Gottfried Wilhelm Leibniz Universität Hannover
Institut für Verteilte Systeme
Distributed Computing & Security Group

Master thesis
Informatics (M.Sc.)

Anomaly detection in streaming data using autoencoders

Student:	B.Sc. Bin Li
First Supervisor:	Prof. Dr. Eirini Ntoutsi
Second Supervisor:	Prof. Dr. Wolfgang Nejdl
Date:	April 27, 2018

Declaration of Authorship

I hereby certify that this thesis has been composed by me and is based on my own work, unless stated otherwise. No other person's work has been used without due acknowledgement in this thesis. All references and verbatim extracts have been quoted, and all sources of information have been specifically acknowledged.

Hanover, April 27, 2018

B.Sc. Bin Li

Contents

List of Figures

List of Tables

Codeverzeichniss

Abstract

Data stream is a data format appears in plenty of big data research scenarios, for example, manufactural sensors, production line data etc. Here anomaly detection plays an important role for use cases like predictive maintenance, event detection, and could potentially avoid large amount of financial costs. However, different from traditional anomaly detection tasks, anomaly detection in streaming data is especially difficult while data comes along the time with latent changes, so that a single static model doesn't fit streaming data all the time. In this paper, we propose a novel autoencoder-based anomaly detection approach specially designed for streaming data. The model takes mini-batches of data from the stream as input, and try to reconstruct it using autoencoder, and the anomaly likelihood is informed by the reconstruction error. Experimental results suggests that our model can sufficiently detect anomaly from data stream and update model online to fit the latest data property.

Chapter 1

Introduction

Anomaly detection is an important problem in data mining, and widely used in the manufacturing industry, commercial world, internet company etc. It could avoid or reduce lose in many scenarios like machine health monitoring, credit card fraud detecting and spam email classification, and could also be used as a preprocessing step to remove anomalies for datasets. There are already plenty of anomaly detection techniques proposed in previous literatures, that solve this problem from variety perspectives, e.g. distance-based methods, clustering analysis, density-based methods etc.

There is no lack of anomaly detection approaches that perform good with respect to different kinds of data, however, majority of them are batch model, which means, all data should be available in advance. This becomes a shortcoming under today's big data background. With the rapid development of hardware in the last decade, the situation of data acquisition and analysis has significantly been changed. Specifically, the IoT application. Assume that we collect data from sensors attached to IoT devices, the data comes continuously and everlasting. During data analysis, we should always consider the volume and velocity of data, which means, on one hand, with traditional batch classifiers, the infinity data stream will lead to out of memory, on the other hand, streaming data usually comes with a high speed that leaving the system few processing time. In addition, the statistical property of data may also change over time, which is formally called 'concept drift'. The model should always learn new knowledge from the stream and update its definition of normal and anomalous automatically. To this end, an anomaly detection system for streaming data should be able to 1) be initialized with only a small subset, 2) process streaming data and make prediction in real-time, 3) adapt data evolution over time.

Malhotra et al. introduced an autoencoder based anomaly detection approaches in [1],[2], and achieved good performance in multiple time series dataset. However, in this approach, they assume that the whole datasets are available beforehand, and didn't considered the aforementioned online learning difficulties. Hence, we enhanced this kind of autoencoder based anomaly detection approaches with the online learning ability by inplementing incremental model updating strategies based on the streaming data.

In this paper, we introduce a novel and robust incremental autoencoder-based anomaly detection model, which designed specifically for time series data in a streaming fashion using Long Short-Term memory (LSTM) units, with also online learning ability for model updating. For each accumulated mini-batch of streaming data, the autoencoder reconstructs it with previous knowledge learned from normal data. Anomaly data (never used for training) is expected to cause significant larger reconstruction error than normal data. In addition, the model update itself online according to performance-based criterions.

Chapter 2

Related works

There are already pretty much researches about anomaly detection, some of them referred to streaming data and online learning. In this section, we list some widely used classical machine learning-based approaches, as well as some autoencoder based researches, and finally we refer the provious work on neural network incremental learning.

2.1 Classical machine learning based approaches

As an important component of data mining and machine learning, anomaly detection has been investigated using plenty efficient models.

LOF model

In anomaly detection, the Local Outlier Factor (LOF) is a common density-based approach. LOF shares some concepts with DBSCAN such as 'core distance' and 'reachability distance', in order to estimate local density. Here, points with substantially lower local density than their neighbors are considered as anomalies. LOF shows competitive performance in many anomaly detection tasks, especially when dealing with data with unevenly density distribution. However, when getting a numerical factor from LOF model, it is actually hard to define a threshold automatically for the judgement of anomaly.

OCSVM model

Another widely used model is the domain based One-class Support Vector Machine. As an unsupervised one-class classifier, OCSVM takes only normal data as input, and generates a decision surface to separate them from the anomaly states. By analyzing anomalies, the datasets are always bias to the normal part, and anomaly appear only rarely. So, this kind of one-class classifiers avoid making balance between the two classes. Besides, they also take advantage of classical support vector machine, with the help of kernel method, they can also deal with linearly not separable data. Although clas-

sical machine learning approaches can handle most of the normal anomaly detection, only few of those approaches could be directly or after some modification used for time series or streaming data, while they ignore the temporal dependency between samples.

2.2 Autoencoder-based anomaly detection approaches

LSTMs-Autoencoders are originally widely used for text generation. Text data are usually embedded into vector as input of autoencoder. And the tasks are either generate temporal relevant text on the decoder side or learn text representation in the hidden layer [1]. As text data are relevant in the sense of words within a sentence or between sentences. It is similar to the streaming data temporal dependency problem.

Sutskever et al. [8] use a deep LSTMs-based sequence to sequence model for language translation. In their work, the deep LSTMs encoder take single sentence as input, and learn a hidden vector of a fixed dimensionality, and then a different LSTMs decoder decodes it to the target sentence. As a translation task, they found that this encoder-decoder architecture can capture long sentences and sensible phrases, especially they achieved better performance with deep LSTMs in compare with shallow LSTMs. In addition, a valuable found is, reversing the order of words in the input sentence makes the optimization problem much easier and achieved better performance. The LSTMs based model outperforms non-LSTMs model on the long input sentence cases (more than 35 words) since its long-term memory ability.

Li et al. [2] did similar research on long paragraph text or even entire document generation using LSTMs-autoencoders. Their main contribute is the hierarchical sentences representation. The model learns words level, sentence level, paragraph level and document level information with each respectively a LSTMs layer, so that the model captures very long-term temporal information. Moreover, they introduced an attention based hierarchical sequence to sequence model that connect the most relative part between encoder and decoder like the works around a final punctuation. They experiment with documents over 100 words, the results show that hierarchical and attention-based hierarchical LSTMs learns even better long-term temporal information than standard LSTMs-encoder-decoder models.

As autoencoders achieves great successes in text data and speech processing, they are also used on time series anomaly detection in terms of temporal dependently data. These models train autoencoders with only normal data, and anomaly data as unknown patterns. Then the autoencoder can only reconstruct normal patterns, large reconstruction error indicates anomaly. An early work [6] uses the vanilla autoencoder to detect abnormal status of the electric power system. In order to capture temporal information, they applied sliding window on the raw data as input. As anomaly scoring method, they evaluated each sliding window with respect to their reconstruction error. As some mea-

sures in the autoencoder output vectors are more sensible to anomalies than others, they use the average absolute deviation of reconstruction error as anomaly score. And the anomaly threshold is chosen by large amount of experiments over normal data.

An important reason of using autoencoder for anomaly detection is its ability of dealing with high-dimensional. Sakurada et al. [7] experimented with time series data that consist of 10-100 variables with no linear correlation. Comparing with reconstruction using PCA or Kernel PCA techniques, using the autoencoder reconstruction error is more easily to recognize anomalies.
In further researches, Malhotra et al. [5][3] develop the application of LSTMs-autoencoder in sequence learning into anomaly detection problem. They proposed stacked LSTM networks model to learn high level temporal patterns. The show that LSTMs outperforms normal RNNs based anomaly detection model and avoid facing to the gradient vanishing problem. They also detect anomaly based on the reconstruction error. The scoring function is based on the parameters of a estimated normal distribution of a validation set. Their experiments show that the model performs good in variety kinds of datasets. A variation of this model [4] has been shown that achieves better performance in the anomaly detection tasks. The author tells that, using a constant as input of decoder instead of read time series value improves the performance of model.

2.3 Online incremental learning with autoencoders

Zhou et al. proposed an online incremental updating method for denoising autoencoders by modifying the hidden layer neurons in order to deal with the non-stationary streaming data properties. The kern ideal are two steps, merging hidden layer neurons if there are information redundancy, and adding hidden layer neurons to capture new knowledge. Their experimental result shows comparable or better reconstruction result than non-incremental approaches with only few data used during initialization. And they show that their incremental feature learning methods performs more adaptively and robustly to highly non-stationary input distribution.
Dong et al proposed a 2-step anomaly detection mechanism with incremental autoencoders. The implemented the system with ensembled autoencoders in multithreads to leverage parallel computing when large volumes of data arrive. Besides their 2-step mechanism check anomaly in the first step and verify anomaly data with previous and subsequent data (to differ between anomalous state and concept drift) to reduce false-positive rate in anomaly detection. In the experimental results, they show that their model outperforms commonly used tree-based anomaly detection model especially when concept drift presents and speed up the online processing speed with mini-batch learning and online learning in multithreads.

Chapter 3

Preliminaries

3.1 Definition of a stream

Definition of a stream(time series, dimensionality, volume, velocity, label) Assuming that there are some devices or data warehouse that generate data continuously with a velocity V (here we only taking about numerical data). The data stream from 1st timestamp until ith timestamp is descripted as: ********
Where *** represent the instance at timestamp t in the data stream. And we assume the volume of data stream is infinity, which means, there are always available data instances generated by the data source. To be more generally, we consider *** as either univariate or multivariate, *** is defined as ************
Where <f*1,. . . ,f*N> is the feature space of the data stream with size N. For each instance ***, the label y*t = 0,1 tells either the instance is normal or abnormal.

3.2 Method of processing the stream

For further online processing and detection, we generate mini-batches upon the data stream. The streaming data is accumulated as window W, and a mini-batch consist of one or more windows. ********
********** Where W*t is a window with length WN start from instance at timestamp t, B*t is a mini-batch consists of BN windows starting from window W*t.

3.3 Defination of outlier

Pointwise A data point (instance) is anomalous if this point is distant from other observations according to some specific measurement metrics. This is used in fine-grained anomaly detection tasks, that need to find out every single anomalous instance, e.g. credit card fraud detection, spam email detection.
Window-based A window is anomalous if the window contains one or more anomalous data points. For most of the window-based anomaly detection algorithm, they only calculate the anomaly score of a given window, it's hard and sometimes not necessary to find

out which data points of this window are anomalous.

3.4 LSTMs and autoencoder

Recurrent neural networks(RNNs) are widely used for speech, video recognition and prediction due to its recurrent property that captures the temporal dependency between data in compare with feed forward networks. However, the volume of RNN's memory is limited, and vanishing gradient is also a difficulty by training RNNs. Therefore, the long short-term memory networks (LSTMs) are a kind of reinforced RNN that is able to remember meaningful information in arbitrary time interval. A LSTM network is a recurrent neural network with neurons being LSTM units.
A single LSTM unit can be unfolded over time. The LSTM unit take a data window as input, one data point at a specific time point for each step. Therefore, the LSTM unit extracts useful and drop useless temporal information for the window of data.
Deep LSTM RNNs are built by stacking multiple LSTM layers. Note that LSTM RNNs are already deep architectures in the sense that they can be considered as a feed-forward neural network unrolled in time where each layer shares the same model parameters. It has been argued that deep layers in RNNs allow the network to learn at different time scales over the input. (TrainingandAnalyzingDeepRecurrentNeural Networks)
An autoencoder is an artificial neural network with symmetrical structure. Normally an autoencoder has at least one hidden layer that consists of less neurons than input and output layers. And the basic aim of autoencoders is to reconstruct its own input and learn a lower dimensional representation (encoding) of input data in the hidden layer. Moreover, the autoencoders are also used for anomaly detection by measuring the reconstruction error between inputs and predictions. Normally the component between input layer and hidden layer is called encoder***of the autoencoder, and the symmetrical component between hidden layer and output layer is called decoder ***. For input X, the objective function is to find weight vectors for encoder and decoder to minimize the reconstruction error.

Chapter 4

Proposed model

4.1 Overview / architecture of the framework

The proposed model is a full flow from data stream generation, anomaly detection with autoencoder-based model and online model incremental updating. The first received batched of streaming data are used for decision of model hyperparameters and the initialization. Hyperparameters includes the hidden layer size, batch size, input window length as well as the number of epochs. Once the hyperparameters are learned, an autoencoder will be constructed and initialized with random weights. A subset of the streaming data is used for initial model training (only normal data used for training). Furthermore, the model is used for online anomaly detection, and will be retrained when the retraining condition is triggered.

4.2 Kafka structured data stream generation

We utilize Apache Kafka as the streaming platform. Kafka is a widely used Publish/Subscribe architecture streaming system. It different from classical message queue technique with its fault tolerant, durable and large capacity properties. In the experimental setting, our data source is static databases, Kafka generate real-time data stream pipeline as data source publish records to the specific topic (the data category mechanisms used in Kafka), and furthermore the stream of records will be consumed by different consumers like our analysis model, visualization model etc. This configuration can be easily scaled up to more complicated and demanding real world use cases. Each record in the Kafka stream pipeline is in the form of [Key, Value, Timestamp], where keys are used for positioning and values carry the data record.

4.3 Autoencoder component initialization

i. EncdecAD based architecture (input, output, hidden layer)
The LSTMs-Autoencoder is consist of two LSTM units, one as encoder and the other one as decoder. The encoder inputs are fix length vectors with shape <Batch*num, Step*num,

Elem*num>, where Batch*num is the number of data windows contained in a mini-batch, Step*num is the numbers of data points within each data window, and Elem*num represents the number of data dimensionality. Here, Batch*num and Step*num are learned as hyperparameter in the process beginning. And on the decoder side, it will output exactly the same format data vector for each mini-batch. As introduced in last section, the LSTM unit copies its cell state for itself as one of the cell input at next timestamp. At the last timestamp of encoder, the cell state of LSTM unit is the hidden representation of the input data vector and copied to the decoder unit as initial cell state, so the hidden information can be passed to the decoder. The size of hidden layer representation vector, namely the size of cell state is another hyperparameter need to be learn in the initialization phase. The larger the hidden vector, the more information can be captured during the process, so it is a feature highly depends on the data. Similar to previous study[sutskever et al 2014], we also train the encoder and decoder with time series in reverse order. For example, if the input data fragment are data points from timestamp t1 to t2, then the decoder will predict data point at t2 at first, and then back to t1 step by step, while this trick makes the gradient escarpment between last state of encoder and first state of decoder smaller and easier to learn.
ii. Reconstruction error, anomaly score, parameters The autoencoder tries to reconstruct the input as decoder output with its knowledge of normal data, so if the input data contains anomalies, the reconstruction error will be obviously large due to the lack of anomalous knowledge.

4.4 Online learning for batch-based outliers

However, if we consider using the model for streaming data, the autoencoder might get outdated because of the relative small and simple initialization dataset and concept drift happed along with time. So the update of model is necessary. The main contribution of this paper is the incremental learning setting of the autoencoder model.

Chapter 5

Related works

5.1 Classical machine learning based approaches

5.2 Autoencoder-based anomaly detection approaches

Chapter 6

Related works

There is already pretty much research based on anomaly detection, some of them referred to deal with streaming data.

6.1 Classical machine learning based approaches

As an important component of data mining and machine learning, anomaly detection has been investigated using plenty efficient models. LOF In anomaly detection, the Local Outlier Factor(LOF) is a common distance-based approach. LOF shares some concepts with DBSCAN such as 'core distance' and 'reachability distance', in order to estimate local density. Here, points with substantially lower local density than their neighbors are considered as anomalies. LOF shows competitive performance in many anomaly detection tasks, especially when dealing with data with unevenly density distribution. However, when use get a numerical factor from LOF model, it is actually hard to define a threshold automatically for the judgement of anomaly.
OCSVM Another widely used model is the domain based One-class Support Vector Machine. As an unsupervised one-class classifier, OCSVM takes only normal data as input, and generates a decision surface to separate them from the anomaly states. By analyzing anomalies, the datasets are always bias to the normal part, and anomaly appear only rarely. So, this kind of one-class classifiers avoid making balance between the two classes. Besides, they also take advantage of classical support vector machine, with the help of kernel method, they can also deal with linearly not separable data. Although classical machine learning approaches can handle most of the normal anomaly detection, only few of those approaches could be directly or after some modification used for time series or streaming data, while they ignore the temporal dependency between samples.

6.2 Autoencoder-based anomaly detection approaches

Autoencoders are widely used in text data and speech processing in order to represent or encoder temporally dependent words using RNN based architecture. LearningPhraseRepresentationsusingRNNEncoder–Decoder forStatisticalMachineTranslation Fur-

thermore, anomaly detection borrows similar idea that train an autoencoder with only normal data, and anomaly data as unknown patterns. Then the autoencoder can only reconstruct normal patterns, large reconstruction error indicates anomaly. Electric Power System Anomaly Detection Using Neural Networks (RMSE, 6000epochs) use the vanilla autoencoder to detect abnormal status of the electric power system. As the signal data acquiring from the power network is time dependently, they apply non-overlapped sliding window upon the input data to capture temporal information. After reconstruction, the difference measurement is based on the first norm of autoencoder output and the desired value. And finally, the anomaly score is on the window level granularity that acquiring from applying smoothing and averaging on the distance vector.
Malhotra et al. proposed a LSTM-based encoder-decoder architecture EncDecAD for sensor data anomaly detection. The model feeds sequential input data with a specific window length to the LSTM neurons in the input layer (encoder), as expect exactly same sequence in the output layer (decoder). And the hidden layer will learn a static representation vector of the temporal window sequence. It is proved in the experiments that the LSTM RNN captures the temporal dependency in sensor time series well and is able to detect subtle anomalies from quasi-periodic time-series and not predictable sequences. As scoring method, they assuming that the reconstruction error obeys gaussian distribution, and used this property to estimate the anomaly score.

6.3 Online incremental learning with autoencoders

Zhou et al. proposed an online incremental updating method for denoising autoencoders by modifying the hidden layer neurons in order to deal with the non-stationary streaming data properties. The kern ideal are two steps, merging hidden layer neurons if there are information redundancy, and adding hidden layer neurons to capture new knowledge. Their experimental result shows comparable or better reconstruction result than non-incremental approaches with only few data used during initialization. And they show that their incremental feature learning methods performs more adaptively and robustly to highly non-stationary input distribution.
Dong et al proposed a 2-step anomaly detection mechanism with incremental autoencoders. The implemented the system with ensembled autoencoders in multithreads to leverage parallel computing when large volumes of data arrive. Besides their 2-step mechanism check anomaly in the first step and verify anomaly data with previous and subsequent data (to differ between anomalous state and concept drift) to reduce false-positive rate in anomaly detection. In the experimental results, they show that their model outperforms commonly used tree-based anomaly detection model especially when concept drift presents and speed up the online processing speed with mini-batch learning and online learning in multithreads.

Chapter 7

Related works

There is already pretty much research based on anomaly detection, some of them referred to deal with streaming data.

7.1 Classical machine learning based approaches

As an important component of data mining and machine learning, anomaly detection has been investigated using plenty efficient models. LOF In anomaly detection, the Local Outlier Factor(LOF) is a common distance-based approach. LOF shares some concepts with DBSCAN such as 'core distance' and 'reachability distance', in order to estimate local density. Here, points with substantially lower local density than their neighbors are considered as anomalies. LOF shows competitive performance in many anomaly detection tasks, especially when dealing with data with unevenly density distribution. However, when use get a numerical factor from LOF model, it is actually hard to define a threshold automatically for the judgement of anomaly.
OCSVM Another widely used model is the domain based One-class Support Vector Machine. As an unsupervised one-class classifier, OCSVM takes only normal data as input, and generates a decision surface to separate them from the anomaly states. By analyzing anomalies, the datasets are always bias to the normal part, and anomaly appear only rarely. So, this kind of one-class classifiers avoid making balance between the two classes. Besides, they also take advantage of classical support vector machine, with the help of kernel method, they can also deal with linearly not separable data. Although classical machine learning approaches can handle most of the normal anomaly detection, only few of those approaches could be directly or after some modification used for time series or streaming data, while they ignore the temporal dependency between samples.

7.2 Autoencoder-based anomaly detection approaches

Autoencoders are widely used in text data and speech processing in order to represent or encoder temporally dependent words using RNN based architecture. LearningPhraseRepresentationsusingRNNEncoder–Decoder forStatisticalMachineTranslation Fur-

thermore, anomaly detection borrows similar idea that train an autoencoder with only normal data, and anomaly data as unknown patterns. Then the autoencoder can only reconstruct normal patterns, large reconstruction error indicates anomaly. Electric Power System Anomaly Detection Using Neural Networks (RMSE, 6000epochs) use the vanilla autoencoder to detect abnormal status of the electric power system. As the signal data acquiring from the power network is time dependently, they apply non-overlapped sliding window upon the input data to capture temporal information. After reconstruction, the difference measurement is based on the first norm of autoencoder output and the desired value. And finally, the anomaly score is on the window level granularity that acquiring from applying smoothing and averaging on the distance vector.
Malhotra et al. proposed a LSTM-based encoder-decoder architecture EncDecAD for sensor data anomaly detection. The model feeds sequential input data with a specific window length to the LSTM neurons in the input layer (encoder), as expect exactly same sequence in the output layer (decoder). And the hidden layer will learn a static representation vector of the temporal window sequence. It is proved in the experiments that the LSTM RNN captures the temporal dependency in sensor time series well and is able to detect subtle anomalies from quasi-periodic time-series and not predictable sequences. As scoring method, they assuming that the reconstruction error obeys gaussian distribution, and used this property to estimate the anomaly score.

7.3 Online incremental learning with autoencoders

Zhou et al. proposed an online incremental updating method for denoising autoencoders by modifying the hidden layer neurons in order to deal with the non-stationary streaming data properties. The kern ideal are two steps, merging hidden layer neurons if there are information redundancy, and adding hidden layer neurons to capture new knowledge. Their experimental result shows comparable or better reconstruction result than non-incremental approaches with only few data used during initialization. And they show that their incremental feature learning methods performs more adaptively and robustly to highly non-stationary input distribution.
Dong et al proposed a 2-step anomaly detection mechanism with incremental autoencoders. The implemented the system with ensembled autoencoders in multithreads to leverage parallel computing when large volumes of data arrive. Besides their 2-step mechanism check anomaly in the first step and verify anomaly data with previous and subsequent data (to differ between anomalous state and concept drift) to reduce false-positive rate in anomaly detection. In the experimental results, they show that their model outperforms commonly used tree-based anomaly detection model especially when concept drift presents and speed up the online processing speed with mini-batch learning and online learning in multithreads.

Bibliography

[1] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder–decoder for statistical machine translation. 2014.

[2] Jiwei Li, Minh-Thang Luong, and Dan Jurafsky. A hierarchical neural autoencoder for paragraphs and documents. 2015.

[3] Pankaj Malhotra, Anusha Ramakrishnan, Gaurangi Anand, Lovekesh Vig, Puneet Agarwal, and Gautam Shroff. Lstm-based encoder-decoder for multi-sensor anomaly detection. 2016.

[4] Pankaj Malhotra, Vishnu TV, Lovekesh Vig, Puneet Agarwal, and Gautam Shroff. Timenet: Pre-trained deep recurrent neural network for time series classi

cation. 2017.

[5] Pankaj Malhotra, Lovekesh Vig, Gautam Shroff, and Puneet Agarwal. Long short term memory networks for anomaly detection in time series. 2015.

[6] Marco Martinelli, Enrico Tronci, Giovanni Dipoppa, and Claudio Balducelli2. Electric power system anomaly detection using neural networks. 2004.

[7] Mayu Sakurada and Takehisa Yairi. Anomaly detection using autoencoders with non-linear dimensionality reduction. 2014.

[8] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. 2014.