AMRITA VISHWA VIDYAPEETHAM

Toxicity Prediction of Daily Medications Using Multi-Task Neural Networks

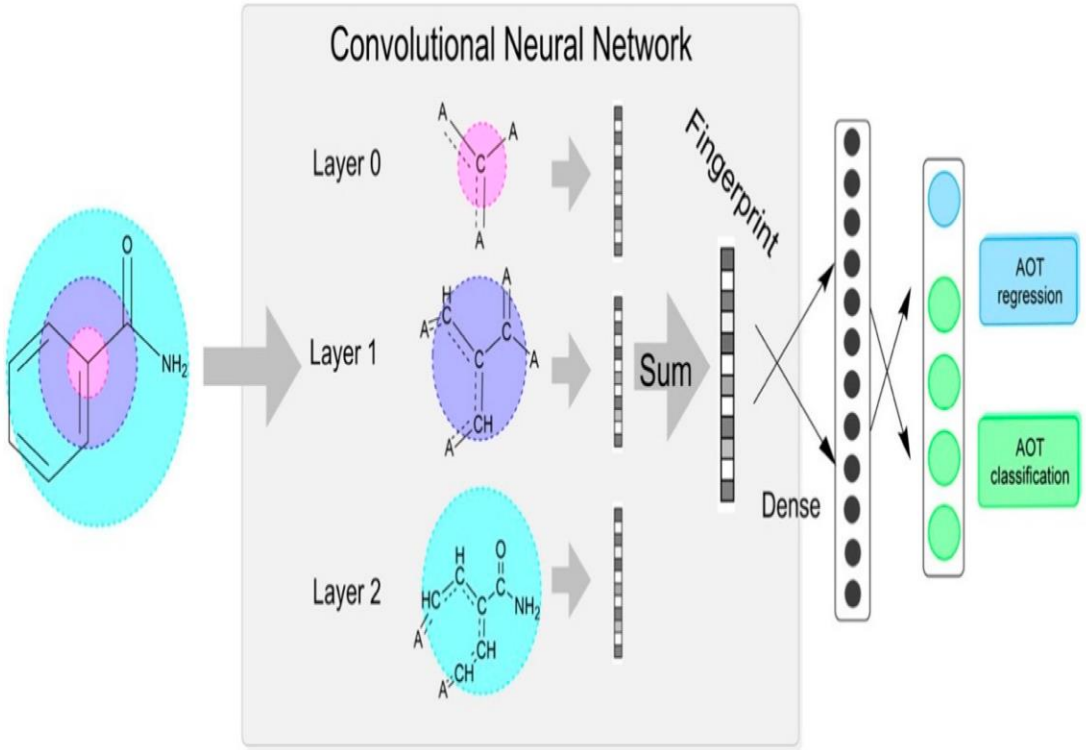
24AIM112 – Molecular Biology and Basic Cellular Physiology

24AIM115 – Ethics, Innovative research, Businesses and IPR

TEAM MEMBERS:

1.	AKHILLESH VARATHAN CS	CB.AI.U4AIM24102
2.	JEFFRIN MERINO J	CB.AI.U4AIM24118
3.	DEEPAK SKANDH K	CB.AI.U4AIM24119
4.	KAVIN M	CB.AI.U4AIM24121

INTRODUCTION



- **Why Toxicity Prediction?**

Identifying toxicity effects early in development is crucial for patient safety.

Thousands of drug candidates fail due to toxicity concerns.

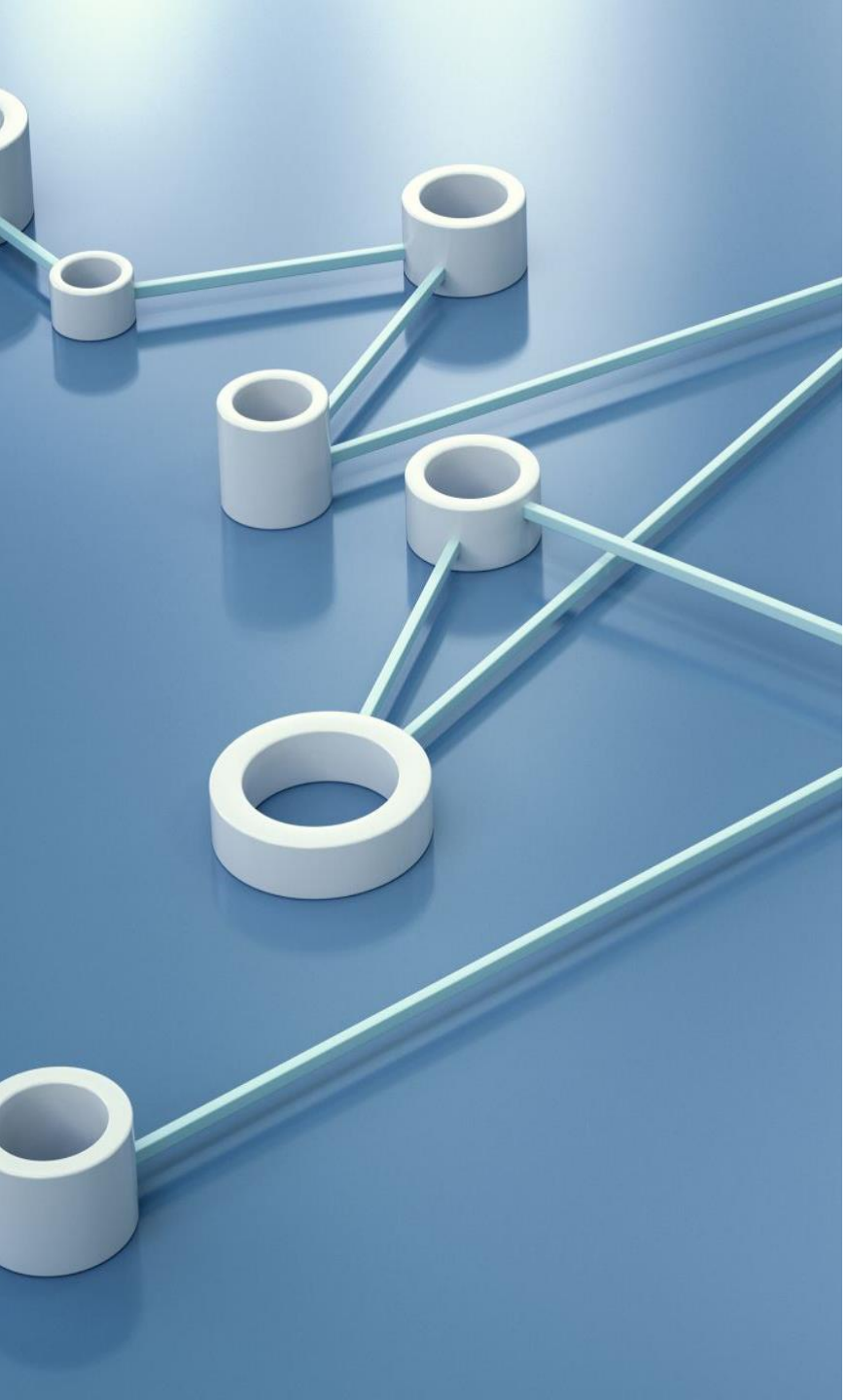
- **Challenges with Daily Medications**

Drugs we consume daily can pose organ-specific risks.

Traditional toxicity testing is time consuming, and often not scalable.

LITERATURE REVIEW

S.No	TOXICITY PAPER	METHOD
1.	Lee et al. (2020) – "Molecular Fingerprint Analysis for Toxicity Screening"	Used traditional machine learning with molecular descriptors.
2.	Zhang et al. (2023) – "Graph Neural Networks for Toxicity Prediction in Drug Discovery"	Highlighted scalability and generalizability in multi-target toxicity tasks.

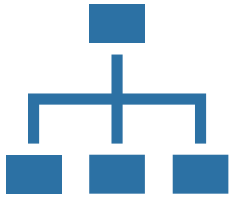


OBJECTIVES

- Developing a neural network that can perform multi-label classification.
- Accessing how well the multi-task neural network works on unseen data.
- Using some regularization techniques such as Dropout, Early Stopping, and Adam optimizer.

PROBLEM OVERVIEW

Molecular Fingerprinting represents chemical structures in fixed size binary vectors.

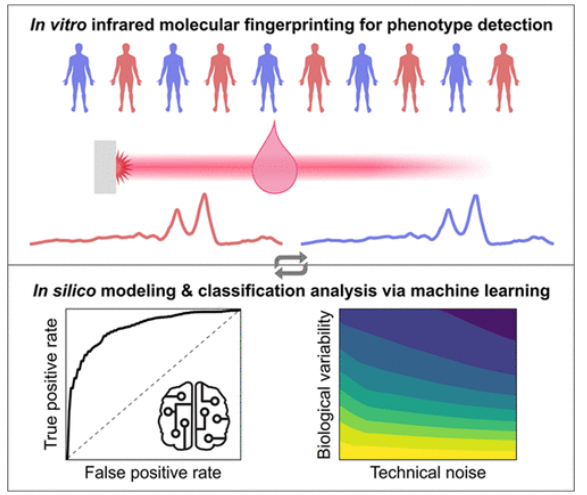


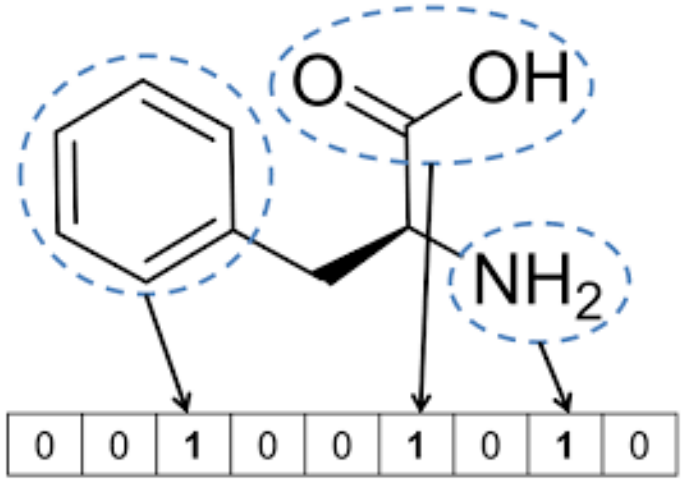
Classification Problem.

Understanding these properties can help in predicting molecular behaviors and potential toxicity in drug development.

DATA OVERVIEW

- The dataset consists of 2048 molecular fingerprint features.
- The features are extracted using the chemical structures.
- The labels are 9 binary categories: CYP450, EYE, ENDOCRINE, RESPIRATION, CARDIO, HEPA, REPRODUCTION, MUTATION, and CARCINOGENS.
- The dataset is split into Train, Val and Test respectively.





PYTHON CODE

```
# Load Training Dataset
train_fingerprints_df = pd.read_csv(r"D:\bio_sem2_project\datasets\merged_dataset\X_train.csv")
train_labels_df = pd.read_csv(r"D:\bio_sem2_project\datasets\merged_dataset\y_train.csv")

# Load Validation Dataset
val_fingerprints_df = pd.read_csv(r"D:\bio_sem2_project\datasets\merged_dataset\X_val.csv")
val_labels_df = pd.read_csv(r"D:\bio_sem2_project\datasets\merged_dataset\y_val.csv")

# Identify Fingerprint and Label Columns
fingerprint_cols = list(map(str, range(2048)))  # 0-2047 molecular fingerprint features
label_cols = ["CYP450", "EYE", "ENDOCRINE", "RESPIRATION", "CARDIO", "HEPA", "REPRODUCTION", "MUTATION", "CARCINOGENS"]

# Convert Data to NumPy Arrays
X_train = train_fingerprints_df[fingerprint_cols].values.astype(np.float32)
Y_train = train_labels_df[label_cols].values.astype(np.float32)

X_val = val_fingerprints_df[fingerprint_cols].values.astype(np.float32)
Y_val = val_labels_df[label_cols].values.astype(np.float32)

# Convert to PyTorch Tensors and Move to Device
X_train_tensor = torch.tensor(X_train).to(device)
Y_train_tensor = torch.tensor(Y_train).to(device)

X_val_tensor = torch.tensor(X_val).to(device)
Y_val_tensor = torch.tensor(Y_val).to(device)

# Create PyTorch Datasets and Dataloaders
train_dataset = Data.TensorDataset(X_train_tensor, Y_train_tensor)
val_dataset = Data.TensorDataset(X_val_tensor, Y_val_tensor)

train_loader = Data.DataLoader(train_dataset, batch_size=64, shuffle=True)
val_loader = Data.DataLoader(val_dataset, batch_size=64, shuffle=False)
```

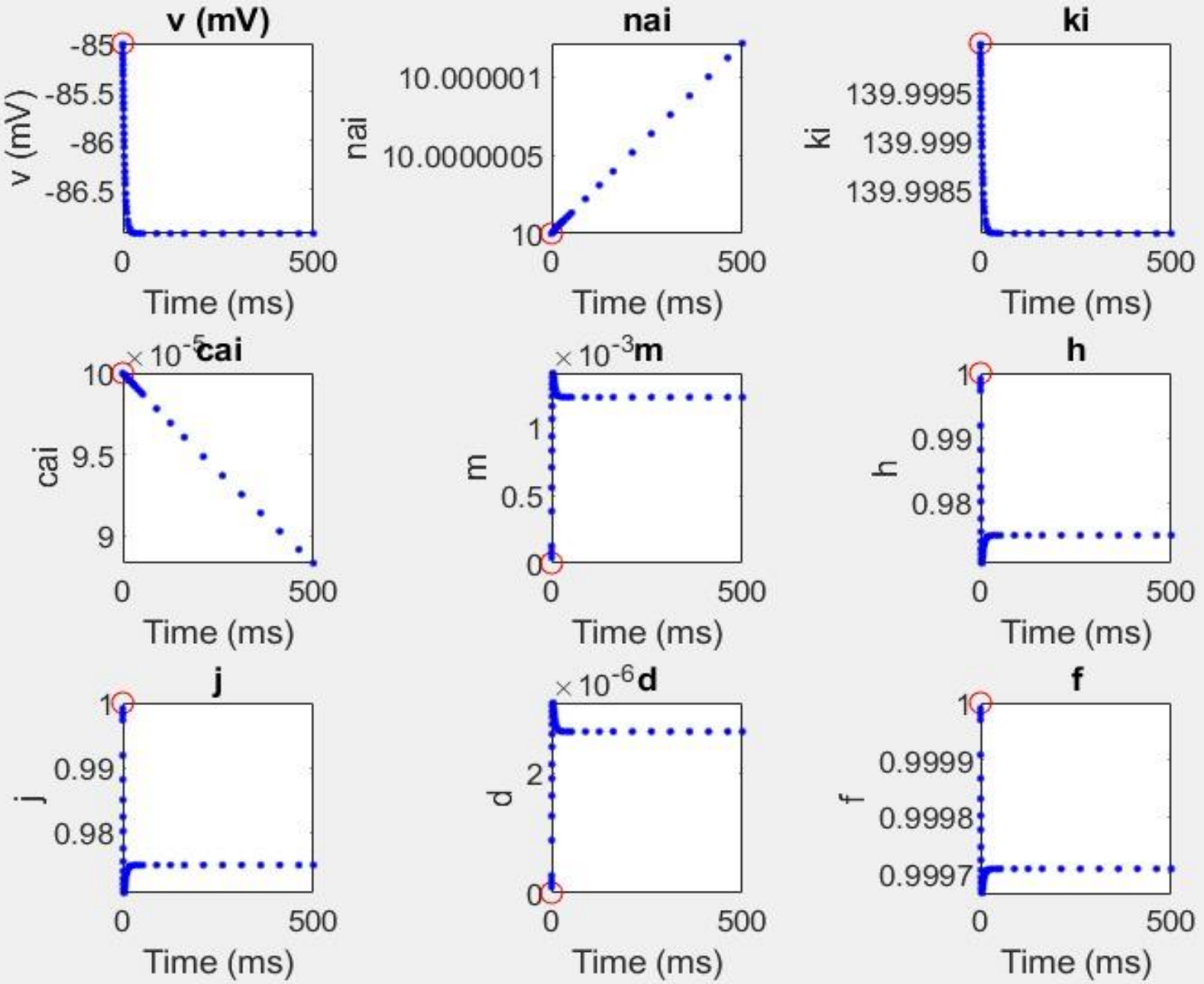
PYTHON CODE

```
# Define Multi-Task Neural Network Model with Dropout
class MultiTaskNN(nn.Module):
    def __init__(self, input_size, output_size):
        super(MultiTaskNN, self).__init__()
        self.fc1 = nn.Linear(input_size, 1024)
        self.fc2 = nn.Linear(1024, 512)
        self.fc3 = nn.Linear(512, output_size)

        self.dropout = nn.Dropout(0.3)  # Drop 30% neurons randomly
        self.relu = nn.ReLU()
        self.sigmoid = nn.Sigmoid()

    def forward(self, x):
        x = self.relu(self.fc1(x))
        x = self.dropout(x)  # Dropout after first layer
        x = self.relu(self.fc2(x))
        x = self.dropout(x)  # Dropout after second layer
        x = self.sigmoid(self.fc3(x))  # Binary classification output
        return x

# Initialize Model, Loss Function, and Optimizer
input_size = 2048
output_size = 9

model = MultiTaskNN(input_size, output_size).to(device)  # Move model to GPU/CPU
criterion = nn.BCELoss()
optimizer = optim.Adam(model.parameters(), lr=0.0005, weight_decay=1e-4)  # Added L2 regularization
```

OUTPUT

Toxicity Type	Accuracy	F1-Score	Sensitivity (Recall)	ROC-AUC Score
Respiratory Toxicity	0.9877	0.9840	0.9877	0.8356
Mutagenicity	0.9616	0.9585	0.9616	0.9694
Eye Irritation	0.9533	0.9508	0.9533	0.9488
Hepatotoxicity	0.9766	0.9682	0.9766	0.8476
Endocrine Disruption	0.9931	0.9908	0.9931	0.8588
Reproductive Toxicity	0.9967	0.9951	0.9967	0.8921
CYP450 Inhibition	0.9654	0.9623	0.9654	0.9380
Cardiotoxicity	0.9954	0.9945	0.9954	0.9713
Carcinogenicity	0.9925	0.9896	0.9925	0.8724
LD50	0.7064	0.7027	0.7064	N/A

OUTPUT


v (mV)
-85
-85.5
-86
-86.5
v (mV)
0
500
Time (ms)
nai
10.000001
10.0000005
10
nai
0
500
Time (ms)
ki
139.9995
139.999
139.9985
ki
0
500
Time (ms)
cai
×10^-5
10
9.5
9
cai
0
500
Time (ms)
m
×10^-3
1
0.5
0
m
0
500
Time (ms)
h
1
0.99
0.98
h
0
500
Time (ms)
j
1
0.99
0.98
j
0
500
Time (ms)
d
×10^-6
2
0
d
0
500
Time (ms)
f
1
0.9999
0.9998
0.9997
f
0
500
Time (ms)

ETHICAL CONSIDERATIONS

- Molecular Datasheets should be accessed legally and transparently
- Predictive Models should be fair and impartial ensuring decisions are not influenced by biases
- Predictive Models shouldn't be misused under any considerations which violate ethical standards such as unethical drug synthesis, false predictions etc.

INTELLECUTAL PROPERTY RIGHTS:

•**Patent No**.: US 2024/0161863 A1 •**Date: May** 16 , 2024 •**Topic:** Interrogatory cell based assays for identifying drug induced toxicity markers. •**Authors**:Niven Narain •**Description**: A platform tech for identifying markers with drug induced toxicity integrating with molecular interactions via organoids.	•**Patent No**: US 2022/0383992 AI •**Date**: Dec,1,2022 •**Topic**: Machine Learning based methods of Analyzing drug-like molecules •**Author**: Kuano kevin •**Description**: machine learning based method to analysis drug structures.

CONCLUSION

- Efficient Toxicity Prediction

1. Our model successfully predicts multiple types of drug toxicity using single neural network.
2. This approach can aid pharmaceutical companies and regulatory agencies in early drug screening and risk assessment.
3. Shared learning across tasks improved accuracy, reduced overfitting.