Data Analytics
Stiftung Universität Hildesheim
Marienburger Platz 22
31141 Hildesheim
Prof. Dr. Dr. Lars Schmidt-Thieme

Thesis
Unsupervised Real-Time Time-Series Anomaly Detection

Abdul Rehman Liaqat

271336, Liaqat@uni-hidesheim.de

Abstract

Anomaly detection is a crucial task for machine learning due to wide-spread usage and type. In particular, it is worth noting that most data arising in industrial setups are of a streaming nature, thus restricting the range of standard anomaly detection tools. This thesis will identify the potential approaches to learn the identification of abnormal behavior from large-scale streaming data. An empirical comparison of state-of-the-art methods will to be extended by a novel technical contribution. In this thesis, the focus is particularly on streaming time-series Anomaly Detection which changes in nature with time and novel contribution will especially try to target this dynamic nature of time-series.

Contents

1 Introduction 10 pages

1. Usage of streaming data
2. Usage of anomaly detection in general
3. Software2.0 development and anomaly detection role
4. Usage of anomaly detection in streaming data
5. problems in the concept and brief intro to our innovation

1.1 Usage of streaming data

With the increase in networking of objects, amount of data being generated is increasing. A big chunk of this data involves streaming data. Streaming data can be defined as data being generated continuously or with a continuous time interval. Statistically, IoT data has grow from 4.4ZB (Zeta Bytes) to 44.4ZB with 50 billions devices. These devices include a small temperature sensor installed in a room to ECG data to satellite communication data. Even videos are kind of streaming data since they include data points (frames) separated by a constant interval.

1.2 Usage of anomaly detection in general

An anomaly is defined as significantly different data point from other data points. The difference is dependent upon the at least two things. One is the data point being an outlier in general among all data points available. Second is that the data point value doens't make sense in the sequence data is being generated hence it is possible for new data point to be extremely different from all previous ones but still make sense to have in the sequence and vice versa. Detecting an anomaly helps in identifying abnormal behavior of a process with potentially useful information.

1.3 Software2.0 development and anomaly detection role

As described in (reference name Software2), going forward software development especially machine learning based software development will be divided into two parts. One part is the preparation of the data and the second part is the design and optimization of algorithms. Both of these steps are used iteratively. Interestingly, as described in the (reference name Software2), more and more time is spent on accumulating, massaging and cleaning datasets. One major part of this process is outlier detection, anomaly detection and novelty detection. By detection of anomalies, it will become sufficiently easy for human expert to find the one data point in a heap of millions data points and then decide whether to remove that data or get more of the same data points. Need of an anomaly detection algorithm to identify faulty devices or prevent potential system failure before it happens. Thus the anomaly detection algorithm should be generic (unsupervised), be ready to predict on live data and be able to consider the important past data points (temporal and spatial pattern detection).

1.4 Usage of anomaly detection in streaming data

1.5 problems in the concept and brief intro to our innovation

1. Stationary and non-stationary time series
2. lack of labelled data
3. Amount of diversity hence calling for a universal detector
4. Concept drift
5. Points of innovation or constraints to consider for innovation

2 Related Work and State of the art 10 pages

1. Time series intro. Prediction and classification.
2. Unsupervised methods on time series
3. Anomaly detection
4. Time series Anomaly detection
5. Unsupervised Time series Anomaly detection
6. General components latest anomaly detection methods and explanation of each
7. HTM based anomaly detection paper review. Problems or way it does it
8. paper 2
9. paper 3
10. general review, problems, bottlenecks and trend

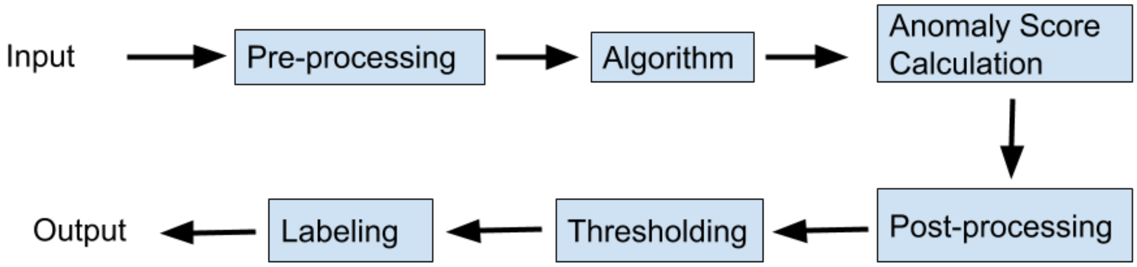

Figure 1: General Anomaly Detection Framework

3 Proposed method 10 pages

3.1 General anomaly score based architecture and detailed explanation

Proposed method can be divided in multiple parts. First part would be the implementation of baselines using latest deep learning architectural components. We use these components to develop two different kind of detectors.

One detector is based upon prediction of a next data point. Let $T_{(t-L-1):(t-1)}$ is the small window taken from time series. The starting point is $t - L - 1$ where L is the window size and last data point is taken from $t - 1$ which is the data point at previous time step. This data window is used to predict the time series value at time t. To predict at time t various different kind of functions as baselines using latest architectural components are designed.

Generally the output of such a function would be

$$f(T_{(t-L-1):(t-1)}) = \hat{f}(t) \tag{1}$$

Thus the function f in prediction based models map data points as: $f : L \mapsto 1$ where L is the window size and output is the prediction of value at current time. For optimization of function f, the difference between time series value at current time step and predicted value at current time is taken. More specifically the objective function will be

$$min|\hat{f}(t) - f(t)| \tag{2}$$

That is to minimize the difference between time series value at time t and predicted time series value at time t using time series value from $T_{(t-L-1):(t-1)}$.

3.2 Describe fully connected prediction architecture

Considering figure 1, each algorithm will be using all parts with few variations and a cusotmized algorithm part.

Pre-processing is playing crucial role in the generalization of the algorithms. There are two different kind of normalization techniques tested. First kind is global normalization

3.3 Describe Convolution based prediction architecture

3.4 Describe LSTM based prediction architecture

3.5 Describe autoencoder based architecture

3.6 Describe fully connected based autoencoder based architecture

3.7 Describe Convolution based autoencoder based architecture

3.8 Describe LSTM based autoencoder based architecture

3.9 Recency concept

3.10 Post-processing

3.11 Thresholding

3.12 Parameterinzing the anomaly score

4 Empirical Formulation and Experiments 5 pages

5 Results 10 pages

6 Conclusion and Discussion 5 pages

7 Experiment Infrastructure

7.1 Experiment Management using MLflow

7.2 Parallel execution using Docker

8 Best practices

Following steps were taken to maximize the efficiency and speed of research:

1. Use version control to track the code and share between different devices.

2. Separate code from data. This will keep the code base small and easy to debug.

3. Separate input data,working data and output data.

   - **Input Data:** Input data-set that never change. For my case it is NAB and other external datasets.
   - **Working Data:** nothing for now.
   - **Output Data:** Results and threshold profiles in my case.

4. Separate options from parameter. This is important:

   - Options specify how your algorithm should run. For example data path, working directory and result directory path, epochs, learning rate and so on.
   - parameters are the result of training data. it includes the score and hyper-parameters.

8.1 Moving from jupyterlab to pycharm

While working with jupyterlab notebook following routine was followed: 1- Load data with sample function 2- Write an algorithm 3- Test the results 4- Write general executeable .py file. 5- Get results on server

Since we needed to track change on two different places, it was becoming harder to track the bugs and improve on efficiency. That's why pycharm was selected to create executeable files and test algorithms at the same time.

9 Reference Usage

10 References

‘