情報処理学会論文誌 Vol.59 No.3 1006–1015 (Mar. 2018)

イベントネットワークにおける syslogを用いた異常検知手法の提案と実データを用いた評価

阿部 博[1,2,a)] 敷田 幹文[3,b)] 篠田 陽一[2,c)]

受付日 2017年6月26日, 採録日 2017年12月8日

概要：大規模なイベントネットワークではネットワーク管理手法の 1 つとして syslog を用いた運用監視が行われる．syslog メッセージに含まれるキーワード検知や閾値による異常検知などネットワークの異常が運用者に通知される．マルチベンダ機器によって構築される特殊なイベントネットワークでは，ログの意味解析やキーワードによる異常検知が行えない環境下であることが多い．本論文ではイベントネットワークで収集される syslog の総量による分析を行い異常を検知する手法を提案する．株式取引で用いられるボリンジャーバンドアルゴリズムを利用し，Interop Tokyo 2016 で構築された ShowNet で収集された syslog の実データを用いて統計学的手法において軽量な計算による異常検出を行い，ボリンジャーバンドアルゴリズムの有効性を評価する．

キーワード：イベントネットワーク，アノマリ検知，syslog，ボリンジャーバンド，ネットワーク監視

The Anomaly Detection Method Analyzing Syslog Data Using Bollinger Bands Algorithm on Event Network

Hiroshi Abe[1,2,a)] Mikifumi Shikida[3,b)] Yoichi Shinoda[2,c)]

Received: June 26, 2017, Accepted: December 8, 2017

Abstract: Network administrator manages and monitors the network using syslog analysis as one of the network management methods on a large event network. The network troubles such as keyword detection included in the syslog message and abnormality detection by the threshold are notified to the network administrator. In a special event network integrated by multivendor network equipment, it is sometimes impossible to analyze the semantics of logs or detect anomalies by keywords. In this paper, we propose a method to detect anomalies by analyzing the total amount of syslog data which collected in the event network using the Bollinger Bands algorithm used in stock trading. We performed anomaly detection by lightweight calculation in the statistical method using real data of syslogs that collected by ShowNet constructed by Interop Tokyo 2016. And we evaluated the effectiveness of the Bollinger Bands algorithm.

Keywords: event network, anomaly detection, syslog, Bollinger Bands, network monitoring

1. はじめに

1.1 背景と目的

大規模な展示会やカンファレンス，シンポジウムといったイベントでは，来場者や参加者に対してインターネットへのアクセスがサービスとして提供されることがある．これらのネットワークを総称してイベントネットワークと呼ぶ．イベントネットワークはマルチベンダのネットワーク機器を用いてシステムが構築されることが多い．1 週間か

[1] 株式会社 IIJ イノベーションインスティテュート
IIJ Innovation Institute Inc., Chiyoda, Tokyo 102–0071, Japan
[2] 北陸先端科学技術大学院大学
Japan Advanced Institute of Science and Technology, Nomi, Ishikawa 923–1292, Japan
[3] 高知工科大学
Kochi University of Tecnology, Kami, Kochi 782–8502 Japan
a) abe@iij.ad.jp/h-abe@jaist.ac.jp
b) shikida.mikifumi@kochi-tech.ac.jp
c) shinoda@jaist.ac.jp

ら 2 週間という短期の準備期間内でネットワークの構築から運用/撤収を行うため，構築されるネットワークが安定稼働するまでに様々なトラブルが発生する．またマルチベンダの機材を用いることにより，機材の互換性に関するトラブルの把握や解決など運用者の経験に基づく問題解決に依存することが多く，トラブル対応の自動化が困難である．

ネットワークの運用では，運用者がネットワーク/サーバ機器/ソフトウェアの動作状況を把握するために syslog を解析する手法が用いられる．運用者にとって，マルチベンダ機器が出力する syslog に対する理解不足や未知のログに対する対応または膨大なログ量の解析に関して，出力されるログのどのキーワードがエラーやアラートであるのかを判断することは難しい．運用者が事前に把握している特定のキーワードに基づくエラーハンドリングは可能ではあるが，ログの急増や未知のログに対処することは困難であり，そもそも膨大なログに埋もれて本来発見したい異常を発見できない場合もある．

通常のネットワーク運用とイベントネットワーク運用の違いは，大規模なイベントネットワークでは多くのマルチベンダ機器や世界で初めて投入される機材やソフトウェアが利用されることがあり，通常の運用では知りえないバクやエラー，未知のログメッセージに遭遇する可能性があり，運用するエンジニアがログの意味自体をその場で理解することが難しい．

本論文では，多数のマルチベンダ機器が混在する大規模なイベントネットワークにおいて，運用者にとって未知のログが多数出現する過酷な環境下であってもログの総量から異常を読み取り，運用者へと効率的に通知可能なアルゴリズム適用の提案を行う．提案手法では株式市場で用いられるテクニカル分析手法を採用し，正規分布に基づく確率理論からログ総量の突発的な上昇や下降を検知する．本手法を用いることでアラートの通知頻度を減少させ，運用者への現実的な異常発生回数の通知が可能となり，トラブルへの初動対応の高速化が行える．

本提案ではマルチベンダ機器が投入される大規模なイベントネットワークの一例として，Interop Tokyo [1] で構築される ShowNet [2] で収集された syslog を用いて実データをもとに異常状態の分析を行う．

1.2 ShowNet

ShowNet は，毎年千葉幕張メッセで開催される Interop Tokyo 内で構築される最新のネットワーク機器や技術を集めた相互接続検証とデモンストレーションを行う実験ネットワーク環境である．また，出展社へのインターネットアクセスをサービスとして提供する側面もあり，実験とサービス提供の 2 面性を有したイベントネットワークである．

ShowNet を構成する機材は，世界で初めて展開されるような最新の製品が多く，機材数は数百を超える．ShowNet には実験ネットワークという一面もあり，試作レベルの機材やソフトウェアが展開される．ShowNet を運用するメンバは，これらの機材やソフトウェアを組み合わせサービスを提供するネットワークを運用構築する．

ShowNet では，毎年異なる機材を用いてシステムやネットワークを構築するため，構築の自動化を行うことが難しい．安定したネットワークを構築するために運用メンバの専門性に特化した高度なスキルが要求される．さらに数年先を見越した最先端の技術導入チャレンジを行うため，運用ノウハウが存在しない技術を利用し，発生したトラブルに運用メンバの経験と勘で対応することが多々発生する．そのため，展示会直前までネットワークが不安定になることがある．また出展社や関係者の不注意により，ネットワークにループが発生するなど，運用メンバが意図しない異常が発生することも多い．

1.3 ShowNet における syslog 監視

ShowNet では，監視システムの 1 つとして syslog を収集し分析するシステムが運用される．数年前までは syslog の可視化は行われずに ShowNet 運用が行われていたため，ネットワーク異常が発生した際には，運用メンバが ping，traceroute，tcpdump などのツールを用いてトラブルの切り分けを行っていた．ログの可視化を行うことで特定時間内に発生したログからキーワードを抽出し，閾値を超えた場合に運用者にアラートを通知することができる．例として図 1 に VMware vRealize LogInsight [3]（以下，LogInsight）を使用した syslog 監視をあげる．LogInsight の機能として，

- ダッシュボードによる条件抽出したログの可視化
- 特定キーワード（OSPF down/BGP down/Storm de-

図 1 VMware vRealize LogInsight を使った syslog 監視例
Fig. 1 Example of the syslog monitoring tool.

tection など）の出現監視

- 特定キーワードのマッチング回数監視（閾値ベース）

などが提供される．キーワードマッチングをトリガとしてトラブルの原因究明を行うわけだが，マルチベンダの機材が多いために運用者は未知のログを扱うことが多く，どのようなキーワードがトラブルの原因になるかを瞬時に判別することは難しい．

ログの意味解析を行い，スコアリングに基づく異常値解析を行うことは可能ではあるが，キーワードの出現頻度だけでは，そのログが正常か異常かの判断を行うことは運用者にとって難しい．機械学習を用いて，教師データを精査し精度を上げ異常検知を行うことも可能ではあるが，イベントネットワークの特徴として開催期間が短すぎネットワーク安定状態における学習データを集めることが難しい．また ShowNet の特性として実験ネットワークの意味合いも強く，debug メッセージや必要のない info メッセージが大量にログ出力され，機械学習を行ううえではノイズとなるデータが多すぎる．

そこで本論文では，ShowNet で集められるマルチベンダ機器から出力される膨大な syslog データの総量を移動平均と標準偏差を用い集計比較することで，計算量が少なく軽量に計算可能なアルゴリズムを利用し，運用者へ異常検知のトリガとなる事象を通知する手法を提案する．

1.4 本論文の構成

2 章では関連研究に関する調査と問題点を提示し，3 章では提案手法を示す．4 章では評価の前提と手法を開示し，5 章では結果について述べる．6 章では得られた結果から考察を行い，7 章でまとめと今後の課題を述べる．

2. 関連研究

時系列データに周期的な規則性がある場合には，データの波形を予測し外れた場合に異常値とする Holt-Winters 法 [4] のような予測アルゴリズムが利用できるが，ShowNet は開催期間が短いためデータの周期性を観測することは困難であり，周期性に頼るアルゴリズムは適さない．

時系列データにおいてイベントが急増したことを検出する手法の 1 つとして，Jon Kleinberg のバースト検知アルゴリズム [5] がある．Kleinberg のバースト検知アルゴリズムでは，解析対象のテキストに含まれるキーワードに対し，確率モデルで定義されたコスト計算を行う．このアルゴリズムはバースト状態よりも定常状態に遷移する特徴があり，一時的なバーストに反応しにくくなるという特性がある．syslog のような時系列のログを解析し異常を検知するには有効な手法であるが，ログの意味解析を行う必要があり，ログの総量が多い場合には計算量が必然的に増加する．

また，バーストの変化点に着目するアルゴリズムとして ChangeFinder [6] の手法がある．ChangeFinder は統計的な処理を行い外れ値ではなく変化点を見つけるアルゴリズムで，ログ総量のような時系列データの値の急増のように定常状態を設定できないデータに対して有効に働く．しかしながら局所的な値の変動に関しては，スコアが平滑化され，時系列に連続する大きな変動ほど異常状態を見つけられない．一瞬の突発的なログ増加に関してはスコアが低くなり異常ではないと判断される可能がある．

マハラノビス距離 [7] は超楕円体で近似したクラスタの重心と特徴点の距離をクラスタの幅で正規化したものである．特徴点がクラスタに帰属する度合いをその値と大きさだけで判断することができ，値が大きいほどクラスタの中心から離れていることを示す．マハラノビス距離による異常検出は，平均値ベクトルと分数といった基本概念を利用し異常値の概念を数式化する．しかし，平均値は異常値の影響を大きく受けるため，中央値やエントロピーを計算して異常検出を行う方法が提案されている．

ARIMA（自己回帰移動平均）モデル [8] は，時系列データに適用されるモデルであり，過去の時系列データから規則性を見つけ出しその規則に基づいて将来の値を求める．ARIMA モデルで予測に用いる変数は予測対象の実績値のみでありデータの取得コストが低い．欠点として移動平均の係数の推移のようにパラメータの調整に時間がかかる点があげられる．

また，本提案の先行研究 [12] として $+2\sigma$ を基準としたボリンジャーバンドの判定と，基準値を超えた幅を固定閾値と比較してレベル分けを行った．本研究では，先行研究の結果をふまえさらに検知の正確さの改善を行う．

3. 提案手法

3.1 提案概要

本研究では，計算が軽量なアルゴリズムとして正規分布に基づいたアルゴリズムを採用する．短期間で構築されるために安定状態を定義しにくいイベントネットワークの特徴を再現するため，ShowNet で収集された syslog を対象とすることで，大規模なイベントネットワークの異常を検知する．

ShowNet では以下のようなログの状態が発生した場合に異常状態であると想定している．

- 機材への攻撃によるログの急増
- 機材の不具合によるログの急増
- 機材の不調によるログの急増
- 機材の設定ミスによるログの急増
- 大量な正常アクセスによるログの急増
- ウィルスやワームが発生することによるログの急増

本提案において syslog 総量急増の検出は移動平均と標準偏差を用いる．具体的なアルゴリズムとしてボリンジャーバンド [9] を用い，ログ総量から異常を検知する．また移動平均アルゴリズムの評価として，単純移動平均と指数移動平均を用い両者の比較を行う．

3.2 ボリンジャーバンド

ボリンジャーバンドは株式の取引で利用されるテクニカル分析手法の 1 つで 1980 年前半に John Bollinger により公表された．移動平均を示す線と，その上下に値動きの幅を示す線を加えた指標のことをいい，価格の大半がこのバンドの中に収まるという統計学を応用したテクニカル分析の 1 つである．

ボリンジャーバンドは正規分布に基づく理論である．統計学の正規分布理論では，**図 2** で示すように，

- 平均値 $\pm\sigma$（標準偏差）に収まる確率は 68.26%
- 平均値 $\pm 2\sigma$（標準偏差）に収まる確率は 95.44%
- 平均値 $\pm 3\sigma$（標準偏差）に収まる確率は 99.73%．

となる．

ボリンジャーバンドの考え方では，株価は 99.73%の確率で $\pm 3\sigma$ のバンド幅に収まると仮定され，株価が $+2\sigma$ のラインを超えた場合や $+3\sigma$ のラインにタッチした場合には株が買われすぎであると判断し，逆に -2σ のラインを株価が下げた場合や -3σ のラインにタッチした場合には株が売られすぎであると判断され，適切な価格へ戻ることが予想できる．

つまり，

- 移動平均 $+3\sigma$（標準偏差）を UpperLimit（上限値）
- 移動平均 -3σ（標準偏差）を LowerLimit（下限値）

として上限値と下限値を採用し，それらの値と株価の比較により適正価格かどうかの判定を行う．

移動平均（単純移動平均）は以下の式として表すことができる．

$$\overline{x} = \frac{1}{n}\sum_{i=0}^{n-1} x_i$$

新しい値を移動平均の計算に加えたい場合には，現在の移動平均の値に対し，新しい値を加え古い値を除くことで求めることができ，総和を求め直す必要はないので軽量な計算で済む．

また標準偏差（σ）は以下の式で求めるられる．

$$\begin{aligned}\sigma &= \sqrt{\frac{1}{n}\sum_{i=0}^{n-1}(x_i-\overline{x})^2} \\ &= \sqrt{\frac{1}{n^2}\left\{n\sum_{i=0}^{n-1}x_i^2-\left(\sum_{i=0}^{n-1}x_i\right)^2\right\}}\end{aligned}$$

標準偏差（σ）に ± 3 をかけた値を，UpperLimit，LowerLimit として用いる．

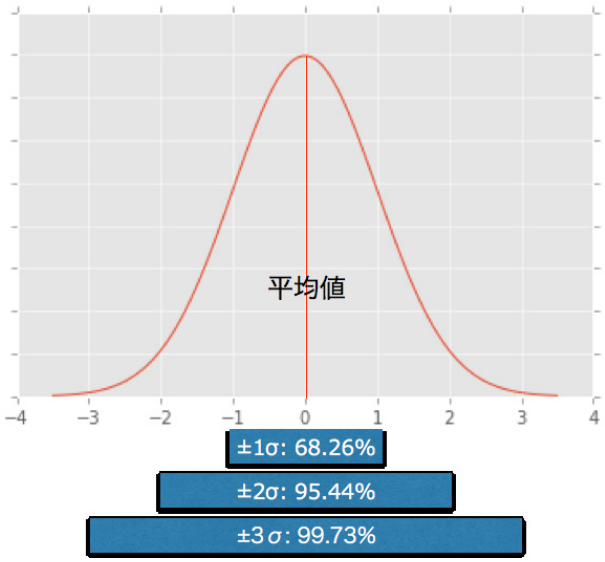


図 2 正規分布と σ の確率
Fig. 2 Normal distribution ratio.

3.3 syslog 総量計算への応用

本提案では syslog の総量の推移に関して株式取引と同様に統計的な推移が存在すると仮定し，移動平均 $\pm 3\sigma$（標準偏差）を超えた場合，もしくは下回った場合には異常状態であると判定する．syslog の総量はシステムが安定していれば株式と同等に一定範囲（$\pm 3\sigma$ の間）で推移すると仮定すると，99.73%の確率で総量は UpperLimit と LowerLimit の間であるバンド内を推移することを意味し，バンドの上限を超える/下限を下回る確率である 0.27%を異常値として検出する．

3.4 アラート発生率と誤検知

ネットワーク運用者は発生したアラートが適切なアラートであるか判断を行う必要がある．アラートが少なければ少ないほど個々の事象を調査する時間は増え，調査の精度を上げることができる．アラートが頻発した場合には，調査を行う人員の数に依存するが調査に費やされる総時間は増加する．ゆえに，実時間で調査を完了させるために本当に通知したいアラートのみ通知するというオペレーションが運用上望ましい．

しかしながら，過度なアラート抑制は誤検知を招く恐れがある．本来通知されなければならないアラートが抑制されるフォールスネガティブが発生した場合に，ネットワーク運用者はアラートに気がつくことができない．逆に，本来通知される必要がないフォールスポジティブなアラートが通知された場合には，必要のない運用オペレーションが発生することになる．ボリンジャーバンドのアルゴリズムでは基準を超えたかどうかのみを判定するため，基準を超えた場合にはボリンジャーバンド的につねに正しいアラート通知となる．ボリンジャーバンドのアルゴリズムのみでは，通知されたアラートが誤検知か否かを判定することは難しい．逆にアラートの抑制を行う場合には，本来通知すべきアラートが通知されないためフォールスネガティブは発生する．

本研究で目標とする 0.27%が異常値として適切な値かどうかは，監視運用者の判断に任せられることになるが，ア

ラートの発生回数が初期の気づきとして運用対応を行う現実的な回数内に収まれば運用に適用可能と仮定する．

4. 評価

4.1 実験概要

実験環境を**表 1** に示す．本実験では，ShowNet に接続された機材が出力した管理ログを収集して，ShowNet 終了後に実験環境において Python を用いて解析を行った．解析対象の syslog の容量は約 6.4 GB で，行数にして約 4,350 万件を対象とした．なお可読できないバイナリ形式で出力されたログに関してはノイズとして扱い除外してから解析を行った．

4.2 手法

本提案では，ログの意味分析を行わずに時間あたりのログ行数の出現回数を集計分析する手法を用いた．ログの総数を分析するにあたり，syslog から分析には必要ではない情報の削除を行った．syslog フォーマット [10] は大きく，ヘッダ部とメッセージ部からなる．ヘッダ部は，

- タイムスタンプ
- デバイス名

を必ず含み，タイムスタンプはローカル時刻でフォーマットは “Mmm dd hh:mm:ss” となる．また，デバイス名はホスト名または IP アドレスとして定義される．メッセージ部は，フリーフォーマットでテキストメッセージが出力される．

ログの総数から分析を行うために本提案では，タイムスタンプとデバイス名の 2 カラムを用いて分析を行った．計算速度を速めるため，メッセージ部の情報を削除したファイルをフィルタプログラムにより csv データとして作成した．事前処理を行った csv ファイルを，Python のデータ解析ライブラリである pandas [11] を用いてログの総量を時系列に解析した．読み込んだ csv データは pandas で定義される DataFrame 形式で処理される．DataFrame は，pandas で定義されるデータ構造の 1 つで，二次元のテーブルとしてデータを定義できる．各行，各列にはラベルをつけることができ，1 行を 1 つのデータとして表計算ソフトウェアのように処理できる．DataFrame 形式で読み込んだデータに対して，基本的な算術計算をメソッドとして呼び出すことができる．本提案では，移動平均と標準偏差を扱うが，これらに関しても，DataFrame に対するメソッド呼び出し（移動平均：mean メソッド，標準偏差：std メソッド）を行うことで実装する．

表 1 実験環境
Table 1 Experiment environment.

OS	CentOS 7.2
開発言語	Python 3.5.1
CPU	Intel(R) Xeon(R) CPU E5-2670 v3 @ 2.30 GHz
メモリ	128 GB

処理を行う前提は以下とする．

- 1 日ごとのログ総量を計算/描画対象とする．
- 1 分単位でデータをグルーピングする．
- 過去 1 時間分のログ総量を移動平均に利用する（0 時台の計算には前日の 23 時のデータを読み込む）．
- 1 分単位のグルーピングにより，移動平均/標準偏差のウィンドウサイズは 1 時間で 60 とする．

総量，移動平均，標準偏差を求めるサンプルコードは**図 3** となる．

各行の処理は以下を意味する．

1) pandas ライブラリの読み込み
2) ログファイル（csv ファイル）の読み込み
3) DataFrame のデータを 1 分単位でまとめて集計
4) mean メソッドで移動平均値を計算
5) std メソッドで標準偏差を計算
6) 標準偏差の 3 倍を計算（$+3\sigma$）
7) 標準偏差の -3 倍を計算（-3σ）
8) 移動平均と標準偏差の足しあわせ（UpperLimit）
9) 移動平均と標準偏差の足しあわせ（LowerLimit）

入力データの例外として，5/27 のみ初日ということで前日からの学習データは存在しない．

これらの計算データから画像を生成した例が**図 4** となる．青い棒グラフである count が 1 分ごとにまとめられたログの総量を表す．総量の上方の赤い折れ線グラフが UpperLimit

```
1  import pandas as pd
2  df = pd.read_csv('./2016-06-06-syslog.log
       ', delim_whitespace=True, ...)
3  count = df.groupby(pd.TimeGrouper('1Min'))
       .count()
4  mean  = count.rolling(window=60).mean()
5  std   = count.rolling(window=60).std()
6  std_plus  = std.apply(lambda x: x * 3)
7  std_minus = std.apply(lambda x: x * -3)
8  upper_limit = mean.add(std_plus)
9  lower_limit = mean.add(std_minus)
```

図 3 サンプルコード
Fig. 3 Sample program.

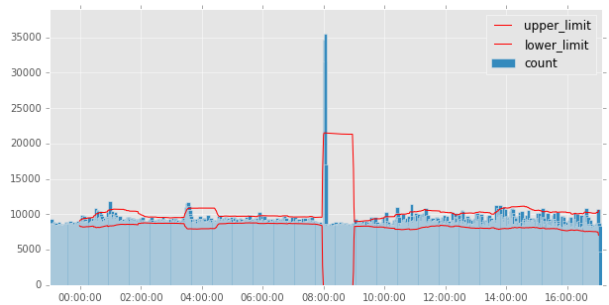


図 4 1 日のログ総量分析例
Fig. 4 Example of total log analysis per day.

表 2 アラートレベル
Table 2 Alert level.

Level	異常値の範囲
Low	3σ 以上 4σ 未満
Middle	4σ 以上 5σ 未満
High	5σ 以上

を意味し，下方の赤い折れ線グラフが LowerLimit を表す．ログの総量が UpperLimit と LowerLimit の範囲を推移する場合には，ログの総量は異常状態ではないと判断する．

本提案では異常値を見つける方法として，ボリンジャーバンドの上限のみにまずは着目した．つまり移動平均 $+3\sigma$ を指す UpperLimit が総量の合計を超えた回数を集計した．このときに UpperLimit を超える確率は，正規分布の $+3\sigma$ のみとなり約 0.14%（0.27%の半分）となる．UpperLimit にのみ着目する理由は，一般的に機器の異常やネットワークの異常時には syslog が大量に出力されるという運用者の経験則に基づく．また正常アクセスであっても，DoS（Denial of Service）のような攻撃を受けた場合にはアクセスログが大量に出力され，上限値超えに気がつくことで正常ではない状態であると判断できる．

さらに異常値判定の精度を考慮しアラート通知の頻発を抑えることを目的として，異常値が $+3\sigma$ を超えた値の範囲に着目し，アラートのレベル分けを行うこととした（**表 2**）．アラートレベルは動的に計算され，Low = 3σ 以上 4σ 未満，Middle = 4σ 以上 5σ 未満，High = 5σ 以上という形で出力される．先行研究 [12] では Low，Middle，High を固定値の幅として計算していたが，本提案では動的にアラートレベルを計算する仕組みを取り入れた．また，移動平均のアルゴリズムの種類により異常値の検出がどのように変化するかの実験を行うため，単純移動平均と指数移動平均の 2 種類の移動平均アルゴリズムを用い，異常検知数がどのように変化するか比較実験を行った．

5. 結果

5.1 アラート発生率

ShowNet の期間中に発生した syslog の総量とボリンジャーバンドの上限を超えたアラートの発生率を**表 3** に示す．

表 3 は，
(1) 日付
(2) ログ総数
(3) 時間スロット数
(4) アラート回数
(5) 発生率
の 5 項目からなる．ログ総数は，日付ごとに集計した syslog の総数を表す．時間スロット数は，ログの集計単位（1 分単位：60 秒）を 1 日（86,400 秒）で割った数

表 3 アラート発生率
Table 3 Total alert information.

日付	ログ総数	時間スロット数	アラート回数	発生率
5/27	192	617	12	1.94%
5/28	181,285	1,440	46	3.19%
5/29	552,579	1,440	50	3.47%
5/30	821,363	1,440	48	3.33%
5/31	617,368	1,440	27	1.88%
6/1	917,368	1,440	47	3.26%
6/2	1,949,738	1,440	38	2.64%
6/3	1,771,956	1,440	28	1.94%
6/4	2,108,661	1,440	38	2.64%
6/5	3,177,122	1,440	24	1.67%
6/6	3,297,654	1,440	24	1.67%
6/7	2,702,382	1,440	24	1.67%
6/8	3,186,363	1,440	24	1.67%
6/9	12,769,834	1,440	24	1.67%
6/10	9,446,694	1,083	22	2.03%
合計	43,500,499	20,420	476	2.33%

（86,400/60 = 1,440）を表す．5/27 の時間スロット数が 1,440 より少ないのは，ログ収集が開始された当日なので，1 分ごとの集計が 192 回しか行われなかったからである．syslog の収集機構は，5/27 の午後には動作していたが，ラックへと積載された各機器の syslog 出力の設定時間が違うため，収集開始時刻も各機器により異なる．そのため ShowNet 全体として syslog を収集開始した時刻を定めることはできない．また 6/10 の時間スロット数が 1,440 より少ないのは，17 時に ShowNet がシャットダウンされ撤収が開始されたためである．

アラート回数は，ボリンジャーバンドの UpperLimit を超えた回数を表し，発生率は時間スロット数に対して UpperLimit を超えた回数をパーセンテージで表したものとなる．ログ総量の合計から算出した ShowNet 期間中の全アラート発生率は 2.33%となり，集計されたログの 97.67%は UpperLimit を超えなかった．

ShowNet の開催期間は主に準備期間（Hotstage）と会期準備期間，会期の 3 つに分けられる．Hotstage は 5/27 から 6/3 まで，会期準備期間は 6/4 から 6/7 まで，会期は 6/8 から 6/10 までとなる．Hotstage 期間中は，新しい機材のつなぎこみやネットワークの構築，ソフトウェアの稼働などが進められ，syslog の総量が日を追うごとに増加する．会期準備期間には，出展者が出展準備のために会場に持ち込んだ端末を ShowNet へと接続し始める．会期準備期間，ならびに会期中には 200 万行から 1,200 万行ほどでログの総量が推移しているが，ログの総量が増加してもアラートの発生率は 1%台から 2%台の間で推移している．

5.2 アラートの回数とアラートレベルの割合

次に表 2 に示したアラートレベルごとのアラート割合を

集計した．**表 4** がアラートレベルごとに分類した集計結果となる．全アラート回数のうち，Low アラートが 50.63%，Middle アラートが 24.79%，High アラートが 26.47%という割合を占める．Low アラートと Middle アラートがアラートの大部分である約 75%を占めている．

5.3 先行研究との比較と指数移動平均との比較

先行研究 [12] で求めたアラート発生回数と，本提案で動的に計算をした 2 つの移動平均（単純移動平均と指数移動平均）のアラート回数を比較した図が**図 5** である．単純移動平均と比較して，指数移動平均はデータに対して指数関数的に重みを減少させるという特徴がある．重みは指数関数的に減少するので，最近のデータを重視するとともに古いデータを完全には切り捨てないという特徴がある．つまり，指数移動平均は直近のデータの動きに早く反応するということであり，単純移動平均よりも異常を素早く発見できる可能性がある．

表 4 アラートレベル集計
Table 4 Aggregation by alert level.

日付	アラート回数	Low	Middle	High
5/27	12	2	4	6
5/28	46	22	14	10
5/29	50	26	13	11
5/30	48	19	19	19
5/31	27	10	8	9
6/1	47	21	11	15
6/2	38	16	13	9
6/3	28	15	4	9
6/4	38	23	4	11
6/5	24	16	2	6
6/6	24	12	4	8
6/7	24	12	8	4
6/8	24	15	5	4
6/9	24	21	2	1
6/10	22	11	7	4
合計	476	241	118	126
割合		50.63%	24.79%	26.47%

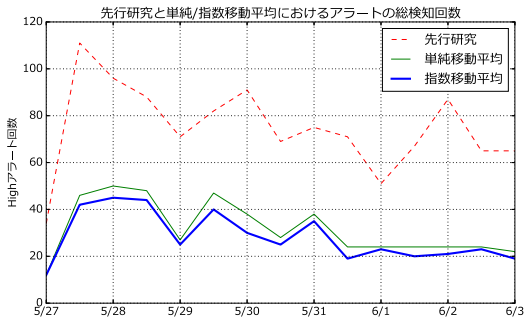


図 5 アラート総検知数比較
Fig. 5 Comparison with previous research.

6. 考察

6.1 アラートの発生率について

5.1 節で示したアラート発生率の結果より，ShowNet における syslog の総量はボリンジャーバンドで仮定した $\pm 3\sigma$ 以内である 0.27%を上回る水準となっており，統計予想回数を上回りアラートが発生していた．これは株式市場ほどイベントネットワークは安定しておらず，純粋なボリンジャーバンドのアルゴリズムでは，正規分布の予想値から外れることを意味する．しかしながら，会期が近づくにつれログの総量は純増しているがアラートの発生率はログの総量に比例して増加はしていない．**表 5** に示すように，全体のログ総量は約 97.60%の確率で $\pm 3\sigma$ の範囲に収まっている．結果として異常通知回数はボリンジャーバンドの統計範囲には収まらなかったが，syslog の総量は $\pm 2.2\sigma$（97.22%）から $\pm 2.3\sigma$（97.86%）の間の分布となり，ボリンジャーバンドアルゴリズムの分布に近似している．σ のパラメータを調整することで syslog 総量による異常検知を行うことは有効であると考えられる．

次に，先行研究 [12] の結果と比較してアラート発生回数の総数は，単純移動平均では 1,123 回から 476 回へと大きく減少した．指数移動平均を用いることで単純移動平均で求めたアラート発生回数の総数である 476 回から 423 回へとさらに減少した．図 5 が示すとおり，日々のアラート発生率は先行研究，単純移動平均，指数移動平均の順に減少している．指数移動平均を異常検出に採用することによりアラート検知回数が大きく減少したことから，より精査された異常をネットワーク運用者へ通知することが可能となる．

表 5 $\pm 3\sigma$ の範囲に収まる確率
Table 5 Probability within $\pm 3\sigma$.

日付	時間スロット数	$+3\sigma$ 以上	-3σ 未満	± 3 の範囲
5/27	617	12	0	98.06%
5/28	1,440	46	2	96.67%
5/29	1,440	50	0	96.53%
5/30	1,440	48	6	96.25%
5/31	1,440	27	3	97.92%
6/1	1,440	47	0	96.74%
6/2	1,440	38	1	97.29%
6/3	1,440	28	0	98.06%
6/4	1,440	38	1	97.29%
6/5	1,440	24	0	98.33%
6/6	1,440	24	0	98.33%
6/7	1,440	24	0	98.33%
6/8	1,440	24	0	98.33%
6/9	1,440	24	0	98.33%
6/10	1,083	22	1	97.88%
合計	20,420	476	14	97.60%

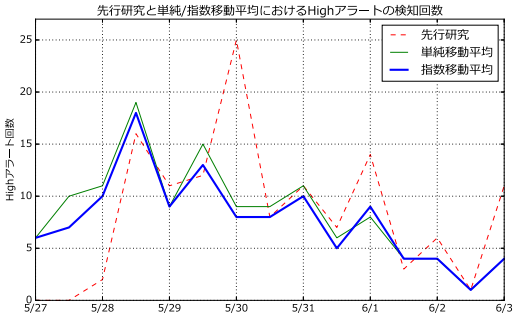


図 6 High アラート検知数比較
Fig. 6 Comparison of high alert detection count.

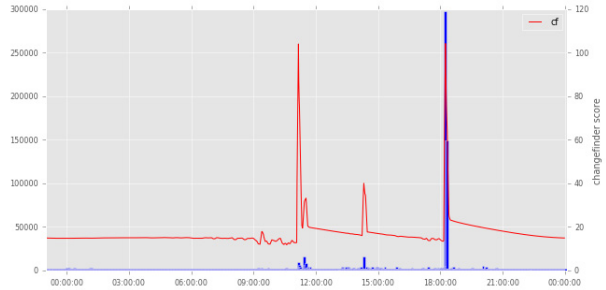


図 7 6/6 の変化点検出
Fig. 7 Changing point detection on 6th June.

6.2 レベル分けの有効性

次に 5.2 節で示したアラート回数とアラートレベルの割合については，Low アラートと Middle アラートが約 75%を占めており，これらのアラートを取り除き High アラートのみを通知すれば，異常通知を減らす意味では有益なフィルタとなりうる．

本論文では，先行研究 [12] と比較して，アラートレベルを固定値の幅で求めるのではなく，動的にレベル分け計算を行う手法を用いた．**図 6** が示すように，結果として High アラートの総数は先行研究と比較して大きく減少していない．先行研究では High アラートは「2σ との差が 1,000 以上離れた値」と定めたが，本研究では 5σ を超えた値とした．先行研究の 1,000 という数値に根拠がなかったわけだが，本研究では 3σ よりもさらに低い確率で発生しうる異常値として 5σ を基準値とした．High アラートの総数は先行研究と単純移動平均ではほぼ差がない状態となり，指数移動平均を利用した場合に微減という結果となった．アラートのレベル分けでは閾値に固定値を用いるよりも，指数移動平均を用いた方がより効果的に大きな変動をとらえられるという結果となった．先行研究では，本来運用者が気がつきたい 1,000 以下の値の中での大きな変動を見つけられないことを意味するが，本研究では 1,000 以下の値の中でも起こりうる大きな変動を検知し通知することが可能となる．

6.3 誤検知への対応と実運用性

本手法ではアラートのレベル分けを行うことで，本来通知すべきアラートを通知しないフォールスネガティブが発生する可能性が高まる．しかしながらレベル分けを行わない場合には，通知されるアラートの総数は，「日に最大 50 件ほど」になる場合があり，ネットワーク運用者が運用を行ううえでアラートの調査負荷が高まる．本提案で用いた指数移動平均を利用したレベル分けを行うことで，High アラートは日に 18 件以下の検知数を記録した．このアラート件数は，レベル分けを行わない「日に 50 回」というアラート数と比較して，少数のネットワーク運用者でも現実的にハンドリング可能な回数である．

ボリンジャーバンドアルゴリズム単体では，誤検知を発見することはできないが，他の監視システムと連携を行い相関をとることで誤検知かどうかを判断することは可能である．また，アラート発生時間のログのみを意味解析し，機械学習を行って誤検知か否かを判定することも可能性の 1 つである．しかしながら，本提案はあくまで syslog の総量を用いた異常検知とネットワーク運用者が実オペレーション可能なアラート通知回数の両立を目標としており，本提案ではその両方が実現可能となっている．

6.4 他の研究との比較

本研究では syslog のヘッダのみを処理対象とし集計を行ったが，他の研究では通常必要なログの意味解析を行い異常を検知する．これにより意味解析を行う計算を省略できるばかりか，集計処理を行う場合にも対象となるデータ量が少なくなり計算を高速に行える．またボリンジャーバンドに必要な計算は移動平均と標準偏差の計算のみであるため，全体的なアルゴリズムの計算量を少なく抑えられる．これは，他の手法に対して計算量が少なく異常検知が行えることを意味する．

本手法と比較するため，ChangeFinder アルゴリズムを適用した計算を行った．**図 7** は 6/6 のデータから ChangeFinder の計算結果を描画したものであり，2 つの大きなスコアの増加が見られる．6/6 では 11 時過ぎと 18 時過ぎにログの総量が大きく変動したことを意味する高いスコアの値となっている．18 時過ぎの大きな変動の原因は，SNMP GET の request と response が対となり 18:10–18:16 の間に数十万行のログが出力されていた．実際には，11 時過ぎの変化よりも 18 時過ぎの変化の方が何倍も大きく，運用者への通知を行いたい事象は 18 時過ぎの変化のはずだが，スコア的にはどちらの時間の変化もほぼ同じ値になっている．ChangeFinder ではログ総量の変化点を見つけることが可能だが，ログ総量がどの程度の割合で増加したか判断することはできず，アラートのレベル分けを実装することは難しく，結果としてネットワーク運用者へアラートの重要度を通知することが困難となる．

6.5 イベントネットワーク特有の問題への対応

イベントネットワーク運用はサービス提供を行うネットワークであり，トラブル対応の速度がネットワーク品質に大きく影響する．ログの意味解析を行う手法の場合，ログの急増により処理しなければならない対象ログも急増し，解析処理自体が遅延してしまい運用者への異常通知が遅れる可能性がある．本手法のようにログの総量にのみ着目し，かつ統計的手法によって異常検出計算も軽量なアルゴリズムを用いることで運用者への異常通知が高速化し，対応速度を重視した運用を行うことが可能となり，イベントネットワーク以外の様々なネットワークやサーバ運用などにも適用することで運用の品質を向上することができる．

一般的に開催されるカンファレンスやシンポジウムの開催期間は長くて 1 週間程度や短くて 2–3 日という中でイベントネットワークが構築されるが，ShowNet は 2 週間という比較的長い時間をかけ構築運用される．本提案手法では，1 時間分のウィンドウ幅でデータが学習され計算される．短期間のイベントネットワークにおいても，学習に必要なデータを短時間で蓄積でき計算することができる本提案アルゴリズムが有効に働く．

6.6 実データの利用

ShowNet の syslog 運用では事前に運用メンバが把握している特定キーワードによる閾値ベースの監視が行われている．syslog 可視化が行われるまでは，ループが発生した場合には解決に数十分から数時間かかる場合もあった．syslog の実データが監視に用いられることにより，異常検知までの時間が数分まで大幅に減少した．閾値ベースの syslog 監視は，発生した事象がイベント回数として可視化され通知されるため運用者にとって有用に働く．しかしながら，運用メンバが把握している特定キーワードにのみ対処可能であり，それ以外のログ異常に関してはやはり運用メンバの勘によるシステム運用がなされている．検知されたからといってそれが本当に検知したいトラブルなのか，ネットワーク機器へ設定を行ったため発生した記録ログなのか判断がつきにくく，最終的には運用者が確認を行うこととなる．また，特定キーワードはイベントごとに変化する可能性があり，さらに監視を行う閾値の適切な調整が必要となり，運用者の経験に頼るシステムでは自動化は困難である．

ShowNet という実際に運用構築される大規模なイベントネットワークにおける，マルチベンダ機材が混在する特殊な環境において収集された syslog の総量を監視し，異常を自動検知することが，ShowNet 運用の属人性の排除を進める一歩となる．さらに本実験が実データを用いて評価されたということは，本実験がシミュレーションではなく，実際の ShowNet の運用自動化を進める大きなアドバンテージとなる．

7. まとめと今後の課題

7.1 まとめ

本論文では，ShowNet という大規模でマルチベンダ機材が大量に投入されるイベントネットワークでの異常検知を，syslog の総量を集計してボリンジャーバンドアルゴリズムを用いて解析し検知する手法を提案した．これによりネットワーク運用者の経験と勘に頼る属人性を排除した自動的な異常検知を行い，ネットワーク運用者へ通知するというシステムの実現が可能となる．また，指数移動平均をベースとした動的なアラートのレベル分けを追加することで，ネットワーク運用者に対して現実的な異常発生回数を通知する機構を実現した．結果として，先行研究よりも良い精度でのアラート検出とレベル分けを行うことが可能となり，ネットワーク運用者へのトラブル対応の高速化の可能性を示した．

本提案は，ログの意味解析をあえて行わず本来は意味解析が有効である機械学習や統計分析では実現できない，軽量で高速に計算できる単純なアルゴリズムが有効に働いた．高速な計算による自動的な異常通知の結果として，トラブルへの初期対応を素早く実行することが可能となり，属人的なイベントネットワーク運用に大きく貢献できる可能性がある．また，本手法はイベントネットワークだけでなく，企業や大学などで行われるネットワーク運用やサーバ運用に対しても有効に働き，多くのネットワークのトラブル解決のための一手法として貢献できる．

7.2 今後の課題

本論文ではイベントの事後に分析処理を行ったが，ログをリアルタイムに処理することで本研究を実運用へと乗せることが可能となり，運用者へリアルタイムでアラートを通知することが可能となる．また，総量にのみ着目するのではなく送信元ホスト別に異常を検出する仕組みを作ることで，運用者に対して異常発生源が送信元ホストであるという直接的な問題解決の提示を行える syslog 監視システムを提案できる．さらに，セキュリティ機器のログを対象にすることにより，セキュリティインシデントに対する異常検知という点で貢献可能である．

参考文献

[1] Interop Tokyo, available from ⟨http://www.interop.jp/⟩.
[2] ShowNet, available from ⟨http://www.interop.jp/2016/shownet/⟩.
[3] VMware vRealize Log Insight, available from ⟨http://www.vmware.com/jp/products/vrealize-log-insight.html⟩.
[4] Kalekar, P.S.: Time series forecasting using holt-winters exponential smoothing, Kanwal Rekhi School of Information Technology 4329008, pp.1–13 (2004).
[5] Kleinberg, J.: Bursty and Hierarchical Structure in

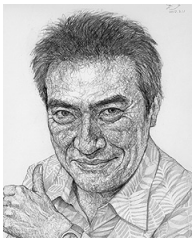
Streams, *Proc. 8th SIGKDD*, pp.91–101 (2002).
[6] Takeuchi, J. and Yamanishi, K.: A Unifying Framework for Detecting Outliers and Change Points from Time Series, *IEEE Trans. Knowledge and Data Engineering*, Vol.18, No.4, pp.482–492 (2006).
[7] Mahalanobis, P.C.: On the generalised distance in statistics, *Proc. National Institute of Sciences of India*, Vol.2, No.1, pp.49–55 (1936).
[8] Box, G.E.P., Jenkins, G.M. and Reinsel, G.C.: *Time Series Analysis Forecasting and Control, 3rd Edition*, Prentice-Hall International, Inc. (1994).
[9] Bollinger, J.: *Bollinger on Bollinger Bands*, McGraw Hill (2002).
[10] Gerhards, R.: RFC 5424, The Syslog Protocol, March 2009, available from ⟨https://tools.ietf.org/html/rfc5424⟩.
[11] pandas, Python Data Analysis Library, available from ⟨http://pandas.pydata.org/⟩.
[12] 阿部 博, 敷田幹文：イベントネットワークにおける syslog を用いた異常検知手法の提案と実データを用いた評価，*IOTS2016* (Dec. 2016).

阿部 博 （正会員）

株式会社 IIJ イノベーションインスティテュート技術研究所研究員．北陸先端科学技術大学院大学篠田研究室博士後期課程所属．ACM 会員．

敷田 幹文 （正会員）

高知工科大学情報学群大学院工学研究科基盤工学専攻情報学コース教授．



篠田 陽一

北陸先端科学技術大学院大学教授．