

Enabling Large-Scale Probabilistic Seizure Detection with a Tensor-Network Kalman Filter for LS-SVM

De Rooij, S. J.S.; Batselier, K.; Hunyadi, B.

DOI

[10.1109/ICASSPW59220.2023.10193615](https://doi.org/10.1109/ICASSPW59220.2023.10193615)

Publication date

2023

Document Version

Final published version

Published in

ICASSPW 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing Workshops, Proceedings

Citation (APA)

De Rooij, S. J. S., Batselier, K., & Hunyadi, B. (2023). Enabling Large-Scale Probabilistic Seizure Detection with a Tensor-Network Kalman Filter for LS-SVM. In *ICASSPW 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing Workshops, Proceedings* (ICASSPW 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing Workshops, Proceedings). IEEE. <https://doi.org/10.1109/ICASSPW59220.2023.10193615>

Important note

To cite this publication, please use the final published version (if applicable). Please check the document version above.

Copyright

Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy

Please contact us and provide details if you believe this document breaches copyrights. We will remove access to the work immediately and investigate your claim.

Green Open Access added to TU Delft Institutional Repository

'You share, we take care!' - Taverne project

<https://www.openaccess.nl/en/you-share-we-take-care>

Otherwise as indicated in the copyright section: the publisher is the copyright holder of this work and the author uses the Dutch legislation to make this work public.

ENABLING LARGE-SCALE PROBABILISTIC SEIZURE DETECTION WITH A TENSOR-NETWORK KALMAN FILTER FOR LS-SVM

S.J.S. de Rooij^{1*}, K. Batselier², B. Hunyadi¹

Delft University of Technology

¹Circuits and Systems, ²Delft Center for Systems and Control

ABSTRACT

Recent advancements in wearable EEG devices have highlighted the importance of accurate seizure detection algorithms, yet the ever-increasing size of the generated datasets poses a significant challenge to existing seizure detection methods based on kernel machines. Typically, this problem is mitigated by significantly undersampling the majority class, but in practice, these methods tend to suffer from too many false alarms. Recent works have proposed tensor networks to enable large-scale classification with kernel machines. In this paper, we explore the use of a probabilistic tensor method, the *tensor-network Kalman filter for LS-SVMs (TNKF-LSSVM)*, for seizure detection, as we hypothesize that using more data will improve the detection performance. We show that the TNKF-LSSVM performs comparably to a regular LS-SVM in detecting seizures when both are trained on the same dataset. Additionally, the TNKF-LSSVM can provide meaningful uncertainty quantification, and it is able to handle large-scale datasets beyond the capabilities of the LS-SVM (i.e., $N > 10^5$). However, for the presented model configuration detection performance does not seem to improve with more input data.

Index Terms— epilepsy, Kalman filter, tensor network, SVM, seizure detection

1. INTRODUCTION

Epilepsy is a brain disorder characterized by abnormal brain activity that can cause seizures, abnormal behaviour, sensations and, sometimes, loss of awareness [1]. It accounts for a significant portion of the global disease burden, affecting around 50 million people worldwide. For up to 70% of the patients, antiepileptic drugs can provide adequate treatment, meaning that the remaining 30% of patients continue to suffer from seizures [1].

Recent work [2]–[4] has focused on the development of wearable devices that can monitor patients in their daily life. Such devices could make a tremendous difference in patients' lives. They could allow for better diary keeping of when

seizures occur, hence helping clinicians optimize treatment strategies. Furthermore, they might even be able to warn a patient when a seizure is about to occur.

For all that to be possible, however, an accurate seizure detection method is necessary. A state-of-the-art seizure detection method for one such device is based on a least-square support vector machine (LS-SVM) with an RBF kernel, using a collection of time, frequency and entropy-based features [2]. A drawback of LS-SVMs, however, is that they do not scale well with the input data size in terms of memory requirements ($\mathcal{O}(N^2)$) and computational complexity ($\mathcal{O}(N^3)$). The authors of [2] dealt with this by using only 24 hours of data to train the model, and by heavily undersampling the non-seizure data. This, however, removes a lot of useful information from the data. Furthermore, it does not make the detector robust to day-to-day changes in brain dynamics, when recent research has shown that seizures can exhibit a multi-day rhythm [5].

We hypothesize that taking all the data into account can make the detector more robust and reduce the false alarm rate, by including more non-seizure examples. One approach to be able to use more data would be to use a deep learning method. Recently, there certainly has been a lot of focus on the development of deep learning methods for seizure detection [6]–[8]. However, these algorithms are typically computationally expensive [9] and prone to overfitting [10]. Notably, kernel machines such as SVMs and Gaussian processes, which have fewer hyperparameters, have been shown to perform just as well if not better than neural networks [11], [12].

Therefore, the current work aims to enable large-scale seizure detection by using a tensor-network Kalman filter to solve the least-square system of the LS-SVM (TNKF-LSSVM), a classification method developed by the authors of [13]. Using a Kalman filter means that the classification method becomes probabilistic and allows for the calculation of an uncertainty bound. Incorporating uncertainty quantification can help to assess the reliability of predictions which would be useful for the clinical implementation of the method. The current work will focus especially on which adaptations need to be made to the TNKF-LSSVM algorithm in order to make it work for imbalanced data, which is typical in seizure detection.

*Seline de Rooij, and thereby this work, is supported by the TU Delft AI Labs program.

1.1. Notation

The following notation conventions will be used. Vectors, matrices and tensors are denoted by boldface lowercase letters (e.g. \mathbf{a}), boldface uppercase letters (e.g. \mathbf{A}) and boldface Euler script letters (e.g. \mathcal{A}), respectively. Scalars are denoted by italic letters. The mean of a variable is denoted by an overline.

2. TNKF-LSSVM

In this section, we will explain how the TNKF-LSSVM algorithm [13] works. This is done by first giving some necessary background information on LS-SVMs (Section 2.1) and tensors (Section 2.2).

2.1. LS-SVM

The least-square support vector machine (LS-SVM) was developed in 1999 by Suykens & Vandewalle [14]. It is an adaptation of the support vector machine classifier [15], where instead of using inequality constraints in the primal optimization function, equality constraints are used. In the dual, this ultimately leads to solving a linear least-square system. We will merely provide the reader with this dual problem formulation, for a detailed derivation please refer to [14], [16].

For a given training set, $\{\mathbf{x}_k, y_k\}_{k=1}^N$, where $\mathbf{x}_k \in \mathbb{R}^F$ are the features and $y_k \in \{-1, 1\}$ the corresponding labels, the dual LS-SVM formulation is given by (1), and the predictor by (2),

$$\begin{bmatrix} 0 & \mathbf{y}^T \\ \mathbf{y} & \mathbf{\Omega} + \mathbf{I}/\gamma \end{bmatrix} \begin{bmatrix} b \\ \boldsymbol{\alpha} \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{1} \end{bmatrix} \quad (1)$$

$$\hat{y}(\mathbf{x}') = \text{sign} \left[\sum_{k=1}^N \alpha_k y_k K(\mathbf{x}', \mathbf{x}_k) + b \right], \quad (2)$$

where $\mathbf{\Omega}(i, j) = y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) = y_i y_j \varphi(\mathbf{x}_i)^T \varphi(\mathbf{x}_j)$ is the kernel matrix, $\boldsymbol{\alpha} \in \mathbb{R}^N$ are the dual variables, b the bias, and γ is a regularization parameter.

Due to the kernel matrix, a feature mapping is introduced, mapping the features to a multidimensional space, $K(\mathbf{x}_i, \mathbf{x}_j) = \varphi(\mathbf{x}_i)^T \varphi(\mathbf{x}_j)$, such that $\varphi(\cdot) : \mathbb{R}^F \rightarrow \mathbb{R}^{F_h}$. This allows the SVM to perform nonlinear classification. The feature mapping is typically expressed by a kernel function. In this paper, we use the RBF kernel:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp \left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|_2^2}{2\sigma^2} \right).$$

2.2. Tensors and tensor-trains

Tensors are multidimensional arrays. A d -way or d -dimensional tensor has d indices. A matrix is a 2-way tensor and a vector is a 1-way tensor. The number of entries in a tensor grows exponentially in d . Therefore, high dimensional problems cannot be handled efficiently by standard numerical methods, as operations and memory usage grow exponentially as well.

Thus, an efficient representation of a tensor is often needed to work with these problems. One such representation, and the one used in this work, is the tensor-train decomposition (TT) [17]. In the tensor-train decomposition a d -way tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_d}$ is decomposed into d auxiliary three-dimensional tensors $\mathcal{A}^{(i)} \in \mathbb{R}^{R_i \times I_i \times R_{i+1}}$ (for $i = 1 \dots d$), where R_i are called the *ranks* of the TT, such that, $\mathcal{A}(i_1, \dots, i_d) = \sum_{r_0, \dots, r_d} \mathcal{A}^{(1)}(r_0, i_1, r_1) \dots \mathcal{A}^{(d)}(r_{d-1}, i_d, r_d)$.

A tensor is decomposed into a TT by a sequential SVD algorithm (TT-SVD) [17]. Truncating these sequential SVDs leads to significant compression of the original tensor.

The tensor-train decomposition can not only be used to decompose large tensors but also to efficiently represent large vectors and matrices. For vectors, this is done by reshaping the vector $\mathbf{a} \in \mathbb{R}^I$ into a multidimensional tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_d}$, such that $\prod_i I_i = I$, and then decomposing the resulting tensor into a TT.

A matrix $\mathbf{B} \in \mathbb{R}^{I \times J}$ can be decomposed in an analogous manner. The matrix is reshaped into $\mathcal{B} \in \mathbb{R}^{I_1 \times J_1 \times \dots \times I_d \times J_d}$, such that $\prod_i I_i = I$ and $\prod_i J_i = J$. This tensor is then decomposed into a Tensor-Train matrix (TTm) which has, in contrast to a TT, four-dimensional cores, $\mathcal{B}^{(n)} \in \mathbb{R}^{R_n \times I_n \times J_n \times R_{n+1}}$. As this allows one to still perform matrix and vector products while in TT format [17], [18].

The compression ratio that can be achieved by a tensor-train is the largest when the sizes of the modes of the resulting tensor are as small as possible. This process of obtaining a tensor with small mode sizes is typically called *quantization* and the corresponding tensor-train is typically called a quantized tensor-train (QTT).

2.3. Tensor-Network Kalman Filter approach

Instead of solving the least-square problem (1) directly, in the TNKF-LSSVM, this system is solved row-by-row using a Kalman filter [13].

Let $\mathbf{C} = \begin{bmatrix} 0 & \mathbf{y}^T \\ \mathbf{y} & \mathbf{\Omega} + \mathbf{I}/\gamma \end{bmatrix}$, $\mathbf{z} = \begin{bmatrix} b \\ \boldsymbol{\alpha} \end{bmatrix}$ and $\mathbf{u} = \begin{bmatrix} 0 \\ \mathbf{1} \end{bmatrix}$, and suppose the residuals have a zero-mean Gaussian distribution $\mathbf{r} \sim \mathcal{N}(0, \sigma_r^2 \mathbf{I})$, then (1) can be written as (3).

$$\mathbf{C} \mathbf{z} + \mathbf{r} = \mathbf{u} \quad (3)$$

This system can be solved optimally row-by-row with a Kalman filter.

$$\begin{aligned} \mathbf{z}[k+1] &= \mathbf{z}[k] + \mathbf{q}[k] \\ \mathbf{u}[k] &= \mathbf{c}[k] \mathbf{z}[k] + r[k]. \end{aligned} \quad (4)$$

Here, $[k]$ denotes the k -th row of \mathbf{C} and \mathbf{u} , and the k -th update of $\mathbf{z} \sim \mathcal{N}(\bar{\mathbf{z}}, \mathbf{P})$. The variable $\mathbf{q}[k] \sim \mathcal{N}(0, \mathbf{Q}[k])$ is introduced to allow for a forgetting factor, to give less weight to samples far in the past ($\mathbf{Q}[k] = (1 - \lambda^{-1}) \mathbf{P}[k]$, $\lambda \in (0, 1]$).

The predictions and their variance can be computed by,

$$\hat{\mathbf{y}}(\mathbf{x}') = \text{sign}([1 \ \boldsymbol{\omega}'] \bar{\mathbf{z}}) \quad (5)$$

$$\sigma_{\hat{\mathbf{y}}}^2 = [1 \ \boldsymbol{\omega}'] \mathbf{P} [1 \ \boldsymbol{\omega}']^T + \sigma_r^2, \quad (6)$$

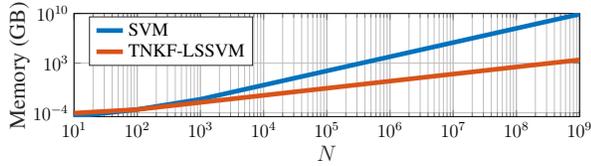


Fig. 1. Memory requirements of the LS-SVM and TNKF-LSSVM classifier plotted against the number of data points.

where $\omega'(k) = y_k K(\mathbf{x}', \mathbf{x}_k)$ for $k = 1 \dots N$ and $\omega' \in \mathbb{R}^{1 \times N}$.

For large datasets where $N > 10^5$, constructing the covariance matrix $\mathbf{P} \in \mathbb{R}^{N+1 \times N+1}$ becomes unfeasible. Therefore, the TNKF-LSSVM algorithm uses TTs and TTm's to decompose the Kalman filter variables ($\mathbf{c}[k]$, $\mathbf{z}[k]$ and $\mathbf{P}[k]$), making it possible to apply the algorithm to large-scale data. Figure 1 showcases the memory efficiency gain obtained by using the TT(m)'s. The memory usage of LS-SVM and TNKF-LSSVM is displayed against increasing N , with a maximum TT-rank of 40 and a feature vector length of $F = 360$ (which corresponds to the current application).

3. SEIZURE DETECTION

3.1. Dataset

In this paper, we use EEG data from the Temple University Hospital Seizure Corpus (TUSZ), version 1.5.2 [19]. To date, this is the largest publicly available seizure dataset. It consists of $\sim 3.87 \times 10^6$ seconds of annotated EEG recordings, around 2.66×10^5 s of which contain seizures. The recordings are from 674 different patients and contain 10 different seizure types. The TUSZ dataset is divided into three separate sets for training, validation and testing: 'train', 'dev' and 'eval'.

3.2. Preprocessing

First, the common electrodes among all the EEG recordings were selected, and a temporal central parasagittal (TCP) montage was applied to each recording. Furthermore, all recordings were resampled to a sampling frequency of 250 Hz. All channels were then filtered using a bandpass filter (0.3-50 Hz). After the filtering, the data was divided into segments of 2 s with 1 s overlap between segments. Segments with a root mean square (RMS) amplitude higher than $150 \mu\text{V}$ or lower than $11 \mu\text{V}$ were discarded from the training set, to remove high amplitude noise and background EEG.

3.3. Feature extraction

Features were extracted from the segments for each channel. The list of the used features is presented in Table 1. These features were taken from literature [2], [20]. The features in the high-frequency (HF) band were extracted before applying the bandpass filter. After extraction, the features were sorted to remove the spatial information and make it suitable for patient-independent classification [21]. Furthermore, all the features were normalized using standard scaling: $x = \frac{x - \bar{x}}{\sigma_x}$.

Table 1. Extracted features [2], [20].

Time domain	1-3. number of zero crossings, maxima and minima
	4. Skewness
	5. Kurtosis
	6. RMS amplitude
	7. Total power
	8. Peak frequency
Frequency domain	9-18. Mean and normalized power in frequency bands: δ (1-3 Hz), θ (4-8 Hz), α (9-13 Hz), β (14-20 Hz), HF (40-80 Hz).

3.4. Additional processing steps for the TNKF-LSSVM

As mentioned in section 2.2, the compression ratio achieved by tensor-trains is most efficient when the mode sizes of the quantized tensor are as small as possible. In this case it means that it should hold that $N + 1 = \prod_i \tilde{n}_i$, where n_i is 'small' and N is equal to the dataset size (+1 is for the bias).

Furthermore, when two TT(m)'s are added or multiplied, the ranks of the TT(m)'s also add up or multiply [17]. This can ultimately lead to an exponential increase in the size of the TT(m)'s. To prevent this, the ranks can be truncated back down using the TT rounding algorithm [17]. Because of this rounding, the TNKF-LSSVM algorithm is no longer permutation independent. It also makes the TNKF-LSSVM classifier more vulnerable to a class imbalance than the regular LS-SVM.

To deal with all of these issues we add the following processing steps to the training stage:

1. Get a new value $\tilde{N} \leq N$ such that $\tilde{N} + 1 = \prod_i \tilde{n}_i$, where $\forall i : \tilde{n}_i \leq 7$.
2. Randomly select $\tilde{N}^- = \lceil \tilde{N}/2 \rceil$ non-seizure and $\tilde{N}^+ = \lfloor \tilde{N}/2 \rfloor$ seizure data points. If $\tilde{N}^+ > N^+$ use an over-sampling technique such as SMOTE [22].
3. Randomly permute the order of the training data and alternate the seizure and non-seizure data points. This way, there should be a non-seizure data point at every uneven row index and a seizure data point at every even row index. This step is very important, as without it we found that the performance of the classifier remains at chance level.

3.5. Postprocessing

To improve the detection rate and reduce false positives a moving average filter, with a memory of 10 segments, was applied to the classifier output. Furthermore, neighbouring seizure events less than 90 seconds apart were 'stitched' together. And seizure events with a duration shorter than 25 seconds were classified as non-seizure [2], [23].

Table 2. Results of the classifiers on the ‘eval’ set. Between brackets the $\pm\sigma$ bounds are provided ($-\sigma, +\sigma$).

Metric	LS-SVM	TNKF-LSSVM small sample	TNKF-LSSVM large sample
Sensitivity	68.4	71.2 (69.8, 72.4)	63.6 (0.4, 100)
Precision	40.6	46.2 (46.6, 46.0)	37.8 (100, 39.1)
F1 score	0.509	0.561 (.559, .563)	0.475 (.008, .562)
FP/24h	80.1	66.3 (63.9, 67.9)	83.5 (0, 124.7)

3.6. Hyperparameter selection and algorithm evaluation

The hyperparameters for the TNKF-LSSVM were selected using gridsearches on the validation set (‘dev’). To evaluate the performance of the seizure detectors, we used the following performance metrics: sensitivity, precision, F1 score and false positive rate per 24 hours. These metrics were computed after the post-processing step using the any-overlap scoring algorithm to determine when a correct detection occurs [24].

Additionally, we plot the ROC curve of the classifiers, based on the classified segments (before post-processing), and show their AUC.

4. RESULTS

The performance of the TNKF-LSSVM algorithm was compared to that of the LS-SVM algorithm. However, since the LS-SVM classifier is not able to handle the entire dataset, the training dataset is first undersampled such that $\tilde{N}^- = 10^4$ and $\tilde{N}^+ = \tilde{N}^-$. The LS-SVMlab software package was used for the LS-SVM classifier, using the `tunelssvm` function to tune the hyperparameters [16].

To be able to compare the performance of the two classifiers the TNKF-LSSVM was first trained on the same sample as the LS-SVM. After that, we trained the model on a larger sample ($\tilde{N}^- = 2.6 \times 10^5$) to see if performance would improve when using more data. The used hyperparameters can be found along with the code on: github.com/sderooij/tnkf_lssvm_seizure_detect.

Table 2 shows the performance of both seizure detectors when applied to the ‘eval’ set. And in Figure 2 the corresponding ROC curve of both classifiers is shown. For the TNKF-LSSVM the results for the $\pm 1\sigma$ bounds (the 68% confidence interval) are included. These bounds were computed by adding or subtracting σ (Equation 6) from the classifiers’ output, prior to using the sign function. The performance is then evaluated again for these modified outputs.

5. DISCUSSION & CONCLUSIONS

From the results, it is clear that when trained on the same training set, the TNKF-LSSVM has similar results to the regular LS-SVM. It even seems to be outperforming the LS-SVM. This could be due to the tensor-trains’ ability to remove

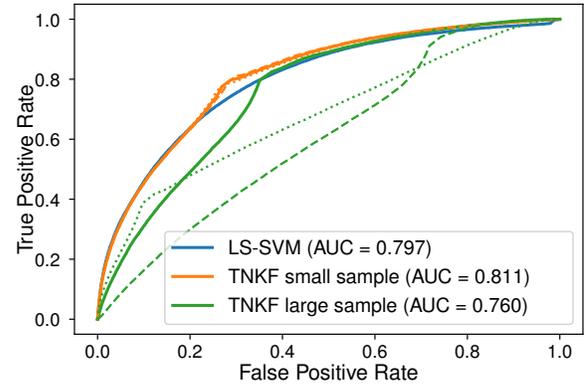


Fig. 2. ROC curve of TNKF-LSSVM and LS-SVM. The dashed and dotted lines show the $-\sigma$ and $+\sigma$ results resp.

redundant, and perhaps noisy, information from the model, thus acting as a regularizer. Though, it also might simply be the case that we were able to find a more optimal set of hyperparameters for the TNKF-LSSVM than the LS-SVM.

The confidence bounds of the TNKF-LSSVM for this smaller sample, which overlap with the nominal curve in the ROC, correlate with the relatively accurate predictions. Whereas we see that when the TNKF-LSSVM is trained on the larger sample, not only does the performance decline, but the uncertainty bound becomes considerably larger. The incorporation of this uncertainty quantification could provide added value in clinical implementation. Because if the bounds grow significantly, it may be an indication that the trained model is no longer fitting the incoming data well, signifying a need to retrain the model. Additionally, it may lead to the development of novel post-processing strategies.

Increasing the size of the training data set did not lead to improved performance. Thus, we may conclude that our original hypothesis was false and that adding more data does not improve the detection rate and reduce the false positive rate. Suggesting a better sampling strategy may be more crucial.

However, another likely possibility is that this is simply a consequence of using the TT(m)’s since the truncations become more ‘severe’ when more data is used. This is perhaps best exemplified by the fact that we had to approximate the covariance matrix \mathbf{P} by a rank one TTm in order to ensure that it remains positive definite. It’s possible that at some point this approximation becomes invalid, causing the algorithm to generate worse predictions, which indicates a possible lack of convergence. This could account for the large confidence bounds. A topic for future research could thus be to study the convergence conditions of the TNKF algorithm, to ensure convergence even for low-rank TT(m)’s.

Concludingly we may state that the TNKF-LSSVM can match, or even beat, the LS-SVM’s performance when both are trained on the same data, while also providing an uncertainty bound. However, being able to train on more data does not improve its performance.

6. REFERENCES

- [1] World Health Organization, "Epilepsy: a public health imperative: summary," WHO/MSD/MER/19.2, 2019, 12 pp.
- [2] K. Vandecasteele, T. De Cooman, J. Dan, *et al.*, "Visual seizure annotation and automated seizure detection using behind-the-ear electroencephalographic channels," *Epilepsia*, vol. 61, no. 4, pp. 766–775, 2020.
- [3] D. Johansson, K. Malmgren, and M. Alt Murphy, "Wearable sensors for clinical applications in epilepsy, Parkinson's disease, and stroke: A mixed-methods systematic review," *Journal of Neurology*, vol. 265, no. 8, pp. 1740–1752, 2018.
- [4] S. Boeckx, W. van Paesschen, B. Bonte, and J. Dan, "Live Demonstration: SeizeIT - A wearable multimodal epileptic seizure detection device," in *2018 IEEE Biomedical Circuits and Systems Conference*, Cleveland, OH: IEEE, 2018, pp. 1–1.
- [5] P. J. Karoly, D. M. Goldenholz, D. R. Freestone, *et al.*, "Circadian and circaseptan rhythms in human epilepsy: A retrospective cohort study," *The Lancet Neurology*, vol. 17, no. 11, pp. 977–985, 2018.
- [6] A. Craik, Y. He, and J. L. Contreras-Vidal, "Deep learning for electroencephalogram (EEG) classification tasks: A review," *Journal of Neural Engineering*, vol. 16, no. 3, p. 031001, 2019.
- [7] A. H. Ansari, P. J. Cherian, A. Caicedo, G. Naulaers, M. De Vos, and S. Van Huffel, "Neonatal Seizure Detection Using Deep Convolutional Neural Networks," *International Journal of Neural Systems*, vol. 29, no. 04, p. 1850011, 2019.
- [8] U. R. Acharya, S. L. Oh, Y. Hagiwara, J. H. Tan, and H. Adeli, "Deep convolutional neural network for the automated detection and diagnosis of seizure using EEG signals," *Computers in Biology and Medicine*, vol. 100, pp. 270–278, 2018.
- [9] N. C. Thompson, K. Greenewald, K. Lee, and G. F. Manso, *The Computational Limits of Deep Learning*, 2022. arXiv: 2007.05558 [cs, stat].
- [10] M. M. Bejani and M. Ghatee, "A systematic review on overfitting control in shallow and deep neural networks," *Artificial Intelligence Review*, vol. 54, no. 8, pp. 6391–6438, 2021.
- [11] A. Garriga-Alonso, C. E. Rasmussen, and L. Aitchison, "Deep Convolutional Networks as shallow Gaussian Processes," in *International Conference on Learning Representations*, 2022.
- [12] B. Hammer and K. Gersmann, "A Note on the Universal Approximation Capability of Support Vector Machines," *Neural Processing Letters*, vol. 17, no. 1, pp. 43–53, 2003.
- [13] M. Lucassen, J. A. K. Suykens, and K. Batselier, *Tensor Network Kalman Filtering for Large-Scale LS-SVMs*, 2021. arXiv: 2110.13501 [cs, LG].
- [14] J. Suykens and J. Vandewalle, "Least Squares Support Vector Machine Classifiers," *Neural Processing Letters*, vol. 9, no. 3, pp. 293–300, 1999.
- [15] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.
- [16] J. A. K. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor, and J. Vandewalle, *Least Squares Support Vector Machines*. WORLD SCIENTIFIC, 2002.
- [17] I. V. Oseledets, "Tensor-Train Decomposition," *SIAM Journal on Scientific Computing*, vol. 33, no. 5, pp. 2295–2317, 2011.
- [18] I. V. Oseledets, "Approximation of $2^d \times 2^d$ Matrices Using Tensor Decomposition," *SIAM Journal on Matrix Analysis and Applications*, vol. 31, no. 4, pp. 2130–2145, 2010.
- [19] I. Obeid and J. Picone, "The Temple University Hospital EEG Data Corpus," *Frontiers in Neuroscience*, vol. 10, 2016.
- [20] B. Hunyadi, M. Signoretto, W. Van Paesschen, J. A. K. Suykens, S. Van Huffel, and M. De Vos, "Incorporating structural information from the multichannel EEG improves patient-specific seizure detection," *Clinical Neurophysiology*, vol. 123, no. 12, pp. 2352–2361, 2012.
- [21] B. R. Greene, G. B. Boylan, R. B. Reilly, P. de Chazal, and S. Connolly, "Combination of EEG and ECG for improved automatic neonatal seizure detection," *Clinical Neurophysiology*, vol. 118, no. 6, pp. 1348–1359, 2007.
- [22] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.
- [23] A. Temko, E. Thomas, W. Marnane, G. Lightbody, and G. Boylan, "EEG-based neonatal seizure detection with Support Vector Machines," *Clinical Neurophysiology*, vol. 122, no. 3, pp. 464–473, 2011.
- [24] V. Shah, M. Golmohammadi, I. Obeid, and J. Picone, "Objective Evaluation Metrics for Automatic Classification of EEG Events," in *Biomedical Signal Processing*, I. Obeid, I. Selesnick, and J. Picone, Eds., Cham: Springer International Publishing, 2021, pp. 223–255.