Unsupervised Pretraining of Echo State Networks for Onset Detection

Peter Steiner[1 [0000−0002−8149−2275]], Azarakhsh Jalalvand[2,3 [0000−0001−8739−1793]], and Peter Birkholz[1 [0000−0003−0167−8123]]

[1] Institute for Acoustics and Speech Communication, Technische Universität Dresden, Dresden, Germany
peter.{steiner, birkholz}@tu-dresden.de
[2] IDLab, Ghent University–imec, Ghent, Belgium
[3] Mechanical and Aerospace Engineering department, Princeton University, USA
azarakhsh.jalalvand@ugent.be

Abstract. Note onset detection – the detection of the beginning of new note events – is a fundamental task for music analysis that can help to improve Automatic Music Transcription (AMT). The method for onset detection always follows a similar outline: An audio signal is transformed into an Onset Detection Function (ODF), which should have rather low values (i.e. close to zero) for most of the time, and pronounced peaks at onset times, which can then be extracted by applying peak picking algorithms on the ODF. Currently, Recurrent Neural Networks (RNNs) and Convolutional Neural Networks (CNNs) define the state of the art. In this paper, we build upon previous work about onset detection using Echo State Networks (ESNs) that have achieved comparable results to CNNs. We show that unsupervised pre-training of the ESN leads to similar results whilst reducing the model complexity.

Keywords: Echo State Networks · Clustering · Note Onset Detection.

1 Introduction

Music analysis is a relatively complex task that is typically split in various sub-tasks, such as note onset detection and multipitch tracking. Musical onsets are the starting points of new note events. Detecting note onsets is the first step for further music analysis tasks, such as Beat Tracking [4] or Automatic Music Transcription [7].

In general, musical onset detection can be treated as a three-step approach [5] including the following steps (1) feature Extraction, (2) reduction – computation of an Onset Detection Function ODF, and (3) peak picking.

Starting from a raw audio signal, mostly spectral features and their derivatives (Spectral Flux) over time are extracted. Since onsets are accompanied by the increase and change of energy, they can be visually recognized in this spectral representation by edges. Thus, the first step is responsible to simplify the onset detection.

The next step (reduction) computes a one-dimensional Onset Detection Function (ODF) from the typically high-dimensional feature vector sequence. In the ODF, onsets are characterized by strong peaks. There are various ways to compute an ODF from the feature vector sequence, with and without machine learning techniques. In [16], we have provided an extensive overview about reduction techniques.

Finally, peak picking algorithms are applied on the ODF to extract the actual onset times. In [6,12,15], neural networks such as bidirectional LSTM networks, Convolutional Neural Networks (CNNs) and Echo State Networks (ESNs) have all achieved state-of-the-art results in learning the ODF from feature vectors. The methods achieved F-Measures of 0.873, 0.903 and 0.886 on the Böck dataset [2], respectively.

Recently, in [14], we have shown that ESNs strongly benefit from a cluster-based input weight initialization, leading to equivalent or superior results compared to a baseline ESN whilst needing significantly less reservoir neurons, which consequently reduces the required amount of free parameters and training time.

In this paper, we built upon our results from [15,14] and investigated whether the unsupervised pre-training improves the performance of ESNs for note onset detection while lowering the model complexity. As the Böck dataset [2] also contains audio files that are not labeled and thus excluded from the standard cross validation, we furthermore investigated whether the pre-training benefits from additional data.

The remainder of this paper is structured as follows. Section 2 introduces the outline of onset detection with the basic ESN and the pre-trained KM-ESN. The experimental setup including the dataset is described in Section 3. In Section 4, we discuss the impact of the pre-training on the onset detection results and compare our result to the state of the art. Finally, we summarize our conclusion and future work in Section 5.

2 Onset Detection with Echo State Networks

Echo State Networks (ESNs) [8] belong to the class of Recurrent Neural Networks (RNNs) together with e.g. Long-Short-Term Memory Cells (LSTMs). However, in contrast to LSTMs, only very small parts of the weights in ESNs are trained using linear regression. Due to the recurrent connections inside the reservoir, which is the core element of an ESN, information from previous inputs is retained for a certain amount of time. This so called memory can be tuned by the hyper-parameters of the reservoir and subsequently act as a short- or long-term memory. As the number of neurons inside the reservoir is typically higher than the input features, the reservoir transforms the input space into a high-dimensional feature space, in which onsets can be separated with hyper-planes from non-onsets. Thus, the Onset Detection Function ODF is a multi-linear function of the reservoir states.

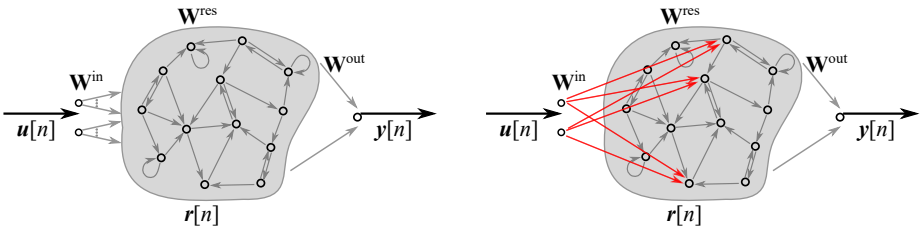
2.1 Feature Extraction

In this paper, we used our findings from [15] and divided the audio signal $s[k]$ (sampling frequency 44.1 kHz) in frames of the lengths {1024, 2048, 4096} samples starting at the same time index and a frame rate of 100 Hz. Each frame was windowed using a Hann window, and afterwards magnitude spectra were computed. As in [6], we applied a triangular filterbank with a logarithmic frequency spacing and 12 filters per octave to the magnitude spectra. We stacked the output of all three filterbanks, applied the $\log_{10}$ to the magnitude plus 1, and finally added the "Super-Flux" features [3] to obtain the final feature set with N^{in} features.

2.2 Echo State Network

The main outline of a basic ESN is depicted in Fig. 1a. It mainly consists of the following components:

- The input weight matrix $\mathbf{W}^{\text{in}}$ passes the N^{in} features to the reservoir with N^{res} neurons.
- The reservoir weight matrix $\mathbf{W}^{\text{res}}$ interconnects the neurons inside the reservoir.
- The output weight $\mathbf{W}^{\text{out}}$ connects the neurons to the output node.



(a) Basic ESN (b) KM-ESN

Fig. 1: Main outline of ESNs. In basic ESN, the input weights are randomly initialized. In KM-ESN, the input weights are the normed centroids of a K-Means algorithm applied to the training data. As there can be more neurons than centroids, not every neuron necessarily receives input information in case of the KM-ESN.

Typically, $\mathbf{W}^{\text{in}}$ and $\mathbf{W}^{\text{res}}$ are initialized from a random uniform distribution between ± 1 and standard normal distribution, respectively. Furthermore, they are typically initialized sparsely, which means that each neuron inside the reservoir receives a very small fraction of $K^{\text{in}} = 10$ input features and previous outputs of $K^{\text{rec}} = 10$ reservoir neurons. As $\mathbf{W}^{\text{res}}$ needs to fulfill the *Echo State Property* [8],

$\mathbf{W}^{\text{res}}$ is normalized to its maximum absolute eigenvalue. The hyper-parameters α_{u} (input scaling) and ρ (spectral radius) are factors to multiply $\mathbf{W}^{\text{in}}$ and $\mathbf{W}^{\text{res}}$ with, respectively.

The key difference between ESNs and typical RNN architectures is that $\mathbf{W}^{\text{in}}$ and $\mathbf{W}^{\text{res}}$ are initialized randomly, with no more optimization during the training. Only the output weights $\mathbf{W}^{\text{out}}$ are trained using linear regression.

With $\mathbf{r}[n]$ representing the reservoir state, Equations (1) and (2) describe the behaviour of ESN.

$$\mathbf{r}[n] = (1-\lambda)\mathbf{r}[n-1] + \lambda \tanh\left(\mathbf{W}^{\text{in}}\mathbf{u}[n] + \mathbf{W}^{\text{res}}\mathbf{r}[n-1] + \mathbf{w}^{\text{bi}}\right) \tag{1}$$

$$y[n] = \mathbf{W}^{\text{out}}\mathbf{r}[n] \tag{2}$$

Equation (1) is a leaky integration of the reservoir states $\mathbf{r}[n]$ with $\lambda \in (0, 1]$ being the leakage. Every neuron in the reservoir has a constant bias input from the bias weight vector $\mathbf{w}^{\text{bi}}$, which is initialized and fixed from a uniform distribution between ± 1 and then scaled with the factor α_{bi}. Equation (2) shows that the one-dimensional output $y[n]$ is actually a linear combination of a given reservoir state $\mathbf{r}[n]$ (i.e. $\mathbf{W}^{\text{out}}$ is a row vector).

For training, all reservoir states are collected in the reservoir state collection matrix $\mathbf{R}$. To add the intercept term for linear regression, every reservoir state $\mathbf{r}[n]$ is expanded by a constant of 1. The desired outputs $d[n]$, which are 0 for non-onsets, 1 for onsets and 0.5 for frames around onsets, are collected into the desired output collection vector $\mathbf{D}$. Afterwards, $\mathbf{W}^{\text{out}}$ is obtained using ridge regression (Equation (3)), to prevent overfitting to the training data. The regularization parameter $\epsilon = 0.01$ penalizes large values in $\mathbf{W}^{\text{out}}$, and $\mathbf{I}$ is the identity matrix. The size of the output weight vector $1 \times (N^{\text{res}} + 1)$ determines the total number of free parameters to be trained in ESNs. The output $y[n]$ corresponds to the onset detection function ODF.

$$\mathbf{W}^{\text{out}} = \left(\mathbf{R}\mathbf{R}^{\text{T}} + \epsilon\mathbf{I}\right)^{-1}\left(\mathbf{D}\mathbf{R}^{\text{T}}\right) \tag{3}$$

The basic ESN described so far, can be extended in several directions. Here, we focus on the K-Means-based initialized ESN (KM-ESN) and on bidirectional architectures.

KM-ESN: Although randomly initialized ESNs have achieved state-of-the-art results in various directions, several publications, such as [10,11] argue that there should exist better approaches that incorporate more prior or biologically plausible knowledge. Consequently, in [14], we have proposed the KM-ESN, which utilizes normalized centroids obtained from a K-Means algorithm applied to all feature vectors of the training data as input weights instead of randomly initialized input weights. The motivation behind the KM-ESN is that

1. using the K-Means-based pre-training tunes the input weights in a way that few reservoir states respond strongly to an input vector instead of many reservoir states responding with similar strengths.
2. a purely random weight initialization leads not to the best results [13],
3. passing feature vectors into the reservoir via $\mathbf{W}^{\text{in}}$ using Equation (1) is closely related to computing the cosine similarity between the feature vectors and all weights of one neuron,
4. unsupervised pre-training allows to train the ESN with additional unlabeled data.

The outline of the KM-ESN is visualized in Fig. 1b. The key difference between the basic ESN and the KM-ESN are the pre-trained input weights. As the number of centroids (K) is task-dependent and needs to be chosen carefully, we theoretically allow that only a part of the reservoir neurons receives input information. However, as the dataset in this paper is relatively complex, we kept $K = N^{\text{res}}$ in our experiments.

Bidirectional reservoirs: In the case of bidirectional reservoirs [9], the input is fed through the ESN normally and reversed in time. The reservoir states of both directions are collected and concatenated. As the output is computed using the concatenated reservoir states, bidirectional ESNs have the double number of free parameters in W^{out}. For example, the number of parameters for a reservoir with 500 neurons in the bidirectional case is 500 in the forward and 500 in the backward path.

2.3 Peak picking

In [16], we have discussed that the output of an ESN after linear regression indicates an onset or non-onset, and would be zero for a non-onset and one for an onset, ideally. We used exactly the same peak picking algorithm as in our previous study. The algorithm itself is a simple threshold-based peak picking algorithm originally proposed in [12]. At first, the ODF was smoothed using a Hamming window with 5 samples. Next, local maxima greater than a tunable threshold δ were detected. The locations of the resulting peaks were considered to be onsets.

3 Experimental setup

3.1 Dataset

For training and evaluation of the ESN models, we used the publicly available dataset[4] introduced by Böck in [2], which consists of all important types of onsets and various musical genres. Because the dataset was already used for several

[4] Download links: https://github.com/CPJKU/onset_db/

evaluations of algorithms for onset detection, we could directly compare the results of our cross-validation with state-of-the-art algorithms.

The dataset consists of 321 audio files together with onset annotations and 84 audio files without annotations that can be ignored according to Sebastian Böck. All audio files are sampled at 44.1 kHz and the total duration of all labelled audio files is 102 min. In total, there are 27 700 onset labels included. The labelled dataset is already split into eight folds for an 8-fold cross validation. We used six folds to train the ESN and one subset as a validation set to tune the hyper-parameters. Afterwards, we rotated the folds and repeated the optimization for another set of training and validation folds. This procedure was repeated eight times until every subset has been used for validation exactly one time. The hyper-parameters for the final evaluation were obtained by determining the lowest mean loss across all eight validation losses.

After fixing the hyper-parameters, the final model was trained using seven folds and tested on the last unseen fold. Again, this was repeated for eight times until each fold has been used for testing exactly one time. The results we report later are the mean values across the eight repetitions.

As we investigated the impact of utilizing additional data for the unsupervised weight initialization, we add the 84 audio files without onset annotations to the training dataset of the K-Means algorithm. We did not split the unlabeled files in eight subsets for cross validation but used all of them for every fold.

3.2 Measurements

As in [16,15], we compared our results with the state-of-the-art algorithms and report different measures using the Madmom library [1] with the same settings as used in [12]. The detected onset times were compared to the reference onset times. If an onset was detected in a time-window of ± 25 ms around a reference, it was considered as a true positive (TP). If no onset was detected in the window around a reference, it was considered as a false negative (FN). If any onset was detected outside the window, it was a false positive (FP). With these notations, the measurements Precision P (4), Recall R (5), F-Measure F (6) are defined.

$$P = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{4}$$

$$R = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{5}$$

$$F = 2 \cdot \frac{P \cdot R}{P + R} \tag{6}$$

In this work, F served as the objective function to determine the peak picking threshold δ.

The K-Means algorithm aims to partition N feature vectors into K sets $S_1, S_2, \ldots, S_K$ and thereby minimizes the within-cluster sum of squares (SSE) in Equation (7)

$$\mathrm{SSE} = \sum_{k=1}^{K} \sum_{\mathbf{u}[n] \in S_k} \|\mathbf{u}[n] - \mu_k\|^2 \ , \tag{7}$$

where μ_k is the k-th centroid. With the SSE, we determined a suitable reservoir size for hyper-parameter tuning of the ESN.

3.3 Implementation and optimization strategy

The algorithm was developed in Python 3 and will be available in our Github repository[5]. The optimization process was conducted using a sequence of grid and line searches as in [15].

The optimization workflow to fix the hyper-parameters of the KM-ESN was slightly different: Before the reservoir design, we computed the average SSE across all eight folds for different K, which is presented in Figure 2. A strong decay could be observed until $K = 200$. Afterwards, SSE continued to decrease with a much slower slope.

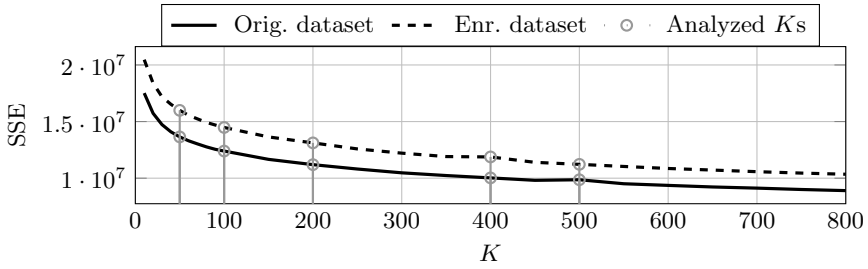


Fig. 2: Summed Squared Error (SSE) as a function of number of clusters (K) for applying K-Means algorithm on Böck music dataset.

Comparing the SSE of the original and the enriched dataset, no strong differences could be observed in the global behaviour. The SSE of the enriched dataset was always larger, because there were more training examples for the K-Means algorithm and the SSE is not normalized to the number of training examples.

We sampled $K \in \{50, 100, 200, 400, 500\}$ and then optimized the hyper-parameters using the workflow from [15]. As we observed that the determined

[5] https://github.com/TUD-STKS/PyRCN

hyper-parameters of the KM-ESN with 200 neurons were similar to larger reservoirs, we did not optimize the hyper-parameters for larger reservoirs with more than 500 neurons. For the experiments in Section 4, we fixed the hyper-parameters of the KM-ESN with 500 neurons and only increased the reservoir size as usual. The same observation also occurred in case of the KM-ESN with the enriched dataset. Table 1 shows the determined hyper-parameters for the basic ESN and for the KM-ESN without and with the enriched dataset.

Table 1: List of all hyper-parameters to be tuned. The values show the search range, the step size and the final values for all investigated architectures.

Hyper-parameter	Range	Step size	Final values		
			basic ESN	KM-ESN	KM-ESN (augm.)
Input scaling α_{u}	$[0.1, 1.5]$	0.1	0.4	0.1	0.1
Spectral radius ρ	$[0, 1.0]$	0.1	0.3	0.6	0.8
Bias scaling α_{bi}	$[0, 1.0]$	0.1	0.2	0.6	0.6
Leakage λ	$[0.1, 1.0]$	0.1	1.0	1.0	1.0
Threshold δ	$[0.2, 0.5]$	0.02	0.42	0.42	0.42

4 Results and Discussion

4.1 Basic ESN vs. KM-ESN

In Figure 3, results of different ESN architectures are summarized. In general, larger reservoirs performed better than small reservoirs. Especially the F-Measure of the basic ESN increased from 0.73 to 0.85. The KM-ESN always outperformed the basic ESN, especially in case of smaller reservoirs. The only exception occurred for the smallest model with 50 neurons, where the basic ESN performed slightly better. A KM-ESN with only 1000 neurons reaches almost the same performance as a basic ESN with 5000 neurons. This shows that the pre-training of the input weights influenced the detection results. In case of reservoir sizes up to 10 000, the performances of the basic ESN and KM-ESN reached a maximum of $F \approx 0.85$ and $F \approx 0.86$, respectively.

In [14], where we have introduced the KM-ESN, an outcome for speech was that we only needed a very limited number of 300 clusters for speech analysis. This is contrary to the results presented here, where even $K = 10\,000$ still improved the results. In contrast to speech, music is relatively complex: Several instruments can play simultaneously, and each instrument can play one or more notes at the same time. For example, a typical piano has 88 notes and the pianist can play up to ten notes simultaneously. This leads to a lot of possible note combinations. Therefore in case of polyphonic music, it would be logical to consider each centroid as the representative of a combination of notes, rather

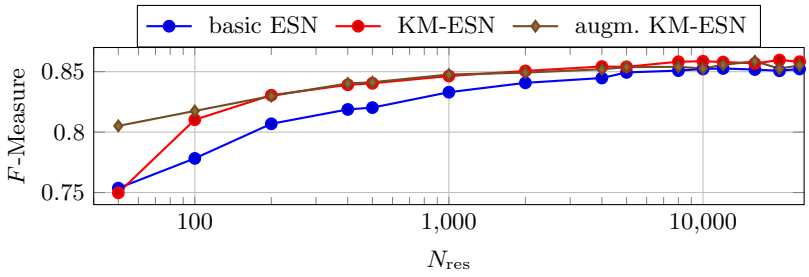


Fig. 3: F-Measures for the unidirectional basic ESN and for the unidirectional KM-ESN with and without the enriched dataset. The KM-ESN always outperformed the basic ESN. The enriched dataset only helped the smallest KM-ESN to improve the results.

than an isolated note. Consequently, a large number of centroids can be expected for music-related tasks.

A potential advantage of the KM-ESN is that it can be pre-trained with basically any meaningful data. NFrom Fig. 3 it can be observed that mostly the smallest KM-ESN benefited from the enriched dataset. If the reservoir size is increased, the KM-ESN with and without the enrichment strategy achieved similar results, and for very large reservoirs, the performance of the KM-ESN with the enriched dataset even slightly decreased. This means that the K-Means algorithm with many centroids found subclusters in the training dataset which were not representative for the test dataset.

4.2 Final results

In Table 2, the final results of this work are compared to the current state-of-the-art algorithms, namely the bidirectional LSTM [6], the best ESN results from [15], and the CNN [12]. The proposed model clearly outperformed the large bidirectional ESN increasing the F-Measure from 0.84 to 0.861 and 0.866 for the unidirectional KM-ESN with 20 000 neurons and for the bidirectional KM-ESN with 10 000 neurons, respectively. Obviously, with a causal model, we have almost reached the performance of the non-causal bidirectional LSTM [6] with less parameters N_{params} to be trained. Comparing the results of single- and multi-layer ESN in [15], one would expect that stacking KM-ESNs can minimize or close the gap between the performance of CNN and ESN with much fewer parameters. As the KM-ESN closely follows the basic ESN, it can be stacked as usual, e.g. as in [15].

Figure 4 shows the time signal, spectrogram and ODF with target and predicted onsets for an example of the test dataset. In all ODFs, we can see that potential onsets are characterized by prominent peaks. In case of the KM-ESN, the peaks are more prominent and fewer peaks in non-onset positions occurred. Thus, the KM-ESN is able to outperform the basic ESN.

Table 2: Performance of large reservoirs evaluated using the 8-fold cross validation on the Böck dataset. All the reference models were evaluated on the same dataset by the authors. With the utilized "Super-Flux" features and the basic ESN, we were able to outperform the reference single-layer ESN model from [15]. With the bidirectional KM-ESN models without and with the enrichment strategy, we almost reached the performance of the bidirectional LSTM [6].

Architecture	Precision	Recall	F-Measure	N_{params}	Mode	Features
Basic ESN 12000u	0.877	0.859	0.854	12 001	uni	Super-Flux
Basic ESN 16000b	0.892	0.826	0.858	32 001	bi	Super-Flux
KM-ESN 20000u	0.875	0.846	0.861	20 001	uni	Super-Flux
KM-ESN 10000b	0.889	0.844	0.866	20 001	bi	Super-Flux
KM-ESN 16000u (augm.)	0.870	0.850	0.860	16 001	uni	Super-Flux
KM-ESN 8000b (augm.)	0.879	0.855	0.866	16 001	bi	Super-Flux
Bidirectional LSTM [6]	0.892	0.855	0.873	20 225	bi	Super-Flux
ESN 1L-24000b [15]	0.881	0.804	0.840	48 001	bi	Spectral Flux
ESN 2L-28000b-5000b [15]	0.920	0.855	0.886	66 002	bi	Spectral Flux
CNN [12]	0.917	0.889	0.903	289 406	–	Super-Flux

5 Conclusion and Outlook

We have successfully extended our approach [15] for onset detection by utilizing the novel KM-ESN [14] with input weights pretrained using the unsupervised K-Means algorithm. With this model, we have removed randomness from the basic ESN and, especially for small reservoir sizes, we have achieved promising results. Specifically, a KM-ESN of only 1000 neurons reached almost the same performance as a bidirectional basic ESN model with 5000. Data enrichment seemed to be not very useful as there was no improvement when increasing the amount of training data for the K-Means algorithm.

In the future, we will extend this approach in various directions: We have shown that a large number of centroids was required for this seemingly simple task of onset detection. This might be due to the complexity of music signals, and is contrary to speech-related tasks, in which we utilized a very low number of centroids. If we can predict or at least robustly estimate the number of required clusters from a given unlabeled dataset task-dependently, we could speed up the reservoir design by sampling fewer K for hyperparameter optimization. Furthermore, it easily allows to switch to sparse KM-ESNs as soon as the reservoir size is getting larger than the estimated K.

With the existing KM-ESN model, we can do transfer learning towards other music-related tasks. As the combination of multipitch tracking and onset detection has been shown to be beneficial for note-based automatic music transcription [7], we can use exactly the same pre-trained set of input weights for onset and for

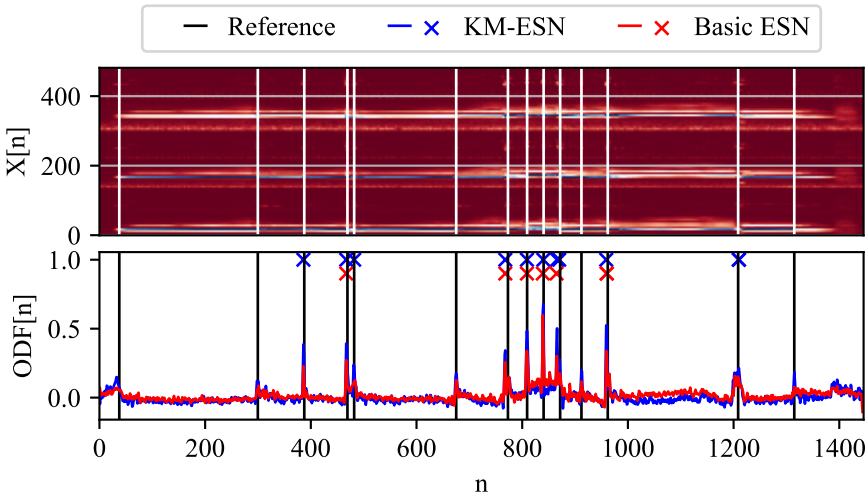


Fig. 4: Features, reference onsets and ODF for a basic ESN and a KM-ESN of a clarinet excerpt from the test set. The onsets were often relatively soft and difficult to recognize. Both, the basic ESN and the KM-ESN have several false negative onsets. However, as the peaks in case of the KM-ESN are more prominent and thus the onsets around $n = \{400, 500\}$ could be detected.

multipitch models. Only the hyper-parameters need to be tuned and the outputs need to be trained task-dependently.

Acknowledgements

The parameter optimizations were performed on a Bull Cluster at the Center for Information Services and High Performance Computing (ZIH) at TU Dresden. The research is also partially funded by the Special Research Fund of Ghent University (BOF19/PDO/134).

References

1. Böck, S., Korzeniowski, F., Schlüter, J., Krebs, F., Widmer, G.: Madmom: A new Python Audio and Music Signal Processing Library. In: Proceedings of the 24th ACM international conference on Multimedia. pp. 1174–1178. ACM (2016)
2. Böck, S., Krebs, F., Schedl, M.: Evaluating the Online Capabilities of Onset Detection Methods. In: Proceedings of the 13th International Society for Music Information Retrieval Conference, ISMIR 2012, Mosteiro S.Bento Da Vitória, Porto, Portugal, October 8-12, 2012. pp. 49–54 (2012), http://ismir2012.ismir.net/event/papers/049-ismir-2012.pdf

3. Böck, S., Widmer, G.: Maximum filter vibrato suppression for onset detection. In: Proceedings of the 16th International Conference on Digital Audio Effects (DAFx). Maynooth, Ireland (Sept 2013). vol. 7 (2013)
4. Böck, S., Schedl, M.: Enhanced beat tracking with context-aware neural networks. In: Proceedings of the International Conference on Digital Audio Effects (DAFx-11). pp. 135–139. Montreal, Quebec, Canada (Sept 19–23 2011), https://www.dafx.de/paper-archive/2011/Papers/31_e.pdf
5. Dixon, S.: Onset Detection Revisited. In: Proceedings of the International Conference on Digital Audio Effects (DAFx-06). pp. 133–137. Montreal, Quebec, Canada (Sept 18–20, 2006), http://www.dafx.ca/proceedings/papers/p_133.pdf
6. Eyben, F., Böck, S., Schuller, B.W., Graves, A.: Universal Onset Detection with Bidirectional Long Short-Term Memory Neural Networks. In: Proceedings of the 11th International Society for Music Information Retrieval Conference, ISMIR 2010, Utrecht, Netherlands, August 9-13, 2010. pp. 589–594 (2010), http://ismir2010.ismir.net/proceedings/ismir2010-101.pdf
7. Hawthorne, C., Elsen, E., Song, J., Roberts, A., Simon, I., Raffel, C., Engel, J., Oore, S., Eck, D.: Onsets and Frames: Dual-Objective Piano Transcription (2017)
8. Jaeger, H.: The "echo state" approach to analysing and training recurrent neural networks. Tech. Rep. GMD Report 148, German National Research Center for Information Technology (2001), http://www.faculty.iu-bremen.de/hjaeger/pubs/EchoStatesTechRep.pdf
9. Jalalvand, A., Demuynck, K., Neve, W.D., Martens, J.P.: On the application of reservoir computing networks for noisy image recognition. Neurocomputing **277**, 237 – 248 (2018). https://doi.org/https://doi.org/10.1016/j.neucom.2016.11.100, http://www.sciencedirect.com/science/article/pii/S0925231217314145, hierarchical Extreme Learning Machines
10. Lukoševičius, M., Jaeger, H., Schrauwen, B.: Reservoir Computing Trends. KI – Künstliche Intelligenz **26**(4), 365–371 (2012). https://doi.org/10.1007/s13218-012-0204-5
11. Scardapane, S., Wang, D.: Randomness in neural networks: an overview. WIREs Data Mining and Knowledge Discovery **7**(2), e1200 (2017). https://doi.org/https://doi.org/10.1002/widm.1200, https://onlinelibrary.wiley.com/doi/abs/10.1002/widm.1200
12. Schlüter, J., Böck, S.: Improved musical onset detection with Convolutional Neural Networks. In: 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 6979 – 6983 (May 2014). https://doi.org/10.1109/ICASSP.2014.6854953
13. Schrauwen, B., Verstraeten, D., van Campenhout, J.: An overview of reservoir computing: theory, applications and implementations. In: Proceedings of the 15th european symposium on artificial neural networks. p. 471-482 2007. pp. 471–482 (2007)
14. Steiner, P., Jalalvand, A., Birkholz, P.: Cluster-based Input Weight Initialization for Echo State Networks. IEEE Transactions on Neural Networks and Learning Systems **submitted** (2021)
15. Steiner, P., Jalalvand, A., Stone, S., Birkholz, P.: Feature Engineering and Stacked Echo State Networks for Musical Onset Detection. In: 2020 25th International Conference on Pattern Recognition (ICPR). pp. 9537–9544 (2020)
16. Steiner, P., Stone, S., Birkholz, P.: Note Onset Detection using Echo State Networks. In: Böck, R., Siegert, I., Wendemuth, A. (eds.) Studientexte zur Sprachkommunikation: Elektronische Sprachsignalverarbeitung 2020. pp. 157–164. TUDpress, Dresden (2020)