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

1.1 Motivation

1.2 Objective

2 Related Work and State of the art

3 Benchmarks

3.1 Autoencoder based models

3.1.1 Fully connected layers

3.1.2 Fully convolution layers

3.1.3 LSTM based

3.2 Prediction based models

3.2.1 Fully connected layers

3.2.2 Fully convolution layers

3.2.3 LSTM based

4 Unsupervised Anomaly detection with recency

5 Usage of RBF loss function

6 Experiments

6.1 Data

6.1.1 Numenta Anomaly Benchmark (NAb)

7 Execution and Results

8 Discussion

9 Experiment Infrastructure

9.1 Experiment Management using MLflow

9.2 Parallel execution using Docker

10 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.

11 Reference Usage

12 References

‘