Enhancing Credit Card Fraud Detection through ML and DL

Mid-Quarter Presentation / Pitch

Ya Wei Tsai

MPCS 2024 Spring 53120 Applied Data Analysis

April 15, 2024

Introduction

Project Objectives:

- Enhance the detection of credit card fraud using advanced machine learning (ML) and deep learning (DL) techniques.
- Develop a robust fraud detection system using ensemble methods and autoencoders.

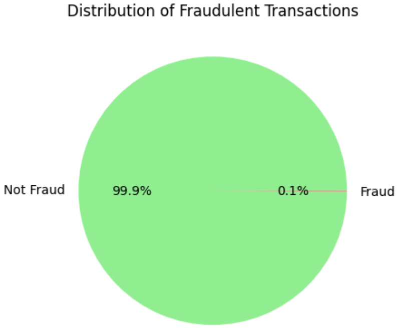
Importance:

- Combat the increasing sophistication of fraudsters.
- Utilize state-of-the-art technology to improve detection and prevent fraud.

Data and Resources

Data Sources:

- Real-life credit card transaction data from Kaggle.
- Preprocessing using a variety of techniques to handle imbalances and enhance model training.



Tools and Libraries

Tools and Libraries:

- Python, Pandas, NumPy for data manipulation.
- TensorFlow, PyTorch for deep learning models.
- Scikit-learn and cuml for ML models on GPU.
- matplotlib for visualization

Progress So Far

Completed Tasks:

- Initial data preprocessing and exploration.
- Basic ML and DL models implementation and preliminary testing.
- Transition to GPU-accelerated algorithms using the cuml library.

Model Evaluation

Metrics:

- Accuracy: The ratio of correctly predicted observations to the total observations.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

  where TP is True Positive, TN is True Negative, FP is False Positive, and FN is False Negative.

- Matthews Correlation Coefficient (MCC): A balanced measure even if the classes are of very different sizes.

$$\text{MCC} = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP) \times (TP + FN) \times (TN + FP) \times (TN + FN)}}$$

  where TP, TN, FP, and FN are the numbers of true positive, true negative, false positive, and false negative predictions, respectively.

CNN Model Code

```python
class FraudCNN(nn.Module):
    def __init__(self, num_features):
        super(FraudCNN, self).__init__()
        self.conv1 = nn.Conv1d(1, 16, kernel_size=3,
                               stride=1, padding=1)
        self.pool = nn.MaxPool1d(2)
        self.conv2 = nn.Conv1d(16, 32, kernel_size=3,
                               stride=1, padding=1)
        self.fc1 = nn.Linear(32 * num_features // 2, 64)
        self.fc2 = nn.Linear(64, 1)

    def forward(self, x):
        x = x.unsqueeze(1)
        x = F.relu(self.conv1(x))
        x = self.pool(x)
        x = F.relu(self.conv2(x))
        x = x.view(x.size(0), -1)
        x = F.relu(self.fc1(x))
        x = torch.sigmoid(self.fc2(x))
        return x
```

Progress So Far - Benchmar Performance

Performance Metrics:

- Accuracy: 0.9995 (A benchmark for model evaluation)
- Matthews Correlation Coefficient (MCC): 0.7931 (Primary focus metric)

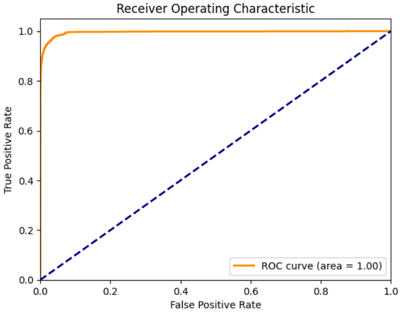


Figure 2: ROC Curve.

Performance Metrics:

- Accuracy: 0.999387
- Matthews Correlation Coefficient (MCC): 0.3528 (Primary focus metric)

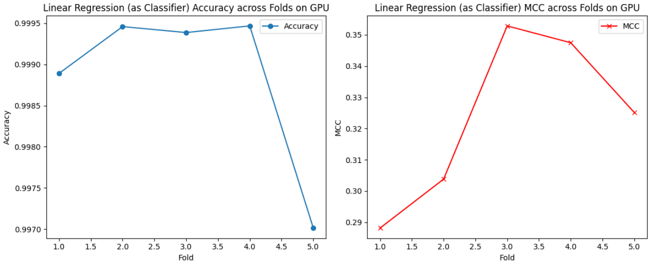


Figure 3: Accuracy And MCC Curve.

Progress So Far - Lasso Regression (Alpha = 0.001)

Performance Metrics:

- Accuracy: 0.9994
- Matthews Correlation Coefficient (MCC): 0.000 (Primary focus metric)

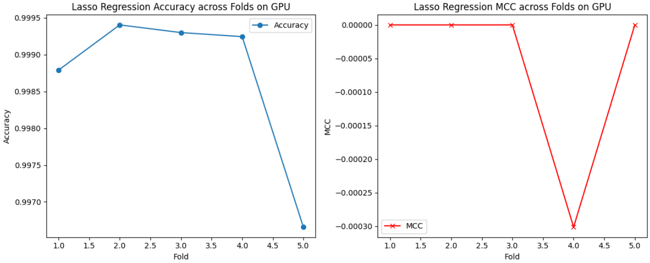


Figure 4: Accuracy And MCC Curve.

Progress So Far - Logistic Regression

Performance Metrics:

- Accuracy: 0.99966
- Matthews Correlation Coefficient (MCC): 0.68293 (Primary focus metric)

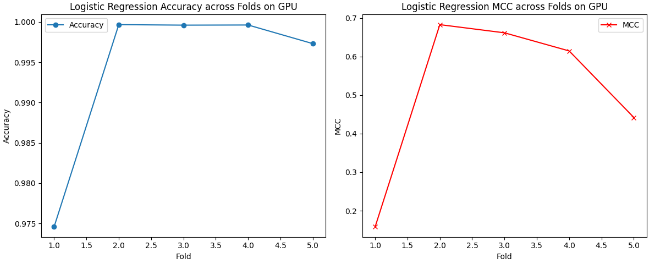


Figure 5: Accuracy And MCC Curve.

Progress So Far - KNN

Performance Metrics:

- Accuracy: 0.99948 (K = 2)
- Matthews Correlation Coefficient (MCC): 0.775969 (Primary focus metric)

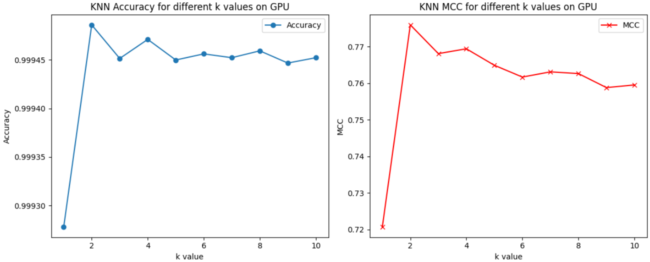


Figure 6: Accuracy And MCC Curve.

Progress So Far - Roadblocks

Challenges/Obstacles:

- Computational limitations encountered with large datasets.
- Handling the severe class imbalance in fraud detection datasets.

Next Steps

Plan of Action:

- Integrate ensemble learning techniques to improve detection accuracy.
- Try to conduct PCA
- Explore and implement advanced deep learning architectures.
- Continue refining data preprocessing to enhance model training.

Anticipated Challenges:

- Adjusting models to cope with the dynamic nature of fraud patterns.
- Using some design like dropout to prevent overfitting problems.

Conclusion

Summary:

- Our project is well-positioned to make significant advancements in fraud detection.
- The integration of ML and DL holds promising potential for developing a highly effective fraud detection system.

Thank You