第2章 ①追加学習なし，②追加学習やRAGで補完，③従来手法と組み合わせたハイブリッド構成

LLMによる異常検出の3つのアプローチ

氏森 充

大規模言語モデル（LLM）は，ログ解析やセキュリティ検出など，多様な異常検出タスクへの応用が注目されています．既存のLLMの能力を最大限に引き出し，システムの要求仕様（導入コスト，精度，リアルタイム性）に合わせるためには，適切なアプローチを選択することが不可欠です．

表1　アプローチ①…追加学習を必要としない手法

手　法	概　要	具体例・特徴
Zero-shot prompting	タスク定義だけを与え，LLMに直接，正常／異常を判断させる	例：「このログに異常があるか判断してください」．事前データ不要で試験が容易
Few-shot prompting	正常／異常の例を数件提示し，それに倣って判定させる	少量の例示で学習効果を再現でき，安定性が向上
Chain of Thought（CoT）	判断手順を明示して段階的に思考させる	例：「平均→偏差→特異点→最終判断」というステップを指示．推論の透明性が高い
自己反省型／評価付きプロンプト	モデル自身に複数案を生成させ，根拠を比較，自己評価させる	誤判断の抑制や説明の一貫性向上に有効
テンプレート化＋条件付きプロンプト	入力形式や出力ラベルを定型化して安定した回答を得る	異常／正常などの表記を固定し，結果の整合性を確保する

表2　アプローチ①の各手法における利点と課題

手　法	主な利点	主な課題・注意点
Zero-shot prompting	• データ準備コストが最も低い • 未知の異常にも柔軟に対応できる	• プロンプト設計の影響が大きく誤判定しやすい • 再現性が低く出力のばらつきが大きい
Few-shot prompting	• 少数例で安定性と精度を改善できる • 事前学習モデルの知識を活用可能	• 例示の選び方によって結果が偏る • トークン数が増えやすくスケールしにくい
Chain of Thought（CoT）	• 推論過程を明示でき説明性が高い • 人間の理解と整合的な出力が得られる	• 長文出力による遅延 • コスト増の可能性 • 段階指示が複雑すぎると誤動作する
自己反省型／評価付きプロンプト	• 誤判断の抑制と一貫性向上が期待できる	• 生成負荷が高くリアルタイム処理には不向き
テンプレート化＋条件付きプロンプト	• 形式が固定化され出力の安定性が高い • システム連携に適する	• 表現の柔軟性が低下し，未知のパターン検出には不向き

表3　アプローチ①の安定性を高めるための方法

課　題	対応策	解　説
プロンプト設計の感度・出力のばらつき	構造化プロンプト／テンプレート化，安定性評価（Prompt Stability），A/Bテスト	出力形式や語彙を固定化しセマンティック安定性の観点で評価・改善を繰り返す
Few-shotの例示選定感度	クラスタリングによる代表例選定，反事例（ネガティブ例）併用，動的プロンプト適応	ログや時系列データをクラスタごとに代表例化し，更新に合わせてプロンプトを動的変更する
出力長・遅延	トークン数制限，段階出力，Early-stopping制御	要約＋詳細回答の2段階生成で遅延を低減する
自己反省型プロンプトの負荷	部分反省／限定反省，反省回数制限，アンサンブル統合	比較回数を減らし，複数生成の統計的統合で安定性を確保する
テンプレート化による柔軟性低下	複数テンプレート保持，自由入力セクション併設	異常の多様性に対応できるようテンプレートを柔軟に切り替え可能にする