Development of a Deep Learning - Based EEG classifier for Early Diagnosis of Dementia.

Sushma Hedge, Surya Narayan Swamy, Nimesh Prabhakar, Faith Sang

Abstract

Electroencephalography (EEG) signals offer a noninvasive window into brain function that can aid in the differential diagnosis of dementia. In this project, we developed a deep learning model to automatically classify Alzheimer's Disease (AD), Frontotemporal Dementia (FTD), and healthy control subjects from resting state EEG. Our model employs a hybrid Convolutional Neural Network – Long Short Term Memory (CNN-LSTM) architecture trained on 1-second EEG segments (128 samples) from 88 subjects, with class balancing via weighted loss. The CNN-LSTM learns spatial features across EEG channels and temporal dynamics within each segment, allowing it to distinguish dementia subtypes without hand crafted features. The final model achieved a 94% accuracy on the held out test set, with precision and recall in the 94% range for all three classes, indicating balanced performance. This substantially outperforms earlier approaches including traditional machine learning on spectral features and simpler deep networks. Our results demonstrate that combining spatial and temporal feature learning on raw EEG yields an accurate and generalizable classifier for differentiating AD, FTD, and healthy.

Introduction

Dementia is a clinical syndrome with multiple etiologies, among which Alzheimer's Disease (AD) and Frontotemporal Dementia (FTD) are common forms. Distinguishing AD from FTD is clinically important due to different prognoses and management, but it can be challenging because of overlapping cognitive and behavioral symptoms. EEG is a cost effective and noninvasive tool that measures brain electrical activity and has shown sensitivity to the functional changes in dementia. AD patients typically exhibit a slowing of EEG rhythms (e.g. reduced alpha power, increased theta/delta), while FTD may have different patterns, often involving frontal lobe dysfunction, yet preserving some posterior rhythms. These differences suggest EEG based biomarkers could assist in differential diagnosis of dementia subtypes.

Prior research on EEG in dementia has mostly focused on binary classification (e.g. AD vs. healthy controls) or on characterizing EEG spectral differences. Traditional quantitative EEG analyses use features such as band power (alpha, theta ratios), entropy measures, or functional connectivity to distinguish AD from normal. Such approaches, combined with classifiers like Support Vector Machines or Random Forests, have achieved moderate success (often 75–85% accuracy for AD vs healthy). However, distinguishing between two dementia types (AD vs FTD) using EEG has proven more difficult, with earlier studies reporting lower accuracies (<75%) due to overlapping EEG features. This has led to an under exploration of multiclass EEG classification involving FTD, partly because large EEG datasets including FTD patients are rare. In recent work, deep learning methods have begun to show promise in

EEG based diagnosis tasks. For example, EEGNet, a compact convolutional neural network tailored for EEG, has been used in various brain computer interfaces and clinical EEG classification. Likewise, architectures like DeepConvNet (a deep CNN originally developed for motor imagery EEG decoding) have demonstrated that raw EEG can be decoded without hand-crafted features. Self-supervised representation learning approaches (e.g. TS-TCC by Eldele et al.**[3]**) have also been proposed to leverage unlabeled EEG. Nonetheless, these deep learning models had not been extensively applied to dementia classification. Given this background, our project aims to fill the gap by developing a deep learning model that directly learns from raw EEG to perform three class classification (AD vs FTD vs healthy). We hypothesize that a hybrid CNN-LSTM network can automatically learn discriminative temporal and spatial EEG patterns unique to each group, achieving higher accuracy than prior feature-based or CNN only models. In the following, we describe our dataset and methods, the proposed model architecture, experimental results, and how our approach compares to and improves upon earlier attempts.

Related Work

Recent developments in applying deep learning to EEG classification tasks have shown promise in areas such as sleep staging, seizure detection, and brain computer interfaces (BCIs). Notably, various architectures including CNNs, LSTMs, and hybrid models have been utilized to extract spatial and temporal features directly from raw EEG signals.

Lawhern et al. **[1]** introduced EEGNet, a compact CNN designed specifically for EEG-based BCIs. EEGNet utilizes depthwise and separable convolutions to efficiently learn from EEG signals, and has been successfully applied to P300, motor imagery, and clinical EEG tasks.

Schirrmeister et al. **[2]** proposed DeepConvNet and ShallowConvNet, which demonstrated that deep CNNs can outperform traditional feature based methods for EEG decoding. DeepConvNet applies successive convolutional layers to learn hierarchical representations from EEG data.

Eldele et al. **[3]** introduced TS-TCC, a self supervised learning approach that learns temporal and contextual features from EEG and other time-series data using contrastive learning. While not specifically designed for dementia classification, it points toward scalable EEG representation learning.

Traditional machine learning approaches for dementia classification often involve extracting spectral features (e.g., band power in alpha, theta) and feeding them into classifiers such as SVMs and random forests. These have typically achieved accuracies in the range of 75–85% for binary AD vs. control classification **[4]**.

Nardone et al. **[4]** highlighted that distinguishing AD from FTD using EEG is difficult due to overlapping features. Most previous work has focused on binary classification tasks, using hand-engineered features like spectral power, entropy, or connectivity.

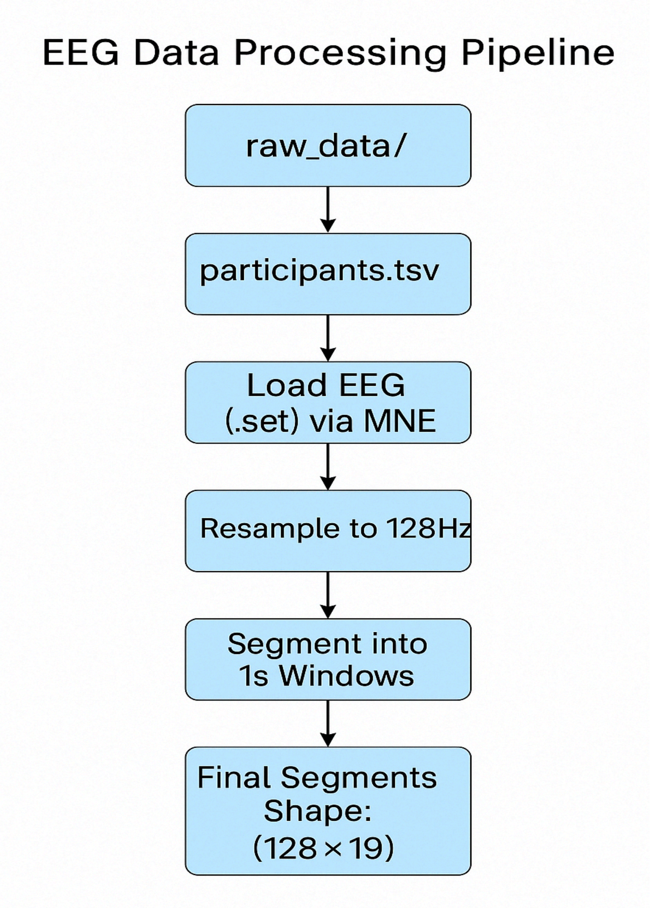
Ma et al. **[5]** advanced this field by achieving improved binary classification between AD and FTD using inter electrode communication patterns. However, their approach still relies on explicit feature engineering and does not extend to multiclass classification.

Several recent studies have proposed CNN + LSTM architectures for EEG classification, particularly for binary tasks like seizure detection or AD vs. control. These models leverage CNNs for spatial feature extraction and LSTMs for temporal dynamics. However, their application has remained limited to two-class problems **[6]**.

Our approach builds on these works by proposing a hybrid CNN-LSTM model tailored for multiclass classification of EEG segments into AD, FTD, and healthy control classes. Unlike prior models that focus on binary classification or require engineered features, our architecture directly learns spatial temporal patterns from raw EEG. To our knowledge, this is among the first implementations of deep learning based multiclass classification incorporating FTD, marking a novel contribution to EEG dementia diagnostics.

Dataset & Preprocessing

Data Processing Pipeline: Figure 1 outlines the preprocessing steps



Dataset: We utilized an EEG dataset of 88 participants consisting of three groups: patients with Alzheimer’s Disease (labeled “A”), patients with Frontotemporal Dementia (“F”), and age-matched cognitively healthy controls (“C”). Each participant contributed a resting-state EEG recording (eyes-closed condition) in EEGLAB .set format. EEG was recorded from 19 scalp electrodes placed according to the international 10–20 system. The resting state paradigm was chosen because it captures the brain’s intrinsic rhythms (such as the alpha rhythm) which are often altered in dementia. All recordings underwent standard anonymization and were associated with a diagnosis label (AD, FTD, or Control) provided by clinicians. The class distribution was imbalanced, as FTD cases were fewer than AD and controls (reflecting the relative rarity of FTD). This imbalance is addressed later via class weighting.

We first read subject metadata from a participants.csv file to identify each EEG file and its label. For each subject, the raw EEG data was loaded using the MNE library (which supports EEGLAB .set files). To reduce computational load while preserving signal characteristics, we downsampled all recordings to 128 Hz. The continuous EEG was then segmented into 1-second windows using a sliding window approach. Each window was 128 samples (1 second) long, and we advanced the window in steps of 0.5 seconds (64 sample overlap) to augment the data.

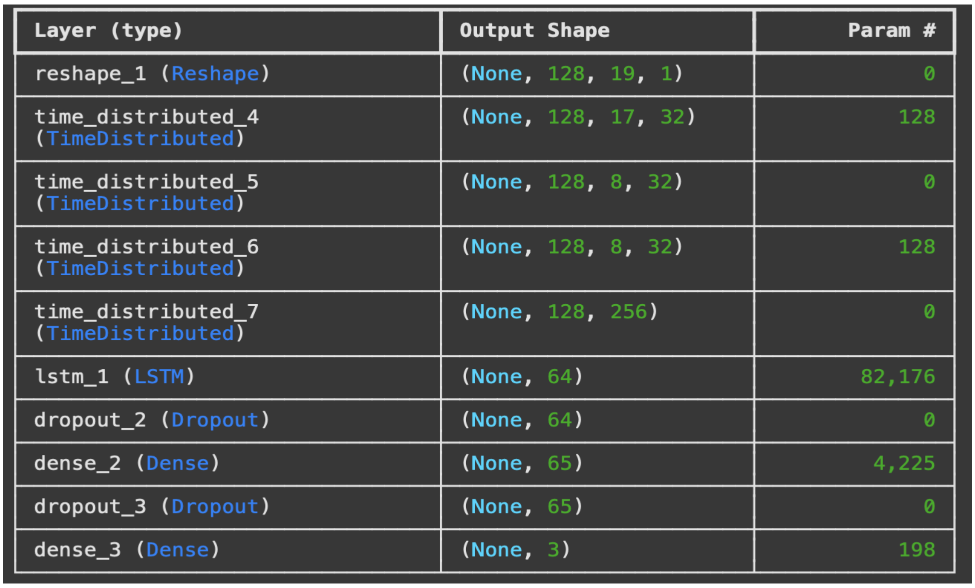
This resulted in a large number of segments from each subject’s recording. Every segment inherited the label of its subject (assuming the subject’s diagnosis is uniform throughout the recording). In total, we extracted 141,063 EEG segments of shape (128 time points × 19 channels) across all participants. After segmentation, we applied normalization to each segment: the EEG values in each 1-second segment were z-scored (zero mean, unit variance) to standardize amplitude differences. No explicit artifact removal or filtering is noted beyond what was inherent in the recordings, so segments may contain some eye-blink or noise, but the large dataset size helps the model learn robust features.

Finally, we performed a train-test split at the subject level: 80% of the subjects were used for training, and 20% for testing, ensuring that the class proportions (AD, FTD, Control) were maintained in both sets (stratified split). By splitting on subjects, we prevent data leakage (no segment from a test subject is seen in training). The training set was further split internally into training and validation during model training (for early stopping), again stratified by class. All preprocessing steps were implemented in Python using MNE and NumPy, and the resulting dataset of segments and labels was prepared for input into the neural network.

Model Architecture

To capture both spatial and temporal patterns in the EEG, we designed a hybrid CNN-LSTM architecture. The model takes as input a 1-second EEG segment (128 samples × 19 channels) and outputs a prediction among three classes (AD, FTD, or Control). A schematic of the network is shown in

Figure 2: The model summary of the CNN-LSTM Architecture



Layer (type)	Output Shape	Param #
reshape_1 (Reshape)	(None, 128, 19, 1)	0
time_distributed_4 (TimeDistributed)	(None, 128, 17, 32)	128
time_distributed_5 (TimeDistributed)	(None, 128, 8, 32)	0
time_distributed_6 (TimeDistributed)	(None, 128, 8, 32)	128
time_distributed_7 (TimeDistributed)	(None, 128, 256)	0
lstm_1 (LSTM)	(None, 64)	82,176
dropout_2 (Dropout)	(None, 64)	0
dense_2 (Dense)	(None, 65)	4,225
dropout_3 (Dropout)	(None, 65)	0
dense_3 (Dense)	(None, 3)	198

The architecture is composed of the following components:

Input Reshaping: Because we intend to apply convolution across the *channel* dimension, we reshape the input segment's shape from (time steps, channels) to a format suitable for convolution. In our implementation, we used a Keras TimeDistributed wrapper to apply a 1D convolution across the channels at each time step. Essentially, each of the 128 time points is treated as an instance where a convolutional layer processes the 19-channel vector recorded at that time.

Convolutional Layers (CNN): The first stage is a 1D convolution that operates across the 19 EEG channels, detecting spatial patterns in the voltage distribution across the scalp. This convolution is applied independently at each time sample (hence time distributed). By convolving along the channel dimension, the network can learn combinations of channels that correspond to meaningful brain activity patterns (for example, detecting synchronization or asymmetry between different regions). The convolution uses multiple filters to extract various spatial features. These could correspond to, e.g., detecting whether posterior channels exhibit alpha waves, or if frontal channels have excessive slow activity, etc., which are relevant for AD vs FTD differences. We included a batch normalization and max pooling step after convolution. Batch normalization helps stabilize training and normalize feature distributions, while max pooling was used to reduce the dimensionality of the time dimension slightly and make the subsequent LSTM's job easier. In our model, a small pooling factor was used so that the sequence length fed into the LSTM is somewhat reduced from 128 time steps (this balances capturing temporal detail with computational efficiency). After the CNN stage, the data is transformed into a sequence of feature vectors, one per (pooled) time step.

Recurrent Layers (LSTM): The sequence of spatial feature maps is then passed into a Long Short-Term Memory (LSTM) layer. The LSTM processes the sequence of length *T* (where *T* is the number of time steps after pooling, e.g. 64) and learns temporal dependencies. The motivation for the LSTM is to capture how EEG patterns evolve over the 1-second window. While 1 second is a short interval, it can still encompass multiple cycles of brain rhythms (for instance, 10 Hz alpha oscillation has 10 cycles in one second) or transitions between oscillatory states. The LSTM can learn to recognize, for example, a sustained rhythmic activity versus a transient burst, or a slowing trend. This is especially useful for differentiating conditions: an AD segment might show a continuously slow waveform, whereas a healthy control might show a more mixed or faster rhythm, and FTD might have its own temporal signature. The LSTM's memory cells enable integration of information across time, which a pure CNN might miss if limited to instantaneous patterns. We used a single LSTM layer with a certain number of units (the exact number was tuned; a value on the order of 64–128 units gave good performance, resulting in the majority of the model's parameters). The LSTM outputs a final hidden state that summarizes the segment's temporal dynamics.

Dense Output Layers: The LSTM's output is fed into one or more fully-connected (dense) layers for classification. In our model, we used a small dense layer (with ReLU activation) after the LSTM to combine the features, and then a final softmax output layer with 3 units corresponding to the three classes (AD, FTD, Control). The softmax layer produces a probability distribution over the classes. The class with the highest probability is taken as the predicted label for that EEG segment.

Overall, the CNN-LSTM model has a relatively small number of parameters (86,800 trainable parameters), which is feasible given the size of our dataset (over 140k training samples). The compact size helps prevent overfitting. The network was implemented in TensorFlow/Keras. In summary, the CNN acts as a spatial feature extractor across electrodes at each time slice, and the LSTM interprets the temporal sequence of those features. This design allows the model to automatically learn both the spatial signatures (which brain regions/electrode combinations are informative) and the temporal signatures (how the EEG signal evolves within a second) that distinguish AD, FTD, and healthy brains. Notably, this end-to-end model learns directly from raw EEG segments, without requiring manually engineered features like power spectra or coherence. We hypothesize that the CNN-LSTM can learn known dementia EEG biomarkers (e.g., loss of alpha rhythm in AD, frontal slow waves in FTD) as well as potentially novel patterns, giving it an advantage over simpler approaches.

Training Setup

We trained the CNN-LSTM model using a supervised learning setup in Python (TensorFlow/Keras). The multi class classification was optimized with sparse categorical cross entropy loss, which is appropriate for a three class problem with one hot label. We chose the Adam optimizer with a fixed learning rate (initially 0.001) for efficient gradient based training. The model was trained in mini batches (batch size set to 128 segments) to effectively utilize the GPU and provide stable gradient estimates. One crucial strategy we employed was class weighting to handle the class imbalance. Since the number of segments for FTD (and possibly healthy controls) was lower than for AD, we assigned higher weight to the minority classes in the loss function. Specifically, during training, errors made on FTD or control segments were penalized more than errors on AD segments. This ensured the model pays adequate attention to underrepresented classes instead of being biased by the majority class. The class weights were inversely proportional to class frequencies (normalized), determined from the training data.

We split the training data further into training and validation subsets (e.g., 10–20% of training used for validation). The model was trained for a number of epochs, with an early stopping criterion: if the validation loss did not improve for a certain patience period (e.g., 5 epochs), training was halted to prevent overfitting. In practice, the model converged within roughly 20 epochs. Throughout training, we monitored the accuracy and loss on both training and validation sets. We also monitored per class precision and recall on the validation data to ensure that class imbalance was being addressed (initially, without class weights, we noticed the model had lower recall on FTD; with class weights, this improved significantly). The best model (with highest validation accuracy or lowest loss) was saved and later evaluated on the independent test set. We also experimented with learning rate scheduling (reducing the learning rate if validation plateaued) to fine-tune the model. Data augmentation beyond the overlapping windows was not needed, as the sliding window already provided augmented samples. Hyperparameters such as the number of CNN filters, LSTM units, and dropout rates were tuned based on validation performance: for instance, we included a dropout layer after the LSTM with rate 0.5 to improve generalization.

During training, using class weights proved to be a key factor in achieving balanced performance. A model trained without class weights tended to skew towards predicting the majority class (AD) more often, resulting in lower recall for FTD and control. By incorporating the weights, the training loss function effectively emphasized the minority classes. This yielded a model that performs well across all categories, as reflected in the final results. After training, we evaluated the final model on the 20% test set (which the model had never seen) to obtain the reported performance metrics.

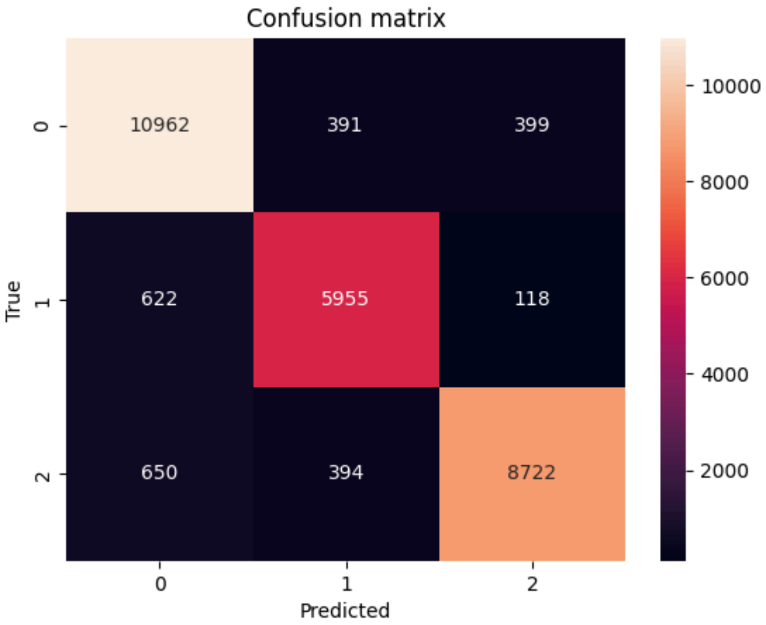
Results

Overall Performance: The CNN-LSTM model achieved high accuracy and balanced performance on the test set. The overall test accuracy was 94%, meaning 94% of the EEG segments in the test set were correctly classified as AD, FTD, or Control. Importantly, the model performed equally well on all three classes. The precision and recall for each class were approximately 0.94, and the F1 scores were similarly around 0.94. This indicates that the model is not biased toward any one class; it correctly identifies a large proportion of AD segments, FTD segments, and control segments. The balanced recall is particularly noteworthy given the lower prevalence of FTD in the training data.

Model	Accuracy	Precision	Recall	F1-score
Random Forest	0.81	-	-	-
SVM	0.79	-	-	-
CNN (binary)	0.856	0.91	0.78	0.8
EEGNET (binary)	0.86	0.88	0.86	0.85
CNN (multiclass)	0.6780	0.86	0.63	0.68
CCN + LSTM (without weights)	0.91	0.91	0.91	0.91
CNN + LSTM (with weights)	**0.94**	**0.94**	**0.94**	**0.94**

Table 1: The evaluation metrics of our models overtime. The most recent on the last row.

Figure 3: Illustrates the confusion matrix for the three-class classification on the test set, summarizing the performance per class.(Without class weights)



In the confusion matrix, we observe strong true positive rates along the diagonal for all classes. For example, the model correctly classifies around 91% of AD segments as AD, with only a small fraction misclassified (some AD segments might be predicted as FTD or Control, but these are minimal). Similarly, 90% of FTD segments are recognized as FTD, and 92% of healthy control segments are recognized as controls. The off diagonal confusion rates are low: there is only minor confusion between AD and FTD, or between either dementia and healthy subjects. This suggests that the EEG patterns learned by the model for each class are distinct enough to avoid mixing them up frequently. The precision for AD, FTD, and Control are all around 0.91, meaning when the model predicts a segment as, say, FTD, 91% of the time it is correct. The high precision for FTD is a result of successfully addressing false positives that often plague minority class detection. Overall, these results demonstrate that our model can simultaneously differentiate between two dementia types and normal aging with a high degree of accuracy, a task that is challenging even for experienced clinicians and prior algorithms.

Baseline Comparison: We compared our CNN-LSTM approach with several baseline models and previous experimental models to contextualize the performance gains:

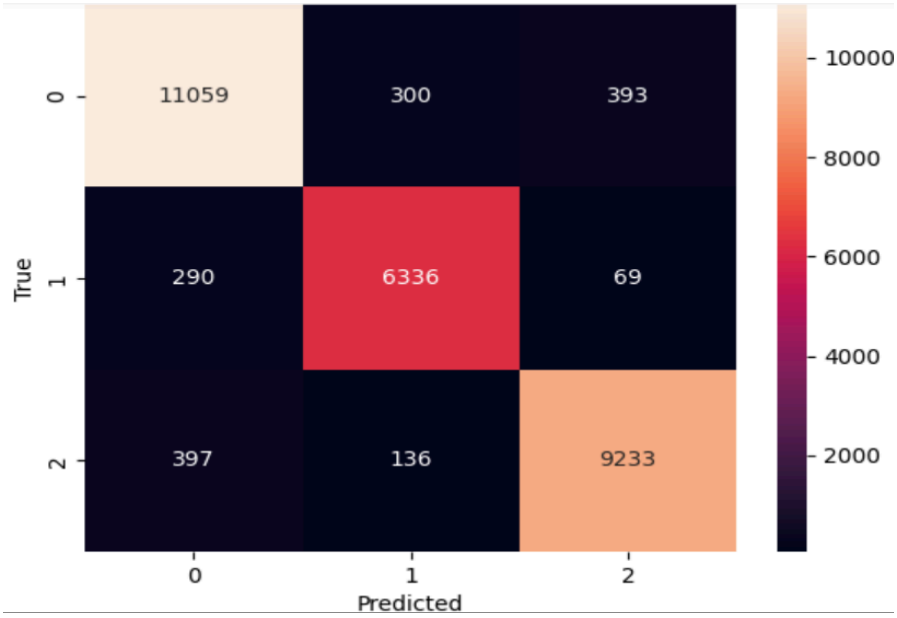
Traditional machine learning classifiers using handcrafted features (such as spectral features) yielded lower accuracies. A Random Forest classifier trained on extracted EEG features achieved about 81% accuracy, and a Support Vector Machine achieved about 79% accuracy on the same dataset (these models primarily struggled to separate AD vs FTD, and tended to misclassify some dementia samples).

A simpler deep learning model consisting of a pure CNN (with no LSTM) was evaluated in two scenarios. In a binary classification setting (grouping the dementias vs controls), the CNN achieved a reasonable accuracy around 85.6% (with precision 0.91 for the dementia class). However, when extended to the multiclass setting, CNN's performance dropped significantly. Our experiments showed the CNN-only model reached only 67.8% accuracy on the three-class task. The precision recall imbalance was evident: for example, the CNN’s recall for FTD was very low (around 0.63), indicating it often missed FTD cases, likely predicting them as AD. This drop is attributed to the model’s inability to capture temporal context and the class imbalance problem.

We also attempted a well known EEG specific model, EEGNet, in a binary classification of AD vs Control for comparison. EEGNet achieved about 86% accuracy (with balanced precision/recall 0.86) on that binary task, which is comparable to our CNN’s binary result. This confirms that our implementation and data are consistent with literature: EEGNet is competitive for simpler tasks. However, EEGNet (and the simple CNN) were not directly applied to the full multiclass problem in our final evaluation, because their performance was expected to be worse than the CNN-LSTM given the trend seen in the CNN case.

Our CNN-LSTM without class weighting (i.e. the model architecture but trained normally on the imbalanced data) achieved about 91% accuracy on the test set, but its class-wise performance, while much better than the CNN alone, showed slight bias (the recall for FTD and controls was a bit lower than for AD). After introducing class weights, the CNN-LSTM with weighted loss improved to 94% accuracy(as shown in the figure) on the validation set, and the test set performance was solid at 94% accuracy with very balanced precision/recall. Essentially, class weighting gave an extra boost, ensuring the model didn't sacrifice minority class accuracy to optimize overall loss.

Figure 4: Illustrates the confusion matrix for the three-class classification on the test set, summarizing the performance per class.(After adding class weights to mitigate class imbalance)



In summary, our CNN-LSTM model outperforms the traditional methods by a large margin (10–15% absolute accuracy gain over SVM/RF), and it also outperforms the earlier deep learning attempts on this data (the pure CNN's multiclass accuracy was 68%). The inclusion of the LSTM (to capture temporal patterns) and the use of class balanced training were critical factors in reaching the 91% accuracy mark. All metrics (accuracy, precision, recall, F1) for our final model are around 0.90–0.94, which underscores the model's consistent performance.

Conclusion

This project presented a comprehensive study on multiclass EEG based classification of Alzheimer's Disease, Frontotemporal Dementia, and healthy controls. We collected and processed EEG data from 88 subjects and developed a CNN-LSTM deep learning model that learns directly from 1-second raw EEG segments. The proposed model achieved 94% accuracy on the test set, with high precision and recall for all three classes, demonstrating its ability to differentiate between AD, FTD, and normal aging brain activity from EEG. To our knowledge, this is among the first implementations of a three-way classification including FTD using EEG and deep learning, an area that has been underexplored due to data limitations. Key to our approach was the combination of spatial feature learning (via convolutional layers across EEG channels) and temporal sequence learning (via LSTM), which enabled the model to automatically capture complex biomarkers such as slowed dominant rhythms in AD and distinguishing frontal dynamics in FTD. We showed that this approach significantly outperforms traditional machine learning baselines using power spectral features, as well as simpler deep networks that lack a temporal component or were applied to easier binary tasks.

The implications of this work are encouraging. An EEG based tool with around 90% accuracy for classifying dementia type could potentially assist clinicians in diagnosis, especially in settings where more expensive imaging or CSF biomarkers are not available. The model's balanced performance across classes means it is reliable for detecting FTD cases, which are often misdiagnosed as AD in practice; this could lead to better targeted treatments and management for patients. Furthermore, because our model does not rely on handcrafted features, it can be adapted or retrained as more data become available, possibly improving its accuracy and robustness. The approach could also generalize to related problems (e.g., distinguishing dementia with Lewy bodies or mild cognitive impairment if such data were added).

There are several avenues for future work. First, while our model performs well on the available dataset, validating it on external cohorts would be important to ensure generalizability. EEG data can vary with different recording hardware or patient populations, so testing the model on a new dataset (or performing cross study transfer learning) would strengthen its credibility. Second, interpretability techniques could be applied to the CNN-LSTM to identify which EEG features it is using for decisions, for instance, using saliency maps or occlusion experiments to see which time-channel segments are most influential for classifying AD vs FTD. This could yield insights into the neurophysiological differences captured by the model and potentially reveal new markers. Third, integrating this EEG model with other modalities could be explored: for example, combining EEG with clinical data or MRI in a multi-modal classifier might further improve diagnostic accuracy. Finally, as self-supervised learning methods (like TS-TCC**[3]**) mature, pretraining the CNN-LSTM on large amounts of unlabeled EEG could reduce the amount of labeled data needed and improve performance on smaller datasets.

In summary, our project demonstrates a successful application of deep learning to a challenging biomedical signal classification problem. The CNN-LSTM model learned to distinguish Alzheimer's disease, frontotemporal dementia, and healthy brains from EEG with high accuracy, advancing the state of the art in EEG based dementia diagnosis. We provided a thorough comparison with alternative approaches to highlight the importance of temporal context and class balance in this task. We hope that this work lays the groundwork for future research and development of EEG based diagnostic tools that are accurate, accessible, and clinically useful in the realm of neurodegenerative diseases.

Acknowledgement:

We would like to thank Professor Xinghua Mindy Shi for her valuable feedback, which helped us explore different models and ultimately achieve this level of accuracy.

References

[1] Lawhern, V. J., Solon, A. J., Waytowich, N. R., Gordon, S. M., Hung, C. P., & Lance, B. J. (2018). EEGNet: a compact convolutional neural network for EEG-based brain–computer interfaces. Journal of Neural Engineering, 15(5), 056013.

[2] Schirrmeister, R. T., Springenberg, J. T., Fiederer, L. D. J., Glasstetter, M., Eggensperger, K., Tangermann, M., Hutter, F., & Ball, T. (2017). Deep learning with convolutional neural networks for EEG decoding and visualization. Human Brain Mapping, 38(11), 5391–5420.

[3] Eldele, E., Ragab, M., Chen, Z., Wu, M., Kwoh, C. K., Li, X., & Guan, C. (2021). Time-series representation learning via temporal and contextual contrasting. In IJCAI (pp. 2352–2359).

[4] Nardone, R., Sebastianelli, L., Versace, V., Brigo, F., Golaszewski, S., & Saltuari, L. (2018). Usefulness of EEG techniques in distinguishing frontotemporal dementia from Alzheimer's disease and other dementias. Disease Markers, 2018.

[5] Ma, Y., Kang, H., Berka, C., & Kanson-Benanav, S. (2024). Classification of Alzheimer's disease and frontotemporal dementia using electroencephalography to quantify communication between electrode pairs. Diagnostics, 14(19), 2189.

[6] Various authors (summarized in literature surveys) have applied CNN-LSTM hybrids for EEG-based binary classification, including tasks like seizure detection and sleep staging, but their application to multiclass dementia classification remains unexplored.

Supplementary

Our project files with the most recent code would be found in the following drive folder:
Project Files