

Identifying Bad Software Changes via Multimodal Anomaly Detection for Online Service Systems

Nengwen Zhao, Junjie Chen, Zhaoyang Yu

Honglin Wang , Jiesong Li , Bin Qiu

Hongyu Xu , Wenchu Zhang , Kaixin Sui , Dan Pei

発表者：西山知弥

背景

- 大規模なオンラインサービスシステムでは,ソフトウェアの変更が頻繁に行われる
- 変更はインシデントの発生やユーザーエクスペリエンスの低下を招く可能性がある
- インシデントの影響を減らし,システムを向上させるためには,悪質なソフトウェア変更を特定することが不可欠

目的

- 悪質なソフトウェア変更をよりよく理解する

アプローチ

- 大手商業銀行の大規模な実世界データに基づく初めての
の実証的研究
- 異種のマルチソースデータを融合し, 悪質なソフトウェアの変更を識別, 早い段階で警告するSCWarnの提案

ソフトウェアの変更に関するRQ

RQ1 : 悪質な変更によって引き起こされるインシデントの割合はどのくらいか？

RQ2 : 悪質な変更の原因と行動は何か？

RQ3 : 悪質な変更を特定する現在の方法は十分か？

RQ1～RQ3 調査方法

- 異なる製品チームによって開発・保守されている5つのサービスシステムを対象
- 各サービスについて、過去2年間のすべてのインシデントチケットとチェンジチケットを収集し分析
- チケットの分析とラベリングについては「Cohen's Kappa係数」[67]の監修の下、チケットに記録されたトラブルシューティングの手順とインシデントの理由の記述を分析

RQ1:悪質な変更によって引き起こされるインシデントの割合はどのくらいか？

- ソフトウェアの変更によって引き起こされたインシデントの数をカウント

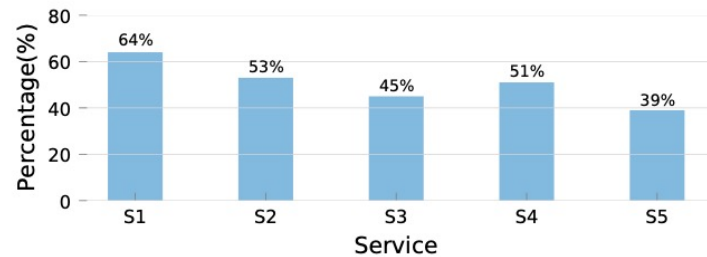


Figure 2: Percentage of incidents induced by changes

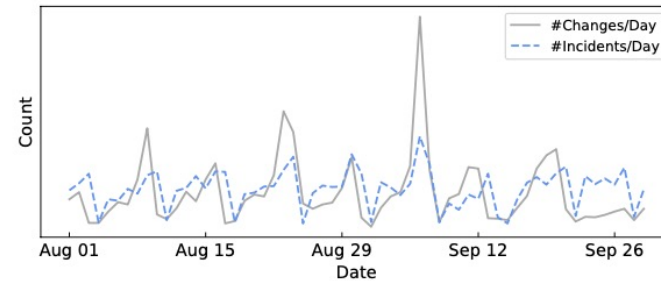


Figure 3: #Incidents per day and #Changes per day

- 変更はインシデントを引き起こす重要な要因
- 39%～64%(平均で50.4%)を占めていた

RQ2 : 悪質な変更の原因と行動は何か？

コードの欠陥, 設定ミス, 互換性のないバージョン, リソースの競合, その他の要因の5種類の原因をまとめた

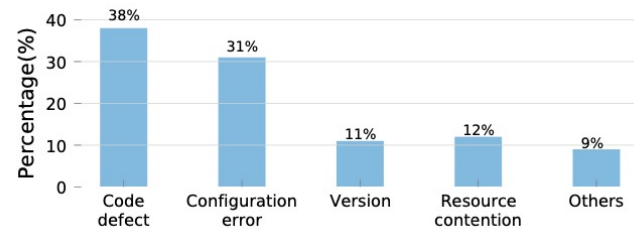


Figure 4: Percentage of different root causes

- コードの欠陥と設定ミスが大きな割合, 69%を占めていた

RQ3 : 悪質な変更を特定する現在の方法は十分か？

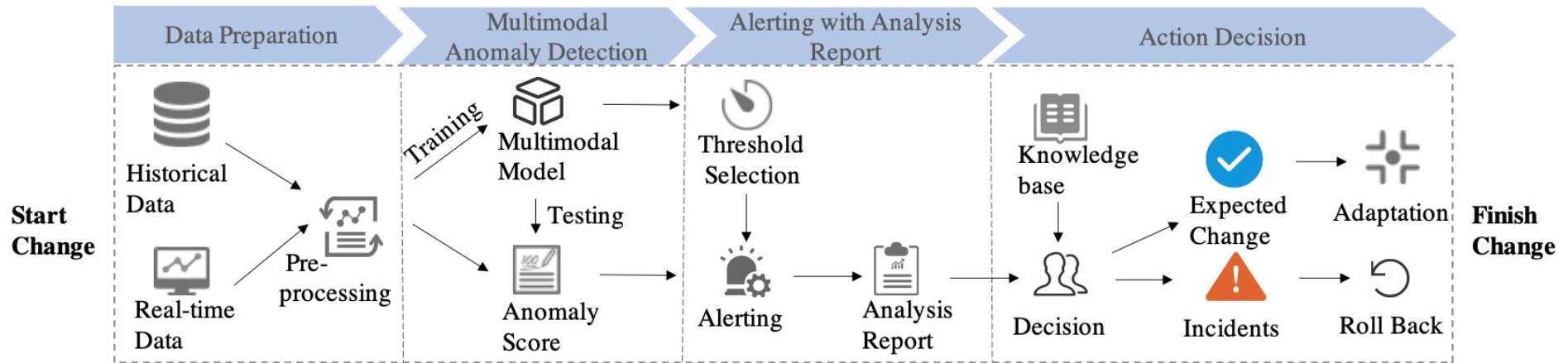
調査した銀行では, 悪質な変更を特定するために, 3- σ [1]戦略を採用して, ビジネスKPI(例: 応答時間)の動作を検査していた

- インシデントチケットの分析やエンジニアとの議論を通じて, 現在の方法は満足できるものではないことがわかった

原因

- ソフトウェアの変更には, 複数のソースデータが含まれるため, ビジネス KPI にのみ注目していると, 他のデータに隠れている異常を特定できない
- 異常検知アルゴリズム自体の欠点

SCWarnの概要



- ログ解析に基づいて異種データを統一的な時系列データに変換
- マルチモーダル学習を用いて, マルチソースのモニタリングデータ(ログやKPI)から異常を検出
- マルチモーダルLSTMが提供する異常スコアが警告ルールに違反した場合, エンジニアが疑わしい変更注意を払うように通知するための警告信号が発せられる

マルチモーダル異常検出

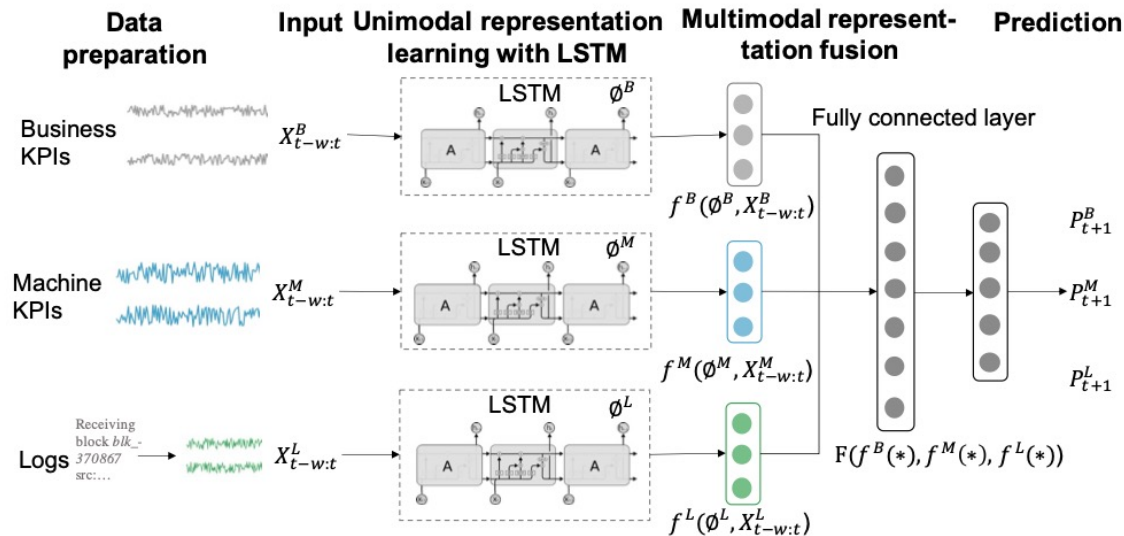


Figure 7: Multimodal anomaly detection model

- SCWarnでは, 各時系列の時間的依存性を捉えるためにLSTMを適用
- マルチモーダルフュージョン[59]を用いてマルチソースデータ間の相互相関を符号化

SCWarnの評価のためのRQ

RQ4 : 悪質なソフトウェアの変更を識別する際のSCWarnの有効性は何か？

RQ5 : マルチモーダルLSTMの有効性は何か？

RQ6 : SCWarnの時間効率はどうか？

RQ7 : SCWarnの結果に与える変数の影響は何か？

RQ4：悪質なソフトウェアの変更を識別する際のSCWarnの有効性は何か？

いくつかの最先端の異常検知手法と比較

Table 3: Precision (P), recall (R), F1-score (F1) and MTTD (minutes) comparison between SCWarn and baselines

Approaches	Dataset $\mathcal{A}$				Dataset $\mathcal{B}$			
	P	R	F1	MTTD	P	R	F1	MTTD
SCWarn	0.91	0.95	0.93	5.1	0.97	0.98	0.97	2.3
Gandalf-AD	0.68	0.95	0.79	6.2	0.77	0.99	0.87	3.1
Funnel	0.77	0.69	0.73	14.0	0.76	0.87	0.81	6.4
Lumos	0.66	0.94	0.78	10.0	0.77	0.93	0.82	10.0
mFunnel	0.69	0.93	0.79	9.0	0.82	0.79	0.80	3.0
mLumos	0.85	0.83	0.84	10.0	0.72	0.99	0.83	10.0

- F1スコア0.93および0.97を達成し, すべての測定値を考慮しても最高の性能を示した
- MTTDは平均3.7分で, 20.4%~63.7%減少しており, 悪質な変更を早期に発見できることがわかりました

RQ5 : マルチモーダルLSTMの有効性は何か？

いくつかの最先端の異常検知手法と比較

Table 4: Effectiveness of multimodal LSTM algorithm compared with existing anomaly detection approaches

		Dataset $\mathcal{A}$				Dataset $\mathcal{B}$			
Data source	Approaches	P	R	F1	MTTD	P	R	F1	MTTD
Business KPIs	Donut	0.65	0.92	0.76	7.3	0.94	0.75	0.83	5.4
	B-LSTM	0.86	0.75	0.80	8.2	0.99	0.88	0.93	6.6
Machine KPIs	LSTM-NDT	0.80	0.71	0.76	5.2	0.85	0.83	0.86	3.5
	OmniAnomaly	0.71	0.99	0.83	5.4	0.88	0.87	0.87	3.2
Logs	DeepLog	0.57	0.96	0.71	11.2	0.55	0.83	0.66	8.9
Multi-source data	M-AE	0.79	0.85	0.81	5.1	0.82	0.90	0.86	2.6
	M-LSTM	0.80	0.95	0.87	5.3	0.99	0.94	0.96	3.2
	Multimodal AE	0.94	0.83	0.88	6.1	0.95	0.93	0.94	4.0
	Multimodal LSTM	0.91	0.95	0.93	5.1	0.97	0.98	0.97	2.3

- 単一のデータソースにのみ依存するアプローチよりも優れている
- また,マルチソースデータを含む他の手法(M-AE,M-LSTM, マルチモーダルAE)よりも優れていた

RQ6：SCWarnの時間効率はどうか？

Table 5: Training time (minutes) and detection time (seconds) comparison

Approaches	Dataset $\mathcal{A}$		Dataset $\mathcal{B}$	
	Training	Detection	Training	Detection
SCWarn	4.84	0.94	3.43	0.86
Gandalf-AD	6.28	0.78	6.03	0.74
Funnel	-	1.44	-	1.04
Lumos	-	1.23	-	1.08
Donut	11.57	1.01	8.32	0.84
B-LSTM	3.45	0.55	2.89	0.49
LSTM-NDT	3.89	0.61	3.01	0.52
OmniAnomaly	28.11	1.34	24.89	1.88
DeepLog	50.65	2.43	46.23	2.77
M-AE	3.11	0.40	2.80	0.26
M-LSTM	4.42	0.49	4.19	0.44
Multimodal AE	3.65	0.67	3.20	0.60

- SCWarnの学習時間は数分程度で許容範囲内
- SCWarnと比較したほとんどのアプローチは,ほぼリアルタイム(1秒未満)で結果を出すことができる
- 全体として, SCWarnはオフラインでのトレーニング時間は許容範囲内で, オンラインでの検出時間は無視できるレベル

RQ7 : SCWarnの結果に与える変数の影響は何か？

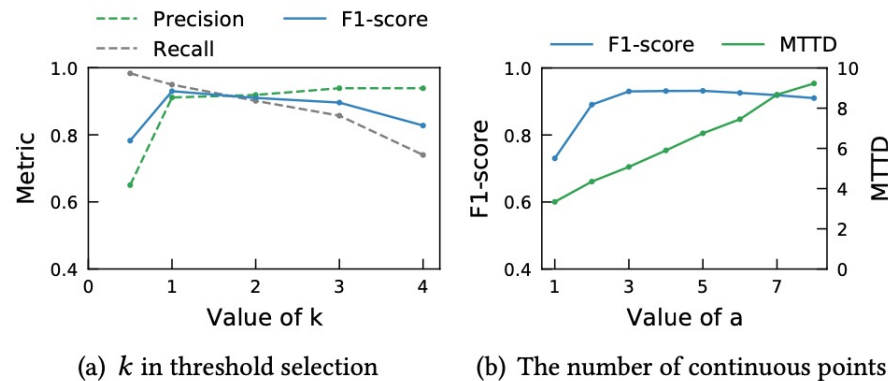


Figure 9: The effect of parameters

k . . . 異常スコアの閾値

小さい場合, 閾値は比較的緩く, Recallは高いがPrecisionは低いという結果に

a . . . 警告戦略における連続異常点の数

a が小さすぎると, 誤警報が発生. 大きくすると, 十分なPrecisionが得られるが, 警告戦略が厳しすぎると, MTTDが長くなる可能性がある

今後の課題

- 検出が困難なものがあること

あまり使用されない関数に隠されたコード欠陥は, 長い間検出されない可能性がある

- 銀行のサービスシステムにSCWarnを導入しただけで, 実世界での大規模な導入は未だ行なっていない

まとめ

- インシデントの約50.4%は変更が原因だった
- 悪質な変更を特定する現在の手法は満足できるものではないことがわかった
- 悪質な変更を正確, タイムリーに特定するためのSCWarnという新しいアプローチを提案し, 有効性が確認された

所感

- 出てくる用語が自分には難しかった
- インシデントの多くの原因が, 悪質な変更ということがデータでわかった
- 再利用を検出する際に参考になるかは, わからなかった