機械学習を用いたログのリアルタイム監視

【論文】

機械学習を用いたログのリアルタイム監視

井上 正人[*1]

Real-Time Log Monitoring Using Machine Learning

Masato Inoue

Abstract

We propose a method for monitoring system logs on Linux servers. We use Jubatus which is a machine learning library to monitor logs on user logins of SSH. First Jubatus learns logs of authorised access. After that, Jubatus can tell logs of authorized access from logs of unatuhorized access. We write Python programs to handle Jubatus and log data. This method provides a simple way for monitoring system logs compared with commercial software.

Key words: Machine Learning, Jubatus, Log Monitoring

1 はじめに

コンピュータにおいては，システムを構成する機器に関する情報やアプリケーションの動作に関して，さまざまなログが記録される．これらのログは時系列で記録されており，一定期間保存され，もし，システムまたはアプリケーションに関して何らかの異常が起こった場合，その原因を明らかにするための重要な手がかりとなるものである．通常，ログはファイルに保存され，定期的に調べられる他，システムに異常があった場合，それを用いて調査される．

一般の設定においては，ログはファイルに逐次保存されていくもので，リアルタイムに監視するものではない．しかし，ログの中には，リアルタイムに監視しておきたいものもある．可用性の高いシステムにおける機器のログや，不正侵入，情報漏洩に関するログなどである． 特にコンピュータシステムに関するログのリアルタイム監視には，商用のソフトウェアの他，オープンソースの Nagios[1)] が使用されるが，どれも設定にはシステムに関する詳細な知識と，ソフトウェアに関する高度な知識が必要で，簡単に使用できるものではない．

今回は，機械学習により，正常なログと異常なログを判別し，異常なログであれば通知するシステムを構築する．機械学習により，複雑な設定やログの詳細な内容を解析することなしに，ある程度，異常・正常を判別できるようになる．

2 Jubatus による機械学習

ビッグデータの時代になり膨大なデータを扱う時代になった．データ量が膨大になるだけではなく扱うデータも複雑になってきている．従来のモデルを作成し解析する手法では，モデルの作成が困難になってきている．高次元のデータにおいては，人間の把握能力を超えたデータの関係を解析することが困難で，十分な分析が行えないケースが出てきている．このため，統計的な手法に基づいてデータを分類したり，新たな入力に対して出力値を予測する機械学習を行うことが有用となっている．

今回は，機械学習のフレームワークとして Jubatus[2)] を利用する，Jubatus は株式会社 Preferred Networks と日本電信電話会社ソフトウエアイノベーションセンタが開発した，オープンソースのソフトウェアで，さまざまなオンライン機械学習の機能を提供する．機能としては

Received November 15, 2016

*1 海上保安大学校　inoue@jcga.ac.jp

・多値分析
・線形回帰
・推薦（近傍探索）
・グラフマイニング
・異常検知（アノマリ）
・クラスタリング

を備えている．本論文においては，異常検知の機能を利用して，ログが正常か異常かを判別する．Jubatus の異常検知のアルゴリズムは LOF(Local Outlier Factor)[3] の手法に基づいている．

3 ログ監視を行うコンピュータについて

ログ監視を行うコンピュータの OS は Linux のディストリビューションの 1 つである CentOS 7[4]を使用した．CentOS 7 には Jubatus をインストールするためのパッケージ(バージョンは 0.9.4)が用意されており，そのパッケージを利用した．また，Jubatus とデータのやり取りやログファイルを扱うために Python をインストールした．Python の他には Java や Ruby によるプログラムの作成も可能である．

コンピュータのハードウェアのスペックは以下の通りである．

CPU：Intel Xeon E3-1240 V2(3.40GHz)
メインメモリ：4GB
ハードディスク：500GB SATA

CentOS 7 のログ管理は，CentOS 6 以前の syslog と異なり systemd-journald で行われている．systemd-journald はバイナリでコンピュータにシステムのログが保存されている．journalctl コマンドを用いると，保存されているログから必要なログをテキストファイルで表示することができる．今回は，すでに保存されているログが対象ではなく，リアルタイムで，新しく出力されるログを監視するので，journalctl コマンドで，新規のログを指定されたファイルに出力するように設定する．

ログの監視は Python のプログラムで行う．ログが出力されるファイルをモニターしておき，出力されたログを Jubatus で評価する．Jubatus においては，データはすべて JSON(JavaScript Object Notation)形式で与える．JSON 形式においては，データはキーと値のペアで与えられる．例えば

{"key1":"value1", "key2":"value2", …}

のように"key1"，"key1"などがキーで"value1"，"value2"などがその値となる．journalctl のログは通常のテキストファイルの形式と，JSON 形式の詳細なログの出力形式がある．前者の場合は，JSON 形式に変換してデータを Jubatus に渡す必要がある．今回は JSON 形式の出力を利用した．"value1"，"value2"などの値は数値またはテキストで与えられるが，今回解析に使用したキーの値はテキスト形式で，数値の部分はなかった．

テキスト形式のアノマリを判断するために N-gram の手法を用いる[5]．たとえば N が 2 場合の 2-gram においてはテキストデータが

That is OK.

であった場合，空白も文字として

"Th"，"ha"，"at"，"t "，" i"，"is"，"s "，" O"，"OK"，"K."

のように 2 文字ずつのデータに分割してから数値化する．これによりテキストデータの統計処理が可能となる．

4 ログ監視のシステム

図 1 にログのリアルタイム監視を行うためのシステムの概略を示す．

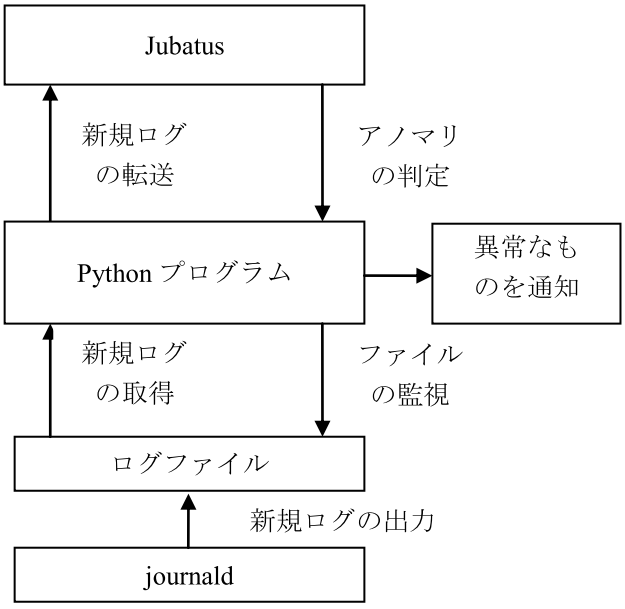

図 1 ログのリアルタイム監視

中心となるのは Python のプログラムで，ログの書き込みの監視，新規に出力されたログの受け取り，受け取ったログの Jubatus への送信，Jubatus から

のスコアの受け取り，スコアの値よってシステム管理者への通知など，中心的な役割を果たす．図１はログのリアルタイム監視を，ログを監視するシステムと同じコンピュータで行っている．システム関連のログについては，コンピュータに異常が発生した場合に，そのコンピュータ自身では異常の発生をうまく通知できないことも考えられる．このような場合，監視を別のコンピュータで行う必要がある．このとき，ログを別のコンピュータに送信する必要がある．これを行うためには Fluentd(td-agent)[6] のソフトウェアを使用する．ログを監視したいコンピュータとログを監視しるコンピュータ双方にこの Fluentd のソフトウェアをインストールしておき，監視対象のコンピュータから出力されたログを監視するコンピュータに送信すればよい．図２に監視対象のコンピュータと監視するコンピュータが異なるシステムの概略を示す．

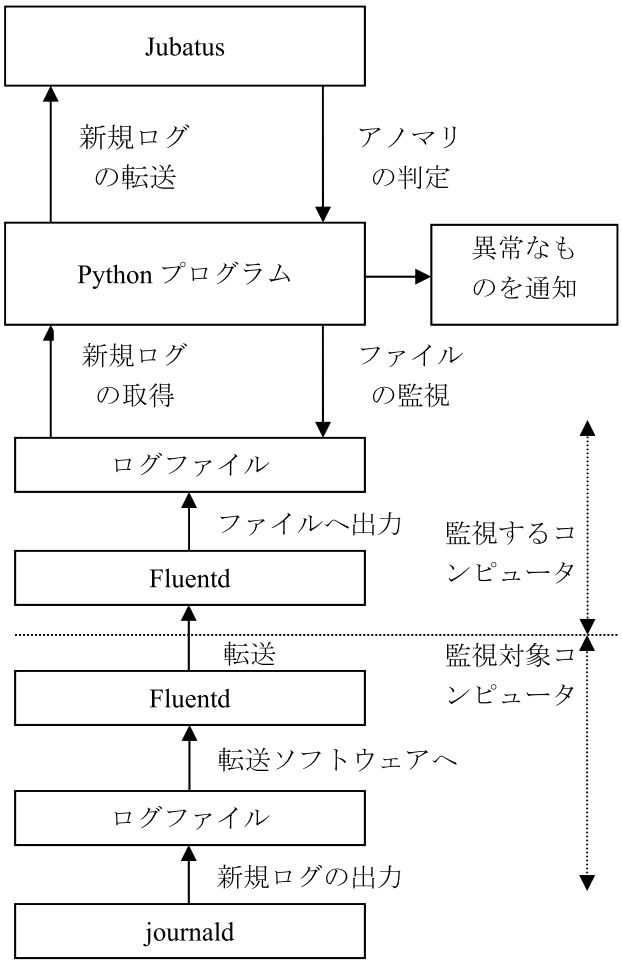

図２ 監視対象コンピュータと監視するコンピュータが異なる場合

このシステムでは，ログを Fluentd 間で転送し，転送された先でログを一度ファイルに書きだして監視している．これを直接，Jubatus に転送し，アノマリの判定を行わせる方法があるが，Fluentd と Jubatus 間のファイルの転送を行うプログラムの作成が難しく，一度ファイルに書きだす方がプログラムが簡単になる．

5 解析するログについて

当初は，メモリの異常や CPU の異常などのシステム的な異常のログを検知して通知するシステムの構築を目指していたが，Linux のシステムは堅牢で，メモリを消費するプログラムや CPU の使用率をフルにするプログラムを実行しても，エラーを出力しなかった．また，正常に動作しているときにログの出力がないため，学習データの取得が困難であった．このため，SSH のログイン時の出力に着目し，正常なログイン時のログとログインの失敗のログで，ログイン失敗時に通知するシステムを構築する．

6 結果

まず，学習データとして，20回正常にログインしたときのログで学習を行う．JSON 形式のログはキーが"MESSAGE"の値のみを使用し，残りは使用しなかった．異常検知のためのアルゴリズムは最近接を利用した LOF である light_lof を用いた．テキストは2-gram で数値化する．学習した後は，ログのスコアの計算のみを行い，正常なログインがあってもデータによる学習は行わない．このデータをもとに，当たに出力されたログに対してスコアを計算する．計算方法は以下の通りである．

新たに出力されたログを数値化したデータを x とし，学習データで x に i 番目に近いデータを y(x, i) とする．最近接 k を取った場合のスコアは

$$r = \frac{1}{k}\sum_{i=1}^{k}\{y(x,i)\ とx\ の距離\}$$

と

$$s = \frac{1}{k}\sum_{i=1}^{k}\left[\frac{1}{k}\sum_{j=1}^{k}\{y((x,i),j)\ とy(x,i)\ の距離\}\right]$$

の比でr/sとなる．1より大きな値ほど異常値となる．最近接は値をいろいろ試して10に設定している．

学習の後に，新たに正常にログインしたときの Jubatus のスコアを表1に示す．10回の正常なログインを表にしている．

表1 正常なログインの値

回目	正常なログイン
1	1.01316928864
2	1.04039347172
3	1.06597709656
4	1.05348265171
5	1.03667426109
6	1.03374111652
7	1.07132065296
8	1.0031144619
9	1.03823316097
10	1.04460585117

すべての値が 1 付近になっていることが分かる．正常なログインの場合は，この 1 種類のログのみである．

次に，ログインが失敗したときの場合であるが，まず，ID は合っているが，パスワードが違っていて失敗している場合，認証失敗のログとパスワードが不整合であるログが連続して表示される． この後，接続を中止しする操作のときに，接続中止のログが書き込まれる．このとき，認証失敗のログと接続を中止したときのログは毎回同じでスコアは

認証失敗：4.59140205383
接続中止：3.35965800285

であった．これらのログのスコアが毎回同じであるのは，出力されるログの"MESSAGE"の部分がすべて同じであるからである．パスワードが不整合であるログのスコアは以下の通りとなる．

表2 パスワードでログインに失敗したときの値

回目	パスワードの不整合
1	2.4686434269
2	2.41592907906
3	2.3161072731
4	2.37118387222
5	2.53163027763
6	2.42310833931
7	2.56015300751
8	2.55610322952
9	2.41233038902
10	2.39591884613

次にユーザ ID が不正でログインに失敗した場合であるが，ユーザが不正，ユーザが不正(認証前)，ユーザが不明，認証失敗，パスワードの不整合の 5 つのログが連続して表示され，接続を中止する操作のときに，接続中止のログが書き込まれる．このうち，ユーザが不正，ユーザが不正(認証前)，ユーザが不明，認証失敗，接続中止のログは毎回同じとなり，スコアは以下のようになる．

ユーザが不正：2.51353573799
ユーザが不正(認証前)：3.63927173615
ユーザが不明：3.45935916901
認証失敗：4.53475999832
接続中止：3.35965800285

また，パスワードの不整合のスコアは表 3 のようになる．

表3　ユーザ ID でログインに失敗したときの値

回目	パスワードの不整合
1	1.84954452515
2	1.77145898342
3	1.69564151764
4	1.7666028738
5	1.8276399374
6	1.70861172676
7	1.7845531702
8	1.77943015099
9	1.79167377949
10	1.83179855347

表 1 の正常にログインしたときのスコアと表 2，表 3 及び，認証失敗，接続中止などのログインに失敗したときの他のすべてのログのスコアを比較して，Python プログラムにおいて，値を 1.4 に設定し，これより小さければ正常とすれば，ログインに失敗したときのログをログインが成功したときのログと比較して異常なログとして判別できる．Python プログラムでメールの送信などの通知を行えば，ログのリアルタイム監視が行える．

7 おわりに

今回，Jubatus を用いて，SSH のログインに対するログのリアルタイム監視をおこなった．これにより，正常なログと異常なログを Jubatus のアノマリの値により，区別し，通知することが可能になる．SSH に対しては，公開鍵方式の認証もあり，これら

のログインなど，他のログに対しての設定も必要であるが，今回は行えなかった．

ログについては，異常時にしか出力されないものもあるが，オプションなどで正常時にも何等かのログを出力するようにすれば，この方法が使える範囲が広がると思われる．

今後，今回の方法を応用して，メールサーバのソフトなど，いろいろなサーバソフトに対してログのリアルタイム監視を行えるシステムを構築してゆきたい．

参考文献

1) Nagios Enterprises LCC, Nagios - Network, Server and Log Monitoring Software, https://www.nagios.org/ (2016 年 11 月 14 日参照)
2) 株式会社Preferred Networksと日本電信電話株式会社, Jubatus : オンライン機械学習向け分散処理フレームワーク — Jubatus, http://jubat.us/ja/ (2016 年 11 月 14 日参照)
3) M. M. Breunig, H.-P. Kriegel, R. T. NG, J. Sander, LOF: Identifying Density-based Local Outliers, *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, (2000), 93-104
4) The CentOS Project, CentOS Project, https://www.centos.org/ (2016 年 11 月 14 日参照)
5) C. E. Shannon, Prediction and Entropy of Printed English, *Bell System Technical Journal*, **30**, (1951), 50-64
6) Fluentd Project, Fluentd | Open Source Data Collector | Unified Logging Layer, http://www.fluentd.org/ (2016 年 11 月 14 日参照). td-agent は Fluentd の安定版