COMP421(2020T2) - FINAL PROJECT REPORT

Au Tsz Kin

Victoria University Wellington

ABSTRACT

TBA

1. INTRODUCTION

In this report, we explore the unsupervised anomaly detection technique using Long short-term memory(LSTM) in temporal data, which LSTM is a variant of recurrent neural networks (RNNs).

1.1. Anomaly Detection

Anomaly detection is an significant problem that has been studied within a wide variety of research areas and application domains. It refers to the problem of finding patterns or instances in data that do not match to expected behaviour. These deviated patterns or instances can be described as anomalies, outliers, exceptions, aberrations, surprises, peculiarities, discordant observations, or contaminants depending on domain and context [1]. Anomaly detection is useful and significant as the detected anomalies in data often translate to significant, actionable, or even critical information in various application domains. For example, anomaly detection techniques can be used in life-critical systems to detect faults, invasion detection for cyber-security, fraud detection for credit cards, insurance, or other finance-related areas.

1.1.1. Challenges of anomaly detection

The definition of anomalies are different across application domains. In the real world, defining a region of a data that include every variation of normal behaviour is not easy; critical anomalies that lie on the edge of the border can be considered normal, thus cause false-positive errors or vice versa.

Anomaly detection is usually considered an unsupervised problem, because anomalies are rare and often not seen before; when new samples arise or system update required, a new type of anomaly might come along,therefore it is impossible to define every variation of anomaly in advance. Annotating anomalies is often a time-consuming process, and domain experts with sufficient knowledge are required to label anomalies. We are performing anomaly detection on time series data in this report, there will be more challenges involved, QIANG YANG et al. [2] stated that time series data remains it own problem, a large variety of time series used for predictions are contaminated by noise, that makes prediction on short-term and long-term more difficult.

The terms "anomaly/anomalies" were used to refer to the patterns or instances that deviated from normal pattern in dataset used.

1.2. Recurrent Neural Network (RNN)

Recurrent Neural Network(RNN) is a class of artificial neural networks that allow previous outputs to be used as current inputs while having hidden states; therefore having the ability to learn long-term temporal patterns. RNNs are unlike Fully-Connected Networks or Convolution Networks, which lack "memories" when processing sequences or time series of data (e.g. climate data, stock market, medical data). In reality, when we understand the meaning of a sentence, each word is not independent, and we also have its context in our minds. Therefore the basic concept of RNN is to maintain some intermediate state information to help understand the context. RNN models are mostly used in the fields of natural language processing and speech recognition.

Fig.1 shows a vanilla RNN, hidden units grouped with state s at time t, each neuron receive inputs from previous neurons at time step $t-1$, and passing the output to the next neuron at $t+1$, the state vector is preserved during its forward computation.

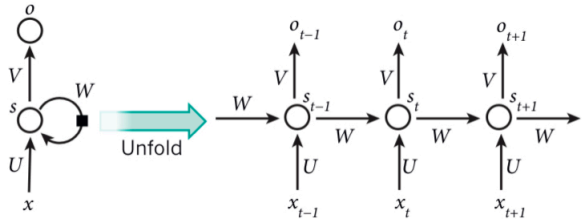


Fig. 1. A vanilla RNN, image used from [3]

1.2.1. Limitation

Training RNNs has proved to be difficult because the back-propagated gradients either grow or shrink at each time

step, so when process sequences that are very long, RNN is prone to problems such as gradient exploding and gradient vanishing[3].

1.3. Long Short-Term Memory (LSTM)

Long Short-Term Memory (LSTM) is a variant of RNN, it was introduced in [4], it has been proved to be a better successor of RNN in learning long-range temporal data, it mainly solved the vanishing gradients problem during long sequence training in RNN. Simply put, compared to vanilla RNNs, LSTM can perform better in in learning long-term dependencies. Other than allow previous outputs to be used as current inputs while having hidden states in RNN. LSTM adds a method that can transmit information with multiple timesteps apart. Think of a conveyor belt/carry track running together when you process the sequences. The information of each node in sequence can be put on the conveyor belt, or taken off from the conveyor belt, of course, you can also update the information on the conveyor belt. In this way, the information long ago is preserved and the loss of information is prevented.

To be exact, the main units of LSTM are introduced in [4] and [5] and the illustrative diagram of a LSTM unit is shown in Fig.2 :

- A central unit called Constant error carousel (CEC), which allows for constant error flow through special self connected units.
- Three gates/multiplicative units that control the flow in CEC;
  - Input gate: prevents CEC from receive irrelevant inputs;
  - Forget gate: the mechanism is introduced in [5] that allow LSTM to "forget" or abandon old and no longer relevant content in memory.
  - Output gate prevents other units receive disturbing information from CEE;

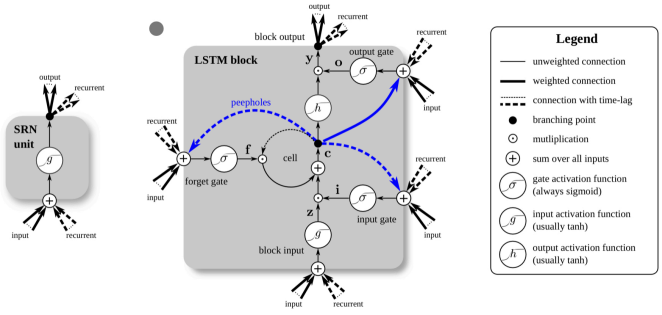


Fig. 2. LSTM with forget gate compare with a simple recurrent network (SRN), image used from [6]

1.4. Selected Paper

The suitable paper selected to match the subject is Akash Singh's master thesis [7]. He is a ML Engineer/Data Scientist; Master in Data Science from KTH Information and Communication Technology. He used an unsupervised approach to accomplish Long short-term memory (LSTM) technique for anomaly detection in time series data since the data are unlabelled.

A RNN model with LSTM units was trained to learn the normal temporal patterns and predict values in future time steps. The anomaly scores is given by modelling prediction error result; therefore determine whether a instance is an anomaly or not. Singh also explored different ways to maintain state in LSTM, and the effect of different number of time steps used on prediction and detection performance.

Three real-world datasets were used in his experiments, the results show that maintaining LSTM state is critical for getting proper result, LSTM RNNs are suitable for general purpose time series modelling and anomaly detection.

1.5. Contribution

TBA

2. METHODS, NETWORK ARCHITECTURE, DATASETS AND FILE DIRECTORIES

Singh's experiments and model was written and implemented using Python programming language with Keras; His algorithm contains two parts. Part one is to train a prediction model by learning normal temporal patterns from dataset, the model can predict future time series. Part two is the anomaly detection, anomaly scores are computed from the prediction errors.

2.1. Prediction Model

The term $lookback$ number and $lookahead$ number were used in the model as input the most recent p value and output q value respectively.

The network contains hidden recurrent layer/layers followed by an output layer. The number of hidden recurrent layers and units are vary for different dataset. Two recurrent layers are fully connected with each other. To avoid overfitting $dropout$ is used between two recurrent layers. The output layer is a fully connected dense NN layer. The number of neuron nodes in the output layer is equal to the $lookahead$ value, with one neuron for each future value predicted. Since the model is used for regression, linear activation is used in the output layer and mean-square error(MSE) as the loss function [7].

The main purpose of the prediction model is to predict multiple time steps ahead, that is, predicting multiple future values; With a $lookahead$ of q at time t the model predicts

the next q future values i.e. t+1, t+2..., t+q. Consider a time series with a scale of 10 minutes. Predicting $lookahead$ value of 6 can give us the behaviour of the time series for next 30 minutes. It is useful for a system to predict possible unusual behaviour, e.g. an extreme value, early alerts can be sent out.

2.2. Anomaly Detection and Data Pre-processing

The prediction errors is used as anomaly indicators, which is the difference between prediction made at time step t-1 and the input value received at current time step t.

The prediction errors from training data are modelled using a Gaussian distribution; the mean and variance are computed using maximum likelihood estimation (MLE).

On new data, the log probability densities (PDs) of errors are calculated and used as anomaly scores: the lower the PD values the greater likelihood of the instance being an anomaly.

A validation set contains both normal instances and anomalies is used to set a threshold on log PD values; it can separate anomalies from normal instances and produce as few false positive errors as possible. Finally, a separate test set containing both normal instances and anomalies is used to evaluate the model.

In order to learn normal time series patterns and optimise prediction performance, only normal data without anomalies is used for training LSTM RNN model. For different dataset, each is divided into four subsets: a training set, N, with only normal values; validation set, V_N, with only normal values; a second validation set, V_A, with normal values and anomalies; and a test set, T , having both normal values and anomalies. There are three main procedures of the LSTM RNN training algorithm:

- 1. Set N with only normal values is used for training prediction model, Bayesian optimization [8] to find the best values for network hyper-parameters. And V_N is used for early stopping to prevent model overfitting.
- 2. Gaussian distribution is used for modelling prediction errors on N. The trained prediction model is applied on V_A. The log PD of errors are calculated from V_A and used as anomaly scores. A threshold is set on the log PD values which separates the possible anomalies from normal values.
- 3. The prediction errors from the test set T is used for the set threshold, therefore it is used as anomaly indicator to identify anomalies from the test set T.

Due to the report length constraint, Singh's full algorithm steps can be viewed in [7].

2.3. Datasets

Instead of application specific datasets or generate artificial datasets; Because it is difficult to judge how well an anomaly detection algorithm would generalize to different type datasets and there is no real-world validity of the actual anomalies and algorithm performance[7]. To guard against these problems, three real-world data sets were used in Singh's experiments: Numentas Machine Temperature Dataset, Power Demand Dataset and ECG Dataset. All three datasets contain real world anomalies annotated by domain experts; And all been used in previous works on anomaly detection.

2.4. Code Directories

The full code of the project is under `/Code`. There are four main parts of the original code that I explored or conduct experiments.

- The configuration of model hyper-parameters is placed under `/Code/configuration/`
- The LSTM model implementation is placed under `Code/models/`
- The LSTM predictor that train model and predict results is `Code/lstm_predictor.py`
- The real-world datasets are stored in `Code/resources/data/`

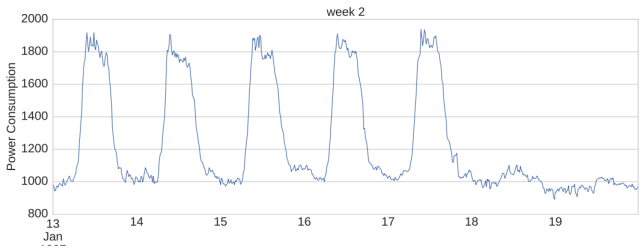
3. THE PROBLEMS AND LIMITATION OF CURRENT SYSTEM

Singh's model are worked quite well for both Power Demand Dataset and ECG Dataset (i.e. electrical activity of the heart). According to Singh's thesis, the model detected all 5 anomalies in the Power Demand test set with the PD threshold of 24 but there is 1 small false positive error. For ECG test set, the model with a threshold of 23 detects all three anomalies with no error. It is because they all have consistent and repetitive patterns, and the LSTM model will be relatively easy to learn these pattern. Fig.3 shows example plots of a normal and an anomalous weekly cycle, the ECG dataset is produce the plot similar to Power Demand Dataset.

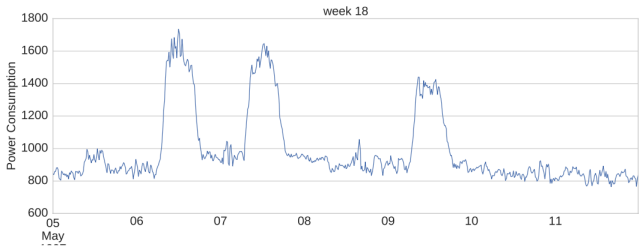
However, there are four anomalies in the machine temperature dataset, the results of the anomaly detection on sets V_A and T are shown in Fig.6 and Fig/8 respectively, orange shaded area in the graph denotes possible anomalies made by the LSTM algorithm. The PD threshold of 11 was necessary to detect the first anomaly in set V_A, but there are quite a few false positives errors. On T the threshold of 11 detected the second anomaly but did not detect the first anomaly, and also incurred a few false positives before the first anomaly.

The result with too many false positives will make the detection unusable. TBA

The machine temperature dataset does not seem to have any repeating pattern and indeed the performance of the Singh's model was found to be insensitive to how the state was maintained as he mentioned on the thesis.



(a) A weekly cycle with high demand on weekends and low demand on weekdays.



(b) A week with anomalies as Monday and Thursday have low demand.

Fig. 3. Power demand normal and anomalous patterns [7]

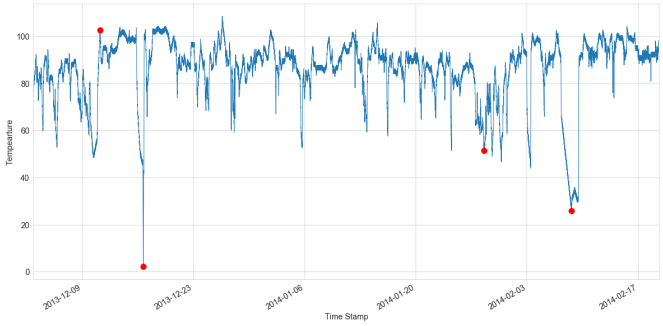


Fig. 4. Normal pattern/values and actual anomalies in the Numentas Machine Temperature Dataset

4. EXPERIMENT RESULT AND CONTRIBUTION

5. CONCLUSION

Fig. 5. Prediction result on machine temperature dataset after modification

Fig. 6. Original validation set results on machine temperature dataset

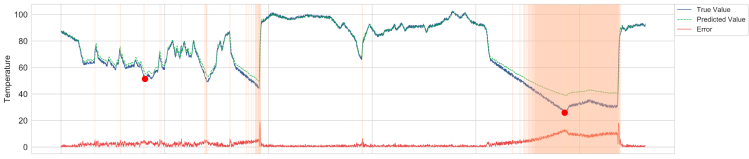


Fig. 7. Validation set results on machine temperature dataset after modification

Fig. 8. Original test set results on machine temperature dataset

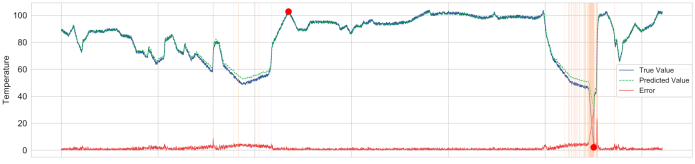


Fig. 9. Test set results on machine temperature dataset after modification

6. REFERENCES

[1] Varun Chandola, Arindam Banerjee, and Vipin Kumar, "Anomaly detection: A survey," *ACM Comput. Surv.*, vol. 41, no. 3, July 2009.

[2] Qiang Yang and Xindong Wu, "10 challenging problems in data mining research," *International Journal of Information Technology & Decision Making*, vol. 5, no. 04, pp. 597–604, 2006.

[3] Yann LeCun, Y. Bengio, and Geoffrey Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–44, 05 2015.

[4] Sepp Hochreiter and Jürgen Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, pp. 1735–80, 12 1997.

[5] Felix A Gers, Jürgen Schmidhuber, and Fred Cummins, "Learning to forget: Continual prediction with lstm," 1999.

[6] Klaus Greff, Rupesh K Srivastava, Jan Koutník, Bas R Steunebrink, and Jürgen Schmidhuber, "Lstm: A search space odyssey," *IEEE transactions on neural networks and learning systems*, vol. 28, no. 10, pp. 2222–2232, 2016.

[7] AKASH SINGH, "Anomaly detection for temporal data using long short-term memory (lstm)," `https://github.com/akash13singh/lstm_anomaly_thesis`, 2017.

[8] Jasper Snoek, Hugo Larochelle, and Ryan P Adams, "Practical bayesian optimization of machine learning algorithms," *Advances in neural information processing systems*, vol. 25, pp. 2951–2959, 2012.