Eindhoven University of Technology
Department of Electrical Engineering
Signal Processing Systems Group

Technische Universiteit Eindhoven University of Technology
Where innovation starts

Classification of fractionated electrograms in epicardial mappings using a recurrent neural network

by

Rob Romijnders

Practical Training Report

Project period: May-Aug 2-16
Report Number: 1287

Commissioned by:
Signal Processing Systems Group

Supervisors: L.M. Eerikainen, R Vullings

The Department of Electrical Engineering of the Eindhoven University of Technology accepts no responsibility for the contents of M.Sc. theses or practical training reports

Classification of fractionated electrograms in epicardial mappings using a recurrent neural network

Internship SPS, Rob Romijnders

Supervision:
L.M. Eerikäinen MSc, *dep. Electrical Engineering, Eindhoven University of Technology*
dr.ir. R. Vullings, *dep. Electrical Engineering, Eindhoven University of Technology*
dr. N.M.S. de Groot *dep. Translational Electrophysiology, Erasmus Medical Centre*

July 29, 2016

Abstract

Atrial fibrillation (AF) is the most common cardiac arrythmia. Epicardial mappings measure the atria in array electrodes to investigate the underlying mechanisms of AF. Currently, clinicians manually annotate the single, double, triple and quadruple deflections in the electrograms. Triple and quadruple deflections, also named fractionated, signify irregularity in the underlying atrial tissue. The manual annotation is both time consuming and error sensitive. This work implements a recurrent neural network (RNN) to automate the classification of these electrograms. Firstly, we implement features reported in literature. Secondly, we propose a bidirectional RNN to improve on these classifications. An extension of this model is also able to annotate the electrograms. Features from literature classify non-fractionated electrograms (singles, doubles) from fractionated electrograms (triples, quadruples) with 0.58 to 0.75 area under curve (AUC). The proposed approach with RNN achieves 0.97 AUC on the same task. The proposed annotation model annotates with 91% accuracy when compared to expert annotations.

1 Introduction

In atrial fibrillation (AF), the atria contract arrythmically. Both clinicians and scientists work to understand the underlying mechanisms of AF. Epicardial mapping measures the atria at high resolution to understand the mechanisms underlying AF [1]. In epicardial mapping, an array electrode maps the electrical activation over 192 channels [2].

The different morphologies in the atrial electrograms represent the activation pattern of underlying atrial tissue. In a single complex, the atrial cells activate synchronized. For doubles, triples and quadruples additional activations occur. At high level, we consider singles and doubles as non-fractionated electrograms and triples and quadruples as fractionated electrograms. By this definition, fractionation relates to the pathology of the underlying tissue [3]. Figure 1 shows examples.

In current clinical practice, experts manually annotate the main deflections for each complex. This work is both time consuming and error sensitive. This work employs recent advancements in machine learning to automate the annotation of atrial electrograms and thus support future research.

Recent advances in machine learning provide new tools for signal classification and annotation. In time series classification, neural networks outperform traditional approaches [4] and modelling of

physiological data [5].

Recurrent Neural Networks (RNN) learn to map sequences to sequences [6]. This allows for both classification and annotation. RNN's have been successfully applied to emotion detection [7], speech recognition [8], image captioning [9], phoneme classification [10], bio-informatics [11], handwriting recognition [12], sports analytics [13], video analysis [14] and attention-based machine translation [15].

This work applies a RNN to classify and annotate atrial electrograms measured during epicardial mapping. Section 2 firstly discusses benchmarks reported in literature, fractionation index (FI), complexity index (CI) and sample entropy (SampEn). Secondly, it introduces our implementation of the RNN. Section 3 reports the results of the experiments on data from epicardial mappings. Section 4 discusses the main findings, limitations and future perspectives.

2 Method

2.1 Data

The data contains unipolar electrograms measured in epicardial mapping. The electrodes measure at 1kHz. The arrays are either circular, with 256 channels, or rectangular, with 128 or 192 channels. Custom-made software sets annotations at the main deflection. Experts manually correct the annotations.

The electrograms are not preprocessed. To align the signals from different measurements, we normalize the signals to unit power, as calculated per channel. All figures in this paper refer to this normalized amplitude.

Typical measurements range between 4 and 10 seconds. Pairs $\{(x^{(i)}, y^{(i)})\}_{i=0}^{N-1}$ are extracted such that every pair contains one complex of 120 ms, $x^{(i)} \in R^{120}$, and the corresponding annotations as $y^{(i)} \in \{l_1, l_2, ..., l_{K^i}\}$. For signals, t indexes time with $t = 0, 1, ..., T-1$. For annotations, l_k, denotes the location of the k-th annotation and K^i denotes the total number of annotations at sample i. For example, $y^i = \{35, 75, 95\}$ is a triple with deflections at 35 ms, 75 ms and 95 ms.

The extracted pairs are randomly allocated to a training set, validation set or test set in an 8:1:1 ratio. In total, we have 61.578 complexes, from epicardial mappings in 36 patients. Extracted pairs per patient contribute to only one of the three sets. N denotes the size of each set. Reported performances in section 3 result from the test set

Set	Non-fractionated		Fractionated	
	Singles	Doubles	Triples	Quad.
Train.	36414 (12263)	12263 (12263)	3235 (12263)	516 (12263)
Val.	4589 (1493)	1493 (1493)	411 (1493)	61 (1493)
Test	4618 (1485)	1485 (1485)	376 (1485)	75 (1485)

Table 1: **Sizes of training, validation and test set** The first line denotes size before resampling and the second line (in parentheses) denotes size after resampling

The classes are highly unbalanced. Singles, doubles, triples and quadruples occur in a ratio 70:23:6:1. Resampling in the training and validation set prevents the model from overfitting. Singles are under sampled and the triples and quadruples are over sampled to equal the amount of doubles [16]. Table 1 outlines the class sizes before and after resampling.

2.2 Benchmarks

CI and FI describe atrial electrograms and were first introduced by Yoshida et al. [17]. CI counts the sign changes in the atrial potential. FI counts the sign changes in the one-step increments of the potential.

$$CI(x^{(i)}) = \frac{1}{T}\sum_{t=1}^{T-1} \mathbb{1}(\mathbb{1}(x_{t-1}^{(i)} \geq 0) = \mathbb{1}(x_t^{(i)} < 0)) \tag{1}$$

$$FI(x^{(i)}) = \frac{1}{T}\sum_{t=2}^{T-1} \mathbb{1}(\mathbb{1}(x_{t-1}^{(i)} \geq x_{t-2}^{(i)}) = \mathbb{1}(x_t^{(i)} < x_{t-1}^{(i)}))) \tag{2}$$

$\mathbb{1}()$ denotes the indicator function

SampEn measures the complexity of a signal. Cirugeda-Roldan et al. introduce SampEn to classify atrial electrograms [18]. The authors hypothesize that more fractionated electrograms show

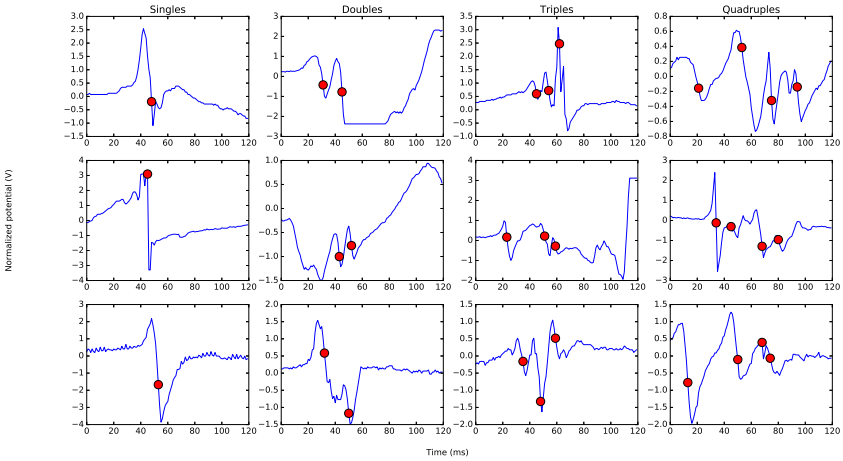

Figure 1: **Examples of atrial electrograms** Potentials are normalized potentials. Red dots are expert annotations

higher entropy. Our implementation follows the equations outlined in [18] and we refer to hyperparameters, r and m, by the same definitions.

The hyperparameters and the distance function for SampEn are set via cross-validation. A random search runs for 300 trials and picks hyperparameters according to the best performance on the validation set.

2.3 Recurrent Neural Network

The choice for a RNN to model the electrograms has two arguments. Firstly, a recurrent neural network processes data of any sequence length, whereas many other models require sequences of fixed length. Secondly, many models require stationary input signals. This does not apply to atrial electrograms.

As a RNN, we choose the Long Short-term Memory architecture (LSTM), first introduced by Schmidhuber et al. [19]. Traditional RNN's suffer from the vanishing gradient problem for models that require long term dependencies [20]. This would be detrimental for our data of 120 ms long.

After its initial publication, Gers et al. included the forget gate [21]. This prevents the cell state to increase in size with every time step. LSTM, in this paper, refers to LSTM with forget gate.

The annotations in the electrograms require precise timing. Gers et al. suggest using peepholes to improve the timing of the predictions [22]. Peepholes connect the various gates with the cell state. This improves the timing of information in the hidden state, upon which the annotation depends.

Annotations center on the steepest part of the complex. Such property cannot be inferred in a unidirectional LSTM. Therefore, we use a bidirectional LSTM (BiLSTM) [12]. The use of a bidirectional LSTM does not hamper the use in clinical practice as it will run offline. A bidirectional LSTM combines two unidirectional LSTM's. One runs forward and another runs backward. The hidden state, as used for classification and annotation, is a concatenation of the forward and backward hidden states.

The architecture consists of a two layered LSTM with state size 60. Stochastic gradient descent updates the parameters according to Adam optimizer with initial learning rate of 0.005 and decay of 0.9 every epoch [23]. Dropout serves as a regularizer with keep probability 0.7 [24]. An implementation of early stopping [25] ends the training procedure when the cost on the validation set starts increasing. State size, learning rate and dropout were set according to performance on the validation set.
Altogether, the LSTM follows equations (3–7) with

classification and annotation following equations (9, 10). We omit superscript (i) to maintain readability.

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \quad (3)$$

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f) \quad (4)$$

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

$$c_t = f_t c_{t-1} + tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \quad (6)$$

$$h_t = o_t \; tanh(c_t) \quad (7)$$

σ is the logistic sigmoid function. i, f, o, c, h are the input, forget and output gate, and cell and hidden state. All these vectors have the same size. $W_{..}$ are the projection matrices. The peephole matrices, $W_{c.}$, are diagonal. The model runs on Python 2.7.11 with modules from Tensorflow 0.9 implementing bidirectional LSTM's [26].

2.3.1 Classification task

A linear layer, combined with a softmax layer, calculates the probability distribution over the four classes. The electrogram is classified as the most probable class.

$$\begin{aligned} m_t &= W_{hm}h_t + b_m \\ p(C_k|h_t) &= \frac{e^{m_{t,k}}}{\sum_{\kappa=1}^{4} e^{m_{t,\kappa}}} \end{aligned} \quad (8)$$

$$K = \underset{k}{\operatorname{argmax}}\, C_k \quad (9)$$

Here C_k is the class of fractionation, single, double, triple or quadruple, and equation 9 is the final prediction.

2.3.2 Annotation task

A second, clinically relevant, task considers the annotation of the electrograms. Where section 2.3.1 predicts the class, $K^{(i)}$, this task considers the annotation on the exact locations l_k. The probability of an annotation follows from 10

$$p_t(annotation) = \sigma(W_{ha}h_t + b_a) \quad (10)$$

Next, $l_k = t$ joins y if

$$p_t(annotation) > threshold \quad (11)$$

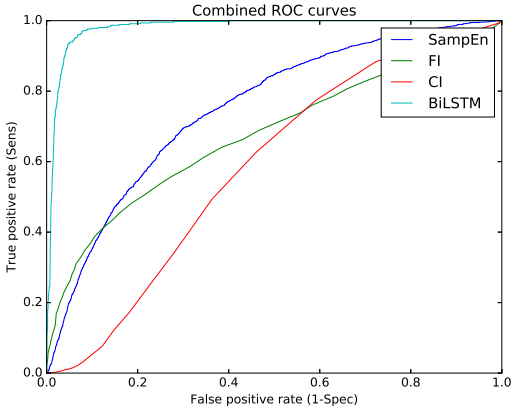

Figure 2: **ROC curve combining four models** This diagram combines ROC curves to classify fractionation using CI, FI, SampEn and the BiLSTM, with 0.58, 0.68, 0.75 and 0.97 AUC respectively

3 Results

All results in this section reflect performance on the test set. Cross-validation and tuning of hyperparameters follows performance on the validation set.

The first results report performances of the benchmark models. This experiment considers classification of non-fractionated (singles and doubles) versus fractionated electrograms (triples, quadruples). Cross-validation shows that $m = 2$, $r = 0.3$ and Euclidean distance function result in best area under curve (AUC) for SampEn. The definition of m and r follow [18]. Figure 2 shows the ROC curve for CI, FI and SampEn. These features achieve 0.58, 0.68 and 0.75 AUC respectively.

Figure 2 also shows the ROC curve for the BiLSTM on the same task, which achieves 0.97 AUC. Since the BiLSTM achieved such high performance on classifying fractionation, the classification task was refined to target singles, doubles, triples and quadruples individually.

Secondly, we report performances for annotating the electrograms and classifying the four classes directly. Figure 3 reports the results in a one-versus-all scheme. Each one-versus-all ROC curve discriminates one class from all other classes. The performances are 0.98, 0.96, 0.97 and 0.99 AUC for single-, double-, triple- and quadruple-versus-all, respec-

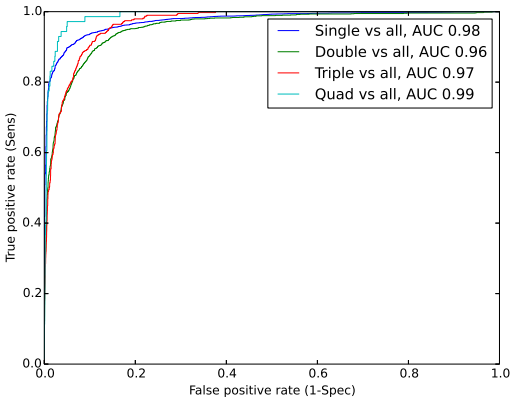

Figure 3: **ROC curve for one-vs-all** The four ROC curves for classifying one versus all other classes

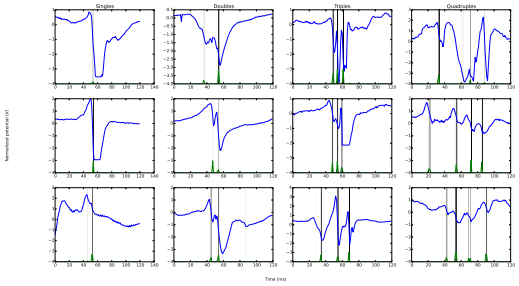

Figure 4: **Examples of atrial electrograms and annotations by the LSTM.** Grey scale and green line illustrate the probability of an annotation.

tively. Table 2 reports the misclassifications in a confusion matrix.

Figure 4 shows results for the annotation task. Both the green plot as the background grey-scale illustrate the probability of an annotation. With the threshold in equation 11 at 0.3, the BiLSTM annotates 91% of annotations within 3 ms from the expert's annotation.

4 Discussion

This work proposes a recurrent neural network to classify and annotate atrial electrograms measured in epicardial mappings. We improve performance on classifying fractionation from benchmarks at

		Model			
		Single	Double	Triple	Quad
Expert	Single	3565	821	134	36
	Double	76	1292	123	20
	Triple	2	58	262	68
	Quad	1	4	9	57

Table 2: **Confusion matrix** row, i, column, j, indicates how many complexes the expert classifies i and the model classifies j

0.58-0.75 AUC to 0.97 AUC. Furthermore, we refine the classification to target singles, doubles, triples and quadruples individually and annotate the deflections. In classification, we achieve 0.96-0.99 AUC and annotate deflections with 91% accuracy.

4.1 Misclassifications

The confusion matrix in table 2 shows unbalanced misclassifications. Misclassifying non-fractionation for fractionation occurs more often than vice versa. This possibly arises from errors in the annotations. When experts manually annotate the electrograms, custom-made software annotates the main deflection and experts add the additional deflections. In this scheme, experts more probably overlook a fractionated for a non-fractionated signal. Moreover, a bug in the custom-made software sometimes discard annotations. This explains why the model seemingly overestimates the fractionation. On revision, the expert agrees that many of the misclassifications are actually correct.

4.2 Limitations

The data extraction limits us to generalize these results. The extracted complexes center around the provided annotation. In this way, the annotations in the test set influence the extraction of the complexes, which biases our results. We partially dilute this bias by randomly perturbing the center of the complex. A better model would annotate the electrograms first and classify based on these annotations. This avoids using the annotations for the extraction.

The BiLSTM annotates the electrograms based on the hidden state, but not based on its past annotations. One can imagine that an annotation

depends on past annotations; one cannot expect two consecutive annotations within 2 ms. A better model would feed the prediction as input during the next time step. This conditions any annotation on its past annotations.

The training procedure ends based on the classification cost, but not on the annotation cost. During training, the annotation task already overfits the data, before the classification cost reaches the stop criterion. This diminishes the performance of the annotation task. Bengio et al. advocate to tailor learning rates in different parts of the network [25]. Better learning rates allow us to converge the cost functions simultaneously and will improve the performance of annotations.

4.3 Future research

The current model gives classifications within the four defined classes. However, the model has no output to indicate alarm in case of artifacts. broken electrodes or other unexpected events. Clinicians will base decisions on the classifications. Therefore, another output of the model should indicate the capability of making a classification at all. Recent work in variational auto encoders show promising results, where the reconstruction loss can grade an anomalous signal [27, 28].

Semi-supervised learning might bring another improvement to the annotation task. Only a small part of the epicardial mappings have been labelled. A semi-supervised learning algorithm allows the model to train on both labeled and unlabeled data. This unlabeled data could improve performance on the small labelled dataset [29].

5 Conclusion

This work proposes a recurrent neural network to classify and annotate atrial electrograms. Classification of single, double, triple and quadruple deflections achieve 0.96 to 0.99 AUC. In another experiment, benchmark from literature achieve 0.58-0.75 AUC. On the same task, the RNN improves performance to 0.97 AUC. Annotation of the deflections achieves 91% accuracy.

References

[1] Ameeta Yaksh et al. "Atrial fibrillation: to map or not to map?" In: *Netherlands heart journal : monthly journal of the Netherlands Society of Cardiology and the Netherlands Heart Foundation* 22.6 (2014), pp. 259–66. DOI: 10.1007/s12471-013-0481-0.

[2] Ameeta Yaksh et al. "A novel intra-operative, high-resolution atrial mapping approach". In: *Journal of Interventional Cardiac Electrophysiology* 44.3 (2015), pp. 221–225. DOI: 10.1007/s10840-015-0061-x.

[3] J. M T De Bakker and F. H M Wittkampf. "The pathophysiologic basis of fractionated and complex electrograms and the impact of recording techniques on their detection and interpretation". In: *Circulation: Arrhythmia and Electrophysiology* 3.2 (2010), pp. 204–213. DOI: 10.1161/CIRCEP.109.904763.

[4] Zhicheng Cui, Wenlin Chen, and Yixin Chen. "Multi-Scale Convolutional Neural Networks for Time Series Classification". In: (2016). arXiv: 1603.06995. URL: http://arxiv.org/abs/1603.06995.

[5] P Mart, Yoshua Bengio, and Georgios N Yannakakis. "Learning Deep Physiological Model of Affect.pdf". In: *IEEE Computational Intelligence Magazine* ().

[6] Yoshua Bengio, Geoffrey E. Hinton, and Yann LeCunn. "Deep learning". In: *Nature Methods* 13.1 (2015), pp. 35–35. DOI: 10.1038/nmeth.3707. arXiv: arXiv:1312.6184v5.

[7] Lang He et al. "Multimodal Affective Dimension Prediction Using Deep Bidirectional Long Short-Term Memory Recurrent Neural Networks". In: *Proceedings of the 5th International Workshop on Audio/Visual Emotion Challenge - AVEC '15* (2015), pp. 73–80. DOI: 10.1145/2808196.2811641.

[8] Hacsim Sak, Andrew Senior, and Françoise Beaufays. "Long Short-Term Memory Based Recurrent Neural Network Architectures for Large Vocabulary Speech Recognition". In: *arXiv preprint arXiv:1402.1128* Cd (2014). DOI: arXiv:1402.1128.

[9] Kelvin Xu et al. "Show, Attend and Tell: Neural Image Caption Generation with Visual Attention". In: *IEEE Transactions on Neural Networks* 5.2 (1994), pp. 157–166. DOI: 10.1109/72.279181. arXiv: arXiv:1211.5063v2.

[10] Alex Graves and Jurgen Schmidhuber. "Framewise phoneme classification with bidirectional LSTM networks". In: *Proceedings of the International Joint Conference on Neural Networks* 4.June (2005), pp. 2047–2052. DOI: 10.1109/IJCNN.2005.1556215.

[11] Sepp Hochreiter, Martin Heusel, and Klaus Obermayer. "Fast model-based protein homology detection without alignment". In: *Bioinformatics* 23.14 (2007), pp. 1728–1736. DOI: 10.1093/bioinformatics/btm247.

[12] Alex Graves. "Generating sequences with recurrent neural networks". In: *arXiv preprint arXiv:1308.0850* (2013), pp. 1–43. DOI: 10.1145/2661829.2661935. arXiv: 1308.0850v5. URL: http://arxiv.org/abs/1308.0850.

[13] Vignesh Ramanathan et al. "Detecting events and key actors in multi-person videos". In: *arXiv* (2015). arXiv: 1511.02917.

[14] Jeff Donahue et al. "Long-term Recurrent Convolutional Networks for Visual Recognition and Description". In: *Cvpr* (2015). DOI: 10.1109/CVPR.2015.7298878. arXiv: 1411.4389v3.

[15] Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. "Effective Approaches to Attention-based Neural Machine Translation". In: *Emnlp* September (2015), p. 11. arXiv: 1508.04025.

[16] MN N Kumar et al. "On the Classification of Imbalanced Datasets." In: *International Journal of Computer Science and technology* 44.8 (2012), pp. 1–7. DOI: 10.5120/6280-8449.

[17] Kentaro Yoshida et al. "Complex electrograms within the coronary sinus: Time- and frequency-domain characteristics, effects of antral pulmonary vein isolation, and relationship to clinical outcome in patients with paroxysmal and persistent atrial fibrillation". In: *Journal of Cardiovascular Electrophysiology* 19.10 (2008), pp. 1017–1023. DOI: 10.1111/j.1540-8167.2008.01175.x.

[18] Eva Cirugeda-Roldán et al. "Characterization of complex fractionated atrial electrograms by sample entropy: An international multi-center study". In: *Entropy* 17.11 (2015), pp. 7493–7509. DOI: 10.3390/e17117493.

[19] Sepp Hochreiter and Jurgen Schmidhuber. "Long Short-Term Memory". In: *Neural Computation* 9.8 (1997), pp. 1735–1780.

[20] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. "On the difficulty of training recurrent neural networks". In: *Proceedings of The 30th International Conference on Machine Learning* 2 (2012), pp. 1310–1318. DOI: 10.1109/72.279181. arXiv: arXiv:1211.5063v2. URL: http://jmlr.org/proceedings/papers/v28/pascanu13.pdf.

[21] Felix Alexander Gers, Jürgen Schmidhuber, and Fred Cummins. "Learning to forget: continual prediction with LSTM." In: *Neural computation* 12.10 (2000), pp. 2451–2471. DOI: 10.1162/089976600300015015. arXiv: arXiv:1011.1669v3.

[22] Felix a Gers, Nicol N Schraudolph, and Jurgen Schmidhuber. "Learning Precise Timing with LSTM Recurrent Networks". In: *Journal of Machine Learning Research* 3.1 (2002), pp. 115–143. DOI: 10.1162/153244303768966139.

[23] Diederik Kingma and Jimmy Ba. "Adam: A Method for Stochastic Optimization". In: *International Conference on Learning Representations* (2014), pp. 1–13. arXiv: 1412.6980.

[24] Nitish Srivastava et al. "Dropout : A Simple Way to Prevent Neural Networks from Overfitting". In: *Journal of Machine Learning Research (JMLR)* 15 (2014), pp. 1929–1958. DOI: 10.1214/12-AOS1000.

[25] Yoshua Bengio. "Practical recommendations for gradient-based training of deep architectures". In: *Lecture Notes in Computer Science* 7700 LECTU (2012), pp. 437–478. DOI: 10.1007/978-3-642-35289-8-26. arXiv: 1206.5533.

[26] Martin Abadi et al. “TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems”. In: (2015), p. 19. arXiv: 1603.04467.

[27] Diederik P Kingma and Max Welling. “Auto-Encoding Variational Bayes”. In: Ml (2013), pp. 1–14. arXiv: 1312.6114.

[28] Pankaj Malhotra and Tam Shroff Com. “LSTM-based Encoder-Decoder for Multi-sensor Anomaly Detection”. In: (2016). arXiv: 1607.00148.

[29] Diederik P Kingma et al. “Semi-Supervised Learning with Deep Generative Models”. In: *arXiv.org* cs.LG (2014), pp. 1–9. arXiv: arXiv:1406.5298v1.