


Proceeding Paper

Application of the Optimised Pulse Width Modulation (PWM) Based Encoding-Decoding Algorithm for Forecasting with Spiking Neural Networks (SNNs) [†]

Sergio Lucas [‡]  and Eva Portillo ^{*‡} 

Department of Automatic Control and Systems Engineering, Faculty of Engineering of Bilbao, University of the Basque Country (UPV/EHU), 48013 Bilbao, Spain; sergio.lucas@ehu.eus

* Correspondence: eva.portillo@ehu.eus

[†] Presented at the 10th International Conference on Time Series and Forecasting, Gran Canaria, Spain, 15–17 July 2024.

[‡] These authors contributed equally to this work.

Abstract: Spiking Neural Networks (SNNs) are recognised for processing spatiotemporal information with ultra-low power consumption. However, applying a non-efficient encoding-decoding algorithm can counter the efficiency advantages of the SNNs. In this sense, this paper presents one-step ahead forecasting centered on the application of an optimised encoding-decoding algorithm based on Pulse Width Modulation (PWM) for SNNs. The validation is carried out with sine-wave, 3 UCI and 1 available real-world datasets. The results show the practical disappearance of the computational and energy costs associated with the encoding and decoding phases (less than 2% of the total costs) and very satisfactory forecasting results (MAE lower than 0.0357) for any dataset.

Keywords: Spiking Neural Networks; Pulse Width Modulation (PWM) based encoding-decoding algorithm; forecasting



Citation: Lucas, S.; Portillo, E. Application of the Optimised Pulse Width Modulation (PWM) Based Encoding-Decoding Algorithm for Forecasting with Spiking Neural Networks (SNNs). *Eng. Proc.* **2024**, *68*, 41. <https://doi.org/10.3390/engproc2024068041>

Academic Editors: Olga Valenzuela, Fernando Rojas, Luis Javier Herrera, Hector Pomares and Ignacio Rojas

Published: 12 July 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In the past few years the use of Artificial Intelligence (AI) techniques has grown exponentially. However, recent studies [1,2] have highlighted the negative impact of this uncontrolled use in the global energy consumption, encouraging the scientific community to employ more efficient hardware and software.

In this sense, Spiking Neural Networks (SNNs) have gained popularity in recent years. These neural models, also known as the third generation of neural networks, are beginning to be considered more power efficient than Artificial Neural Networks [3,4]. The reason for this statement lies on how SNNs work. In SNNs the information is encoded in temporal spike sequences or spike trains, trying to mimic more closely the functioning of human brain. The use of spikes as working unit involves that: (a) SNNs are an event-driven, hence, when a neuron does not receive a spike, that neuron can be considered inactive leading to a decrease in the computational and energy cost of the algorithm; and (b) the cost of multiplying the inputs by the weights can be reduced considerably in hardware implementations [5]. In addition, SNNs present important advantages to be implemented in neuromorphic hardware with ultra-low power consumption.

Despite possessing intrinsic features to manage temporal data, SNNs are mainly applied in classification problems [6–8] rather than in regression or forecasting tasks, where the variable of time is relevant. This fact is based on two limitations. The first limitation is that traditionally there has been a lack of algorithms that can decode accurately the SNNs output into real values [9]. Notice that the decoding is a necessary process in regression and forecasting problems but not in classification ones. The second limitation is related to the difficulty of applying supervised training strategies, such as backpropagation, in SNNs.

In this sense, in [10] the first supervised training general methodology based on SNNs that can be applied to one-step ahead forecasting problem independently of the characteristics of the application field is presented.

This methodology [10] to overcome the aforementioned limitations applies: (i) a novel temporal encoding-decoding algorithm based on Pulse Width Modulation (PWM) [9], and (ii) a Surrogate Gradient (SG) method [11] that enables to employ a supervised training strategy.

Regarding the PWM based encoding-decoding algorithm applied in [10], this algorithm [9] was originally designed to provide substantial improvements in terms of precision in the encoding and decoding phases with respect to predecessor algorithms, being considered the efficiency of the algorithm as a non-crucial criterion. However, in terms of ultra-low energy consumption, if the encoding-decoding algorithm applied is not efficient enough, the efficiency advantages of SNNs may be countered. Thus, it is essential to apply the most efficient possible encoding-decoding algorithm.

In [12] an optimisation of the PWM based encoding-decoding algorithm is presented. This optimisation yields significant improvements in the accuracy of the encoding and decoding phases and reduces the computational and energy costs.

In this sense, this paper presents one-step ahead forecasting based on the application of the optimised PWM algorithm to SNNs. The aim of this work is to achieve significant improvements in accuracy and computational and energy costs of the SNNs based forecasting methodology by applying a more efficient and accurate encoding-decoding algorithm.

The rest of this paper is organized as follows: Section 2 gives a brief summary about SNNs. In Section 3 the methodology for one-step ahead forecasting based on the PWM optimised version applied to SNNs is presented. Section 4 describes the experimental setup used to assess the performance of the optimised algorithm in the SNNs training. Section 5 presents the results and the discussion is included in Section 6. Finally, Section 7 concludes the paper.

2. Spiking Neural Networks

For many years, neuroscience and AI have been cooperating to try to design and develop an algorithm that would try to reach the potential of human brain. This is the premise on which Artificial Neural Networks (ANNs) were created. However, the performance and efficiency of ANNs are not so close to biological networks.

SNNs try to mimic biological neurons more closely than ANNs encoding the information in temporal spike sequences or spike trains, such as the nervous system. This means that the information is contained in the number and timing of the spikes, but not in its shape.

Spiking neuron models are defined by differential equations. There is a wide variety of neuron models based on differential equations [13], but generally speaking the operation in discrete time of any spiking neuron model can be described by the following three equations:

$$H^t = f(V^{t-1}, X^t) \quad (1)$$

$$S^t = \Theta(H^t - V_{th}) \quad (2)$$

$$V^t = H^t(1 - S^t) + V_{rest} \cdot S^t \quad (3)$$

Equation (1) describes how the neuron is charged or discharged depending on if the neuron receives as input (X^t) a spike or not, respectively. This equation also depends on a function (f) that is different for each spiking neuron model and on the potential of the neuron in the previous time instant (V^{t-1}).

Equation (2) applies the Heaviside step function (Θ) to compare the potential of the neuron in the current time instant (H^t) with a threshold (V_{th}). If the value of H^t is higher than the threshold, the neuron will emit a spike, otherwise it will emit a zero.

Equation (3) describes the potential of the neuron for the next time instant. If the neuron does not emit a spike in the current time instant, the remaining potential is equal to

H^t . However, if a spike is emitted, the potential is reset to a value known as V_{rest} and this will be the initial potential value of the neuron for the next time instant.

The most well-known spiking neuron models are Integrate-and-Fire (IF), Leaky Integrate-and-Fire (LIF) and Spike Response Neuron Model (SRNM). The neuron model used in this paper is the LIF model [14] since it is the neuron model applied in [10]. The LIF model is the most widely neuron model used in SNNs due to its simplicity and low computational cost [15] and its functioning is usually abstracted into a resistor-capacitance circuit. In Figure 1 an example of a SNN using LIF neurons in hidden and output layer is shown. As in the case of ANNs, in SNNs the neurons in the input layer are only responsible for the forward propagation of the encoded data.

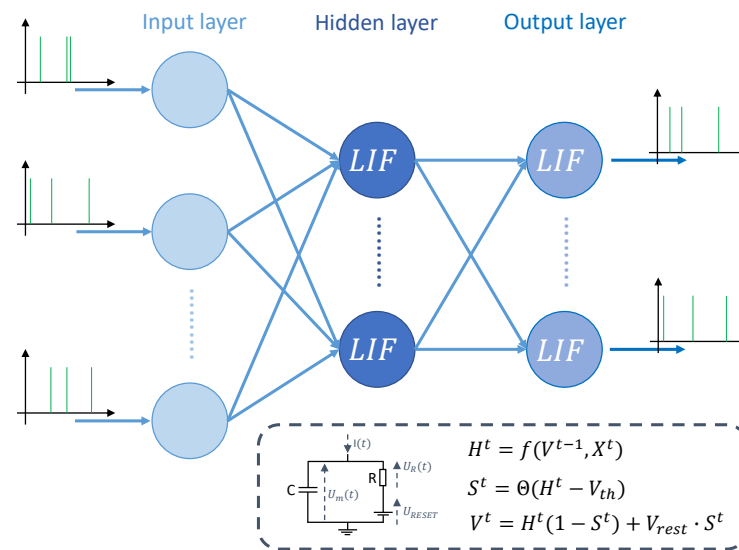


Figure 1. Example of a SNN structure.

3. Application of the Optimised PWM Algorithm to SNNs for One-Step Ahead Forecasting

3.1. Optimised PWM Based Encoding-Decoding Algorithm

In SNNs it is necessary to encode all the real value information into spikes before forwarding the data through any SNN. This is done by an encoding algorithm. The PWM based encoding-decoding algorithm is based on the PWM principles, emitting spikes where there is any intersection between the time-series signal and the carrier signal, commonly represented by a sawtooth. The carrier signal is used as a temporal reference during the encoding and decoding phases, hence, defining correctly this signal is essential. The carrier signal depends on two hyperparameters:

- Number of carriers (nc), straightly related to the carrier frequency.
- Number of points per carrier (npc), which is related to the resolution of the algorithm (see Figure 2A, where npc is four points represented in orange colour).

Regarding the comparison between the functioning of the original PWM based encoding-decoding algorithm (V_{or}) [9] and its optimisation (V_{opt}) [12], two main differences can be highlighted:

1. V_{or} processes point by point the complete time-series (signals formed by $nc * npc$ values) to encode or decode. However, V_{opt} encodes and decodes each time instant separately by calculating the intersection between the line that joints two values of the time-series and the line that defines the sawtooth (see Figure 2A). This change leads to simplified operations and reductions in memory and computational costs by more than half.
2. As shown in Figure 2A, V_{or} always emits the spike at the npc instant immediately after the intersection, while V_{opt} emits the spike at the closest npc instant. This

modification involves a improvement in the accuracy up to double of the encoding and decoding phases.

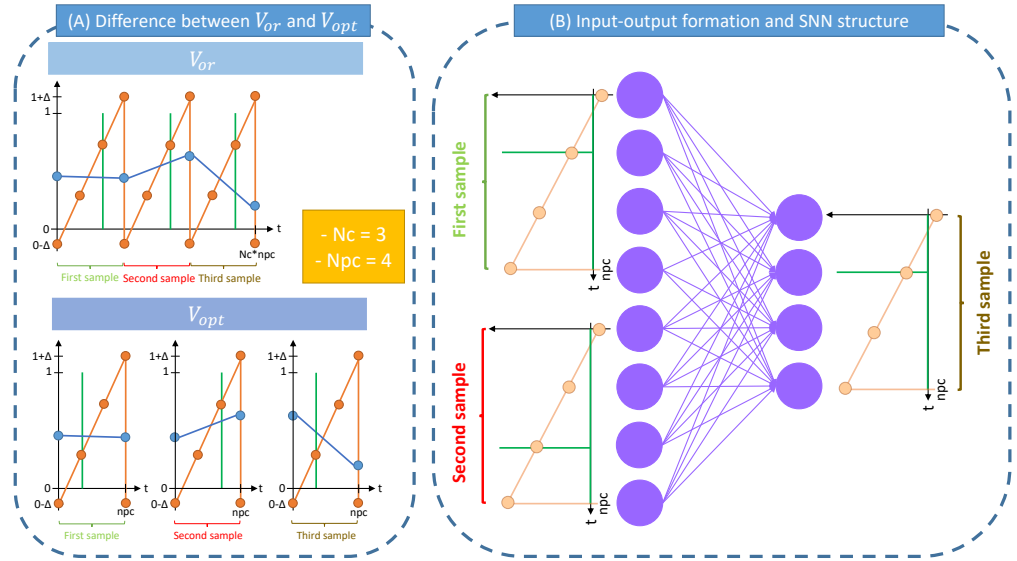


Figure 2. Encoding process and input-output pairs formation.

3.2. Input-Output Pairs Formation and SNN Structure

Once the time-series is encoded, the samples can already be propagated through the SNN. As explained in [10], in this methodology the samples are formed by the information that is contained within each sawtooth. Hence, the input layer of the SNN is formed by the number of previous values (Np) to make the forecast multiplied by npc , and the output layer consists of npc neurons since it is a one step-ahead forecasting. Figure 2B shows an example of the proposed structure being Np equal to 2 and npc equal to 4.

3.3. Surrogate Gradient (SG) Method

In ANNs, the supervised training algorithm is Back Propagation (BP). BP employs partial derivatives of the error committed by an ANN to update its parameters. If BP is applied to SNNs, the weights are updated following the next equations:

$$w_{ij,l} = w_{ij,l} - \eta \Delta w_{ij,l} \quad (4)$$

$$\Delta w_{ij,l} = \frac{\partial E}{\partial w_{ij,l}^t} = \frac{\partial E}{\partial S_{i,l}^t} \cdot \frac{\partial S_{i,l}^t}{\partial H_{i,l}^t} \cdot \frac{\partial H_{i,l}^t}{\partial w_{ij,l}^t} \quad (5)$$

In Equation (5), the term $S_{i,l}^t$ represents the output of a SNN, which is defined by Equation (2) where Θ is the Heaviside step function. If Equation (2) is derived at the time instant where the spike is emitted, this derivative will tend to infinite, making it impossible to apply the BP itself to SNNs.

In [10] BP is applied to SNNs employing a SG method. SG methods apply the differential equations of a spiking neuron model (Equations (1), (2) and (3)) during the samples propagation; however, during backpropagation they apply a surrogate gradient function ($\theta(x)$) instead of $\Theta(x)$. This new function is continuous in time, removing the problem of the non-differentiability and enabling the application of BP in SNNs.

4. Experimental Setup

In this section the validation of the SNNs training methodology with V_{opt} is presented. The validation lies in retraining the best selected SNN for each of the datasets applied in [10] and comparing the new results with the original ones.

4.1. Datasets

As in [10], the validation of the training methodology with V_{opt} is carried out with the same 5 datasets so as to assess that the methodology does not lose its generality with the introduction of V_{opt} . The 5 datasets are:

- A sine-wave signal, whose parameters are: frequency $f_o = 100$ Hz, amplitude $A = 3$, sampling frequency $f_s = 60 \cdot f_o$ and the number of cycles equal to 20.
- The Mackey-Glass Time Series Dataset (MGTSD), which is available in [16].
- The power consumption of the zone 3 in Tetouan. This dataset is available in [17] and the values from 00:00 1 January 2017 to 23:50 31 January 2017 are selected for the validation phase.
- The PM₁₀ hourly monitoring dataset measured by the station of London Bloomsbury in the Greater London Area, which is available in [18]. The period from 10:00 21 February 2017 to 20:00 3 June 2017 is selected for the validation phase.
- The ARTIC dataset that is available in [19] and the voice selected is in the file “arctic_a0001.wav” from the subfolder “US English bdl (made)”. The samples selected for the validation phase are from 11,600 to 14,750.

4.2. Performance Metrics Formulation

The validation of the proposed methodology with V_{opt} is assessed from two different perspectives: (i) the accuracy of the forecasting results; and (ii) the computational and energy cost of the algorithm.

Regarding the accuracy, the Mean Absolute Error (MAE) between the decoded signal (\hat{y}_i) and the original signal (y_i) is applied. This metric is defined by the following equation:

$$MAE = \sum_{i=1}^N |y_i - \hat{y}_i| \quad (6)$$

Concerning the computational and energy cost, the SNNs trained with each of the encoding-decoding algorithms (V_{or} and V_{opt}) are tested 50 times and the average of the computational and energy costs are computed. The time library of Python is used to measure the computational cost, while for the energy cost the PyJoules library from Python is used. Both costs are measured with a PC Intel(R) Core(TM) i7-4770K CPU @ 3.50GHz Linux 6.2.0-32-generic.

4.3. SNNs Selected for the Validation

The best selected SNN in [10] for each of the datasets is chosen to validate the introduction of V_{opt} in the training methodology. A summary of the hyperparameters from the SNNs selected is presented in Table 1. Information about the time-series splitting and the parameters of the LIF neuron are also detailed in Table 1.

Table 1. SNN selected main hyperparameters.

Dataset	Main Hyperparameters		Time-Series Splitting	LIF Neuron Parameters
	Np	npc		
Sine-wave	5	128		
MGTSD	1	32	Training : 70% Validation:20% Test: 10%	Tau = 100 $V_{threshold} = 1$ $V_{reset} = 0$ Learning rate = 0.001
Tetouan energy	1	128		
PM ₁₀ concentration	1	64		
ARCTIC	1	64		

5. Results

In this section the results for the datasets are presented. As stated above, the validation of the proposed methodology is carried out from two different perspectives: (i) the accuracy of the forecasting results; and (ii) the computational and energy cost of the algorithm.

5.1. Accuracy of the Forecasting Results

In the particular case of SNNs, the forecasting error can be understood as the sum of the error made by applying an encoding-decoding algorithm and the error committed by the SNN itself.

Table 2 is shown in order to illustrate the error made by applying the PWM based encoding-decoding algorithm in the proposed methodology. In this table, MAE is applied to estimate the difference between the original time-series and the time-series obtained after being encoded and decoded with V_{or} and V_{opt} , respectively. As expected, the modifications introduced in V_{opt} lead to improve the accuracy in the encoding and decoding phases, reducing the MAE of any of the time-series by at least 40%.

Table 2. MAE achieved encoding and decoding the original time-series.

	MAE between Decoded Time-Series and Original Time-Series		
	V_{or}	V_{opt}	Difference (%)
Sine-wave	0.0047	0.0017	63.83%
MGTSD	0.0196	0.0068	65.31%
Tetouan energy	0.0048	0.002	58.33%
PM ₁₀ concentration	0.0115	0.0066	42.61%
ARCTIC	0.0107	0.0057	46.73%

Regarding the error committed by the SNN, in Figure 3A the decoded SNN output is compared with the target used, being a suitable indicator to assess how well the SNN has learned the data that has been propagated through it during the training. For this part of the validation only the test set is used in order to assess the generalization power of the SNNs trained when they face unfamiliar data. As seen in Figure 3A, in terms of the training performance of each time-series there is no significant difference between applying V_{or} or V_{opt} , being the worst case the PM₁₀ concentration dataset where the maximum MAE difference between V_{opt} and V_{or} is 0.0038. Hence, it can be highlighted that employing a more accurate target during the training does not involve a relevant improvement in the learning.

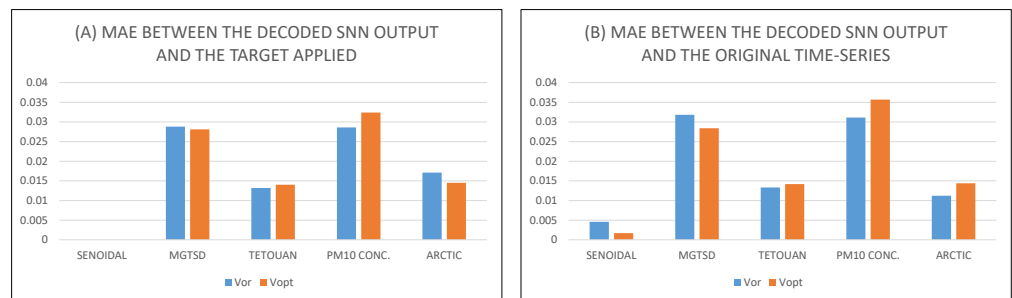


Figure 3. Forecasting measures applying V_{or} and V_{opt} .

Finally, Figure 3B shows the MAE between the decoded SNN output and the original time-series in order to infer the accuracy of the forecasting with the proposed methodology.

As stated above, this graphic can be considered the result of combining the metrics of Table 2 and Figure 3B.

Regarding sine-wave dataset, it is the only case in which the SNN has learned perfectly their targets (MAE in Figure 3A are equal to 0 for both V_{or} and V_{opt}). Thus, the lowest forecasting error is achieved applying V_{opt} since its target is more accurate (see Table 2).

Concerning MGTSD dataset, a better learning is done with V_{opt} target which leads to have slightly better forecasting metrics.

In the case of Tetouan energy consumption, PM₁₀ concentration and ARCTIC datasets, despite employing more accurate targets using V_{opt} the SNNs are not able to learn those targets as well as they do with V_{or} , resulting in slightly better forecasting metrics with the latter encoding-decoding algorithm. These results can be based on the fact that V_{opt} in some cases introduces more noise to the training that may lead to overfitting. For example, in Figure 4 the original time-series, the targets applied and the decoded SNN outputs are shown for the ARCTIC dataset. In this figure can be seen that in the case of using V_{opt} there are areas (sections A, B and C), specially located around local maximums and minimums, where the decoded SNN output cannot match to either the original time-series or the target. Despite making a good forecasting in the rest of the time-series, punctual errors, such as those occurred in these sections, penalize a great deal in the computation of the MAE. In addition, there may be cases in which if in these areas there is a greater distance between the decoded SNN output and the target than between the decoded SNN output and the original signal, the penalty in MAE computation is such that the error committed during the training is higher than the forecasting error, such as with ARCTIC dataset.

In any case, broadly speaking it can be observed that there is not a meaningful difference in terms of accuracy in the forecasting between applying V_{or} or V_{opt} .



Figure 4. Comparison among the original time-series, the target and the decoded SNN output with ARCTIC dataset.

5.2. Computational and Energy Costs

Figure 5 shows the computational and energy costs obtained with V_{or} (left side of the figure) and with V_{opt} (right side). In this part of the validation, the whole time-series (training, validation and test sets) is used to estimate the cost in order to have a larger sample size. In addition, in Figure 5 the costs related to the encoding phase, the propagation of the samples through the SNN and the decoding phase are identified separately.

Regarding the computational and energy costs obtained with V_{or} , the encoding and decoding phases represent more than 50% of total computational and energy costs. However, when V_{opt} is applied these costs are drastically reduced, being the use of SNN the main computational and energy cost with a consumption of more than 98% of total costs. Hence, it is clear that there are relevant advantages in terms of efficiency when V_{opt} is applied.

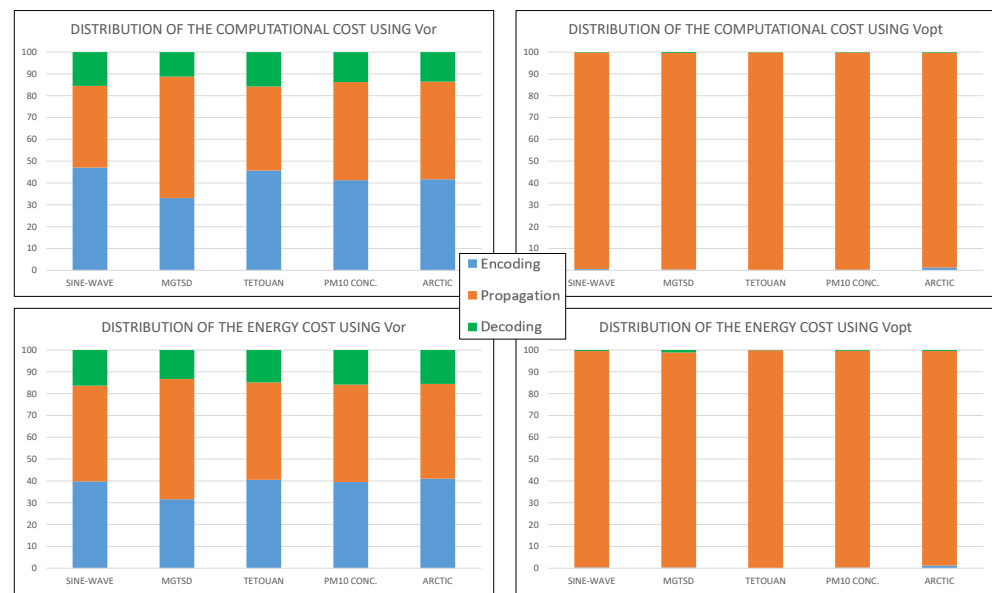


Figure 5. Computational and energy costs distribution applying V_{or} and V_{opt} .

6. Discussion

In view of the results, it can be concluded that the optimised version of the PWM based encoding-decoding algorithm is suitable to be applied with the training methodology explained in [10]. In addition, the validation is carried out with 5 different time-series from 5 different application fields, demonstrating that the use of V_{opt} does not entail a relevant loss of performance in the training methodology.

Regarding the accuracy in the forecasting process, the modifications introduced in V_{opt} lead to improve the accuracy of the encoding and decoding phases and, hence, to provide the SNNs with more accurate targets. This improvement in the accuracy is higher than 42.61% for every time-series applied in this paper. However, Figure 3A has proved that employing a more accurate target during the training does not mean a relevant improvement in the learning of the SNNs, being cases in which the learning process is slightly worse with V_{opt} than with V_{or} . One possible reason to this fact is that the target used with V_{or} is more stable, while the target used with V_{opt} is capable of adapting to all the changes that are produced in the time-series, introducing noise to the SNNs learning process. Nonetheless, in terms of forecasting accuracy there is not a relevant advantage of using V_{or} or V_{opt} , achieving satisfactory results with both algorithms.

Regarding the computational and energy costs, the encoding and decoding phases when V_{or} is applied represent approximately more than 50% of the total costs. This is due to the interpolation carried out during the encoding phase and to encode and decode point by point the whole time-series. However, one modification introduced to V_{opt} is the simplification of the operations during the encoding and decoding phases. Due to this simplification the computational and energy costs of encoding and decoding with V_{opt} are reduced to 2% of the total costs. This is a significant and crucial advantage to use V_{opt} because it boosts the great efficiency advantages of SNNs.

7. Conclusions

The efficiency of the encoding-decoding algorithm is crucial in order to exploit the efficiency advantages of SNNs. In this sense, in this paper an optimisation of the PWM based encoding-decoding algorithm is introduced to one-step ahead forecasting training methodology for SNNs. The methodology is validated with 5 datasets from 5 different application fields.

The results of applying the optimised PWM based encoding-decoding algorithm has shown that the computational and energy costs of the encoding and decoding phases are

practically negligible, while the costs of the original version represent 50% of the total costs. In addition, the methodology provides satisfactory forecasting results, obtaining $MAE \in [0.0017, 0.0357]$ for any dataset.

Author Contributions: S.L. and E.P. contributed equally to this manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This work has been supported by grant IT1726-22 funded by the Basque Government, grant PID2020-112667RB-I00 funded by MCIN/AEI/10.13039/501100011033, NEUROTIP project funded by Programme Euskampus Missions Euskampus Foundation, project PIBA_2020_1_0008 funded by Department of Education of the Basque Government, and project REDGENERA (RED2022-134588-T) funded by Ministry of Science, Innovation and Universities.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: All the databases applied in this work are available or their main characteristics have been detailed during the manuscript so that they can be reproduced.

Conflicts of Interest: The authors declare no conflicts of interest.

References

- de Vries, A. The growing energy footprint of artificial intelligence. *Joule* **2023**, *7*, 2191–2194. [CrossRef]
- Suetake, K.; Ikegawa, S.-i.; Saiin, R.; Sawada, Y. S3NN: Time step reduction of spiking surrogate gradients for training energy efficient single-step spiking neural networks. *Neural Netw.* **2023**, *159*, 208–219. [CrossRef] [PubMed]
- Deng, L.; Wu, Y.; Hu, X.; Liang, L.; Ding, Y.; Li, G.; Zhao, G.; Li, P.; Xie, Y. Rethinking the performance comparison between SNNS and ANNS. *Neural Netw.* **2020**, *121*, 294–307. [CrossRef] [PubMed]
- Fang, W.; Chen, Y.; Ding, J.; Yu, Z.; Masquelier, T.; Chen, D.; Huang, L.; Zhou, H.; Li, G.; Tian, Y. SpikingJelly: An open-source machine learning infrastructure platform for spike-based intelligence. *Sci. Adv.* **2023**, *9*, eadi1480. [CrossRef] [PubMed]
- Bu, T.; Ding, J.; Yu, Z.; Huang, T. Optimized Potential Initialization for Low-Latency Spiking Neural Networks. *Proc. AAAI Conf. Artif. Intell.* **2022**, *36*, 11–20. [CrossRef]
- de Abreu, R.S.; Silva, I.; Nunes, Y.T.; Moiola, R.C.; Guedes, L.A. Advancing Fault Prediction: A Comparative Study between LSTM and Spiking Neural Networks. *Processes* **2023**, *11*, 2772. [CrossRef]
- Qasim Gilani, S.; Syed, T.; Umair, M.; Marques, O. Skin Cancer Classification Using Deep Spiking Neural Network. *J. Digit. Imaging* **2023**, *36*, 1137–1147. [CrossRef] [PubMed]
- Aghabrar, H.; Kiani, K.; Keshavarzi, P. Improvement of pattern recognition in spiking neural networks by modifying threshold parameter and using image inversion. *Multimed. Tools Appl.* **2023**, *83*, 19061–19088. [CrossRef]
- Arriandiaga, A.; Portillo, E.; Espinosa-Ramos, J.I.; Kasabov, N.K. Pulsewidth Modulation-Based Algorithm for Spike Phase Encoding and Decoding of Time-Dependent Analog Data. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *31*, 3920–3931. [CrossRef] [PubMed]
- Lucas, S.; Portillo, E. Methodology based on spiking neural networks for univariate time-series forecasting. *Neural Netw.* **2024**, *173*, 106171. [CrossRef] [PubMed]
- Neftci, E.O.; Mostafa, H.; Zenke, F. Surrogate Gradient Learning in Spiking Neural Networks: Bringing the Power of Gradient-based optimization to spiking neural networks. *IEEE Signal Process. Mag.* **2019**, *36*, 51–63. [CrossRef]
- Lucas, S.; Portillo, E.; Guérin, L.; Cabanes, I. Extensión del algoritmo de codificación-decodificación basado en PWM para Redes Neuronales de Impulsos. In Proceedings of the XLIV Jornadas de Automática, Zaragoza, Spain, 6–8 Septiembre 2023; pp. 168–173. [CrossRef]
- Han, C.S.; Lee, K.M. A Survey on Spiking Neural Networks. *Int. J. Fuzzy Log. Intell. Syst.* **2021**, *21*, 317–337. [CrossRef]
- Gerstner, W.; Kistler, W.M.; Naud, R.; Paninski, L. *Neuronal Dynamics: From Single Neurons to Networks and Models of Cognition*; Cambridge University Press: Cambridge, UK, 2014; pp. 1–577. [CrossRef]
- Yamazaki, K.; Vo-Ho, V.K.; Bulsara, D.; Le, N. Spiking Neural Networks and Their Applications: A Review. *Brain Sci.* **2022**, *12*, 863. [CrossRef] [PubMed]
- Waheeb, W. Mackey-Glass Time Series Dataset. 2016. Available online: https://figshare.com/articles/dataset/Mackey-Glass_time_series/4233584 (accessed on 10 January 2023).
- Salam, A.; Hibou, A.E. Power Consumption of Tetouan City Data Set. 2021. Available online: <https://archive.ics.uci.edu/ml/datasets/Power+consumption+of+Tetouan+city> (accessed on 27 February 2023).

18. Department for Environment Food & Rural Affairs. UK Air Information Resource. Available online: <https://uk-air.defra.gov.uk/data/> (accessed on 15 March 2023).
19. Black, A.W. CMU_ARCTIC Speech Synthesis Databases. Available online: http://festvox.org/cmu_arctic/ (accessed on 7 February 2023).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.