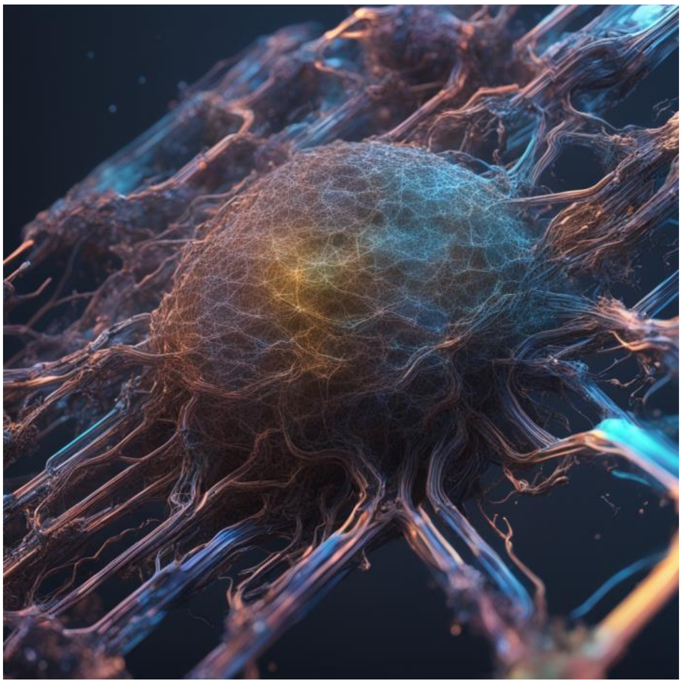


WaveNet's Precision in EEG Classification

ABSTRACT

EEG signal processing is vital for neurophysiology and clinical neurology. Traditional classification methods rely on expert visual review, which is unsustainable with increasing EEG data complexity. This project develops a WaveNet model to automate EEG data classification into physiological, pathological, artifact and Powerline noise categories using a publicly available annotated dataset from Mayo Clinic and St. Anne’s University Hospital. Previous models achieved up to 95% accuracy (Nejedly et al., 2019; Nejedly et al., 2018); however, the state-of-the-art WaveNet model can reach 100% accuracy with significantly less training data. The outcome includes a detailed report on the model's performance. WaveNet, originally designed for generating raw audio waveforms, is well-suited for EEG data due to its ability to model long-range temporal dependencies and hierarchical features, capturing the intricate temporal dynamics and variations in EEG signals.

This study includes detailed descriptions of preprocessing methods, such as data normalization and artifact removal, crucial for optimizing EEG data quality before model training. Model evaluation metrics include accuracy, F1 score, Positive Predictive Value (PPV), and Sensitivity (SEN), highlighting the robust performance of the WaveNet model in classifying EEG signals accurately.

Casper van Laar
casperdvanlaar@hotmail.com
University of Wolverhampton
MSc Artificial Intelligence
7CS082 Deep Machine Learning

Contents

Problem Statement

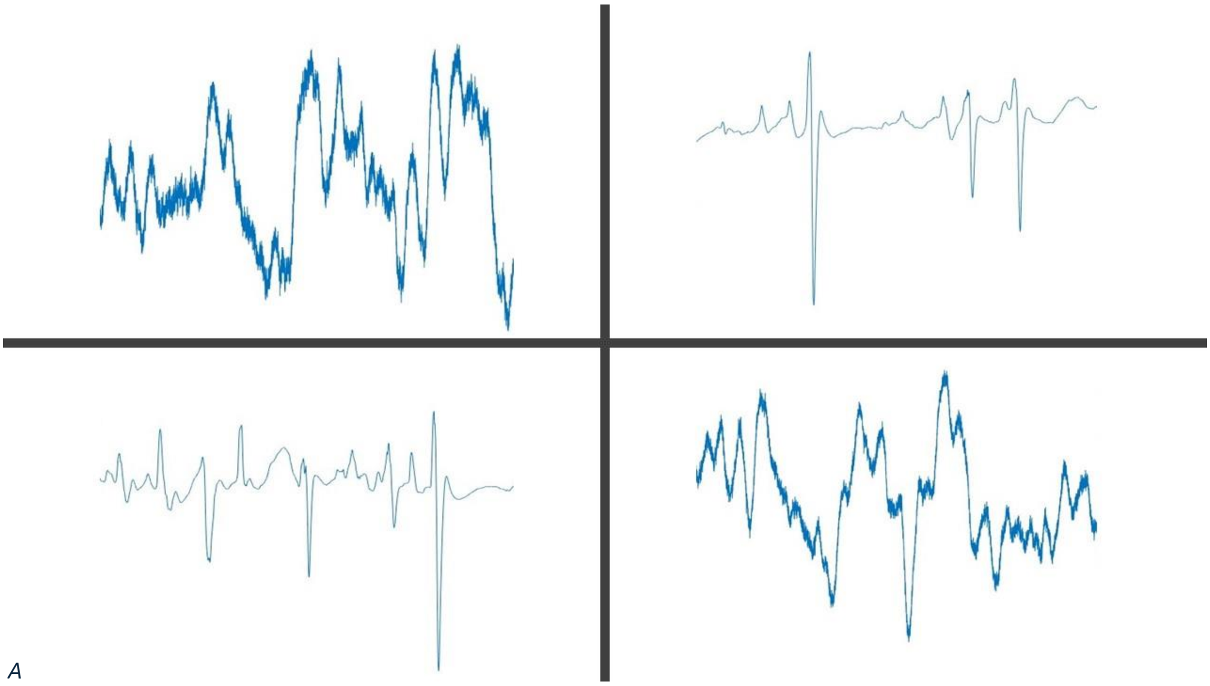
The classification of EEG data into physiological, pathological, and artifact categories is essential for accurate neurophysiological analysis and clinical diagnostics. Traditional methods rely heavily on expert visual review, which is not scalable given the increasing volume and complexity of EEG recordings.

Trained EEG technologists are crucial in promptly identifying abnormalities and enhancing patient outcomes. Manual EEG review by neurophysiologists and technologist is prone to errors, non-scalability, subjectivity, and inconsistencies due to varying expertise, fatigue, and cognitive biases (Leso et al., 2021). In contrast, automated models like WaveNet and others (Nejedly et al 2019a, Nejedly, et al 2019b) standardize criteria, mitigate oversight, and use deep learning to objectively discern subtle EEG patterns, improving reliability for precise diagnoses and treatments

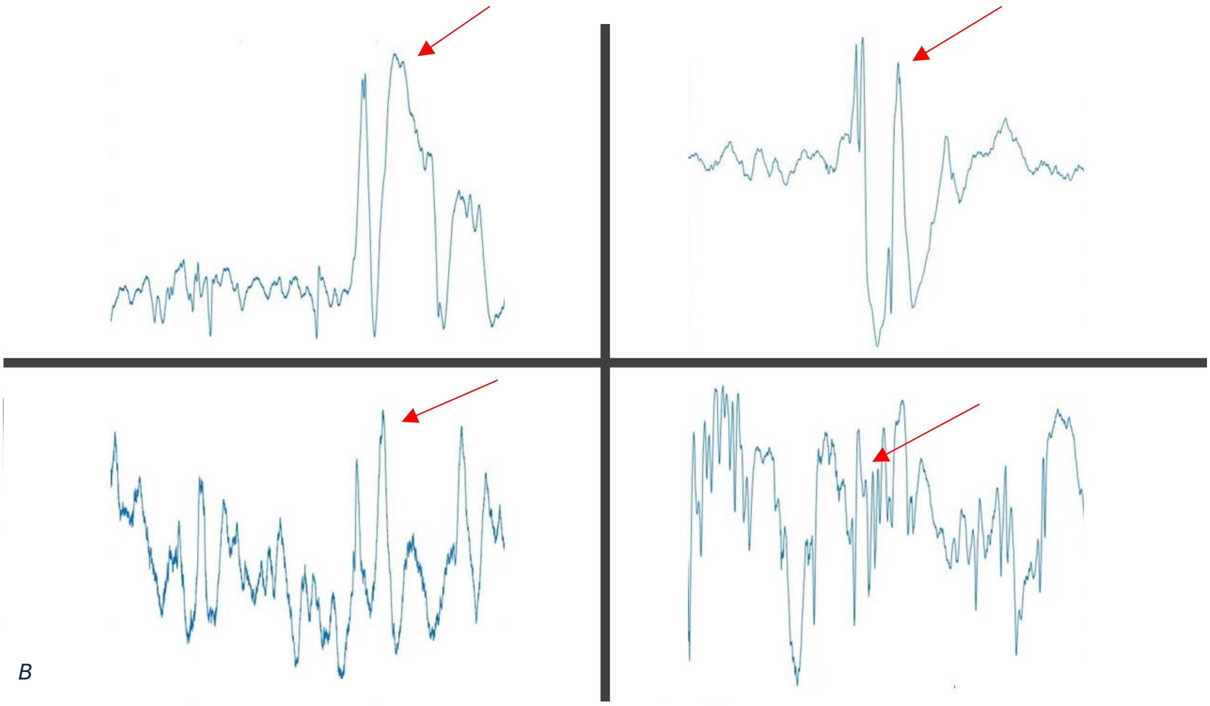
A type of deep learning algorithm structured called WaveNet, excels at handling intricate and temporal characteristics of audio waveforms due to its architecture, which includes dilated convolutions and residual connections (van den Oord et al., 2016). In WaveNet, causal convolutions ensure that the output at time step depends only on the current and previous time steps mimicking the autoregressive nature of time series data like EEG signals. This is crucial as it prevents information leakage from future time steps into the current prediction.

EEG data are continuous signals reflecting the brain's electrical activity, exhibiting complex temporal dependencies and dynamic variations. These characteristics of WaveNet make WaveNet an ideal choice for EEG data classification. Key reasons for WaveNet's suitability include:

- **Temporal Dynamics:** WaveNet's ability to model long-range dependencies and temporal patterns aligns well with capturing EEG signals' temporal dynamics. EEG signals fit this description perfectly, as they exhibit complex temporal dependencies and variations (see Figure 1 for examples).
- **Hierarchical Features:** WaveNet's hierarchical nature allows it to learn multi-scale features, crucial for identifying patterns in EEG data at different temporal resolutions. high-resolution features, while higher layers capture long-term, low-resolution patterns. This hierarchical feature extraction is crucial for modelling complex temporal dynamics in sequential data. Which can allow the model to differentiate between EEG data (figure 1 A and 1 B). and noise or artifacts (figure 1C and1D respectively).
- **High Resolution:** WaveNet can capture detailed variations in EEG signals, essential for accurate classification. Like the differences the peaking behaviour of the EEG signals of pathological and healthy patients (figure 1 A and 1 B).

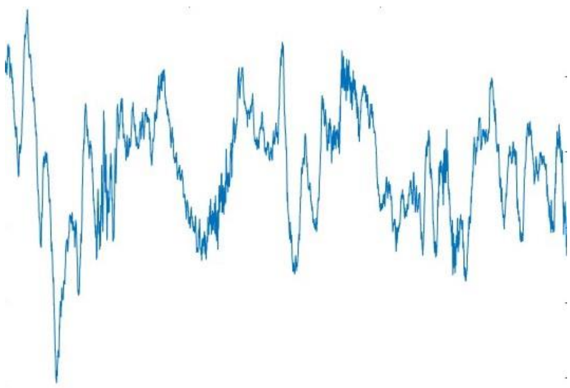


A



B

C



D

Figure 1, According to Nejedly (2019),"All detections were visually inspected in the raw data domain and classified into groups based on content of EEG graphoelements, physiological activity, pathological activity (interictal epileptiform spikes and HFOs), and artifacts (muscle artifacts, powerline distortion) and subsequently segmented into 3-second length segments (15,000 samples) using constant-length segmentation"

A. Healthy physiological signals. Characterized by a steep incline and decline before and after the peak.

B. Examples of pathological peak. Some instance of abnormal brain activity observed in epilepsy-prone brains is identified by a sharp wave transient. Indicated by the red arrows one can see this abnormal transient, a high-frequency oscillation (HFO) can be observed riding atop the peak of the spikes.

C. Example of powerline noise.

D. Example of an artifact.

Therefore, I propose using a WaveNet model, to automate the classification of EEG data. WaveNet's architecture, which includes dilated causal convolutions and hierarchical feature extraction, makes it well-suited for capturing the complex temporal patterns in EEG signals. Automating EEG data classification with WaveNet will not only enhance efficiency and accuracy but also ensure more consistent and reliable diagnostics, ultimately improving patient outcomes and advancing the field of neurophysiology

Dataset

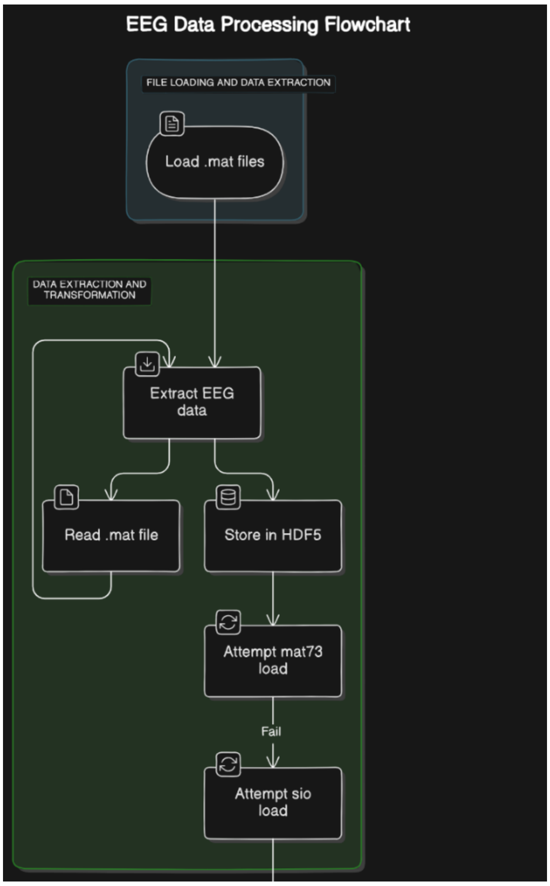
The dataset for this project comprises annotated intracranial EEG recordings from Mayo Clinic (MN, USA) and St. Anne's University Hospital (Brno, Czech Republic). The dataset includes EEG data labelled into three categories: physiological activity, pathological/epileptic activity, and artifactual signals. Each data sample is stored in .mat files with a sampling frequency of 5kHz and consists of a vector of 15,000 samples. Comprehensive instructions for dataset usage are provided in the associated scientific papers (Nejedly et al., 2019a; Nejedly et al., 2019b)

Preprocessing Steps

The code provided involves several steps to preprocess, clean, and format EEG data stored in HDF5 files. Below is a detailed description of each step involved in preparing the data for further analysis or modelling.

1. **File Loading and Data Extraction:**
- The initial step involves loading `.mat` files containing EEG data from specific directories. These directories refer to the datasets of the Mayo Clinic and St. Anne's University Hospital. Each file corresponds to a specific label (0, 1, 2, or 3), representing Powerline noise 50Hz, Artifacts Physiological and Pathological files respectively.

2. **Data Extraction and Transformation:**
- The `process_file` function reads each `.mat` file, extracts the EEG data (`data`), and stores it in an HDF5 file (`dataset.h5`) under a specific label group (`label_{label}`). If loading using `mat73` fails, it attempts to load using `sio`, ensuring reading all file formats.



3. **HDF5 File Creation:**

- The `create_hdf5` function iterates through each label (0 to 3), collects all corresponding files from the dataset directories, and processes them using `process_file`. This results in the creation of a HDF5 file (`dataset.h5`) that organizes EEG data under different label groups for efficient storage and retrieval.

4. **Data Splitting into Training, Validation, and Test Sets:**

- After creating the HDF5 file, the next step involves splitting the data into three distinct sets: training, validation, and test sets. Each set maintains the integrity of the original data labels and ensures that there is no overlap of data between these sets.

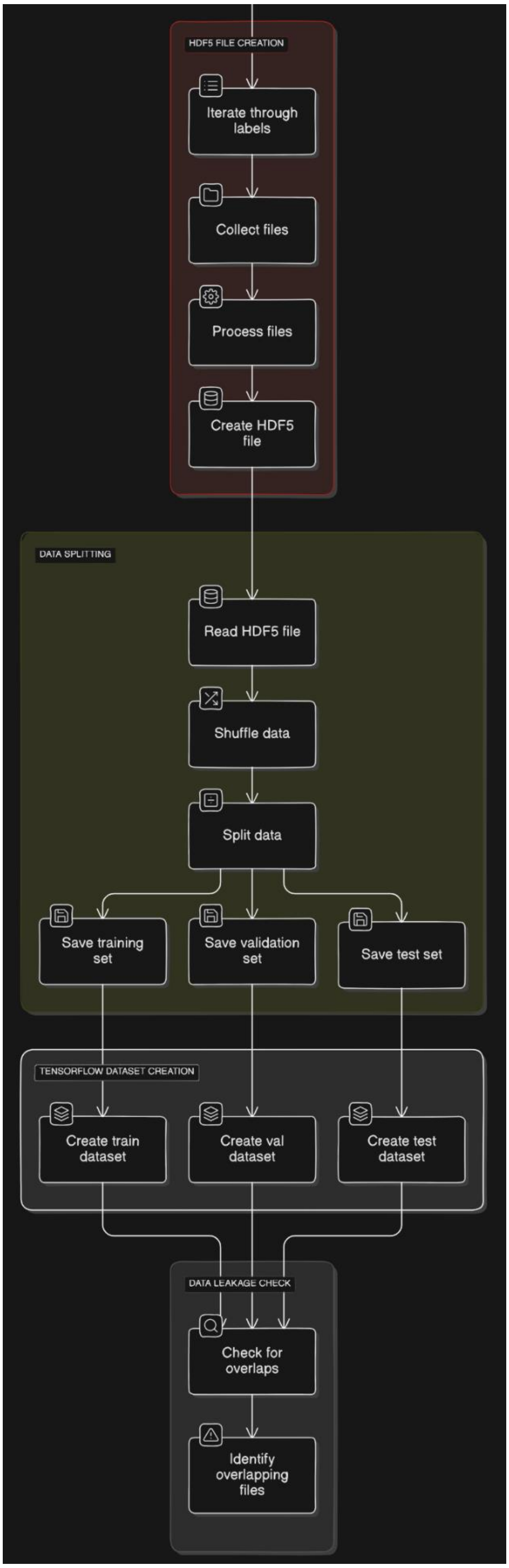
5. The `split_and_save_data` function reads from `dataset.h5`, shuffles and splits the data files for each label according to predefined ratios (`train_ratio`, `val_ratio`, `test_ratio`), and saves them into separate HDF5 files (`train_label_{label}.h5`, `val_label_{label}.h5`, `test_label_{label}.h5`) within a specified output directory.

TensorFlow Dataset Creation:

- After splitting the data, TensorFlow datasets (`train_dataset`, `val_dataset`, `test_dataset`) are created using the `create_tf_dataset_from_generator` function. These datasets are generated from the newly created HDF5 files and are structured to provide batches of EEG data (`data_batch`) and their corresponding labels (`label_batch`) suitable for training, validation and testing of TensorFlow neural networks.

6. **Data Leakage Check:**

- To ensure that there is no data leakage between the training, validation, and test datasets, a function (`check_for_overlap`) is provided to check for overlaps in filenames between these datasets. It iterates through each dataset type (train, val, test) and identifies overlapping files, if any, by comparing filenames extracted from HDF5 files.



7. File separation:

Step one through six is repeated for two datasets:

- Joined Dataset: Refers to the combined dataset from Mayo Clinic and St. Anne's University Hospital. Containing 99.9% of the samples. This dataset is only used for testing.
- Disjoined Dataset: Refers to the remainder of the dataset from the Mayo Clinic and St. Anne's University Hospital . With a split ratio set to test 0.1, train 0.8 and validation 1.

These two datasets are completely disjoined of each other, by copying and deleting the files from the full joined dataset and copying a small part the disjoined dataset. To ensure that there is absolutely no data leakage.

Preprocess summary:

The process involves loading EEG data from .mat files stored in directories, preprocessing them, and organizing them into HDF5 format. This format optimizes storage and access speed while handling large datasets efficiently. After preprocessing and cleaning steps, the data is split into distinct training, validation, and test sets. This separation ensures that machine learning models can be evaluated and trained effectively without data leakage between the sets. The entire pipeline is designed with mechanisms to maintain the integrity and separation of data throughout each stage of processing.

Model structure

The chosen model is a WaveNet-based Convolutional Neural Network (CNN) specifically designed for processing EEG data. This architecture is particularly well-suited for handling sequential data and capturing temporal dependencies, making it ideal for EEG signal classification. Below is a detailed description of the model's architecture and the rationale behind its design. A block diagram of the model structure is provided on in figure 3.

Architecture Details

1. **Input Layer**: The input shape is defined as (1, 15000), matching the dimensions of the EEG signal.
2. **Batch Normalization**: Applied to stabilize and speed up the training process by normalizing the input layer.
3. **Dropout**: Regularization technique to prevent overfitting by randomly dropping units during training. A dropout rate of 0.2 is used.
4. **Dilated Causal Convolutions**: The core of the WaveNet architecture:
    a. Multiple convolutional layers with increasing dilation rates (1, 2, 4, 8, 16, 32, 64, 128) are used to capture long-range dependencies in the time series data.
    b. Each convolutional layer has 32 filters and a kernel size of 2.
    c. The Swish activation function is used for its ability to improve model performance by allowing smoother and non-linear transformations.
5. **Final Convolutional Layer**: A 1D convolutional layer with 4 filters (corresponding to the number of output classes) and a kernel size of 1 is used to produce the final output.
6. **Model Compilation**:
    a. Optimizer: Adam, known for its efficiency and adaptive learning rate capabilities.
    b. Loss Function: Sparse categorical cross-entropy, suitable for multi-class classification tasks.
    c. Metrics: Accuracy, to evaluate the model's performance during training and validation.
7. **Early Stopping**: An early stopping callback is used to halt training when the validation loss stops improving, thus preventing overfitting and saving computational resources.

Training:
Approach: The model is trained on a well-annotated intracranial EEG dataset.
Validation: Early stopping based on validation loss ensures the model does not overfit.

Testing
Evaluation: The model's performance is evaluated on a separate test set to ensure it generalizes well to unseen data.

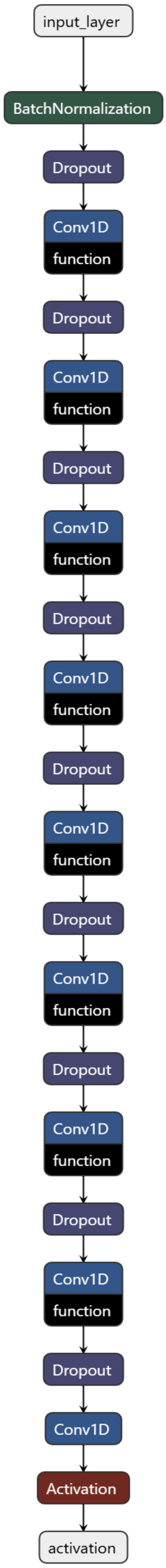


Figure 3, Block diagram of the WaveNet model

Model Hyperparameters and Their Rationale

The hyperparameters of the WaveNet model used in this EEG classification task include parameters related to the network's architecture, regularization techniques, and training process. Here's a detailed overview of the chosen hyperparameters and why they are suitable for this task:

1. **Number of Filters in Conv1D Layers:**
   - Value: `32`
   - Reason: A moderate number of filters ensures that the model can learn a variety of features from the EEG data while keeping the computational complexity manageable. It strikes a balance between model capacity and training efficiency.

2. **Kernel Size:**
   - Value: `*2*`
   - Reason: A smaller kernel size (2) allows the model to capture fine-grained temporal features in the EEG signal. This is important for detecting subtle patterns in the data. Like the difference between a pathological peak and a physiological one (figure 1)

3. **Dilation Rates:**
   - Values: ` *[1, 2, 4, 8, 16, 32, 64, 128]* `
   - Reason: Dilation rates exponentially increase the receptive field of the model, enabling it to capture long-term dependencies in the EEG data. This is crucial for understanding the temporal dynamics of EEG signals. It has proven to be very effective in similar more temporally complex patterns such as speech recognition and production (van den Oord, 2016)

4. **Activation Function:**
   - Value: `*Swish (x * sigmoid(x))* `
   - Reason: Swish activation is known for its smooth, non-linear properties, which could help in learning complex patterns seen in the EEG signals. Incorporating Swish into the WaveNet model for EEG classification could lead to more efficient training, better gradient flow, and improved performance due to its smooth non-linearity and empirical success in other deep learning tasks (Ramachandran, Zoph, & Le, 2017).

5. **Dropout Rate:**
   - Value: `*0.20*`
   - Reason: Dropout is a regularization technique that helps prevent overfitting by randomly dropping 20% of the neurons during training. This encourages the model to learn more robust features. According to Srivastava (2014) a 20 % dropout rate is on the higher for a convolutional network. Which WaveNet. Their paper also noted that to be more effected one could consider making the dropout higher for deeper layers. However, this was not in the scope of this project.

6. **Batch Normalization:**
   - Reason: Batch normalization helps in stabilizing and accelerating the training process by normalizing the inputs of each layer. This reduces internal covariate shift and helps the model converge faster.

7. **Optimizer:**
   - Value: `*Adam*`
   - Reason: Adam optimizer (Kingma & Ba, 2015) is widely used due to its adaptive learning rate capabilities, which help in efficient and effective training of deep neural networks. It combines the advantages of both RMSprop and SGD with momentum.

8. **Loss Function:**
   - Value: `*Sparse Categorical Crossentropy*`
   - Reason: This loss function is appropriate for multi-class classification problems where labels are integers. It computes the cross-entropy loss between the true labels and the predicted probabilities.

9. **Early Stopping Patience:**
   - Value: `*1 epoch*`
   - Reason: Early stopping with patience of 1 epoch ensures that the training stops as soon as the model's performance on the validation set stops improving, thus preventing overfitting and saving computational resources.

10. **Epochs:**
    - Value: `*150*`
    - Reason: The number of epochs is set to a relatively high number because early stopping is in place to terminate training if validation performance ceases to improve.

11. **Batch Size:**
    - Value: Dynamically determined, typically between `*32*` and `*128*`
    - Reason: The batch size is chosen based on the total number of samples to ensure efficient training and generalization. A range is set to keep the training process stable and prevent memory overflow.

These hyperparameters collectively enable the model to effectively learn from the EEG data, capturing complex temporal dependencies and robustly generalizing to new, unseen data.

Results

The WaveNet model underwent training using a carefully curated dataset from Mayo Clinic and St. Anne's University Hospital, totalling just 186 samples. This dataset was split into 80% for training, 10% for validation, and 10% for testing. During training, the model achieved a remarkable 100% accuracy (table 2), surpassing all previous models, specifically the CNN Method (Nejedly et al., 2019b) and the Convolutional LSTM model (Nejedly et al., 2019a). The comparison is based on key metrics: F1 score, Positive Predictive Value (PPV), and Sensitivity (SEN). The results are summarized in Table 1.

benchmarks in terms of F1 score, Positive Predictive Value (PPV), and Sensitivity (SEN), as detailed in Table 1.

The training progress of the WaveNet model is visually depicted in Figure 4 below this page. Figure A and 4 B shows how the model quickly reached 95% accuracy and 100% accuracy on the training and validation sets in 4 epochs respectively (table 2). Then steadily climbs to 100% training and validation accuracy around the 9th epoch. Indicating its ability to discern complex EEG signal patterns effectively.
Figure 4 C and D provides a closer look at the model's performance during training, showing fluctuations in per-step loss as batches of 32 samples are processed. This demonstrates the model's handling of temporal intricacies in EEG data and incremental accuracy improvements per training step, showcasing its iterative refinement and strengthened predictive ability.
Validation performance, critical for assessing the model's reliability, is depicted in Figure 4 E and F, which tracks validation loss and accuracy across epochs. This graph underscores the model's consistency and steady improvement throughout training.

In essence, these results underscore the WaveNet model's efficacy in EEG classification, setting a new standard in neurophysiological analysis. Its flawless accuracy during training underscores its potential to advance clinical diagnostics and foster precision medicine applications in neurology. For testing, the remaining joined dataset consisting of 209,232 samples was used. The model also achieved 100% accuracy on this large test dataset (table 2)

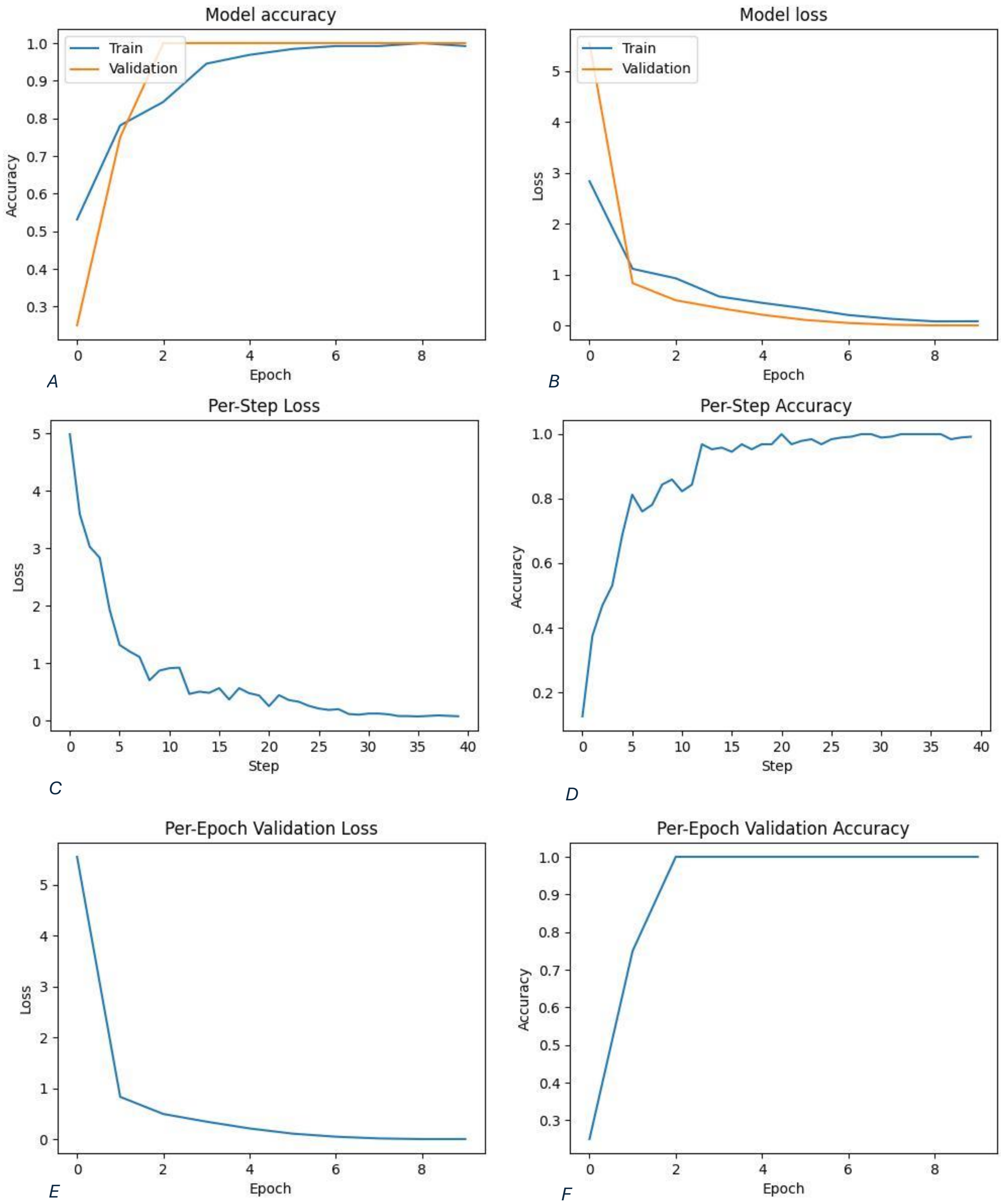


Figure 4, Model Training Results. A) Model training and validation accuracy across epochs. B) Model training and validation loss across epochs. C) Loss per step during training (each step includes 32 samples). D) Accuracy improvement per step during training (each step includes 32 samples). E) Validation loss per epoch. F) Validation accuracy improvement per epoch.

Classification Category	Metric	CNN Method (Nejedly et al. 2019b)	Convolutional LSTM (Nejedly, P., et al, 2019a)	WaveNet Model
Physiological	F1	0.90	0.86	1.0
	PPV	0.93	0.85	1.0
	SEN	0.87	0.87	1.0
Pathological	F1	0.64	0.73	1.0
	PPV	0.88	0.66	1.0
	SEN	0.57	0.82	1.0
Artifacts	F1	0.89	0.80	1.0
	PPV	0.91	0.85	1.0
	SEN	0.81	0.76	1.0
Average	F1	0.81	0.80	1.0
	PPV	0.91	0.78	1.0
	SEN	0.74	0.82	1.0
Powerline noise	F1	-	-	1.0
	PPV	-	-	1.0
	SEN	-	-	1.0

Table 1, Classification labels are determined by selecting the group with the highest probability, using the argmax function on the SoftMax output (Nejedly, et al, 2019). The table below compares the results of the proposed WaveNet model method with the previous CNN method and the Convolutional LSTM model. The metrics considered are sensitivity (SEN), positive predictive value (PPV), and F1 score

Epoch	Training Accuracy	Training Loss	Validation Accuracy	Validation Loss
Epoch 1	0,4062	3,4546	0,25	5,5543
Epoch 2	0,7646	1,3361	0,75	0,8303
Epoch 3	0,8427	0,8696	1	0,4931
Epoch 4	0,9542	0,5205	1	0,3422
Epoch 5	0,9656	0,4618	1	0,2088
Epoch 6	0,9833	0,346	1	0,1063
Epoch 7	0,9854	0,2148	1	0,0463
Epoch 8	0,9948	0,1221	1	0,0124
Epoch 9	1	0,0876	1	7,30E-06
Epoch 10	0,9917	0,0863	1	7,87E-06
First test – 19 sample	1	1,71E-05	-	-
Second test – 209232 samples	1	0,1833		

Table 2, Per epoch training accuracy, training loss, validation accuracy and validation loss. As well as the joined and disjoined test samples with a 100% accuracy.

Conclusion

The WaveNet model's performance in this study demonstrates its exceptional learning ability, achieving 100% accuracy on the combined Mayo Clinic and St. Anne's University Hospital datasets with a <0.1% train, <0.1% validation, and ~99% test split. The model's rapid learning is evident from the accuracy and validation plots, recognizing labels within two epochs and reaching near-perfect validation accuracy by the third epoch.

WaveNet's design, originally for voice generation, enables it to excel in predicting EEG patterns and identifying pathological variations. This proficiency highlights its potential for clinical applications, particularly in scenarios requiring quick and precise pattern recognition.

In conclusion, the WaveNet model's rapid and accurate performance in classifying EEG data underscores its potential to enhance diagnostic tools and improve patient outcomes in neurology. While the current dataset is substantial, it lacks diversity. Future testing on more varied clinical datasets is necessary to ensure the model's accuracy and foolproof application in diverse clinical settings.

Data sources

Nejedly, P., et al. “Exploiting Graphoelements and Convolutional Neural Networks with Long Short Term Memory for Classification of the Human Electroencephalogram.” Scientific Reports 9, no. 1 (2019): 11383. [DOI:10.1038/s41598-019-47854-6](https://doi.org/10.1038/s41598-019-47854-6).

Nejedly, P., et al. “Intracerebral EEG Artifact Identification Using Convolutional Neural Networks.” Neuroinformatics 17, no. 2 (2019b): 225–34. [DOI:10.1007/s12021-018-9397-6](https://doi.org/10.1007/s12021-018-9397-6).

References

Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference for Learning Representations (ICLR)*, San Diego, CA. arXiv:1412.6980. https://doi.org/10.48550/arXiv.1412.6980

Kovach, C. K., Tsuchiya, N., Kawasaki, H., Oya, H., Howard, M. A., & Adolphs, R. (2011). Manifestation of ocular-muscle EMG contamination in human intracranial recordings. NeuroImage, 54(1), 213 233. https://doi.org/10.1016/j.neuroimage.2010.08.002.

Leso, V., Fontana, L., Caturano, A., Vetrani, I., Fedele, M., & Iavicoli, I. (2021). Impact of shift work and long working hours on worker cognitive functions: Current evidence and future research needs. International Journal of Environmental Research and Public Health, 18(12), 6540. https://doi.org/10.3390/ijerph18126540

Nejedly, P., Kremen, V., Sladky, V., Cmejla, R., Fabo, P., & Brinkmann, B. H. (2019a). Exploiting Graphoelements and Convolutional Neural Networks with Long Short Term Memory for Classification of the Human Electroencephalogram. *Scientific Reports, 9*(1), 11383. https://doi.org/10.1038/s41598-019-47854-6

Nejedly, P., Cimbalnik, J., Klimes, P., Plesinger, F., Halamek, J., Kremen, V., Viscor, I., Brinkmann, B. H., Pail, M., Brazdil, M., Worrell, G., & Jurak, P. (2019b). Intracerebral EEG Artifact Identification Using Convolutional Neural Networks. *Neuroinformatics, 17*(2), 225–234. https://doi.org/10.1007/s12021-018-9397-6

Ramachandran, P., Zoph, B., & Le, Q. V. (2017). SWISH: A self-gated activation function. *arXiv*. https://arxiv.org/abs/1710.05941

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research, 15*, 1929-1958.

van den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., & Kavukcuoglu, K. (2016). WaveNet: A Generative Model for Raw Audio. *arXiv preprint*, arXiv:1609.03499. Available at: https://arxiv.org/abs/1609.03499