Carnegie Mellon University

Transcriptomic Biomarkers for Predicting the Progression from Mild Cognitive Impairment to Alzheimer's Dementia

02-518/718 Computational Medicine

Kitiyaporn Takham, Mahitha Chaturvedula

Introduction - Motivation behind project

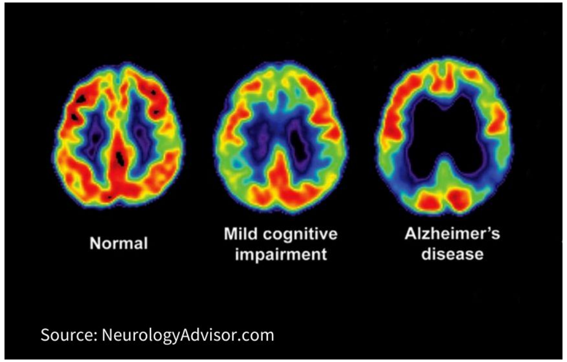
I. Currently no effective treatment for Alzheimer's Disease (AD)

II. Mild cognitive impairment (MCI) - initial phase of memory decline for patients and is often underdiagnosed due to its confusion with AD [1]

   A. MCI has been said to designate an early, yet abnormal, state of cognitive impairment [2]

III. By clearly defining expression markers for MCI and AD, might eliminate confusion between the two stages

   A. Also increase the diagnosis of both stages of dementia, leading to earlier detection and preventive medical care for dementia patients



Introduction - Gaps in current research

I. Many studies have suggested that biomarkers obtained by comparing AD to controls can be used to predict conversion from MCI to AD [3,4]

II. Recent studies have indicated that this change has a non-linear trajectory whereby cognitively unimpaired (CU) individuals progress to MCI, and then further progress to AD [3,4]

III. Recent imaging study found that molecular biomarkers predicting CU-to-MCI conversion are not as helpful as they are for MCI-to-AD conversion [5]

Carnegie Mellon University

Introduction

What are we hoping to accomplish?

Aim 1: Identify a set of gene expression biomarkers for high-risk and low-risk MCI to AD dementia prognosis

Carnegie Mellon University

Introduction

What are we hoping to accomplish?

Aim 1: Identify a set of gene expression biomarkers for high-risk and low-risk MCI to AD dementia prognosis

Aim 2: Compare the performance of supervised models between machine learning and deep learning approaches

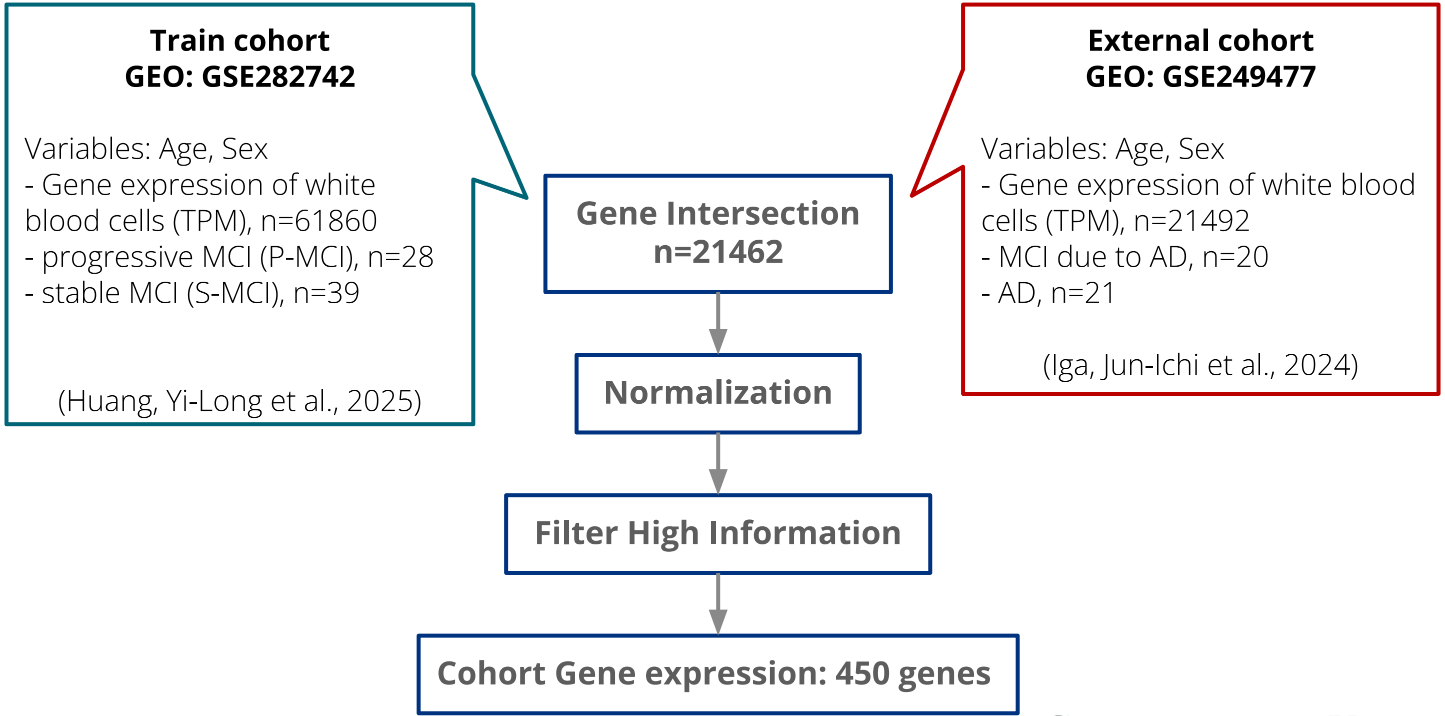
SVM

Support vector machine

CNN

Convolutional neural network

Data



Train cohort
GEO: GSE282742

Variables: Age, Sex
- Gene expression of white blood cells (TPM), n=61860
- progressive MCI (P-MCI), n=28
- stable MCI (S-MCI), n=39

(Huang, Yi-Long et al., 2025)

External cohort
GEO: GSE249477

Variables: Age, Sex
- Gene expression of white blood cells (TPM), n=21492
- MCI due to AD, n=20
- AD, n=21

(Iga, Jun-Ichi et al., 2024)

Methods - SVM

Architecture - Scikit Learn Python library

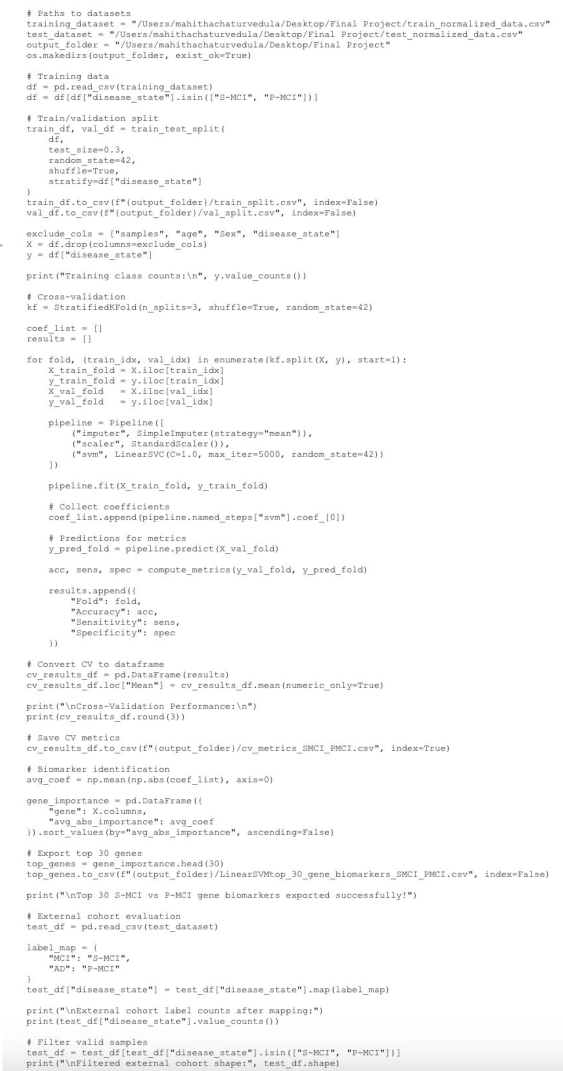
Cross-validation: StratifiedKFold(K = 5)
LinearSVC (C=1.0, max_iter=5000, random_state=42)

Training performance

	Accuracy	Sensitivity	Specificity
Cross-Validation Mean	0.732	0.727	0.743

Biomarker identification

- Using the **average absolute value of SVM coefficients**
- Rank top 30 genes



```
# Paths to datasets
training_dataset = "/Users/mahithachaturvedula/Desktop/Final Project/train_normalized_data.csv"
test_dataset = "/Users/mahithachaturvedula/Desktop/Final Project/test_normalized_data.csv"
output_folder = "/Users/mahithachaturvedula/Desktop/Final Project"
os.makedirs(output_folder, exist_ok=True)

# Training data
df = pd.read_csv(training_dataset)
df = df[df["disease_state"].isin(["S-MCI", "P-MCI"])]

# Train/validation split
train_df, val_df = train_test_split(
    df,
    test_size=0.3,
    random_state=42,
    shuffle=True,
    stratify=df["disease_state"]
)
train_df.to_csv(f"{output_folder}/train_split.csv", index=False)
val_df.to_csv(f"{output_folder}/val_split.csv", index=False)

exclude_cols = ["samples", "age", "Sex", "disease_state"]
X = df.drop(columns=exclude_cols)
y = df["disease_state"]

print("Training class counts:\n", y.value_counts())

# Cross-validation
kf = StratifiedKFold(n_splits=3, shuffle=True, random_state=42)

coef_list = []
results = []

for fold, (train_idx, val_idx) in enumerate(kf.split(X, y), start=1):
    X_train_fold = X.iloc[train_idx]
    y_train_fold = y.iloc[train_idx]
    X_val_fold   = X.iloc[val_idx]
    y_val_fold   = y.iloc[val_idx]

    pipeline = Pipeline([
        ("imputer", SimpleImputer(strategy="mean")),
        ("scaler", StandardScaler()),
        ("svm", LinearSVC(C=1.0, max_iter=5000, random_state=42))
    ])

    pipeline.fit(X_train_fold, y_train_fold)

    # Collect coefficients
    coef_list.append(pipeline.named_steps["svm"].coef_[0])

    # Predictions for metrics
    y_pred_fold = pipeline.predict(X_val_fold)

    acc, sens, spec = compute_metrics(y_val_fold, y_pred_fold)

    results.append({
        "Fold": fold,
        "Accuracy": acc,
        "Sensitivity": sens,
        "Specificity": spec
    })

# Convert CV to dataframe
cv_results_df = pd.DataFrame(results)
cv_results_df.loc["Mean"] = cv_results_df.mean(numeric_only=True)

print("\nCross-Validation Performance:\n")
print(cv_results_df.round(3))

# Save CV metrics
cv_results_df.to_csv(f"{output_folder}/cv_metrics_SMCI_PMCI.csv", index=True)

# Biomarker identification
avg_coef = np.mean(np.abs(coef_list), axis=0)

gene_importance = pd.DataFrame({
    "gene": X.columns,
    "avg_abs_importance": avg_coef
}).sort_values(by="avg_abs_importance", ascending=False)

# Export top 30 genes
top_genes = gene_importance.head(30)
top_genes.to_csv(f"{output_folder}/LinearSVMtop_30_gene_biomarkers_SMCI_PMCI.csv", index=False)

print("\nTop 30 S-MCI vs P-MCI gene biomarkers exported successfully!")

# External cohort evaluation
test_df = pd.read_csv(test_dataset)

label_map = {
    "MCI": "S-MCI",
    "AD": "P-MCI"
}
test_df["disease_state"] = test_df["disease_state"].map(label_map)

print("\nExternal cohort label counts after mapping:")
print(test_df["disease_state"].value_counts())

# Filter valid samples
test_df = test_df[test_df["disease_state"].isin(["S-MCI", "P-MCI"])]
print("\nFiltered external cohort shape:", test_df.shape)
```

Methods - CNN

Architecture - Pytorch

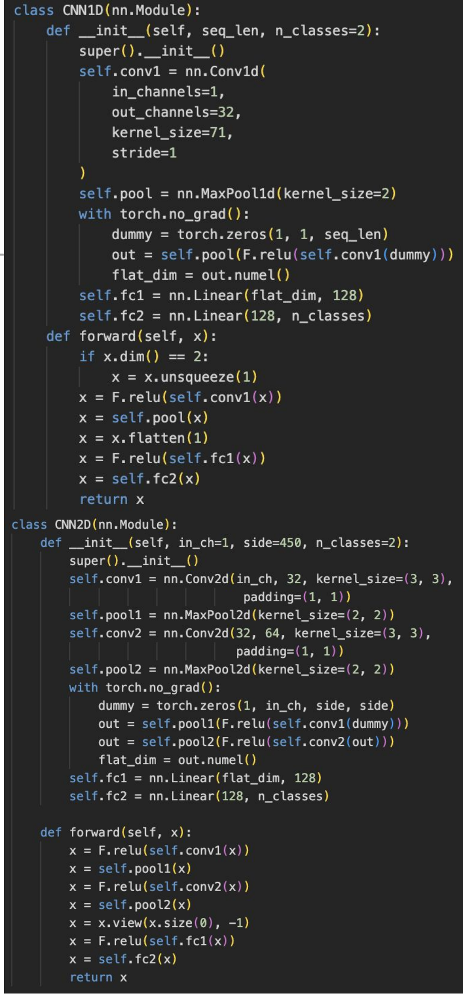
Loss function: Cross Entropy Loss
Optimizer: Adam
Hyper-parameters: lr=0.001, epoch=100
Cross-validation: StratifiedKFold(K = 5)
CNN1D() & CNN2D()

Training performance

	Accuracy	Sensitivity	Specificity
CNN1D Cross-Validation Mean	0.642	0.367	0.850
CNN2D Cross-Validation Mean	0.688	0.493	0.818

Biomarker identification

- Using the **Integrated Gradients**
- Rank top 30 genes

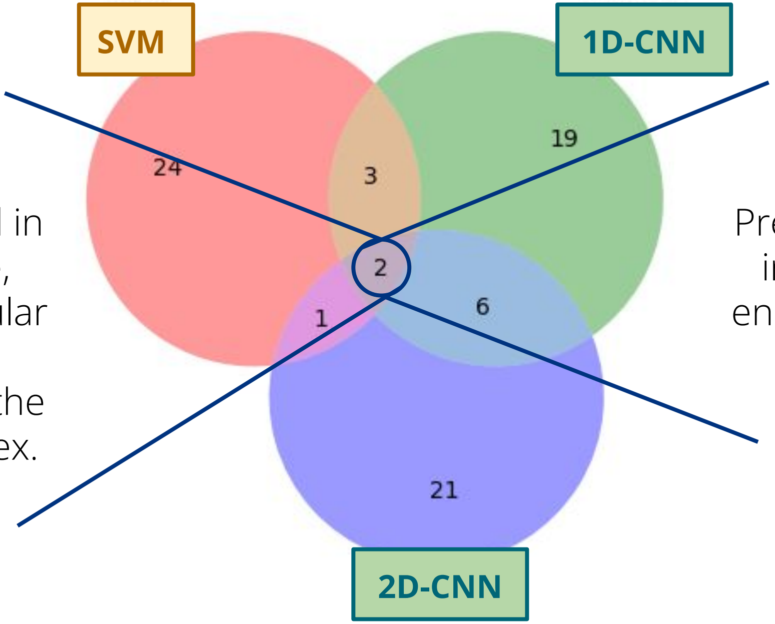


```python
class CNN1D(nn.Module):
    def __init__(self, seq_len, n_classes=2):
        super().__init__()
        self.conv1 = nn.Conv1d(
            in_channels=1,
            out_channels=32,
            kernel_size=71,
            stride=1
        )
        self.pool = nn.MaxPool1d(kernel_size=2)
        with torch.no_grad():
            dummy = torch.zeros(1, 1, seq_len)
            out = self.pool(F.relu(self.conv1(dummy)))
            flat_dim = out.numel()
        self.fc1 = nn.Linear(flat_dim, 128)
        self.fc2 = nn.Linear(128, n_classes)
    def forward(self, x):
        if x.dim() == 2:
            x = x.unsqueeze(1)
        x = F.relu(self.conv1(x))
        x = self.pool(x)
        x = x.flatten(1)
        x = F.relu(self.fc1(x))
        x = self.fc2(x)
        return x
class CNN2D(nn.Module):
    def __init__(self, in_ch=1, side=450, n_classes=2):
        super().__init__()
        self.conv1 = nn.Conv2d(in_ch, 32, kernel_size=(3, 3),
                               padding=(1, 1))
        self.pool1 = nn.MaxPool2d(kernel_size=(2, 2))
        self.conv2 = nn.Conv2d(32, 64, kernel_size=(3, 3),
                              padding=(1, 1))
        self.pool2 = nn.MaxPool2d(kernel_size=(2, 2))
        with torch.no_grad():
            dummy = torch.zeros(1, in_ch, side, side)
            out = self.pool1(F.relu(self.conv1(dummy)))
            out = self.pool2(F.relu(self.conv2(out)))
            flat_dim = out.numel()
        self.fc1 = nn.Linear(flat_dim, 128)
        self.fc2 = nn.Linear(128, n_classes)

    def forward(self, x):
        x = F.relu(self.conv1(x))
        x = self.pool1(x)
        x = F.relu(self.conv2(x))
        x = self.pool2(x)
        x = x.view(x.size(0), -1)
        x = F.relu(self.fc1(x))
        x = self.fc2(x)
        return x
```

Results

Venn Diagram of top 30 genes biomarker



IGKV2D-29

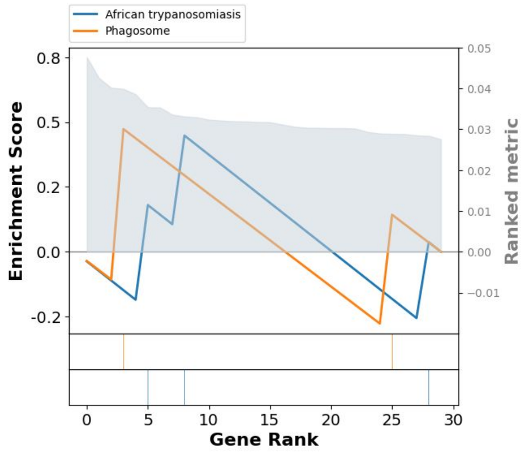
Predicted to be involved in the immune response, which is in the extracellular region and plasma membrane as a part of the immunoglobulin complex.

FAM118A

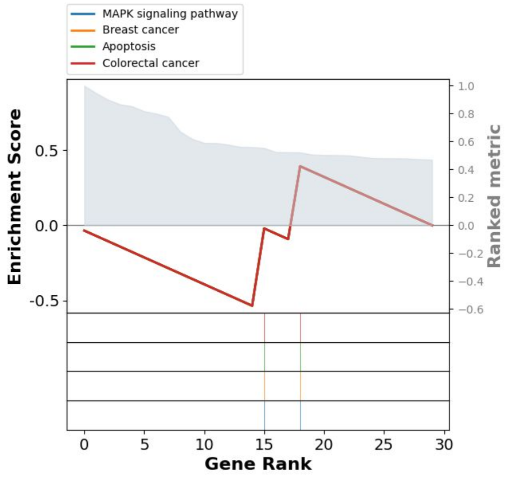
Predicted to be located in the membrane to enable identical protein binding activity.

Results

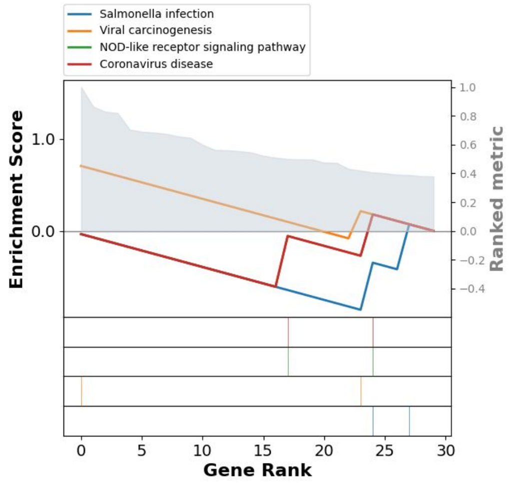
GSEA with top 30 genes



SVM



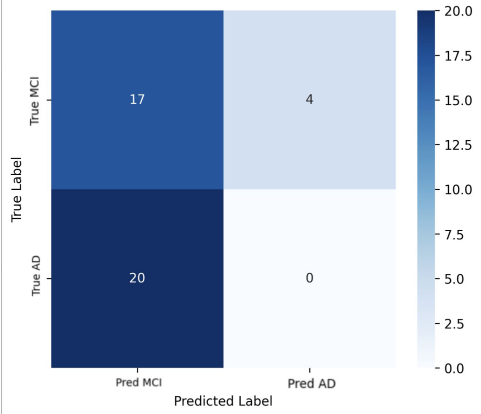
1D-CNN

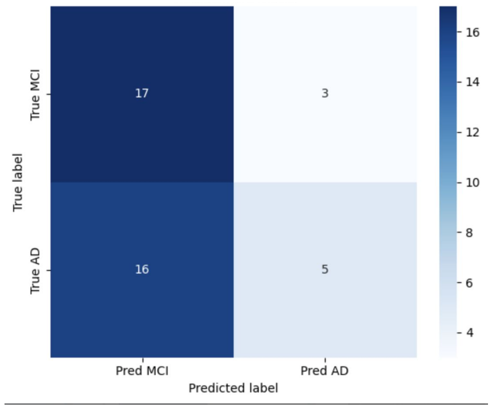


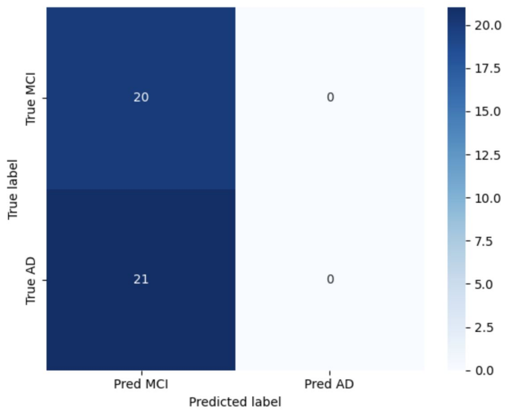
2D-CNN

Results

Confusion Matrix on External cohort







SVM

1D-CNN

2D-CNN

Results

Evaluation Performances

	SVM	1D-CNN	2D-CNN
Accuracy	41.46%	53.66%	48.78%
Sensitivity	80.95%	23.81%	0%
Specificity	0%	85%	100%

Conclusion

Aim 1: Identify a set of gene expression biomarkers for high-risk and low-risk MCI to AD dementia prognosis

What we found: Fam118A and IGKV2D-29 are two biomarkers that we can use to categorize MCI to AD dementia prognosis

Aim 2: Compare the performance of predictive models between machine learning and deep learning approaches

What we found: After comparing evaluation performances, 1D-CNN is likely the best model to use, but features from the SVM and 2D-CNN models can be incorporated to improve model accuracy, sensitivity, and specificity

References

1. Mian M, Tahiri J, Eldin R, Altabaa M, Sehar U, Reddy PH. Overlooked cases of mild cognitive impairment: Implications to early Alzheimer's disease. Ageing Res Rev. 2024 Jul;98:102335. doi: 10.1016/j.arr.2024.102335. Epub 2024 May 12. PMID: 38744405; PMCID: PMC11180381.
2. Petersen RC. Mild cognitive impairment as a diagnostic entity. J Intern Med. 2004 Sep;256(3):183-94. doi: 10.1111/j.1365-2796.2004.01388.x. PMID: 15324362.
3. Di Costanzo A, Paris D, Melck D, Angiolillo A, Corso G, Maniscalco M, et al. Blood biomarkers indicate that the preclinical stages of Alzheimer’s disease present overlapping molecular features. Sci Rep. 2020;10:15612.
4. Li X, Wang H, Long J, Pan G, He T, Anichtchik O, et al. Systematic analysis and biomarker study for Alzheimer’s disease. Sci Rep. 2018;8:17394.
5. Karaman BK, Mormino EC, Sabuncu MR. Machine learning based multi-modal prediction of future decline toward Alzheimer’s disease: an empirical study. PLoS ONE. 2022;17:e0277322.
6. Mian M, Tahiri J, Eldin R, Altabaa M, Sehar U, Reddy PH. Overlooked cases of mild cognitive impairment: Implications to early Alzheimer's disease. Ageing Res Rev. 2024 Jul;98:102335. doi: 10.1016/j.arr.2024.102335. Epub 2024 May 12. PMID: 38744405; PMCID: PMC11180381.
7. Huang YL, Tsai TH, Shen ZQ, Chan YH et al. Transcriptomic predictors of rapid progression from mild cognitive impairment to Alzheimer's disease. Alzheimers Res Ther 2025 Jan 3;17(1):3. PMID: 39754267
8. Iga JI, Yoshino Y, Ozaki T, Tachibana A, Kumon H, Funahashi Y, Mori H, Ueno M, Ozaki Y, Yamazaki K, Ochi S, Yamashita M, Ueno SI. Blood RNA transcripts show changes in inflammation and lipid metabolism in Alzheimer's disease and mitochondrial function in mild cognitive impairment. J Alzheimers Dis Rep. 2024 Dec 23;8(1):1690-1703. doi: 10.1177/25424823241307878. PMID: 40034360; PMCID: PMC11863738.
9. T. Khorshed, M. N. Moustafa and A. Rafea, "Deep Learning for Multi-Tissue Cancer Classification of Gene Expressions (GeneXNet)," in IEEE Access, vol. 8, pp. 90615-90629, 2020, doi: 10.1109/ACCESS.2020.2992907
10. Mostavi M, Chiu YC, Huang Y, Chen Y. Convolutional neural network models for cancer type prediction based on gene expression. BMC Med Genomics. 2020 Apr 3;13(Suppl 5):44. doi: 10.1186/s12920-020-0677-2. PMID: 32241303; PMCID: PMC7119277.