i useed this:

X\_jma shape: (85332, 3, 10)

y\_jma shape: (85332,)

# Define the CNN-GRU model  
def build\_cnn\_gru\_model(input\_shape):  
 model = Sequential()  
  
 # CNN layers  
 model.add(Conv1D(filters=64, kernel\_size=1, activation='relu', input\_shape=input\_shape)) # Changed kernel\_size to 1  
 model.add(BatchNormalization())  
 model.add(MaxPooling1D(pool\_size=1)) # Adjusted pool\_size to avoid downsampling issues  
  
 model.add(Conv1D(filters=128, kernel\_size=1, activation='relu')) # Changed kernel\_size to 1  
 model.add(BatchNormalization())  
 model.add(MaxPooling1D(pool\_size=1)) # Adjusted pool\_size  
  
 # GRU layers  
 model.add(GRU(units=128, return\_sequences=True))  
 model.add(Dropout(0.2))  
 model.add(GRU(units=64, return\_sequences=False))  
 model.add(Dropout(0.2))  
  
 # Dense layers  
 model.add(Dense(units=32, activation='relu'))  
 model.add(Dropout(0.2))  
 model.add(Dense(units=1, activation='linear')) # Regression output  
  
 # Compile the model  
 model.compile(optimizer=Adam(learning\_rate=0.001), loss='mse', metrics=['mae'])  
  
 return model

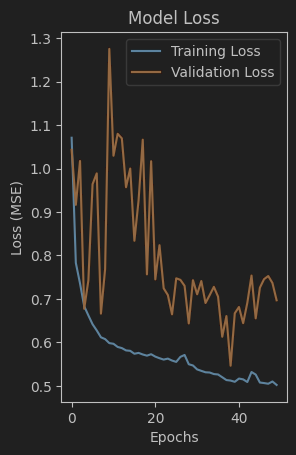
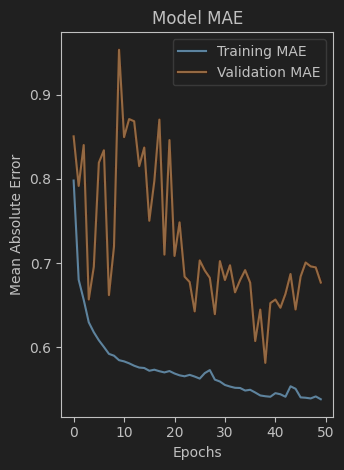
# Define the input shape after preprocessing  
input\_shape = (window\_size, X\_jma.shape[2]) # (3, 10)

# Train the model  
history = model.fit(  
 X\_jma, y\_jma,  
 validation\_split=0.2,  
 epochs=50,  
 batch\_size=64  
)

and I got this:

Training Loss: 0.7158

Training MAE: 0.6664



From the training and validation curves, we can analyze the performance and suggest improvements for your CNN-GRU model:

**Observations:**

1. **Training Loss and Validation Loss**:
   * The training loss shows a consistent downward trend, which indicates the model is learning.
   * However, the validation loss fluctuates significantly, suggesting potential overfitting or that the model isn't generalizing well to unseen data.
2. **Training and Validation MAE**:
   * The training MAE decreases consistently.
   * The validation MAE follows the same trend as the loss, with fluctuations, reinforcing the possibility of overfitting or insufficient generalization.

**Recommendations to Improve Model Performance:**

1. **Increase Time Steps (Window Size)**:
   * The current window size is 3. This might not capture enough temporal information for the GRU to learn meaningful patterns.
   * Try increasing the window size to 5 or 10 time steps and observe how it impacts performance.
2. **Regularization**:
   * Increase dropout rates in the GRU and Dense layers (e.g., from 0.2 to 0.3 or 0.4) to prevent overfitting.
   * Consider adding L2 regularization in Dense and GRU layers.
3. **Batch Size**:
   * Experiment with smaller or larger batch sizes. For example, try 32 or 128 instead of 64, as this can affect learning dynamics.
4. **Learning Rate**:
   * The current learning rate is 0.001. Lower it to 0.0005 or 0.0001 to allow the model to converge more smoothly.
5. **Early Stopping**:
   * Implement early stopping based on validation loss to avoid overfitting and stop training when the model stops improving.
6. **Feature Engineering**:
   * Revisit the features included in X\_jma. Ensure all relevant temporal and spatial features are included. You could also derive new features like moving averages or time-based aggregates.
7. **Normalization**:
   * Confirm the data normalization scales all features appropriately and that there is no data leakage between training and validation sets.
8. **Model Complexity**:
   * Consider adding more layers or units in GRU layers to allow the model to learn more complex patterns. For instance:
     + Increase the first GRU layer to 256 units and the second to 128 units.
     + Add another Conv1D layer with 256 filters before the GRU layers.
9. **Augmentation**:
   * Introduce data augmentation for time-series data, such as adding noise, shifting time, or scaling values slightly to improve generalization.

**Next Steps:**

* Apply these suggestions incrementally and monitor how each change affects the training and validation performance.
* Start by increasing the window size and tuning the dropout rates as these are likely to have the most immediate impact.