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

2 Related Work and State of the art 10 pages

1. HTM based anomaly detection paper review. Problems or way it does it
2. paper 2
3. paper 3
4. general review, problems, bottlenecks and trend

3 Proposed method 10 pages

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

‘