FAST UNIVERSITY OF COMPUTER AND EMERGING SCIENCES

Final Year Research - II

"IDENTIFICATION OF NEUROLOGICAL DISEASES USING DEEP LEARNING METHODS"

Supervisor: Sir Shoaib Rauf

15th May 2025

Group Members

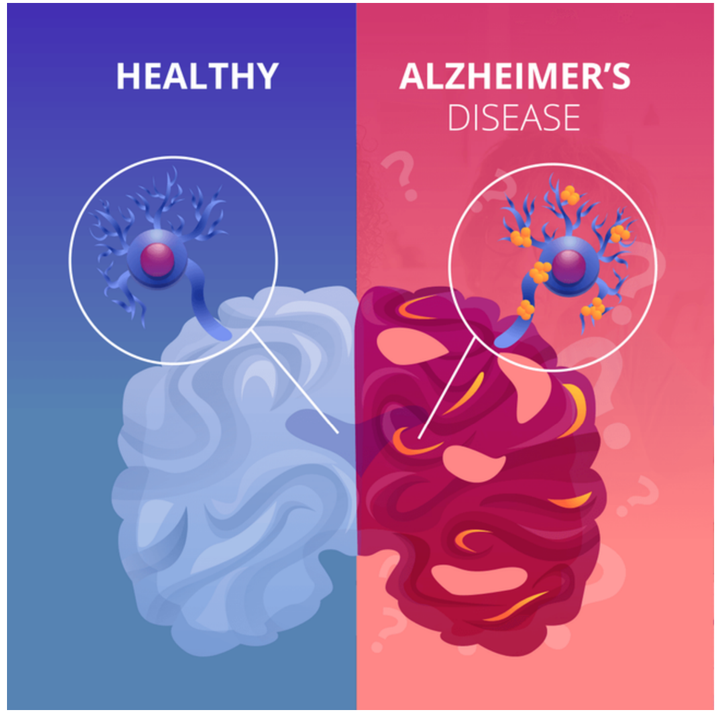
MUHAMMAD
TAHIR
21K-4503

INSHA
JAVED
21K-3279

HASAN
IQBAL
21K-3297

OVERVIEW OF ALZHEIMER’S DISEASE

- A neurodegenerative disorder causing progressive memory loss and cognitive decline.The most common cause of dementia worldwide.
- Over 55 million people live with dementia globally, and this number is expected to double by 2050 (WHO).
- **Symptoms:**
  - Memory loss, difficulty with problem-solving, confusion, and mood changes.
- **Problem Statement:**
  - "Traditional GWAS methods face challenges like high-dimensional data and the inability to capture SNP interactions effectively."
  - There is a need for efficient computational frameworks to identify significant SNPs in AD.



RESEARCH PAPER

Deep learning-based identification of genetic variants: application to Alzheimer’s disease classification. Briefings in Bioinformatics, 23(2), 1–11.

Jo, T., Nho, K., Bice, P., Saykin, A. J., & For The Alzheimer’s Disease Neuroimaging Initiative (2022).

DATASET

- Alzheimer’s Disease Neuroimaging Initiative **(ADNI)**.
- 1000 participants and more than 5000 SNPs of each patient.

METHODOLOGY

Genome Fragmentation

Divide the genome into non-overlapping fragments for manageable analysis, to identify phenotype associated fragments.

Phenotype Influence Score (PIS) Calculation

Use CNNs to identify phenotype associated SNPs.

Classification Model Development

Train a CNN to classify AD vs. normal individuals using selected SNPs.

KEY OBJECTIVES ACHIEVED IN FYP-I:

1. **Dataset Preprocessing:** Cleaned and prepared the dataset for analysis.
2. **Determining Optimal Fragment Size:** Tested CNN, LSTM, LSTM-CNN, and Attention-based model to find the best fragment size for classification.
3. **Dataset Fragmentation & Association Analysis:** Applied the Sliding Window Association Test to identify phenotype-associated fragments.
4. **Identifying SNPs within Significant Fragments:** Calculated PIS scores for each fragment to determine relevant SNPs.
5. **Validation of Top 100 SNPs:** Verified the top 100 SNPs using the API: https://clinicaltables.nlm.nih.gov/api/snps/v3/search?terms={rs_id}

KEY OBJECTIVES ACHIEVED IN FYP-II:

1. **Model Development:** Built a deep learning-based CNN model to classify Alzheimer's Disease (AD) and cognitively normal individuals using selected SNP features.
2. **Model Comparison:** Evaluated traditional machine learning models (e.g., Random Forest) alongside the CNN to determine the best-performing approach based on metrics like accuracy.
3. **Model Interpretability:** Applied SHAP (Shapley Additive exPlanations) analysis to interpret the CNN model and highlight the most influential SNPs contributing to AD classification.
4. **Research Paper Finalization:** Compiled and finalized a comprehensive research paper detailing the methodology, results, and findings for submission.

PREPROCESSING

- **Data Loading:** Reads SNP data, removes unnecessary columns, and cleans up missing values.
- **Imputation** for missing data using KNN and simple mean imputation.
- **Population stratification** controlled by selecting non-Hispanic European ancestry.
- **Preprocessing:** Separates SNP data into features and labels, handling missing labels for consistency.

```
[ ]  def load_data(file):
         data = pd.read_csv(file, sep=" ", header=0)
         data = data.iloc[:, 5:]
         return data.dropna(axis=1, how='all')

     def impute_data(data, imputation_method):
         if imputation_method == 'simple':
             imputer = SimpleImputer(strategy='mean')
         elif imputation_method == '1nn':
             imputer = KNNImputer(n_neighbors=1)
         elif imputation_method == '5nn':
             imputer = KNNImputer(n_neighbors=5)
         elif imputation_method == '10nn':
             imputer = KNNImputer(n_neighbors=10)
         return pd.DataFrame(imputer.fit_transform(data), columns=data.columns)

     def preprocess_data(data):
         features = data.iloc[:, 1:]
         labels = data.iloc[:, 0]
         labels = labels.apply(lambda x: 'rs_ith' if x == '.' else x)
         return features, labels
```

EXPERIMENT TO DETERMINE AN OPTIMAL FRAGMENT SIZE

Objective:

- Analyze model performance across different fragment sizes to determine the optimal size for SNP-based classification.

Models Used:

- LSTM
- CNN
- LSTM-CNN (Hybrid)
- Attention-based models

Key Finding:

- Optimal Fragment Size: 40.

fragment_size	avg_accuracy	avg_training_time	model
10	0.6501254592	9.189088415	CNN
20	0.6492663868	14.59478652	CNN
40	0.6463565674	18.58840328	CNN
100	0.6483333111	36.34310599	CNN
10	0.651573336	5.32543056	LSTM
20	0.6518146485	7.433329347	LSTM
40	0.6520930011	25.26999418	LSTM
100	0.6525489968	57.25330214	LSTM
10	0.6520076989	4.055529379	LSTM-CNN
20	0.6524131045	8.00008474	LSTM-CNN
40	0.6520542405	13.67273265	LSTM-CNN
100	0.6537254675	34.43393836	LSTM-CNN
10	0.6511968881	7.055575721	ATTENTION
20	0.6513706333	12.44334502	ATTENTION
40	0.6516666431	45.19068482	ATTENTION
100	0.6507842903	307.0336405	ATTENTION

STEP 1: DATA FRAGMENTATION AND MODEL TRAINING

Key Steps:

- **Fragmentation:** The dataset is divided into non-overlapping fragments of size 40 (or other sizes).
- **Model Training:** For each fragment, a hybrid CNN-LSTM model is built and trained, with convolutional layers extracting features and LSTM layers capturing temporal dependencies.
- **Selection:** Fragments with accuracy greater than 0.5 are selected and combined into the final dataset for further analysis.

```
f create_fragments(data, column_names, fragment_size):
  """Divide genome data and column names into non-overlapping fragments o
  num_fragments = data.shape[1] // fragment_size
  data_fragments = []
  name_fragments = []

  for i in range(num_fragments):
      start = i * fragment_size
      end = start + fragment_size
      data_fragments.append(data.iloc[:, start:end].values)
      name_fragments.append(column_names[start:end])

  return np.array(data_fragments), name_fragments

port tensorflow as tf
```

STEP 2: FINDING ASSOCIATED SNPS

```
[ ] filee = "/content/drive/My Drive/FYP Resources 2024/FYP I Evaluation/Results/REALphenotype_associated_fragments.csv"
    window_size = 200
    classifier = "dl"
    timestamp = datetime.datetime.now().strftime('%Y_%m%d_%H%M%S')
    num_results = 10
```

Reads data and preprocesses it into features and label

```
[ ] data = pd.read_csv(filee, header=0)
    features, labels = preprocess_data(data)
    labels = labels.astype('int')
```

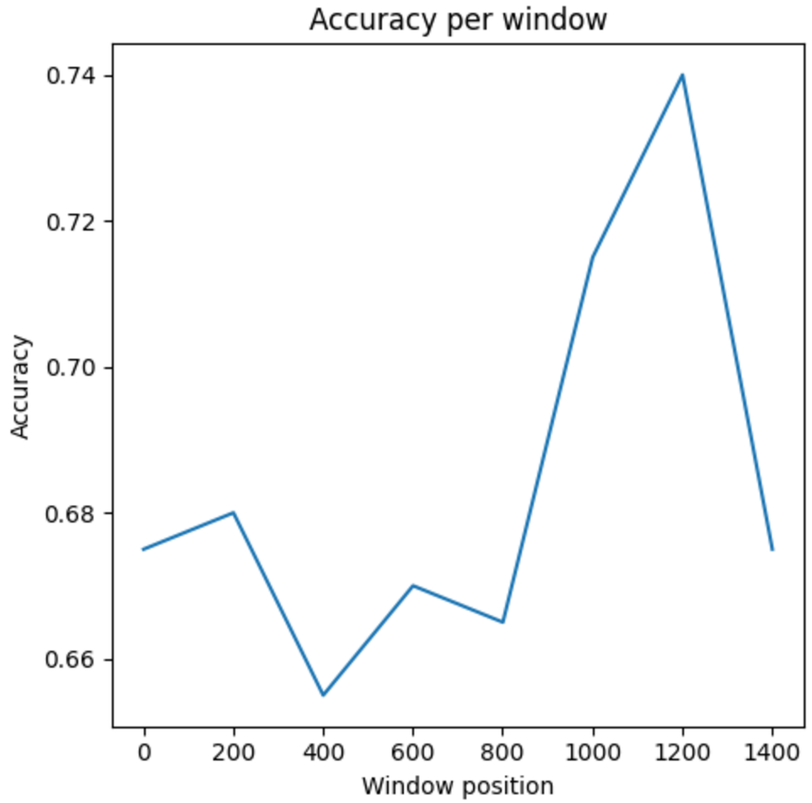
Trains and evaluates models on feature subsets (windows) to find the Phenotype Influence Scores *(PIS)*

```
[ ] accuracies, window_indices = train_and_test_windows(features, labels, window_size, classifier)

    plot_accuracies(window_indices, accuracies, f'accuracies_{timestamp}.png')

    window_accuracies = list(zip(accuracies, window_indices))
    top_accuracies = sorted(window_accuracies, key=lambda x: x[0], reverse=True)[:num_results]

    # using the sorted top_accuracies
    highest_accuracy, corresponding_window = top_accuracies[0]
    print(f"\nHighest accuracy: {highest_accuracy:.4f}, Corresponding window index: {corresponding_window}\n")
```



- **Window Creation**: Split data into overlapping windows.
- **Model Training:**
- **CNN:** Train CNN for each window.
- **Random Forest:** Train Random Forest on each window.
- **Evaluation**: Test on 80% training, 20% testing, and calculate accuracy.(HOLD-OUT method)
- **Purpose**: Assess accuracy for different genomic feature windows

Select top window indices based on Phenotype Influence Scores (PIS) and train the model:

```
top_window_indices = [acc[1] for acc in top_accuracies]
top_window_indices.sort()

feature_indices = []
for i in top_window_indices:
    feature_indices.extend(list(range(i, min(i+window_size, len(features.columns)))))
selected_features = features.iloc[:, feature_indices]

clf_rf, clf_dl, feature_importances = train_and_test_selected_features(selected_features, labels)

feature_names = selected_features.columns

top_indices = feature_importances.argsort()[-num_results:][::-1]
top_features = [(feature_names[i], feature_importances[i]) for i in top_indices]
```

```
epoch 14 | loss: 0.60384 | train_auc: 0.73077 | valid_auc: 0.73977 |  0:00:36s
epoch 15 | loss: 0.624   | train_auc: 0.73937 | valid_auc: 0.70784 |  0:00:37s
epoch 16 | loss: 0.62426 | train_auc: 0.74044 | valid_auc: 0.72392 |  0:00:39s
epoch 17 | loss: 0.61091 | train_auc: 0.73297 | valid_auc: 0.72266 |  0:00:40s

Early stopping occurred at epoch 17 with best_epoch = 12 and best_valid_auc = 0.76172
```

RESULT OF FYP-I

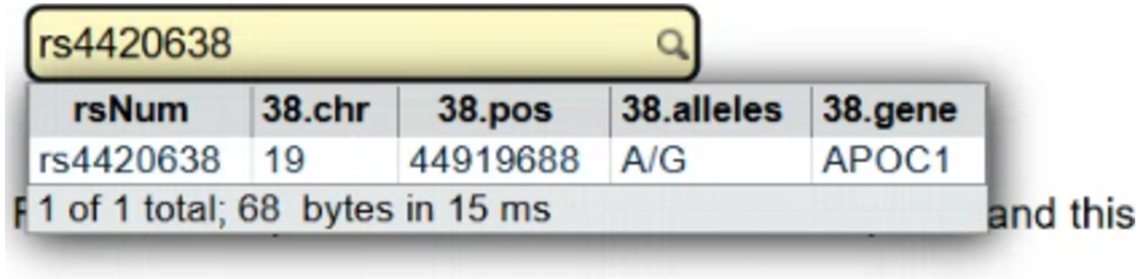
- **Window Indices:** Identified top-performing window indices based on accuracy.
- **Feature Selection:** Selected features from the identified windows for further analysis.
- **Model Training:** Trained both Random Forest (RF) and Deep Learning (DL) models on selected features.
- **Feature Importance:** Extracted and ranked feature importance using model outputs.
- **Top Features:** Identified top features based on their importance scores for further investigation.

No.	SNP ID
0	rs10414043
1	rs56131196
2	rs4420638
3	rs438811
4	rs12721051
5	rs769449
6	rs6857
7	rs283811
8	rs142042446

API Integration: Used the National Library of Medicine's SNP API to verify SNP details.

APOC1 and Its Role in Alzheimer's Disease:

- APOC1 (Apolipoprotein C1) is a gene located on chromosome 19, and it plays a crucial role in lipid metabolism and inflammation.
- Recent studies have highlighted the potential involvement of the APOC1 region in Alzheimer's disease (AD) due to its association with amyloid plaques and tau pathology, key hallmarks of Alzheimer's disease.



and this

API:

'https://clinicaltables.nlm.nih.gov/api/snps/v3/search?terms={rs_id}&q={rs_id}'

- **DOI: 10.4103/1673-5374.130117**
  APOE and APOC1 gene polymorphisms are associated with cognitive impairment progression in Chinese patients with late-onset Alzheimer's disease

- **DOI:10.1001/archneur.61.9.1434**
  APOE and APOC1 Promoter Polymorphisms and the Risk of Alzheimer Disease in African American and Caribbean Hispanic Individuals

No.	SNP ID	importance	position	alleles	gene
0	rs10414043	0.0127228	19:44912455	G/A	APOC1
1	rs56131196	0.0125826	19:44919588	G/A	APOC1
2	rs4420638	0.00923104	19:44919688	A/G	APOC1
3	rs438811	0.00822085	19:44913483	C/T	APOC1
4	rs12721051	0.00815601	19:44918902	C/G	APOC1
5	rs769449	0.00815306	19:44906744	G/A	APOE
6	rs6857	0.00808767	19:44888996	C/T	NECTIN2
7	rs283811	0.00804162	19:44885242	A/C, A/G	NECTIN2
8	rs142042446	0.00716833	19:44883210	TAA/TAATAA	NECTIN2

RESULT OF FYP-II

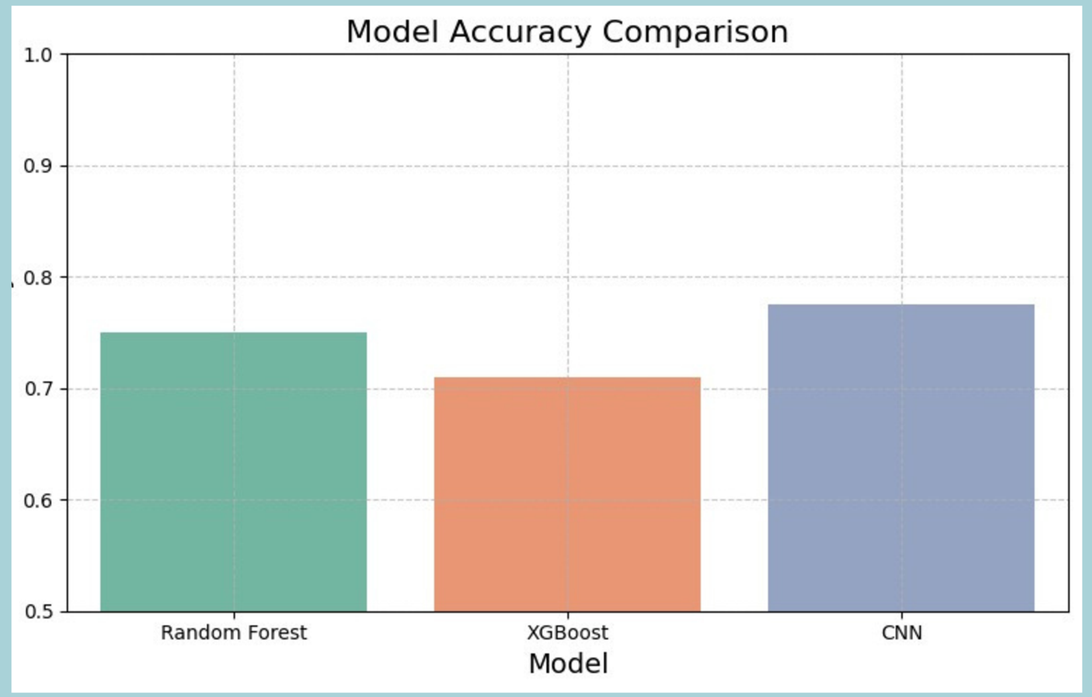
Model Development and Classification Performance:

- Developed a convolutional neural network (CNN) to classify Alzheimer's Disease (AD) and cognitively normal individuals. The CNN model achieved an accuracy of 78.54% and an AUC of 0.8157, outperforming traditional machine learning models.

Comparison with Traditional Machine Learning Models:

- Evaluated Random Forest (RF) and XGBoost against the CNN model.
- **CNN**: 78.54% accuracy, AUC = 0.8157
- **XGBoost**: 71% accuracy, AUC = 0.7263
- **Random Forest:** 75% accuracy, AUC = 0.7712

The CNN outperformed all baseline models, demonstrating its effectiveness on high-dimensional SNP data.



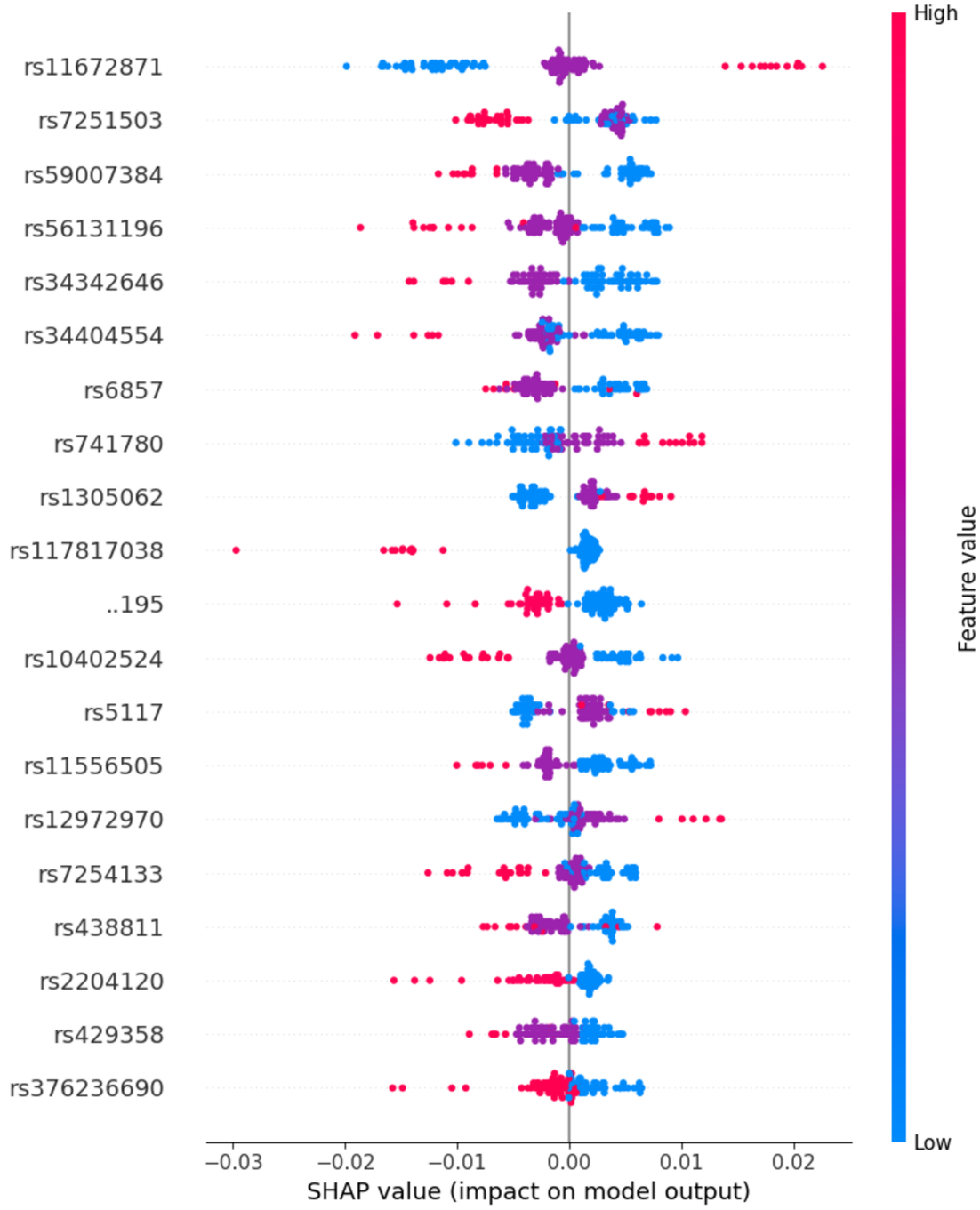
SHAP (Shapley Additive exPlanations) analysis

Model Interpretability and Feature Importance:

- Conducted a **SHAP (Shapley Additive exPlanations) analysis** to improve interpretability. The CNN model emphasized SNP regions known to be biologically significant in AD, aligning with established literature.

SHAP Analysis Insights:

- The SHAP summary plot illustrated the direction and impact of influential SNPs on model predictions. Red-highlighted SNPs contributed positively toward AD classification, while blue SNPs indicated lower risk, enhancing confidence in the model's decision-making process.



They also found another variant near the EXOC3L2/BLOC1S3/MARK4 genes was associated with 1.18 times increased odds of Alzheimer's.

Results Two loci were identified to have genome-wide significance for the first time: rs744373 near *BIN1* (odds ratio [OR],1.13; 95% confidence interval [CI],1.06-1.21 per copy of the minor allele; $P = 1.59\times10^{-11}$) and rs597668 near *EXOC3L2*/*BLOC1S3*/*MARK4* (OR, 1.18; 95% CI, 1.07-1.29; $P = 6.45\times10^{-9}$). Associations of these 2 loci plus the previously identified loci *CLU* and *PICALM* with AD were confirmed

Genome-wide Analysis of Genetic Loci Associated With Alzheimer Disease

(doi:10.1001/jama.2010.574)

Genomic Variants, Genes, and Pathways of Alzheimer's Disease: An Overview

(doi:10.1002/ajmg.b.32499)

Future Work

Model Optimization: Experiment with lightweight or hybrid architectures (e.g., CNN-RNN combinations, transformers) to reduce computational costs while maintaining performance.

Population Diversity: Validate and retrain models on diverse population datasets to improve generalizability and reduce bias across ethnic groups.

SNP Interaction Modeling: Investigate SNP-SNP interactions using graph-based neural networks or attention-based methods to capture non-linear and complex associations.

Clinical Integration: Collaborate with clinicians to test model usability in real-world diagnostic workflows and assess clinical utility in early-stage detection and risk stratification.

ACKNOWLEDGMENT

We would like to express our sincere gratitude to:

- **Our Supervisor, Sir Shoaib Rauf**, for his invaluable guidance and support throughout our research.
- **The Jury Members**, for their insightful feedback and recommendations that helped shape our research.
- **FAST NUCES**, for providing the resources and platform to conduct our research.

Thank you for your time and consideration!