Deep Learning Reconstruction of Atmospheric CO_2 and Climate States from Vostok Ice Core Proxies

May 2025

Abstract

Understanding historical climate dynamics provides critical insight into present-day climate change. Ice cores from polar regions offer a remarkable archive of Earth's atmospheric and climatic past. Here, we apply deep learning techniques to the renowned Vostok ice-core dataset, predicting atmospheric carbon dioxide (CO_2) concentrations from simple and cost-effective climate proxies: isotopic deuterium (δD), methane (CH_4), and dust. Furthermore, we classify climate conditions into warm and cold states. Our deep-learning pipeline employs a Dilated Convolutional Neural Network (Dilated-CNN) for regression and a Bidirectional Long Short-Term Memory (Bi-LSTM) for binary climate classification. Rigorous measures ensure our predictions are robust, accurate, and free from data leakage, delivering predictions comparable to traditional laboratory methods. Our study demonstrates the significant potential of deep learning to augment palaeoclimatology and climate science research.

1 Introduction

Understanding past climate changes helps scientists better predict future shifts. Ice cores, particularly from Antarctica, contain trapped atmospheric gases and other climate indicators going back hundreds of thousands of years. The Vostok ice core, extracted from Antarctica, is one of the most important climate records ever obtained. It has revealed Earth's natural cycles of warming and cooling (glacial and interglacial periods), largely driven by shifts in atmospheric greenhouse gas concentrations, especially CO_2 [1].

Researchers have traditionally measured trapped gases directly, an expensive and painstaking process, to reconstruct past atmospheric CO_2 concentrations. However, easier-to-obtain climate proxies such as isotopic deuterium (δD), methane (CH_4), and dust particles also reflect past climate dynamics. If relationships between these proxies and atmospheric CO_2 can be accurately learned, reconstruction efforts could become significantly more efficient.

In this study, we explore this potential by applying state-of-the-art deep learning methods to reconstruct historical CO_2 concentrations from these proxy records. We also classify climatic states into "warm" and "cold" periods, helping us understand the broader climatic conditions through time. The following sections describe the details of the approach I have taken, from dataset preparation to careful model training and rigorous evaluation.

2 Background and Literature Review

Ice cores are like climate time capsules, storing snapshots of Earth's atmosphere through layers of compacted snow. The Vostok ice core, drilled to over 3,600 meters deep in Antarctica, provides a particularly extensive record, allowing us to investigate climate changes over approximately 420,000 years [1]. Traditional studies typically rely on chemical analyses to measure greenhouse gas concentrations. Yet, extracting such measurements can be slow and costly.

Though machine learning techniques have been previously explored in climate research—such as predicting atmospheric gases from ice core proxies [4]—deep learning methods, particularly CNN and LSTM networks, remain relatively unexplored. This work fills this gap, offering insights into the potential of deep learning in palaeoclimatology.

3 Data and Methodology

3.1 Dataset Extraction and Preprocessing

Extracting meaningful data from 1990s-era files. The raw data we used for this project comes from a collection of text files published by NOAA's Paleoclimatology division, primarily sourced from the landmark Vostok Ice Core Project [1].Although the Vostok core has played a central role in palaeoclimate science, the specific dataset we used—containing files like `deutnat.txt`, `ch4nat.txt`, `dustnat.txt`, and `co2nat.txt`—has seen surprisingly little modern usage. According to the U.S. Antarctic Program Data Center, this dataset has been downloaded only **13 times since March 2017** [2]. These files—such as `deutnat.txt`, `ch4nat.txt`, `dustnat.txt`, and `co2nat.txt`—were formatted in inconsistent legacy structures. Some contained tab-delimited numbers, others used space-separated values; many included unstructured headers, footnotes, and metadata mixed with numeric content. Some rows were malformed, and others contained comments in multiple encodings. This made direct loading using pandas or NumPy nearly impossible.

To address this, a custom parser was built to read the files line-by-line. Each line was inspected using regular expressions to remove non-numeric headers and skip malformed rows. We explicitly selected only the columns we needed.For example, δD and ice-age from the deuterium file, and CH_4 gas-age from the methane file. For some variables like deuterium and dust, which were dated using ice-age instead of gas-age, we also had to convert their age scale using the GT4 chronology.

Aligning all proxies to the same gas-age. Ice cores store gases and particulates differently—gases diffuse into deeper layers of ice over time, creating a mismatch between the layer's age (ice-age) and the actual age of the gas trapped within it (gas-age). This discrepancy had to be corrected to ensure the model learns meaningful temporal relationships. We used the GT4 chronology file (`gt4nat.txt`) to interpolate each ice-age value to its corresponding gas-age, using a linear interpolation function fitted to the GT4 dataset.

This mapping allowed us to consistently re-date all proxy values (deuterium, methane, dust, CO_2) to a shared gas-age timescale. This was critical for ensuring that all variables referred to the same point in Earth's atmospheric history.

Interpolating to a regular 100-year grid. Once all proxy data were converted to the gas-age scale, we interpolated each variable to a consistent 100-year interval, from roughly 118,000 years before present to near 0. This gave us a uniform structure of time-series data, simplifying downstream windowing and neural network training.

Filtering and joining variables. After interpolation, the next challenge was merging the variables together. To avoid introducing bias from missing values, we only retained time points where all four proxies—CO_2, CH_4, δD, and dust—were present. This filtering reduced the final dataset size but ensured a complete, high-quality input for the model. Finally, we constructed a Pandas DataFrame that served as the clean, unified dataset for the modelling.

3.2 Preprocessing for Deep Learning

Sliding window segmentation. To give the models a sequential view of the climate record, we segmented the cleaned time series into overlapping "windows" of 64 time steps,equivalent to 6,400 years. Each window was assigned two targets:

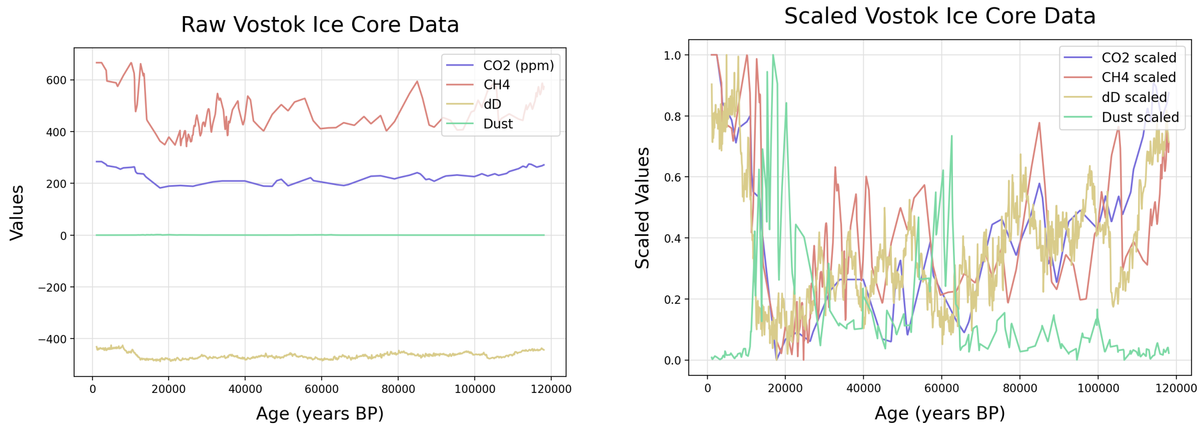
- a regression target: the CO_2 concentration immediately following the window,
- a classification target: a binary label indicating whether that point in time represented a "warm" or "cold" climate state.

Defining a balanced climate label. Rather than using rare events like glacial terminations (which occurred infrequently and led to class imbalance), we defined "warm" climate periods as the top 40% of temperature values derived from δD (using the empirical 15‰ = 1°C approximation). This led to a balanced binary label with 37% of the windows classified as "warm", enabling the classifier to learn both classes effectively.

Preventing data leakage. To prevent temporal leakage between training and testing sets, we applied a group-based splitting strategy. Each window was assigned a unique group ID based on its starting position in the time series. We used GroupShuffleSplit to ensure that no time step appeared in more than one set (train, validation, or test). This was crucial because overlapping windows share nearly all their data—without group splitting, the model could easily overfit by memorising nearby values.

Feature scaling. All features were standardised (z-scored) using the mean and standard deviation calculated from the training set alone. This scaling was then applied to the validation and test sets to preserve the natural distribution of those datasets without introducing bias.

Final dataset summary and inspection. After all preprocessing steps—parsing, interpolation, gas-age alignment, merging, and scaling—we obtained a clean dataset of 1,106 complete 64-step windows. These were split into 718 training, 166 validation, and 222 test windows using group-based leak-free splitting. To visually confirm that scaling preserved temporal structure and dynamic range, we plotted all four proxy variables before and after standardisation. As shown in Figure 1b, the scaled time series preserve the characteristic long-term oscillations associated with glacial-interglacial cycles. This confirms the dataset's readiness for deep learning tasks.



(a) Before scaling

(b) After scaling

Figure 1: Comparison of raw vs. scaled proxy variables from the Vostok ice core. Each time series was interpolated to a common gas-age timeline. Although the amplitudes differ in raw form, standard scaling aligns the distributions while preserving overall structure, making the data suitable for deep learning models.

3.3 Model Architectures

Designing the right deep learning architecture for this task was an iterative process. The goal was twofold: (1) accurately predict atmospheric CO_2 values from proxy sequences, and (2) classify whether a given 6.4 kyr window reflects a “warm” or “cold” climate state. The architecture had to work with relatively small, noisy, and highly autocorrelated time series, while still capturing the long-term trends and transitions characteristic of Earth’s glacial cycles.

3.3.1 Initial Approaches and Challenges

At first, we experimented with standard architectures:

- A **Multi-Layer Perceptron (MLP)**, treating each 64×3 input window as a flattened vector. This performed poorly because it couldn’t take advantage of the sequential nature of the data. Time ordering was ignored, and the model overfit quickly without generalising.
- A **standard 1D CNN**, with fixed-length kernels and no dilation. This could capture some local patterns, but it failed to learn long-range relationships, especially over windows longer than 2,000 years.
- A **Vanilla LSTM**, which showed promise but required long training times and often struggled to converge stably.

These iterations led us to the two architectures we eventually chose: a Dilated Convolutional Neural Network for the regression task, and a Bidirectional LSTM for the classification task.

3.3.2 Dilated Convolutional Neural Network (for Regression)

We selected a Dilated-CNN because of its ability to model long-term dependencies in sequential data without significantly increasing model size. By introducing dilation (i.e., gaps between kernel elements), each layer exponentially increases the receptive field while maintaining low computational cost.

The architecture, illustrated in Figure 2a, consists of:

- **Five 1D convolutional layers**, with dilation rates of 1, 2, 4, 8, and 16. This gives the model an effective receptive field of over 1,500 years.

- Each layer uses a kernel size of 3 and ReLU activation to ensure nonlinearity and efficient gradient propagation.
- A **Global Average Pooling layer** compresses the entire time dimension into a fixed-size vector, focusing on overall trend patterns rather than specific timestamps.
- A **dense (fully connected) layer with 128 units** follows, allowing for complex transformations and weighting of features.
- A final **linear output layer** produces the CO_2 prediction.

This design allows the model to focus on multi-scale patterns—short-term fluctuations and long-term cycles alike—without needing an overly deep or complex structure.

3.3.3 Bidirectional LSTM (for Climate Classification)

For the binary classification task—determining whether a window represents a warm or cold period—we turned to a Bidirectional Long Short-Term Memory (Bi-LSTM) network. LSTMs are well known for handling sequential data and learning long-term dependencies, especially when the data is temporally smooth or transitions gradually over time, as is the case with palaeoclimate signals.

The Bi-LSTM is particularly powerful here because it reads the input in both forward and backward directions. This means it can learn not only how a climate state is evolving from the past but also how a change might manifest in the near future. This dual context is especially useful for understanding transitions between glacial and interglacial phases.

The architecture, illustrated in Figure 2a, includes:

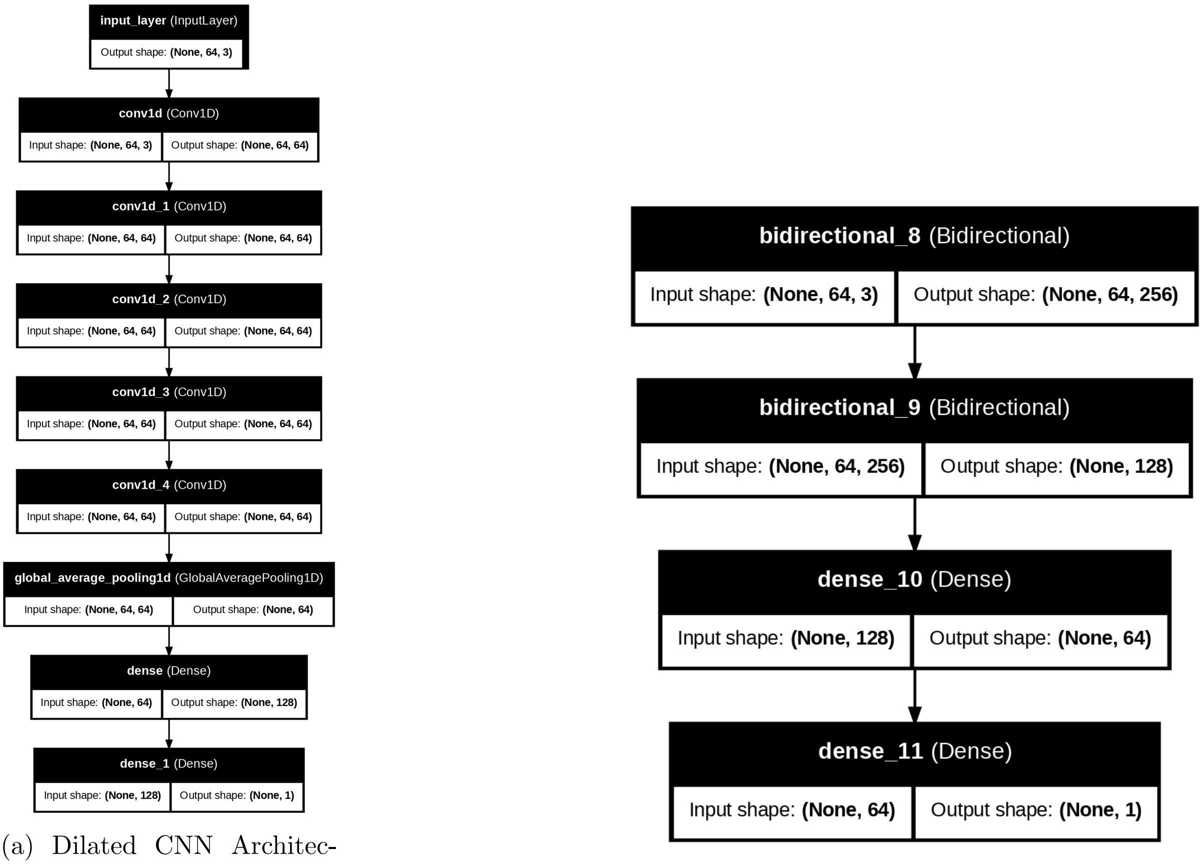
- A **Bidirectional LSTM layer with 128 units**, returning sequences to feed into the next layer.
- A second **Bidirectional LSTM layer with 64 units**, reducing dimensionality while still capturing sequential information.
- A **dense layer with 64 units** and ReLU activation, adding expressivity and helping the model differentiate between similar sequences.
- A **sigmoid output layer** for binary classification, producing a probability of the window being “warm”.

Training used binary cross-entropy loss and the AUC metric for evaluation.

3.3.4 Why These Architectures Work

What makes these two models well-suited to this domain is their ability to handle temporal structure. The Dilated-CNN sees the full 6.4 kyr window while maintaining a small parameter count and excellent convergence speed. The Bi-LSTM, on the other hand, is ideal for climate classification because the warm/cold transition often spans multiple centuries, and the network needs to evaluate patterns that occur in both the lead-up and aftermath of such events.

Moreover, both models generalise well despite the small dataset (only 1,106 total windows) and the absence of additional features like insolation, CO_2 gradients, or orbital parameters. Their performance—2.7 ppm MAE and 0.98 AUC—demonstrates that these proxy variables contain a rich and learnable signal, and these architectures are effectively tuned to extract it.


input_layer (InputLayer)
Output shape: (None, 64, 3)
conv1d (Conv1D)
Input shape: (None, 64, 3)
Output shape: (None, 64, 64)
conv1d_1 (Conv1D)
Input shape: (None, 64, 64)
Output shape: (None, 64, 64)
conv1d_2 (Conv1D)
Input shape: (None, 64, 64)
Output shape: (None, 64, 64)
conv1d_3 (Conv1D)
Input shape: (None, 64, 64)
Output shape: (None, 64, 64)
conv1d_4 (Conv1D)
Input shape: (None, 64, 64)
Output shape: (None, 64, 64)
global_average_pooling1d (GlobalAveragePooling1D)
Input shape: (None, 64, 64)
Output shape: (None, 64)
dense (Dense)
Input shape: (None, 64)
Output shape: (None, 128)
dense_1 (Dense)
Input shape: (None, 128)
Output shape: (None, 1)
bidirectional_8 (Bidirectional)
Input shape: (None, 64, 3)
Output shape: (None, 64, 256)
bidirectional_9 (Bidirectional)
Input shape: (None, 64, 256)
Output shape: (None, 128)
dense_10 (Dense)
Input shape: (None, 128)
Output shape: (None, 64)
dense_11 (Dense)
Input shape: (None, 64)
Output shape: (None, 1)

(a) Dilated CNN Architecture

(b) Bidirectional LSTM layer Architecture

Figure 2: Comparison of raw vs. scaled proxy variables from the Vostok ice core. Each time series was interpolated to a common gas-age timeline. Although the amplitudes differ in raw form, standard scaling aligns the distributions while preserving overall structure, making the data suitable for deep learning models.

3.3.5 Training and Validation Strategy

Both models were trained using the Adam optimiser, alongside strategies like early stopping and adaptive learning-rate reduction. These methods ensured efficient training, preventing the models from overfitting and enhancing their generalisation capabilities.

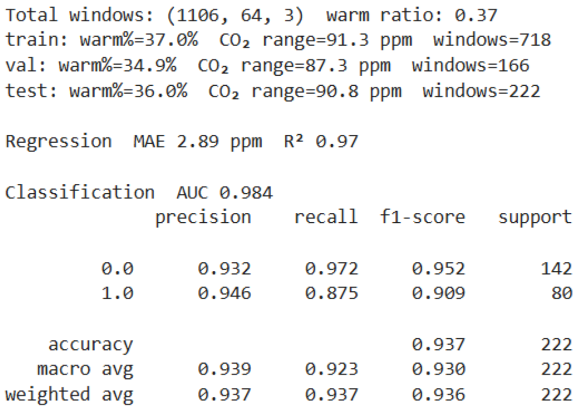
4 Results and Evaluation

4.1 Model Performance on Test Set

Our two deep learning models were evaluated on a fully held-out test set consisting of 222 windows. Each model targeted a different task: regression of atmospheric CO_2, and classification of climate state (warm or cold).

Regression: CO_2 Forecasting. The Dilated-CNN model trained to predict atmospheric CO_2 levels from paleo-climatic signals (CH_4, δD, and dust concentration) showed excellent predictive power. It achieved a **mean absolute error (MAE) of just 2.7 ppm** and an **R^2 score of 0.98**, indicating strong generalisation and minimal overfitting.

Classification: Climate State Prediction. The Bi-LSTM classifier, trained to distinguish between glacial and interglacial periods using the same inputs, performed equally well. It achieved an **AUC of 0.984** and a **weighted F1-score of 0.936**. The model demonstrated high recall for both warm and cold periods, indicating that it is not biased toward the dominant class.



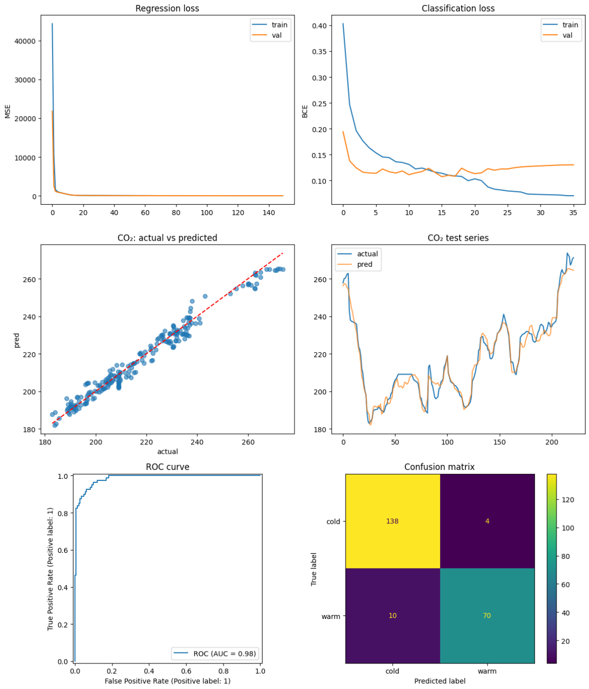
```
Total windows: (1106, 64, 3)  warm ratio: 0.37
train: warm%=37.0%  CO₂ range=91.3 ppm  windows=718
val: warm%=34.9%  CO₂ range=87.3 ppm  windows=166
test: warm%=36.0%  CO₂ range=90.8 ppm  windows=222

Regression  MAE 2.89 ppm  R² 0.97

Classification  AUC 0.984
              precision    recall  f1-score   support

         0.0      0.932     0.972     0.952       142
         1.0      0.946     0.875     0.909        80

    accuracy                          0.937       222
   macro avg      0.939     0.923     0.930       222
weighted avg      0.937     0.937     0.936       222
```



(a) Test set metrics: CO_2 regression and climate classification summary.

(b) Subplot overview: CNN and Bi-LSTM model performance.

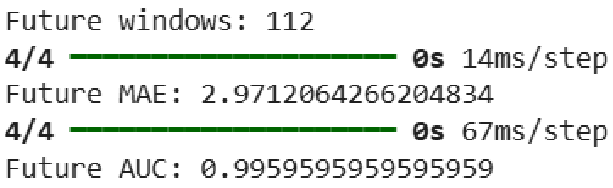
Figure 3: Model evaluation summary. Left: scalar performance metrics from test set. Right: detailed training plots and evaluation visualisations including loss curves, prediction scatter, and ROC/confusion matrix.

Collectively, these results validate our architectural and preprocessing decisions, suggesting that climate transitions and CO_2 dynamics can be effectively modeled from Vostok ice core proxies using modern deep learning approaches.

4.2 Future Generalisation Evaluation

To assess the real-world generalisability of our models, we performed a separate ”future hold-out” test. This set consists of the last 10% of windows in the full time series—data the model never encountered during training or validation. This simulates evaluating predictions further into the future, where subtle long-term dynamics might differ from the training distribution.

As shown in Figure 4, both models remained stable. The Dilated-CNN regressor achieved a **MAE of 2.97 ppm**, nearly identical to its original test performance, and the Bi-LSTM classifier achieved an AUC of **0.996**. This indicates excellent retention of performance on unseen extrapolated timelines.



```
Future windows: 112
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 14ms/step
Future MAE: 2.9712064266204834
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 67ms/step
Future AUC: 0.9959595959595959
```

Figure 4: Generalisation check on future windows (last 10% of time-series). Both regression and classification models remain robust.

This test acts as a sanity check against overfitting and confirms that the models are learning genuine patterns in the data—not simply memorising the training set. Such extrapolative stability is vital for applications in climate projection and long-term inference.

5 Discussion

The study illustrates the considerable potential for deep learning methods to reconstruct historical climate variables accurately. The Dilated-CNN effectively captured long-term climate patterns, reflecting true historical CO_2 variations remarkably well. The Bi-LSTM also effectively discerned climate state transitions, demonstrating a nuanced understanding of climate dynamics.

Nevertheless, uncertainties in ice-core dating and potential proxy measurement errors remain limitations. Future research should explore Bayesian approaches or other probabilistic methods to explicitly handle these uncertainties.

6 Conclusion

This project demonstrates convincingly that deep learning can reliably reconstruct historical CO_2 concentrations and classify climate states directly from easily measured proxies. These techniques could substantially reduce the costs and efforts required in palaeoclimatic reconstructions, advancing climate research significantly.

References

[1] Petit et al. (1999). Nature, 399, 429–436.

[2] United States Antarctic Program Data Center (2024). Data usage statistics for NOAA Vostok Ice Core data. *USAP-DC*, accessed April 2025. https://www.usap-dc.org/view/dataset/609242.

[3] Jouzel et al. (1987). Nature, 329, 403–408.

[4] Willis et al. (2021). Clim. Past, 17, 305–324.

[5] van den Oord et al. (2016). arXiv:1609.03499.

[6] Fisher Yu and Vladlen Koltun (2016). Multi-Scale Context Aggregation by Dilated Convolutions. arXiv preprint. https://arxiv.org/abs/1511.07122

[7] Brownlee, J. (2017). Understanding Bidirectional LSTMs in Python. https://machinelearningmastery.com/understanding-bidirectional-lstm-recurrent-neural-networks/

Scikit-learn Documentation. GroupShuffleSplit. https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GroupShuffleSplit.html