Study of Autoencoder Neural Networks for Anomaly Detection in Connected Buildings

Adrien Legrand*, Brad Niepceron†, Alain Cournier‡ and Harold Trannois§
Université de Picardie Jule Vernes
France
Email: *adrien.legrand@zenika.com, †niepceron.brad@gmail.com, ‡harold.trannois@u-picardie.fr,
§alain.cournier@u-picardie.fr

***Abstract*—Nowadays, buildings are equipped with safety devices, to prevent and identify critical events, such as fires using smoke detection devices or intrusions using alarm systems. The expansion of the Internet of Things (IoT) involves new perspectives, especially in terms of data variety and quantity from which critical event detection can benefit.**

These new perspectives have made it feasible to develop data-driven methods to detect problematic situations that may occur in connected buildings using anomaly detection without necessarily setting up a specific system for each of them. Recent research works in deep learning have shown that autoencoder neural networks have a great capability in analyzing high-dimensional data and can be used for the purpose of anomaly detection in a supervised or unsupervised way, as anomaly labeled data are typically not available.

In this paper, we investigate and compare the capability of different kinds of autoencoder neural networks in the detection of anomalies in a large scale Smart Home dataset and we propose a method for evaluating partially labeled anomaly detection models.

***Index Terms*—Internet of Things, Anomaly detection, Autoencoder, Recurrent Neural Network, Convolutional Neural Network.**

I. Introduction

In recent years, the significant increase in the number of connected objects permitted the rapid emergence of the Internet of Things (IoT). The concept of IoT has been introduced by Kevin Ashton [1] in 1999, and refers to the connection of objects by radio-identification technology (RFID). Since then, IoT and connected objects are in continuous evolution. The IEEE IoT Initiative [2], defined the Future Internet as being a network of networks, including physical objects networks, allowing all kind of new services.

This growth involves the emergence of a large number of connected objects and with them, a massive amount of data. This enormous amount of data can thus be used for the development of data-driven methods to answer to all the new challenges of the IoT.

We attempt to take advantage of this proliferation of devices to analyze the data they retrieve in order to develop a method for the detection of unusual situations in the context of connected buildings. Detection of unusual activities in connected buildings has become a great concerns with the growth of the IoT over the past few years. In the environment of a connected building, anomaly detection is used to treat and analyze data coming from various sensors present in the connected devices.

This detection problem refers to the task of pointing out rare observations in a dataset [3]. These observations are called anomalies, outliers, novelties [4] or discords [5] and can be seen as objects or subsequences of a data whose occurrence frequency in the dataset is low (unusual) or zero (novel) in a dataset. Anomalies can be induced for several reasons such as errors due to malfunctioning of sensing systems or malicious activities. The high interest behind anomaly detection lies in the fact that in a wide range of application domains, anomalies can carry critical and significant informations. In the case of anomaly detection in connected buildings, it can lead to an optimization of the costs and overall better safety.

In this paper, we investigate the use of deep learning based anomaly detection algorithms by analyzing an the REFIT dataset [6]. The remaining sections of this paper are organized as follows. The problem statement and the motivation behind our researches are described in Section II. Section III provides information about the background knowledge and presents related works. The details of our research methodology, our experiments and results are provided in Section IV. Finally, conclusions are drawn in Section V.

II. Motivating scenario and problem statement

Let us consider a scenario where a user (owner, inhabitant, manager...) wants to monitor the state (heating system, isolation, etc.) of a connected building. In this context, the objective is for the user to be able to detect anomalies in an IoT system. These anomalies can be characteristic of sudden structural problems in the building, poorly functioning appliances (malfunctioning heaters, power leak...), sudden changes in users' habits (behavior issues, intrusions...) etc... Examples of real life anomalies are shown in III-A. Detecting such scenarios allows the user to react to these anomalies and limit any potential damages that may come from it.

Machine learning techniques have been widely used to deal with such problems. However typical machine learning (ML) architectures for the IoT are designed for specific use cases or requires manually created rules [7] [8] [9]. As a specific anomaly is considered as a use case, a specific model is needed for each of them. Such a limitation greatly hamper their reuse, evolution and maintenance, thus leading to independent development and low integration into IoT solutions. Moreover, buildings are subject to important heterogeneity in the way

978-1-5386-8509-9/18/$31.00 ©2018 IEEE

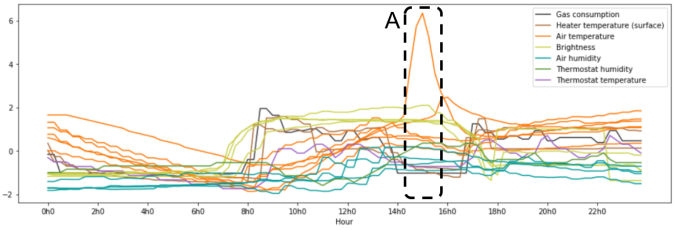

Fig. 1. Temperature peak (values are scaled) : all air temperature sensors are correlated but, around 15h, one sensor stops to be correlated with the others (contextual anomaly) and show a large unusual value (peak / collective anomaly) for 1h30. The anomalous window is in block A.

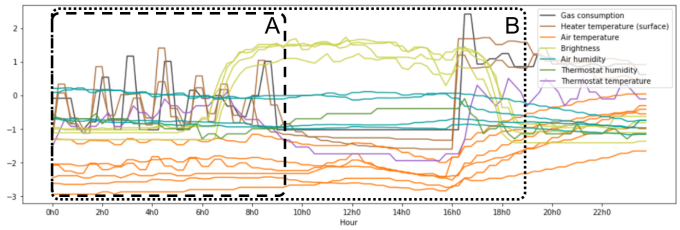

Fig. 2. Low temperature despite heaters functionning (values are scaled) : This window shows an activity of the heaters and gas consumption during the night and the temperature remains at its lowest point in the dataset (peak anomaly and contextual anomaly). Peak anomaly is in block B and contextual anomaly (low temperature despite functionning heaters) is in block A.

their sensors are deployed (number, nature and location of sensors, etc...).

Additionally, the majority of work in the field of anomaly detection applied to connected buildings is, in our knowledge, not concerned by all types of anomalies described in III-A. In this paper, we aim to compare two well-known methods, although underutilized in the field of connected buildings, to detect all types of anomalies.

III. Related work

In this section, we first introduce the background knowledge needed to fully understand our experimentation as well as our choices in the types of models to compare. We describe the different kinds of anomalies that may appear in connected buildings, then we describe the architectures adapted to detect the said anomalies that we have selected based on works present in the state of the art.

A. Anomalies

Anomalies are considered as patterns in data that do not conform to a predefined notion of a normal behavior. They can appear in different types :

- Peak / Point anomalies - This is the simplest type of anomaly, they refer to isolated points from a data instance's feature (in our case, sensor) with a value very different from other points of that feature. In the context of connected building sensor values, such anomalies are very unlikely to appear. Indeed, given the inertia present in the majority of sensors, a peak type anomaly will be very unlikely to be represented by a single and isolated data instance, except with very noisy sensor values (example taken from the REFIT dataset in Fig.1).
- Contextual anomalies - Here, the anomalies are characterized as abnormal individual instances in a specific context. This notion of context has to be formerly induced by the structure of the data and has to be part of the problem formulation [3]. An anomalous contextual behavior can be identified by the behavior of a data instance in a particular context. A data instance might be a contextual anomaly for a given context but appear as a normal behavior in a different one. Taking a dataset representing the evolution of an indoor temperature throughout a year, a contextual anomaly can be for example identified by an abnormal high value in winter that could be totally normal in summer. A contextual anomaly can be seen in Fig.2 showing an example of the REFIT dataset.
- Collective anomalies - If a group of related data instances is anomalous with respect to the rest of the dataset, the group is considered as a collective anomaly. Though, a single instance of the abnormal group may not be an anomaly. In our case, a peak anomaly is likely to be reflected by both values and variations of sensors.

B. Autoencoder neural networks

Autoencoder networks have been developed as solutions for a great variety of problems, such as data denoising or also anomaly detection. While classic neural networks are typically used for classification or regression problems, thus producing an output usually having a dimension different from the input, the main purpose of an autoencoder is to output a reconstruction of the input data [10].

The key point of an autoencoder is the dimension reduction taking place in it. Indeed, an autoencoder reconstructing perfectly its input wouldn't have much interest. Combining dimension reduction and neural network algorithms allow the model to extract structural information and relationships between the features of the input which differs from usual dimension reduction algorithm, such as Principal component analysis (PCA) [11]. Indeed, the nature of neural networks and their activation functions allow to do it in a non-linear way. Non-linear dimension reduction is to be used in our case, since dependencies between sensor values of a connected building may be subject to a great number of factors, different inertias and noises.

Over training, an autoencoder neural network learns to approximate two functions. We consider here a dataset $\mathbf{X} = (\mathbf{x}, \ldots, \mathbf{x}_j, \ldots, \mathbf{x}_n)$ with $\mathbf{x}_j = (x_{j,1}, x_{j,2}, \ldots, x_{j,d}) \in \mathrm{R}^d$. The encoding function f($\mathbf{x}$), for any $\mathbf{x} \in \mathbf{X}$, execute the dimension reduction and compress the data to a vector $\mathbf{h}$ with $\mathbf{h} \in \mathrm{R}^n$, n¡d. The decoding function g($\mathbf{h}$) recreate an approximation of the original input $\mathbf{x}' \in \mathrm{R}^d$.

The principal principle in using autoencoders for anomaly detection is to measure the distance between $\mathbf{x}$ and $\mathbf{x}'$ and set a threshold beyond which the distances and input vectors associated with them would be considered as anomalies. Another approach would be to compute a score based on this distance that would give a probability for an input to be an anomaly.

Though, this basic feed-forward type of autoencoder isn't suited to capture temporal dependency between values in time series data and spatial dependency in image data [12]. Anomalies such as collective anomalies wouldn't be evaluated. This leads to a need of combining the ability of other deep learning algorithms to autoencoders. The following described models are the ones we compare in our experiment.

1) Convolutionnal autoencoders: Autoencoders can benefit from the potential of convolutional neural networks (CNNs) [13]. CNNs belong to a class of feed-forward artificial neural networks that use convolution and pooling as main operations to extract structural dependencies in the data. They have mostly been used for 2D image classification but can also be applied for processing of 1D data [14]. This motivated the development of convolutional autoencoders to reconstruct time series using the spatial dependency extraction feature of a CNN and provide more reliable performances than conventional autoencoders. In practice, the input of a recurrent autoencoder is not a simple vector but a window of vectors corresponding to a "snapshot". A comparison between classic autoencoders and 1D CAEs done in the context of connected building [15] shown that CAE are more efficient in that domain.

2) Recurrent autoencoders: Cyclic connections offered by recurrent neural networks (RNN) make them a very powerful method to process sequential data while retrieving temporal dependencies [16]. In the last few years, incredible successes have been made applying them to a variety of topics such as speech recognition [17], acoustic captioning [18], video recognition [19]... They have the ability to use previous time step activations as inputs and store informations from these input sequences in their internal state. Storing temporal dependencies is the core of this network efficiency and combining it with an autoencoder allows to achieve high-quality reconstruction of time-series data. The input of a recurrent autoencoder is not a simple vector but a window of vectors corresponding to consecutive data instances in the time series, as shown in Fig. 3. Recent works about using Long short-term memory (LSTM) based autoencoder on sensor issued multi variate time series have shown their efficiency in that domain [20]. We have also used LSTM cells in the recurrent autoencoders we tested, since they presents numerous enhancements that conventional recursive neurons lack [21].

IV. Experiment

A. REFIT dataset

We used the Personalised Retrofit Decision Support Tools for UK Homes Using Smart Home Technology (REFIT) Smart Home dataset [6] as a test to compare the anomaly detection capability of the models described.

The REFIT Smart Home dataset includes data from various sensors that have been placed in 20 houses as well as a detailed description of the said sensors and houses. The data have been gathered over a period large enough to allow the training of our models.

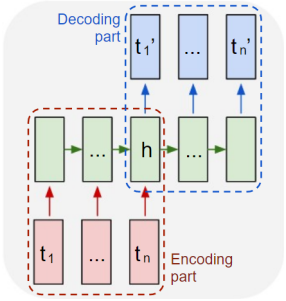

Fig. 3. Schema of a recurrent autoencoder encoding and decoding consecutive timesteps of a window to a single vector h.

In this dataset, we used the house containing the most sensor. In this home's data, we have selected the longest period during which the most sensors were active in order to test the models in a high dimensional context. The selected period runs from 04/11/2014 to 01/03/2015. After having selected this time range, we carried out a first study concerning the data of the sensors.

Our first observation was that some sensors were not useful for our case (very few or no variations, no pattern or data seeming completely random ...). Thus, we worked with 19 sensors of this house that were active during the period that we selected.

Our second observation was that a majority of the sensors showed, unsurprisingly, a strong daily seasonality.It should be noted that there are probably other seasonalities (for example yearly), but, having restricted our dataset to a certain time range, this information would not bring added value.

Once this first study was done, we aligned the data so that each data instance matched the data that each sensor emitted over a 15-minute period and we scaled them.

Based on our second observation, we have enriched each instance of data in our dataset with the timestamp of the day associated with it. More precisely, we associated a preprocessing of the timestamp of each data instance:

$$sinpart = \sin \frac{2 * \pi * timestamp}{24 * 3600}$$

$$cospart = \cos \frac{2 * \pi * timestamp}{24 * 3600}$$

Thay way, 23h59:59 and 00h00 (day timestamp of 86399 and 0) are much closer numerically and using both sinus and cosinus allow each timestamp to have a unique combination. Having similar numerical values for similar temporal contexts helps our models to extract temporal dependencies.

B. Evaluation

The platform on which we performed our tests was google compute engine. Each of the evoked models have been tested and realized using Tensorflow using the Keras wrapper on a Nvidia V100 graphic card equipped instance.

We used a standard methodology to elaborate our models. For each meta parameters specific to each type of model, we selected a list of values that seemed the most relevant and tested all the possible combinations. In addition, we have

TABLE I
SUMMARY OF THE BEST MODELS OF BOTH AUTOENCODER TYPES.

Type	Number of timesteps used	Botteleneck size	% dimension reduction	Fiability index	Absolute fiability index	Training time (s)	Loss
Rec.	48	200	80.15	1.339	**1.146**	1309	0.140
	48	150	85.12	1.282	1.112	1159	0.147
	48	250	75.20	1.279	0.943	1433	0.137
Conv.	96	384	80.95	**1.381**	0.897	977	0.052
	48	192	80.95	1.251	0.606	593	0.051
	24	192	61.90	1.143	0.475	**412**	0.039

varied for both types of models the parameters specific to autoencoders i.e. bottleneck size and size of the input windows to influence the dimension reduction.

The present anomalies in the dataset are not indicated. Plus, our input data being time series windows, we couldn't use unsupervised anomaly detection algorithm evaluation methods such as Excess Mass (EM) and Mass Volume (MV) [22]. Indeed, EM and MV are based, among other things, on the probability of occurrence of scores corresponding to all possible inputs and Lebesgue measures estimated with Monte-Carlo approximation. In our case it would be difficult to calculate such approximations since, in the set of all possible values of a window, a very large majority of these would be noise that would not be representative of possible inputs.

To evaluate our models' performances, we manually labeled some windows that corresponded to real life abnormal scenarios (such as the anomaly presented in III-A). Since we can not be sure that we labeled every anomalies, we could not determine the performances of our models with classical evaluation methods such as Receiver Operator Characteristic (ROC) or Precision-Recall (PR) [23].

However, having a dataset with enough labeled anomalies allows us to establish a good estimate of the performance of our models, given the generalization capacity of artificial neural networks.

We have therefore developed two indexes that reflects the visual criteria typically used to evaluate a model. Usually, in the case of a correctly trained autoencoder, the distribution of the error between the original data and the recreated data has a shape that is similar to a normal distribution.

The indexes (called here "Fiability index" and "Absolute fiability index") we have established evaluates the ability of a model to place anomalies on the right side of the error distribution.

To evaluate a model a, we consider here a set of anomaly errors $\mathbf{a} \in \mathbb{R}^n$ included in a set of reconstruction errors $\mathbf{e} \in \mathbb{R}^m$.

$$d = \frac{\sum_{i=0}^{n}(anomaly_i - mean(errors))}{n} reg \quad (1)$$

$$Fiability_a = d\frac{\frac{u+v}{2}}{m} \quad (2)$$

$$AbsFiability_a = d\frac{w}{m} \quad (3)$$

The term reg in 1 correspond to a regulation factor intended not to let the performance of the models, in terms of pure input recreation, bias the evaluation of their capacity to detect

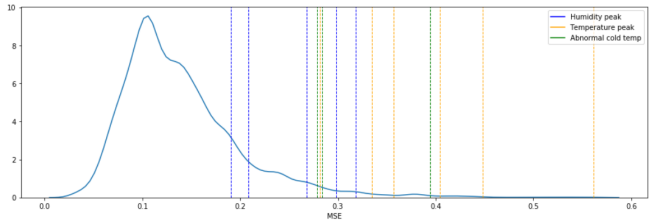

Fig. 4. Error distribution and anomaly placement of the best recurrent model.

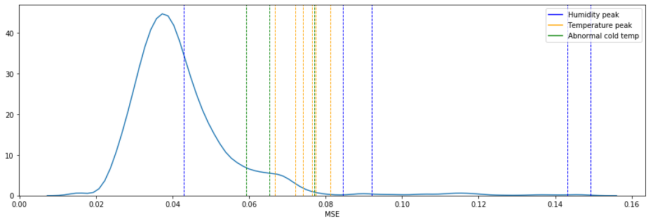

Fig. 5. Error distribution and anomaly placement of the best convolutional model.

the anomalous data instances. Indeed, as demonstrated by our experiment, the ability of an autoencoder to recreate its input with the smallest possible error (i.e. loss minization) is not directly correlated with its ability to detect errors. The terms u and v in 2 correspond respectively to the count of errors $<$ mean(**a**) and the count of errors $<$ median(**a**). The term w in 3 correspond to the count of errors $<$ min(**a**).

Both indexes evaluates the overall ability of a model to place anomalies as far as possible to the right of the error distribution.

However, where the Fiability index gives more importance to the magnitude of the distance between anomalies and the average of the error, the Absolute fiability index gives more importance to all anomalies being located on the right of the distribution.

It should be noted that these indices take advantage of the basic ability to approximate and generalize of the neural networks and is not to be recommended in the evaluation of other anomaly detection algorithms.

C. Observations

It is essentially the Absolute fiability index that allow us to judge which model performs the best in terms of anomaly detection. Indeed, as shown in the Table I, the Fiability index of both recurrent and convolutional autoencoders are of the same order of magnitude. However, recurrent autoencoders have a better overall Absolute fiability index. We can confirm,

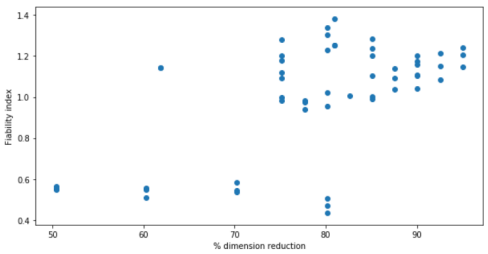

Fig. 6. Graph showing the Fiability indexes of each of the models that we tested according to the % of dimension reduction.

looking at both best models' distributions on Fig.4 and Fig.5, that the recurrent autoencoder is more efficient if we consider every anomalies.

Beside the fact that our experiment shows information about which type of AE performs best in anomaly detection in the context of connected buildings, the information compiled in the Table I gives us indications on the most important parameters to consider in the models. As we mentioned earlier, a smaller loss does not necessarily means a better anomaly detection. The most important parameter are the size of the input windows and the bottleneck size, since they directly influence the dimension reduction. We can see on Fig.6 that results are much better when a sufficient dimension reduction is made. The results are seemingly better between 75% and 95% which are very substantial compression rates.

V. Conclusion

We have selected two types of auto-coders that are sensitive to different types of anomalies. We have tested many recurrent and convolutional models on a set of coherent, high-dimensional academic data. To assess the results, we proposed two indices that take advantage of the basic ability of neural networks to approximate and generalize. The study of the results obtained allowed us to determine that the recurrent autoencoders had a better generalization capability and were the most effective for detecting anomalies in our data set. Extrapolating this result, it can be said that recurrent autoencoders are, to date, the best candidates in the field of neural networks applied to the detection of anomalies in connected buildings. Further tests on different datasets have yet to be carried out to corroborate this statement.

The fact that both types of models are able to produce decent or good results with input of such dimensions (19 sensors * number of timesteps) and with such dimension reduction rates confirms that there are very strong correlations and dependencies in the connected building sensor data.

However, the training time of each model (c.f. Table I), strongly correlated with the size of their input and the number of neurons and generally higher with recurrent autoencoders, is substantial. This constitutes for the moment a barrier to the implementation of such models in anomaly detection systems and may require more investigation.

VI. Acknowledgements

The authors thank the company Zenika and especially the agency of Lille for funding this work. The authors also thank the REFIT Smart Home dataset instigators and maintainers for allowing the use of their data.

References

[1] S. Li, L. D. Xu, and S. Zhao, "The internet of things: a survey," *Information Systems Frontiers*, vol. 17, Apr. 2015.

[2] R. Minerva, A. Biru, and D. Rotondi, "Towards a definition of the internet of things (iot)," *IEEE Internet of Things Initiative*, May 2015, (Date last accessed 14-Nov.-2017). [Online]. Available: http://iot.ieee.org/images/files/pdf/IEEE_IoT_Towards_Definition_Internet_of_Things_Revision1_27MAY15.pdf

[3] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection : A survey," *ACM Computing Surveys (CSUR)*, 2009.

[4] V. J. Hodge and J. Austin, "A survey of outlier detection methodologies," *Artificial Intelligence Review*, 2004.

[5] E. Keogh, J. Lin, S.-H. Lee, and H. V. Herle, "Finding the most unusual time series subsequence: algorithms and applications," *Knowledge and Information Systems*, 2007.

[6] S. Firth, T. Kane, V. Dimitriou, T. Hassan, F. Fouchal, M. Coleman, and L. Webb, "REFIT Smart Home dataset," 6 2017. [Online]. Available: https://figshare.com/articles/REFIT_Smart_Home_dataset/2070091

[7] S. Oniga and J. Süto, "Human activity recognition using neural networks," *Control Conference (ICCC), 15th International Carpathian*, 2014.

[8] P. Mitra, R. Ray, R. Chatterjee, R. Basu, P. Saha, S. Raha, R. Barman, S. Patra, S. S. Biswas, and S. Saha, "Flood forecasting using internet of things and artificial neural networks," *Information Technology, Electronics and Mobile Communication Conference (IEMCON), 2016 IEEE 7th Annual*, 2016.

[9] A. Akbar, F. Carrez, K. Moessner, and A. Zoha, "Predicting complex events for pro-active iot applications," *IEEE 2nd World Forum on Internet of Things (WF-IoT)*, 2015.

[10] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. The MIT Press, 2016, ch. Autoencoders.

[11] J. Almotiri, K. Elleithy, and A. Elleithy, "Comparison of autoencoder and principal component analysis followed by neural network for e-learning using handwritten recognition."

[12] J. Masci, U. Meier, D. Ciresan, and J. Schmidhuber, "Stacked Convolutional Auto-Encoders for Hierarchical Feature Extraction," *Artificial Neural Networks and Machine Learning – ICANN 2011*, 2011.

[13] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. The MIT Press, 2016, ch. Convolutional Networks.

[14] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural network overfitting." *Journal of Machine Learning Research 15*, 2014.

[15] C. Fana, F. Xiaob, Y. Zhaoc, and J. Wanga, "Analytical investigation of autoencoder-based methods for unsupervised anomaly detection in building energy data," *Applied Energy 211*, 2018.

[16] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. The MIT Press, 2016, ch. Sequence Modeling: Recurrent and Recursive Nets.

[17] A. Graves, A. rahman Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013.

[18] H. Sak, A. Senior, K. Rao, O. İrsoy, A. Graves, F. Beaufays, and J. Schalkwyk, "Learning acoustic frame labeling for speech recognition with recurrent neural networks," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, April 2015, pp. 4280–4284.

[19] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, T. Darrell, and K. Saenko, "Long-term recurrent convolutional networks for visual recognition and description," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015, pp. 2625–2634.

[20] P. Malhotra, A. Ramakrishnan, G. Anand, L. Vig, P. Agarwal, and G. Shroff, "LSTM-based Encoder-Decoder for Multi-sensor Anomaly Detection," *ICML Anomaly Detection Workshop*, 2016.

[21] A. Graves, *Supervised Sequence Labelling with Recurrent Neural Networks*. The MIT Press, 2012, ch. Long Short-Term Memory.

[22] N. Goix, A. Sabourin, and S. Clémençon, "On Anomaly Ranking and Excess-Mass Curves," *18th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2015.

[23] N. Goix, "How to Evaluate the Quality of Unsupervised Anomaly Detection Algorithms?" *ICML Anomaly Detection Workshop*, 2016.