If you're observing a loss value that's relatively small (indicating that the model should be doing a good job at reconstruction) but the reconstructed data looks very different from the original when plotted, it suggests a possible disconnect somewhere. Here are a few diagnostic steps and considerations to help narrow down the issue:

Check the Scale of the Data:
When you normalize data using `StandardScaler`, it's important to apply the inverse transformation to the output of your autoencoder to see the reconstructed data in the same scale as the original.

python Copy code

```
reconstructed_data_transformed = train_gen.scaler.inverse_transform(reconstructed_data.reshape(reconstructed_data.shape[0], -1)).reshape(reconstructed_data.shape)
```

Plot `reconstructed_data_transformed` against the original data.

Visualization:
Sometimes, a graph might be misleading if there are a few outliers or if the y-scale is too broad. Ensure the scale of the y-axis is appropriate and try using different visualization techniques.
Consider plotting a histogram or distribution of the reconstruction error for individual data points. This can give insight into the spread of errors.

Model Architecture and Parameters:
It's possible that the model isn't complex enough or is too complex for the data. Experiment with different numbers of LSTM units or layers.
Consider the activation function used in the last layer of the autoencoder. For normalized data with a mean of 0 and a standard deviation of 1, using a 'sigmoid' activation might not be the best choice since its output is bound between [0,1]. A 'linear' activation might be more appropriate.

Training:
Look at the training curve. If the loss stabilizes quickly, the model might be stuck in a local minimum. Try different optimizers or learning rates.
Check the loss value throughout the training epochs. Does it plateau? Does it oscillate?

Loss Calculation:
The loss of 0.5 may seem small, but depending on the scale of your data and the exact nature of the MSE, it might still represent significant discrepancies between your original and reconstructed data.
Additionally, check if the MSE is being correctly computed. If you're using normalized data for training, ensure the inverse-transformed data isn't being compared to normalized reconstructions when computing the loss.

Check the Data:
Always a good step: verify that the input data (`reshaped_data`) looks as expected. Small discrepancies early on can propagate through and manifest as larger issues down the line.

Model Evaluation:
Consider using other metrics in addition to MSE for evaluating your autoencoder's performance. For instance, a perceptual loss or cosine similarity might give a more intuitive sense of how well the autoencoder is reconstructing the data.

After going through these steps, if the issue still persists, you might need a deeper dive into the data or the architecture. Autoencoders can sometimes produce results that are not immediately intuitive, and it can be a matter of fine-tuning or rethinking the approach.