    ]
```

```
    for data_type, filepath in templates:
```

```

few_shot_examples.py
save_few_shot_template(data_type, filepath)

print()
print("=" * 70)
print("✓ すべてのテンプレートファイルを生成しました")
print("=" * 70)
print()
print("【生成されたファイル】")
print("  ・ few_shot_examples/network_examples.txt")
print("  ・ few_shot_examples/log_examples.txt")
print("  ・ few_shot_examples/sensor_examples.txt")
print()
print("【使用方法】")
print("  これらのファイルを編集してカスタムFew-shotサンプルを作成できます")
print("  異常検知プログラムで --few-shot-file オプションを使用して読み込みます")
print()

if __name__ == "__main__":
    main()

```