Identification of Irregular Conditions in Brake Sensor Time Series using Machine Learning

Jochen Röth

Hochschule Darmstadt, FBMN/FBI

Introduction

The early detection of anomalies in time series will be a crucial part in preventive maintenance and in the Internet of Things context. The Long Short-Term Memory (LSTM), a variety of Neural Networks, have excelled in the use of sequence data. The strength of the LSTM architecture lies in its ability to remember individual events over very long, unknown periods of time and avoid the Vanishing Gradient Problem that occurs with Feedforward and Recurrent Neural Networks.

The algorithm design provides a semi-supervised approach that trains the selected LSTM architecture on training data without anomalies. The predicted data points on data with anomalies can then be compared with the actual ones. An anomaly score is then calculated by the Maximum Likelihood Estimator based on the residuals. The hyperparameters of the LSTM architecture are determined by Bayesian Optimization.

Theoretical Background

One advantage of neural networks is that they can approximate almost any function. The Universal Approximation Theorem formalizes this as follows:

$$F(x) = \sum_{i=1}^{N} v_i \varphi(w_i^T x + b_i) \tag{1}$$

with

$$|F(x) - f(x)| < \epsilon \tag{2}$$

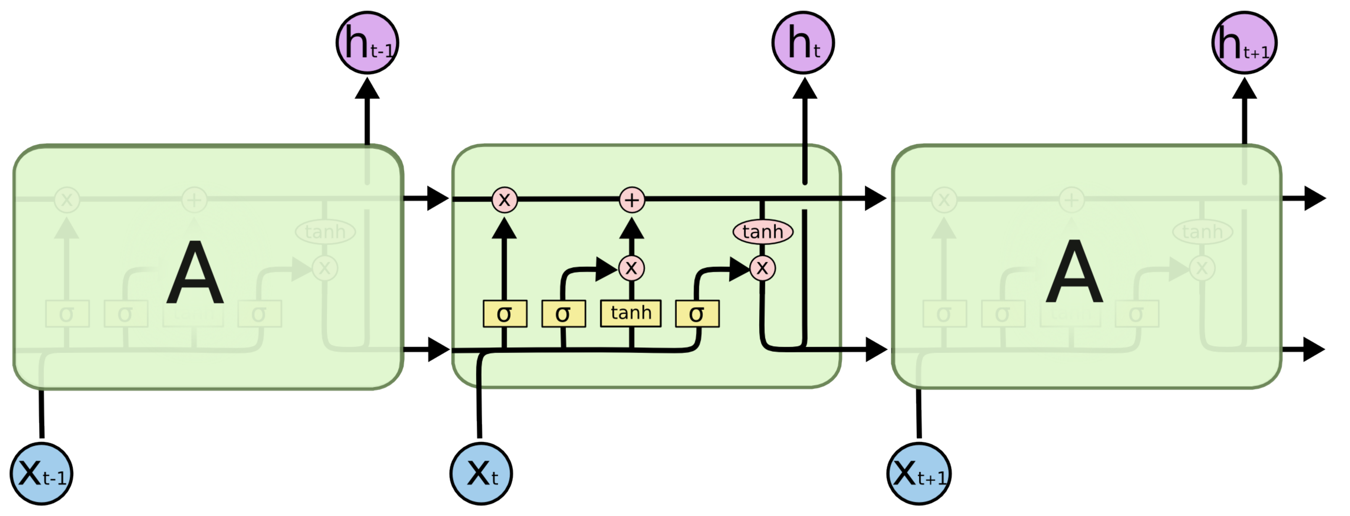

Figure 1:LSTM Cell [1]

The strength of the LSTM architecture lies in its ability to remember individual events over very long, unknown periods of time and avoid the Vanishing Gradient Problem that occurs with Feedforward and Recurrent Neural Networks.

Data

The data set contains the condition monitoring of hydraulic systems. It is based on on multi sensor data of a test bench. The test bench consists of a primary and a secondary cooling-filtration circuit connected by an oil tank. The system repeats periodically constant load cycles for 60 seconds and measures process values such as pressures, flow rates and temperatures. In total, the dataset has 17 input features. During continuous operation, the states of four hydraulic components are modified quantitatively. The four error types cooler and valve condition, pump leakage and hydraulic accumulator are graded with several degrees of severity. In addition, there is one flag which indicates whether the test bench is running stable.[2]

Methodology

The implementation of the LSTM based anomaly detection algorithm is subdivided into the following steps.[3]

1. The training of the algorithm takes place on a training data set T_N. This data set has no anomalies and is suitable for ensuring the highest possible predictive quality for the model.
2. A validation set V_N with no anomalies for early stopping. This dataset should prevent the trained model to overfit on T_N and is used a proxy for the generalization error.
3. A validation set V_A with anomalies and correct values is for model tuning and setting the threshold of anomaly score.
4. A unseen test or evaluation set E_A, with anomalies and correct values, that describes the generalizability of models.

The trained model make predictions on the validation data V_N. The resulting residual vectors are used to determine the parameters with Maximum Likelihood Estimation for a univariate Gaussian Distribution $\mathcal{N}(\mu, \sigma^2)$. Here p^t is used as an anomaly score and small values indicate a higher likelihood of an anomaly.

In order to detect an anomaly, it must now be determined whether a predicted data point represents an anomaly or not. The validation set V_A are used to learn the threshold τ. For this the F_β score can be optimized. F_β takes into account recall and precision.

Hyperparameter Optimization

The hyperparameters in a Neural network are used to define the LSTM architecture. These cannot be learned by training, but must be specified in advance. This is a crucial step and can be done with Bayesian Optimization.

No.	Hyperparmater	Type	Domain	Description
1	$dropout_{L1}$	continuous	(0.0-0.8)	Layer 1 dropout
2	$dropout_{L2}$	continuous	(0.0-0.8)	Layer 2 dropout
3	$dropout_{L3}$	continuous	(0.0-0.8)	Layer 3 dropout
4	$dropout_{L4}$	continuous	(0.0-0.8)	Layer 4 dropout
5	$dropout_{L5}$	continuous	(0.0-0.8)	Layer 5 dropout
6	$learning\_rate$	continuous	(1.0e-7 - 0.1)	Learning rate
7	$units$	discrete	(32,64,128,256,512)	Number of neurons
8	$batch\_size$	discrete	(1,8,16,32,64)	Batch size
9	$look\_back$	discrete	(5,10,15,20,30,40,50)	Window size
10	$epochs$	discrete	(25,50,100,150,200)	Number of epochs
11	$layers$	discrete	(3,4,5,6)	Number of layers

Table 1:Hyperparameter and Domains for Bayesian Optimization

Results

In addition to the optimized hyperparameters determined by Bayesian Optimization, various optimizers and activation functions have been tested. The sigmoid activation functions had a better performance than ReLU. Furthermore the performance of the ADAM optimizer was better than that of RMSprop.

The Bayesian Optimization results in a model architecture with the following hyperparameters selection: One input layer, two stacked LSTM layers. The first LSTM Layer with 32 units, the second with 16 units. After that a dense layer and an output layer. The look back window was set to **5**. The number of epochs were set to at least 20. The lower bound of the learning rate was set to $\mathbf{1.0^{-7}}$ and will be reduced by the factor **0.2** if it hits a learning plateau on the validation data.

The MSE for all 17 input features and one output feature is $\mathbf{3.2824^{-5}}$. The following visualizations show that the validation curves converge already after approximately 7 epochs. According to the central limit theorem, a normal distribution can be assumed for the large residual sample.

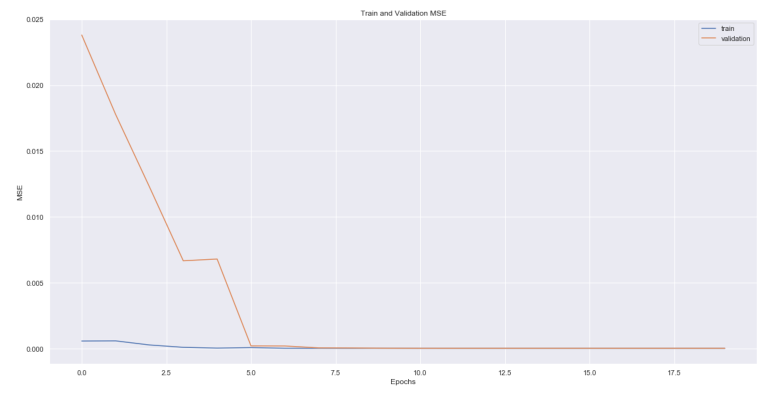

Figure 2:Train and Validation MSE

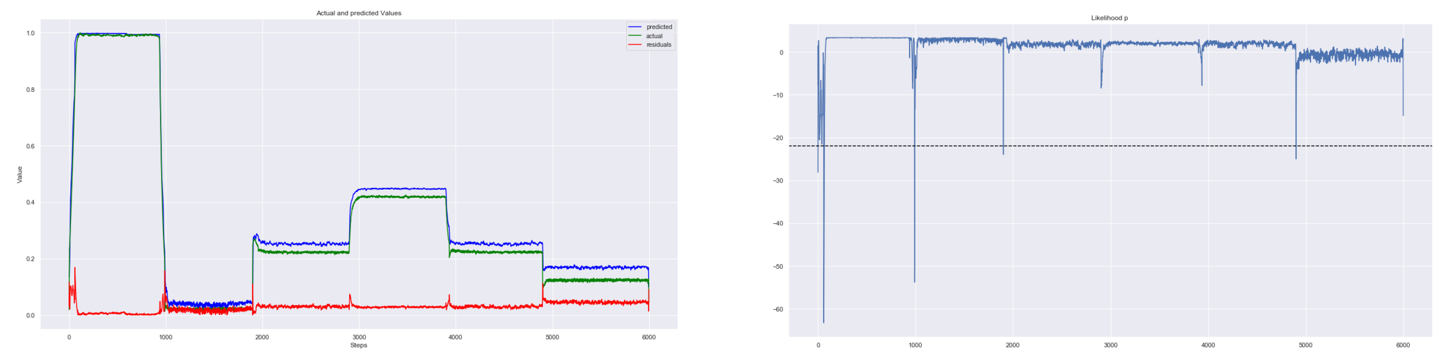

Figure 3:Collective anomalies with anomaly scores

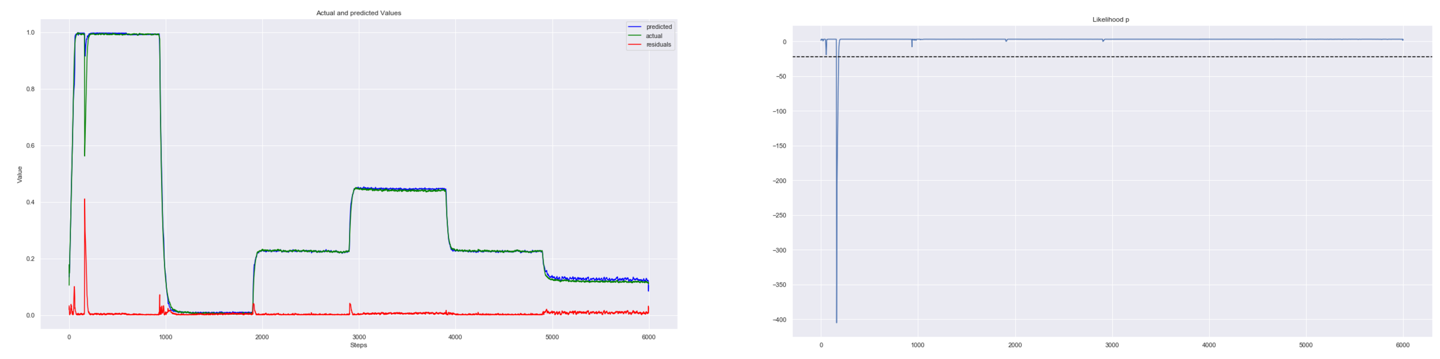

Figure 4:Point anomaly with anomaly scores

The resulting confusion matrix for all evaluated anomalies looks as .

		Actual		
		True	False	Total
Predicted	True	**5**	**5**	**10**
	False	**14**	**9**	**23**
	Total	**19**	**14**	**33**

This results in a $F1-Score$ of **0.3448**.

Conclusion

+ The model was able to learn the feature representation of all 17 input features.
+ The results shown here indicate that the selected algorithm design is able to detect point anomalies.
+ Bayesian Optimization is able to find important and less important hyperparameters.
+ LSTM is good for time series prediction. However, it is less suitable for anomaly detection.
+ Simpler functions can be better approximated without regularization, more complex functions need regularization like dropout.
- The results shown here indicate that the selected algorithm design is not able to detect collective anomalies or a concept drift.
- The look back window size is quite small, large look back windows lead to exploding gradients. Therefore the advantages of an LSTM model doesn't count in auto regression tasks like time series prediction.
- A Bidirectional LSTM did not improve the results.

References

1 C. Olah. Understanding LSTM Networks, http//colah.github.io/posts/2015-08-Understanding-LSTMs/, access:02-02-2019,2015.

2 N. Helwig, E. Pignanelli, and A. Schutze. Condition monitoring of a complex hydraulic system using multivariate statistics. Conference Record - IEEE Instrumentation and Measurement Technology Conference, 2015 - July: 210 - 215, 2015.

3 T. J. O'Shea, T. C. Clancy, and R. W. McGwier. Recurrent Neural Radio Anomaly Detection. 2016.