Model Report

1. Model Code and Documentation

Model Architecture:

1. Input Layer: A dense layer with 64 units and LeakyReLU activation.

2. Output Layer: Softmax activation for multi-class classification.

Code Snippet:

```python
model = models.Sequential()

model.add(layers.Dense(units=64, input_shape=(input_dim,), activation='linear'))

model.add(layers.LeakyReLU(alpha=0.01)) model.add(layers.Dense(units=num_classes,

activation='softmax'))


# Compile the model

model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
```

2. Updated Model Performance Report

- Test Accuracy: 90.52%

Classification Report:

```
          precision    recall  f1-score   support


       0      0.91      1.00      0.95    115336

       1      0.00      0.00      0.00     12077
```

```
    accuracy                           0.91     127413
   macro avg        0.45       0.50       0.48     127413
weighted avg        0.82       0.91       0.86     127413
```

3. Citation Report

Libraries and Resources:

- pandas: Data manipulation and cleaning
- numpy: Numerical operations
- tensorflow/keras: Model architecture and training
- sklearn: Preprocessing, evaluation metrics

4. Summary

Summary of Results:

- The model achieved a test accuracy of 90.52% with a strong performance on class 0.
- However, class 1 (the minority class) was poorly predicted, indicating class imbalance.
- Future improvements could focus on balancing techniques like class weighting or oversampling theminority class.