言語モデルを用いたパラメータ異常検出：複数パラメータの組み合わせに対する異常

内田博規[1] 富永圭太郎[2] 板井秀樹[2]
李玉潔[1] 中藤良久[1]
[1]九州工業大学大学院
[2]パナソニックシステムデザイン株式会社
uchida.hironori182@mail.kyutech.jp
nakatoh@ecs.kyutech.ac.jp

概要

本研究では、テキスト内に含まれる複数のパラメータの組み合わせに基づく異常を検出する手法に関する実験結果を報告する。これまでの研究では、BertMaskedLM 手法が高い有効性を持つことが示されている。しかし、先行研究ではモデルを 1 から学習させており、既存の学習済みモデルを用いて追加学習を行う方が精度向上が期待できると予測可能である。そこで、本研究では、先行研究で提案された改善手法による結果と、有名な学習済み BERT モデルを比較、さらに学習済み BERT モデルに同様の改善手法を適用した場合の結果を調査する。

1 はじめに

センサ、画像、音など、さまざまな情報がデータ化される現代社会において、パラメータ異常検知システムの需要は高まっている。本研究では、特にシステム開発に焦点を当て、テキストログを用いた異常検知の研究を行っている。しかし、ログには個人情報が含まれる場合があるほか、ラベル付けには専門的な知識が必要であることから、パラメータ異常検知用のデータセットが存在しない。この課題に対応するため、独自に課題設定とデータセットを用意して研究を進めている。用意したログは、2 つの文字列型パラメータと 1 つの整数型パラメータを持つものであり、これら 3 つのパラメータの組み合わせに基づく異常検出を目的としている。異常検出モデルとして、我々は BertMaskedLM をベースとした手法を提案している[1]。この手法では、対象ログを用いてモデルを 1 から学習しており、出現するパラメータが既知であることを前提として、事前にそれらを Tokenizer に登録している。これまでの研究では、異なる組み合わせにおいて整数型パラメータの範囲が重なっている場合、精度が低下することが報告されている[2]。また、Tokenizer に登録する辞書の語彙数を増やすことで精度が改善されることが確認されている。そこで、本研究では、語彙数が多い辞書を持つ大規模な学習済み BERT モデルと、先行研究手法の性能を比較検討する。

本研究の主な貢献は以下の通りである。

1. 大規模モデル（学習済み BERT）によるパラメータ異常検知の精度の調査
2. 対象ログを用いた追加学習による BERT モデルの精度評価

2 関連研究

ログの異常パターンには、シーケンス異常、パラメータ異常、時刻飛び、ログの欠落など多様な種類が存在する。しかし、データセットの不足により、これまでの研究は主にシーケンス異常または単一行の異常検知に限定されてきた。また、多くのシーケンス異常向けの手法では、精度向上のために不必要なパラメータ情報を欠落させることが多く、この情報選択の妥当性が疑問視されている[3]。本章では、特にパラメータを使用するシーケンス異常検出の研究に限定して紹介する。

2.1 NeuralLog

ログ解析を必要としない新しいアプローチであり、ログメッセージを直接意味ベクトルに変換し、Transformer モデルを使用して異常を検出する。この手法では、BERT を使用してログメッセージの意味を抽出し、Transformer ベースの分類モデルによって

異常を識別する。しかし、ログを分割する際、分割された文字列が数値の場合にそのデータが削除されるプロセスが含まれている。そのため、NeuralLog は整数型パラメータの異常検出において性能が低いと予測される[4]。

2.2 LAnoBERT

BERT を使用してログキーシーケンスを学習し、より柔軟な埋め込みを生成する手法である。この手法は、ログパーサーに依存せず、正規表現に基づいたシンプルな前処理を採用することで、情報損失を最小限に抑える工夫がなされている[4]。

3 提案手法

本章では、我々が提案する BertMaskedLM を用いた異常検知手法について説明する。本手法は本論文で初めて述べるものではなく、これまでの研究において実験されてきたものである[1, 5, 6]。
提案手法には、以下の条件が必要である。
条件: パラメータの候補が既知である。

3.1 異常検知までの流れ

１．テキストログから調査対象のログを抽出
正規表現などを用いて任意のログ構造を持つ行のみを抽出する。（本研究では抽出後のデータを使用）
２． Tokenizer にデータセットを辞書登録
提案手法では WordPieceTokenizer を使用する。学習データセットを用いて辞書に登録する際、先行研究に基づき Int 型パラメータの登録に次の 2 点の対応を行う。
1） サブワード分割を防ぐため、特殊文字として 1 つずつ登録する。
2） 辞書登録の順序性を無くすため、ランダムに登録する。
上記の対応によって精度向上が報告されている。
３．Tokenizer による文字列→ID 変換
ログを 1 行毎入力し、TokenID 列を作成する。
４．BertMaskedLM の学習方法
TokenID 列をランダムにマスクし、そのマスク部分をランダムに学習対象とするかどうかを選択する。マスキング手順の詳細は以下の通りである：
- トークン列の 15%をマスク対象として選択
- 選択されたトークンの 80%を、特別なトークン[MASK]に置き換え
  例: I love programming.→I [MASK] programming.
- 10%をランダムな別のトークンに置き換え
  例: I love programming. → I apple programming.
- 残りの 10%はもとのトークンを保持。
  例: I love programming. → I love programming.

５．学習済み BertMaskedLM を用いた異常検知
１）検知したい箇所を Mask してモデルに入力
２）モデルから Mask 部分のすべての予測値を取得
３）予測値からパラメータ候補 Token 部分のみ抽出
４）候補数分の予測値を正規化
５）閾値を計算(候補数の逆数)
６）閾値より小さい場合、異常と判定
例：異常判定
String 型(以下は本実験で使用した State Param の時):
- 候補パラメータ： State = {A, B, C}
- 閾値：1/3
- 正規化: P’(A)=Normalize(P(A), P(B), P(C))
- 異常判定: P’(A) < 1/3

Int 型(以下は本実験で使用した ValueParam の時)：
- 候補パラメータ： 0 < Value < 10000
- 閾値：1/10000
- ソフトマックス: P(N)=Softmax(Predictions)
  （ここで、N は MASK されたパラメータの ID）
- 異常判定: P’(A) < 1/10000

3.2 提案手法に対する効果的な改善方法

先行研究で実験された改善方法を以下に示す。
1. Tokenizer に 1~10000 の数値を個別登録[1]
2. Tokenizer に登録する数値をランダムする[1]
3. Tokenizer に対象データ以外のログを登録[6]

これらの対応を行うことで Value パラメータに対する異常精度が F1-Score で 0.541 から 1.0 まで大幅に改善した。以下に簡単な考察を示す。
考察
・Tokenizer に 1~10000 の数値を個別登録(f1=0.677)
WordPieceTokenizer では、数値がサブワード分割されるため、例えば 101 や 1001 が同じ「###1」として登録される。この問題の回避が原因だと考える。
・Tokenizer に登録する数値をランダム化(f1=0.891)
辞書 ID と数値の順番の関係を断ち切ることで各 TokenID 同士のベクトル距離が遠くなり、より別の Word として認識可能になったと考える。
・Tokenizer に対象データ以外のログを登録(f1=1.0)
TokenID の多様性によりベクトル距離が多様化、学習時のランダムマスクの際の TokenID が多様化、などが考えられる。

4 実験目的

これまでの研究から、Tokenizer および BertMaskedLM を多様性のあるログで学習させる方が、精度向上に効果的であることが示されている[6]。しかし、学習には多大なリソース（学習時間や計算資源）が必要であるため、大規模データセットを使用した検証は十分に行われてこなかった。

本研究では、大規模データセットで事前学習された BERT モデルを使用した場合の異常検知精度と、提案手法による異常検知精度の比較を行う。さらに、学習済み BERT モデルに対し、提案手法で用いた Tokenizer への登録方法および追加学習を適用した場合の精度も評価する。

5 実験条件

5.1 データセットの生成方法

本実験では以下の条件を基にランダムなパラメータ選択でログを生成した。

ログは”state=<*1>, value=<*2>, <*3>”のテンプレート構造を持つ。<*>には変動するパラメータが 1 つ入る。以下に各パラメータ候補を示す。

<*1>：[A, B. C]

<*2>：1~10000 の整数値

<*3>：[InfoA, InfoB, InfoC, InfoD]

データセットは[7]から取得可能である。

5.2 正常ログと異常ログの定義

本実験では、3 つのパラメータの組み合わせに基づく異常検知を目的としている。正常な組み合わせを表 1 に示し、表 1 に記載されていない組み合わせを異常と定義する

また、課題をより困難にするため、以下の条件を追加した：

- Type2 の Value 範囲を他の 2 タイプと重複させる。
- Info パラメータに関しても、同じ値を持つ複数の組み合わせを用意する。

表 1 正常なパラメータの組み合わせ

Type	State	Value	Info
1	A	1~150	InfoA or InfoB
2	B	80~2501	InfoD
3	C	500~1501	InfoA or InfoC or InfoD

5.3 テストデータセットの作り方

本実験では、各パラメータの影響を調査するために、State、Value、Info の 3 つのパラメータのうち 1 つを正常な組み合わせからずらした 3 種類のテストデータセットを用意して比較検討を行う。

5.4 比較する手法

我々が提案する BertMaskedLM 手法は、Tokenizer 部分とモデル部分に大きく分けられる。本実験で使用した手法はすべて、BertWordPieceTokenizer と BertForMaskedLM モデルの組み合わせで構成されている。学習済み BERT モデルに対しては、BertTokenizer を使用したが、その構造は BertWordPieceTokenizer と同じである。学習済みモデルには、大規模データセットで事前学習された "bert-base-uncased"を採用した[8]。

本実験で比較する手法の概要を以下に示す。

手法 1 基準モデル（先行研究[1]）

Tokenizer とモデルを 1 から学習

手法 2 基準モデル（先行研究[1]）

Tokenizer とモデルを 1 から学習。Tokenizer に登録する辞書に学習用データ＋BGL[9]データを使用

手法 3 学習済みモデル：追加学習なし

手法 4 学習済みモデル： Tokenizer のみ追加学習

手法 5 学習済みモデル：モデルのみ対象ログで追加学習

手法 6 学習済みモデル： Tokenizer およびモデルの両方を対象ログで追加学習

5.5 評価指標

本実験では、各手法の精度を評価するために以下の指標を使用する。

- F1-Score：適合率と再現率のバランスを評価する指標である。
- 誤警報率(FAR)：全ての正常なサンプルに対して、誤って異常と識別された正しくない陽性の割合を表す。
- 過少報告率(URR)：実際に異常であるサンプルのうち、検出されなかった割合を表す。

指標間の関係

FAR が低ければ、FP が減少し、Precision が高くなるため、F1-Score の向上に貢献する。

URR が低ければ、FN が減少し、Recall が高くなるため、F1-Score の向上に貢献する。

表 2　State パラメータの実験結果

Type	F1	FAR	URR	Epoch
手法 1	0.9331	0	0.1254	50
手法 2	0.9332	0	0.1252	1
手法 3	0.6059	0.4904	0.3496	無
手法 4	0.9332	0	0.1252	20
手法 5	0.3542	0.4492	0.6869	無
手法 6	0.9333	0.0002	0.1248	3

表 3　Value パラメータの実験結果

Type	F1	FAR	URR	Epoch
手法 1	0.8908	0.0379	0.1667	50
手法 2	1	0	0	10
手法 3	0.6608	0.9495	0.0409	無
手法 4	0.7518	0.1396	0.3141	10
手法 5	0.6607	0.9595	0.0361	無
手法 6	1	0	0	20

表 4　Info パラメータの実験結果

Type	F1	FAR	URR	Epoch
手法 1	1	0	0	1
手法 2	1	0	0	1
手法 3	0	0	1	無
手法 4	0	0	1	60
手法 5	0.6710	1	0	無
手法 6	0.6821	0.7348	0.1095	30

6　実験結果

表 2~4 にパラメータ毎の実験結果を示す。各実験において最も精度の高い Epoch の結果を示している。

6.1　State パラメータ

表 2 の結果から、対象のログを学習しない BERT モデルでは精度が低いことが分かる。一方で、手法 4 および手法 6 のように Tokenizer に対象ログデータを登録することで、精度が大幅に改善したことが確認できる。

6.2　Value パラメータ

表 3 の結果から、対象のログを学習しない BERT モデルでは精度が低いことが確認できる。一方で、手法 4 および手法 6 のように Tokenizer に対象ログデータを登録することで、精度が大幅に改善した。また、Value パラメータでは、BertMaskedLM モデルを対象ログで追加学習することで、精度がさらに向上した。

6.3　Info パラメータ

表 4 の結果から、事前学習済み BERT モデルを使用した場合、精度が極端に低いことが確認できる。また、他のパラメータとは異なり、Tokenizer やモデルを追加学習しても精度は向上しなかった。
一方で、手法 1 や手法 2 を含む先行研究[1]では、BertMaskedLM を 1 から学習することで、Info パラメータに対して非常に高い精度（F1-Score で 1.0）を達成している。これについては、Transformer がシーケンスの終わりに向かってより多くの情報を蓄積する特徴によると考察されている。
事前学習済み BERT モデルの精度が低下した要因として、以下の点が考えられる：

- 事前学習した文章と対象ログの長さの違い
- 登録した Token が多く、予測候補が大きすぎる

これらの問題を解決し、精度を改善するためには、新しい学習方法の検討や異なる異常度計算方法の導入が必要である。

結論

本研究では、代表的な学習済み BERT モデルと提案手法の比較検討を行った。先行研究では、学習データに対象ログとは無関係なログを追加することで精度が向上することが報告されていた[6]。これを踏まえ、大量の文章で事前学習された BERT モデルと提案手法の性能を比較した。その結果、対象ログを学習前の BERT モデルは低い精度を示した。一方、提案手法と同じ Tokenizer およびモデルの学習を行うことで大きな精度の向上が見られた。しかし、Info パラメータにおいては、精度の改善が見られなかった。これらの結果から、以下のことが分かる。
1. Tokenizer 及びモデルに対象ログを学習させることは効果的である（学習には工夫が必要[1, 6]）
2. 大量の学習データモデルの方が必ずしも良い精度を示さない（リソース問題のデメリットも持つ）

最後に本研究は、設計した特定のログ構造に基づいて異常検知を行ったものであり、実際の開発現場で使用されるログを用いたさらなる研究が重要である。今後、本研究の成果がセンサデータやサーバアクセスログ等のセキュリティ領域やアクチュエータ等の機械領域を含むさまざまな分野で応用されることを期待する。

謝辞

本研究は JST SPRING JPMJSP2154 の助成を受けたものです.

本研究は Panasonic System Design の助成を受けたものです.

参考文献

1. H. Uchida, K. Tominaga, Hideki. Itai, Y. Li and Y. Nakatoh, “Multi-Parameter Log Anomaly Detection with an Unsupervised Learning Approach”, 2024 International Symposium on Parallel Computing and Distributed Systems (PCDS), Sep. 2024, DOI: 10.1109/PCDS61776.2024.10743635

2. V.-H. Le and H. Zhang, “Log-based Anomaly Detection Without Log Parsing,” in Proceedings of the 2021 36th IEEE/ACM International Conference on Automated Software Engineering (ASE), 2021, pp. 492–504, DOI: 10.1109/ASE51524.2021.9678773.

3. S. Huang, Y. Liu, C. Fung, R. He, Y. Zhao, H. Yang, and Z. Luan, "HitAnomaly: Hierarchical Transformers for Anomaly Detection in System Log," IEEE Transactions on Network and Service Management, vol. 17, no. 4, pp. 2064–2075, December 2020. DOI: 10.1109/TNSM.2020.3034647

4. Lee, Y., Kim, J., and Kang, P., “LAnoBERT: System log anomaly detection based on BERT masked language model,” Applied Soft Computing, vol. 146, Oct. 2023, 110689. doi: 10.1016/j.asoc.2023.110689.

5. H. Uchida, K. Tominaga, Hideki. Itai, Y. Li, Y. Nakatoh, “Improvement of Multi-Parameter Anomaly Detection Method: Addition of a Relational Token Between Parameters,” Cognitive Robotics, in review

6. H. Uchida, K. Tominaga, Hideki. Itai, Y. Li and Y. Nakatoh, “Log Parameter Anomaly Detection System: Evaluating Accuracy with Noisy Training Data”, IEEE International Conference on Consumer Electronics (ICCE), 2025, accepted

7. https://github.com/hiro877/Datasets-for-Anomaly-Detection-Used-in-Research

8. J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” Oct. 2018, [Online]. Available: http://arxiv.org/abs/1810.04805

9. S. He, J. Zhu, P. He, R. M, and M. Lyu, “Loghub: A Large Collection of System Log Datasets towards Automated Log Analytics,” Arxiv Website, 2020. [Online]. Available: https://arxiv.org/abs/2008.06448. [Accessed: 28 May 2024].