InPrime: Indonesian Journal of Pure and Applied Mathematics,
Vol. 3, No. 2 (2021), pp. 112 – 121, doi: 10.15408/inprime.v3i2.21879
p-ISSN 2686-5335, e-ISSN: 2716-2478

Web Traffic Anomaly Detection using Stacked Long Short-Term Memory

Fathu Rahman, Taufik Edy Sutanto, and Nina Fitriyati*
Program Studi Matematika Fakultas Sains dan Teknologi
Universitas Islam Negeri Syarif Hidayatullah Jakarta
Email: fathu.rahman17@mhs.uinjkt.ac.id, taufik.sutanto@uinjkt.ac.id, *nina.fitriyati@uinjkt.ac.id

Abstract

An example of anomaly detection is detecting behavioral deviations in internet use. This behavior can be seen from web traffic, which is the amount of data sent and received by people who visit websites. In this study, anomaly detection was carried out using stacked Long Short-Term Memory (LSTM). First, stacked LSTM is used to create forecasting models using training data. Then the error value generated from the prediction on test data is used to perform anomaly detection. We conduct hyperparameter optimization on sliding window parameter. Sliding window is a sub-sequential data of time-series data used as input in the prediction model. The case study was conducted on the real Yahoo Webscope S5 web traffic dataset, consisting of 67 datasets, each of which has three features, namely timestamp, value, and anomaly label. The result shows that the average sensitivity is 0.834 and the average Area Under ROC Curve (AUC) is 0.931. In addition, for some of the data used, the window size selection can affect the sum of the sensitivity and AUC values. In this study, anomaly detection using stacked LSTM is described in detail and can be used for anomaly detection in other similar problems.

Keywords: time-series data; sliding window; web traffic; window size.

Abstrak

Salah satu contoh deteksi anomali adalah mendeteksi penyimpangan perilaku dalam penggunaan internet. Perilaku ini dapat dilihat dari web traffic*, yaitu jumlah data yang dikirim dan diterima oleh orang-orang yang mengunjungi situs web. Pada penelitian ini, deteksi anomali dilakukan menggunakan* Long Short-Term Mermory (LSTM) *bertumpuk. Pertama, LSTM bertumpuk digunakan untuk membuat model peramalan menggunakan data latih. Kemudian nilai* error *yang dihasilkan dari prediksi pada data uji digunakan untuk melakukan deteksi anomali. Kami melakukan optimasi* hyperparameter *pada parameter* sliding window. Sliding window *adalah data sub-sekuensial dari data runtun waktu yang digunakan sebagai* input *pada model prediksi. Studi kasus dilakukan pada dataset* web traffic Yahoo Webscope S5 *yang terdiri dari 67 dataset yang masing-masing memiliki tiga fitur yaitu* timestamp*,* value*, dan* anomaly label*. Hasil menunjukkan bahwa rata-rata sensitivitas sebesar 0.834 dan rata-rata* Area Under ROC Curve (AUC) *sebesar 0.931. Selain itu, untuk beberapa data yang digunakan, pemilihan* window size *dapat mempengaruhi jumlah dari nilai sensitivitas dan AUC. Pada penelitian ini, deteksi anomali menggunakan LSTM bertumpuk dijelaskan secara rinci dan dapat digunakan untuk deteksi anomali pada permasalahan lainnya yang serupa.*

Kata kunci: *data runtun waktu*; sliding window; web traffic; window size.

1. INTRODUCTION

The Along with the internet and computer technology, internet networks significantly impact society and economic development [1][2]. Web servers can do exchange large amounts of information, and various services are provided [3]. However, with the utility increase of internet service, the

Submitted July 29th, 2021, Revised September 17th, 2021, Accepted for publication September 21st, 2021.

potential for crime through the internet network has become more diverse. Various attacks on the internet network can cause serious damage to the operation of web services and cause economic and social losses [4], [5].

One of the efforts in establishing an internet network security system is anomaly detection. Anomaly detection is a process of finding deviant behavior in internet usage [6]. Internet usage behavior can be seen from web traffic, namely the amount of data sent and received by people who visit websites. The basic principle of anomaly detection is to define normal user behavior well so that attack behavior on the internet network can be identified correctly [7].

Radford et al. [8] conducted an anomaly detection study on a public dataset from the University of New Brunswick's Canadian Institute for Cybersecurity (CIC) and the Information Security Center of Excellence (ISCX) using Long Short Term Memory (LSTM). They resulted in the Area Under ROC Curve (AUC) of 0.84. In the same year, Kim and Cho [9] conducted an anomaly detection study on the Yahoo Webscope S5 web traffic dataset using a convolutional neural network and LSTM (C-LSTM), resulting in a sensitivity of 0.897. Braei and Wagner [10] surveyed several anomaly detection methods on the Yahoo Webscope S5 web traffic dataset and produced an AUC of 0.8121 using a stacked LSTM.

LSTM is an effective model for studying sequential data with a long-term relationship and represents the relationship between current events and past events [11]. Therefore, in this study, anomaly detection was carried out on web traffic data using a stacked LSTM by developing the method used in Braei and Wagner's research [10]. We use several window sizes as hyperparameter optimization on the sliding window as our generic contribution in this work. For making forecasting models, sliding window is a sub-sequence of time series data that is used as an input model to predict the value of the next time series data. The specified sub-sequence length is called window size. Sensitivity and AUC values are used to evaluate the anomaly detection results and determine the effect of window size selection on the sum of these two values.

2. METHOD

The LSTM is one type of Recurrent Neural Network (RNN) that can make a network able to maintain its long-term dependence between data at a certain time and some data at previous times. The main component of the LSTM layer is a unit called a memory block. In each unit, there are three gates, namely input gates, output gates, and forget gates. These gates function to create, read, and reset information on the cell state. Cell state in LSTM is a component that brings information from one unit to another unit [12]. Figure 1 illustrates the working principle of the LTSM unit at each timestamp t.

Each gate and cell state can be formulated as follows:

$$f_t = \sigma_1\left(W_{xf}x_t + W_{hf}h_{t-1} + b_f\right), \quad (1)$$

$$i_t = \sigma_2(W_{xi}x_t + W_{hi}h_{t-1} + b_i), \quad (2)$$

$$\tilde{c}_t = \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c), \quad (3)$$

$$c_t = f_t c_{t-1} + i_t \tilde{c}_t, \quad (4)$$

$$o_t = \sigma_3(W_{xo}x_t + W_{ho}h_{t-1} + b_o), \quad (5)$$

$$h_t = o_t \tanh(c_t), \tag{6}$$

where W and b are weights and biases, x are inputs, f, i, and o are forget gates, input gates, and output gates, $\tilde{c}$ is a candidate of cell state, c is cell state, and h is hidden output [11].

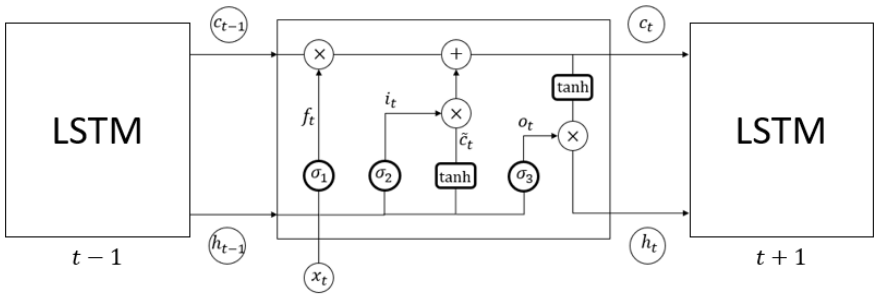

Figure 1. Structure and working principle at LSTM unit [12].

The information carried by the previous cell state c_{t-1} will be decided to be forgotten or passed by multiplying the values of c_{t-1} and the forget gate f_t by the inputs x_t and h_{t-1}. It can do this because the sigmoid function on the forget gate will produce a value in the interval [0,1]. If the value of the forget gate is equal to 0, the result is that the information from $c_{t-1}$is not passed. On the other hand, if the value of the forget gate is more than 0, then c_{t-1} will be forward the information.

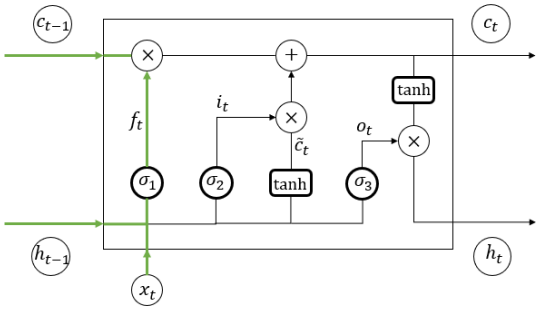

Figure 2. The cell state process c_{t-1} through forget gate.

Then the new information from x_t and h_{t-1} will be converted into candidate cell state $\tilde{c}_t$ using activation function tanh. The new information will be decided whether to use or not bypassing the input gate i_t like how the forget gate works (figure 3). After that, the value of c_{t-1} is updated by adding up the information c_{t-1} that has passed the forget gate and the new information x_t and h_{t-1} that have passed the input gate (figure 4).

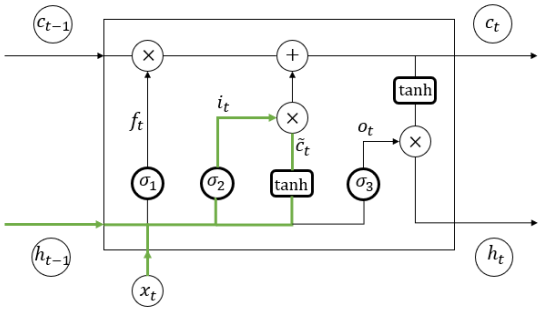

Figure 3. The process of forming candidate cell state $\tilde{\boldsymbol{c}}_{\boldsymbol{t}}$.

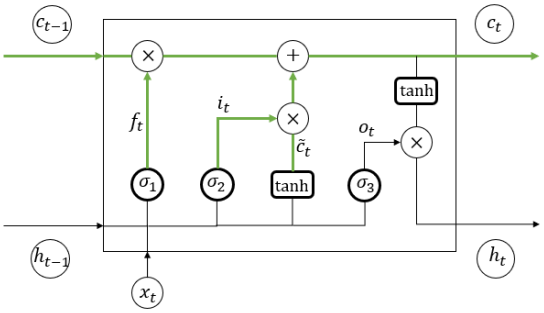

Figure 4. The process of updating information on cell state $\boldsymbol{c}_{\boldsymbol{t}}$.

Finally, the hidden output value h_t is calculated by multiplying the output gate value and $\tanh(c_t)$. This step illustrates in Figure 5.

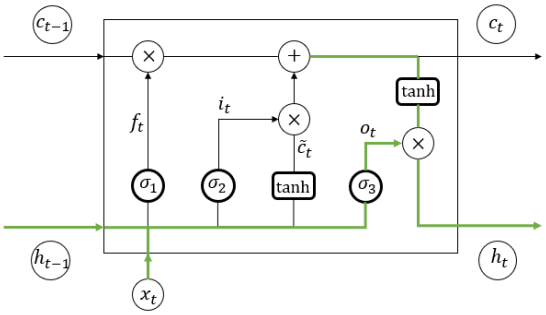

Figure 5. The process of calculating the hidden output value $\boldsymbol{h}_{\boldsymbol{t}}$.

To start the LSTM model, it is necessary to determine the initial value of h_0, c_0, and the weight of W on each LSTM gate and the output layer randomly from a small value [13]. Figure 6 illustrates how to make the LSTM model.

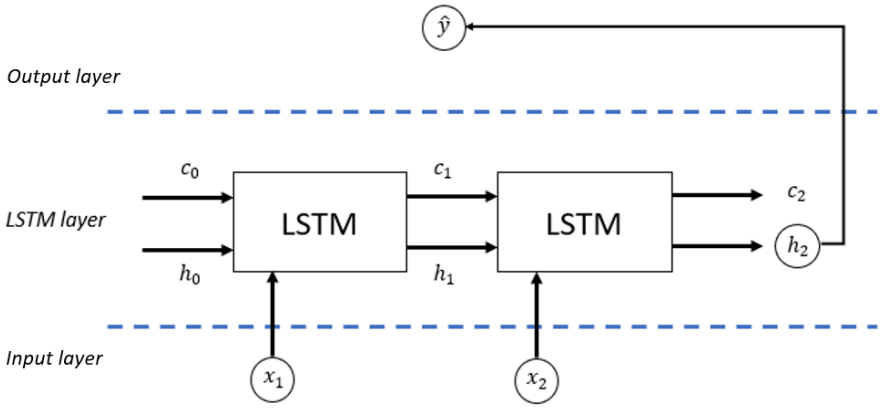

Figure 6. The example of LSTM model.

The LSTM process will be carried out by using equations 1 to 6 with the following steps:

1. In the first LSTM unit, calculate the values of f_1, i_1, $\tilde{c}_1$, c_1, o_1, and h_1 with the input (x_1, h_0, c_0).
2. In the second LSTM unit, calculate the values f_2, i_2, $\tilde{c}_2$, c_2, o_2, and h_2 with the input (x_2, h_1, c_1).
3. Predict the value of $\hat{x}_i$ using equation $\hat{x}_i = f(\boldsymbol{w} \odot \boldsymbol{x} + b)$ and the linear activation function $a(z) = z$.
4. Calculate loss function value using the mean squared error (MSE) and update all values in W using gradient descent.
5. Repeat step 1 to step 4 until the MSE reaches the optimal minimum value.

In performing anomaly detection, LSTM is used to form a forecasting model. Let $X = \{x_1, x_2, \dots, x_N\}$ be time series data, where N is the number of data. In the forecasting model, it is necessary to specify a window size w to apply a sliding window to the data, with $w < N$. This means that w consecutive time-series data is used as input to the LSTM to predict the next x_i value. Let ψ be forecasting function using LSTM, then the value of x_i can be predicted by the following formula:

$$\begin{gathered} \psi: \mathbb{R}^w \to \mathbb{R}, \\ \hat{x}_i = \psi\big((x_{i-w}, x_{i-w+1}, \dots, x_{i-1})\big). \end{gathered} \tag{7}$$

To form a forecast model, there is no specific value for the window size [11]. We use several window sizes i.e. 10, 20, 30, 40, 50, 60, 70, 80, 90, and 100 to get the best evaluation results. Referring to Malhotra et al. [14], the researcher used a stacked LSTM containing two hidden LSTM layers. Stacked LSTM is an architecture of a neural network that consists of several layers of LSTM in its hidden layer. Figure 7 illustrates an architectural example of a stacked LSTM.

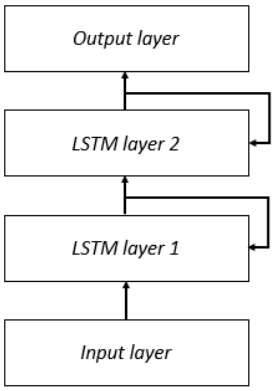

Figure 7. Architectural Example of a Stacked LSTM.

Loop lines in the LSTM layer indicate the process as described in detail in Figures 1 to 6. All hidden output h in the lower LSTM layer (LSTM layer 1) will be used as input for the LSTM layer above (LSTM layer 2).

After forming a forecasting model using stacked LSTM on the training data, the model predicts the test data and calculates the error value. These all the error values are a Gaussian distribution $\mathcal{N} = \mathcal{N}(\mu, \sigma)$. Parameters μ and σ are estimated using the mean and standard deviation of the error values. Based on this distribution, we calculate the probability density function $p_i = p(e_i)_{e^{(i)} \sim \mathcal{N}(\mu,\sigma)}$ from data x_i for any error value $e_i = |x_i - \hat{x}_i|$. Define the anomaly detection binary function as

$$\phi_{binary}(x_i) \mapsto \begin{cases} anomaly, & if\ p_i \leq \delta \\ normal, & other \end{cases}, \tag{8}$$

where $\delta \in \mathbb{R}$ is anomaly boundary.

3. EXPERIMENT

This study uses the real Yahoo Webscope S5 web traffic dataset, consisting of 67 web traffic time-series data containing about 1400 timestamps. Anomaly detection was carried out on each of these data using the Python 3 program with the help of modules from Tensorflow [15] and Keras [16]. In the preprocessing, the data is normalized, and a sliding window is applied. The data is divided into training data and test data with ratios of 70% and 30%. Training data is used to create a forecasting model using the stacked LSTM and 10% of the training data is used to validate the model. Then the forecasting model is used to predict the value in the test data to detect anomaly as described previously. The anomaly detection process is only carried out on data containing anomalies in the test data.

4. RESULT AND DISCUSSION

After normalization and sliding window, each data was obtained with $N - w$ data rows where N is the number of rows of data before sliding window and w is the window size used. The data also has a column of sequential data ($\boldsymbol{x_i^w}$), the value data to be predicted based on the sequential data (x_i), and the category of whether the value is an anomaly or not (Anomaly). Table 1 is the results of the sliding window performed on Data 1 with $w = 10$.

Table 1. The example of sliding window results.

$\boldsymbol{x_i^w}$	$\boldsymbol{x_i}$	**Anomaly**
[0, 0.117, 0.219, 0.287, 0.224, 0.115, 0.107, 0.087, 0.169, 0.108]	0.123	0
[0.117, 0.219, 0.287, 0.224, 0.115, 0.107, 0.087, 0.169, 0.108, 0.123]	0	0
[0.219, 0.287, 0.224, 0.115, 0.107, 0.087, 0.169, 0.108, 0.123, 0]	0.039	0
[0.287, 0.224, 0.115, 0.107, 0.087, 0.169, 0.108, 0.123, 0, 0.039]	0.023	0
[0.224, 0.115, 0.107, 0.087, 0.169, 0.108 0.123, 0, 0.039, 0.023]	0.08	0
[0.115, 0.107, 0.087, 0.169, 0.108, 0.123, 0, 0.039, 0.023, 0.08]	0.112	0
[0.107, 0.087, 0.169, 0.108, 0.123, 0, 0.039, 0.023, 0.08, 0.112]	0.147	0
[0.087, 0.169, 0.108, 0.123, 0, 0.039, 0.023, 0.08, 0.112, 0.147]	0.093	0
⋮	⋮	⋮
[0.26, 0.045, 0.042, 0.103, 0.241, 0.223, 0.203, 0.175, 0.251, 0.206]	0.142	0

The training data obtained from the preprocessing results form a forecasting model with a stacked LSTM. The model is then used to predict the value $(\hat{x}_i)$ in the test data. Table 2 is the prediction results (x_i) carried out on the test data from Data 1 with $w = 10$.

Table 2. Test Data 1 and the prediction results.

$\boldsymbol{x_i}$	$\boldsymbol{\hat{x}_i}$
0	0.053
0.026	0.056
0.055	0.065
0.159	0.078
0.127	0.099
0.109	0.115
0.085	0.122
0.043	0.116
⋮	⋮
0.142	0.168

The error data calculated from the prediction results are then used to detect anomalies. In the following table, several values of the probability density function ($p(e_i)_{e^{(i)}\sim\mathcal{N}(\mu,\sigma)}$), observed anomalies (anomalies) and the results of anomaly predictions (anomaly predictions) would be presented in the table 3. The anomaly detection results are then evaluated by calculating each data's sensitivity and AUC values for several window sizes. Figure 8 presents a sum of the sensitivity values and AUC to the window size of the sample data used.

Table 3. The result of anomaly detection for the test Data 1

$p(e_t)_{e^{(t)}\sim\mathcal{N}(\mu,\sigma)}$	Anomaly	Anomaly Prediction
4.771999	0	0
4.019029	0	0
3.433676	0	0
5.024857	0	0
4.703183	0	0
3.571779	0	0
4.436188	0	0
5.162097	0	0
5.084145	0	0
⋮	⋮	⋮
3.829487	0	0

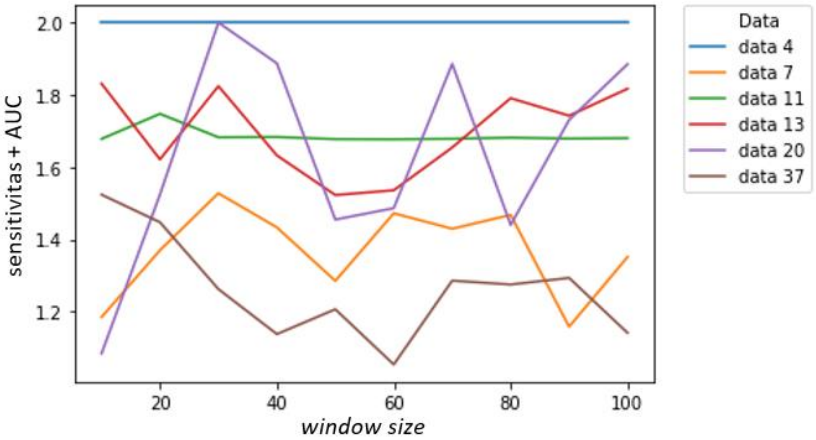

Figure 8. The line chart for window size versus the sum of sensitivity and AUC values.

Based on Figure 8, we can see that window size selection affects the number of sensitivity and AUC values for some data. From these results, we conclude that it is necessary to do window size hyperparameter optimization to get the optimal value of window size. The largest number of sensitivity values and AUC determine the best anomaly detection. The average sensitivity and AUC of all the best evaluation results are 83.3892% and 93.1228%.

5. CONCLUSION

This paper succeeds in evaluating the anomaly detection performance on the real Yahoo Webscope S5 web traffic dataset. The result shows that the average sensitivity and AUC are 83.3892% and 93.1228%, respectively. The model can detect the anomaly about 83.3892% correctly from all anomalies in the observed data and has a probability of about 93.1228% to predict the probability density function from the data indicated anomaly is smaller than the observed normal data. In addition, for some data, the selection of window size can affect the sum of the sensitivity and AUC values. Further improvements are possible by optimizing the hyperparameters used in the LSTM architecture which might increase the sensitivity and AUC values.

REFERENCES

[1] K. H. Kim and S. B. Cho, "Modular bayesian networks with low-power wearable sensors for recognizing eating activities," *Sensors (Switzerland)*, vol. 17, no. 12, 2017, doi: 10.3390/s17122877.

[2] C. A. Ronao and S. B. Cho, "Human activity recognition with smartphone sensors using deep learning neural networks," *Expert Syst. Appl.*, vol. 59, pp. 235–244, 2016, doi: 10.1016/j.eswa.2016.04.032.

[3] S. Y. Huang and Y. N. Huang, "Network traffic anomaly detection based on growing hierarchical SOM," in *Proceedings of the International Conference on Dependable Systems and Networks*, 2013, doi: 10.1109/DSN.2013.6575338.

[4] M. Ahmed, A. Naser Mahmood, and J. Hu, "A survey of network anomaly detection techniques," *Journal of Network and Computer Applications*, vol. 60. pp. 19–31, 2016, doi: 10.1016/j.jnca.2015.11.016.

[5] C. A. Ronao and S. B. Cho, "Anomalous query access detection in RBAC-administered databases with random forest and PCA," *Inf. Sci. (Ny).*, vol. 369, pp. 238–250, 2016, doi: 10.1016/j.ins.2016.06.038.

[6] P. García-Teodoro, J. Díaz-Verdejo, G. Maciá-Fernández, and E. Vázquez, "Anomaly-based network intrusion detection: Techniques, systems and challenges," *Comput. Secur.*, 2009, doi: 10.1016/j.cose.2008.08.003.

[7] C. Torrano-Gimenez, A. Perez-Villegas, and G. Alvarez, "An Anomaly-Based Approach for Intrusion Detection in Web Traffic," *J. Inf. Assur. Secur.*, 2010.

[8] B. J. Radford, L. M. Apolonio, A. J. Trias, and J. A. Simpson, "Network traffic anomaly detection using recurrent neural networks," *arXiv*. 2018.

[9] T. Y. Kim and S. B. Cho, "Web traffic anomaly detection using C-LSTM neural networks," *Expert Syst. Appl.*, vol. 106, pp. 66–76, 2018, doi: 10.1016/j.eswa.2018.04.004.

[10] M. Braei and S. Wagner, "Anomaly detection in univariate time-series: A survey on the state-of-the-art," *arXiv*. 2020.

[11] H. D. Nguyen, K. P. Tran, S. Thomassey, and M. Hamad, "Forecasting and Anomaly Detection approaches using LSTM and LSTM Autoencoder techniques with the applications in supply chain management," *Int. J. Inf. Manage.*, vol. 57, 2021, doi: 10.1016/j.ijinfomgt.2020.102282.

[12] D. Singh *et al.*, "Human activity recognition using recurrent neural networks," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2017, vol. 10410 LNCS, pp. 267–274, doi: 10.1007/978-3-319-66808-6_18.

[13] C. C. Aggarwal, *Neural Networks and Deep Learning*. Springer International Publishing, 2018.

[14] P. Malhotra, L. Vig, G. Shroff, and P. Agarwal, "Long Short Term Memory networks for anomaly detection in time series," in *23rd European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, ESANN 2015 - Proceedings*, 2015.

[15] E. B. Martín Abadi, Ashish Agarwal, Paul Barham *et al.*, "TensorFlow: Large-scale machine learning on heterogeneous systems." 2015, [Online]. Available: tensorflow.org.

[16] F. Chollet, "keras." GitHub, 2015, [Online]. Available: https://github.com/fchollet/keras%7D%7D.