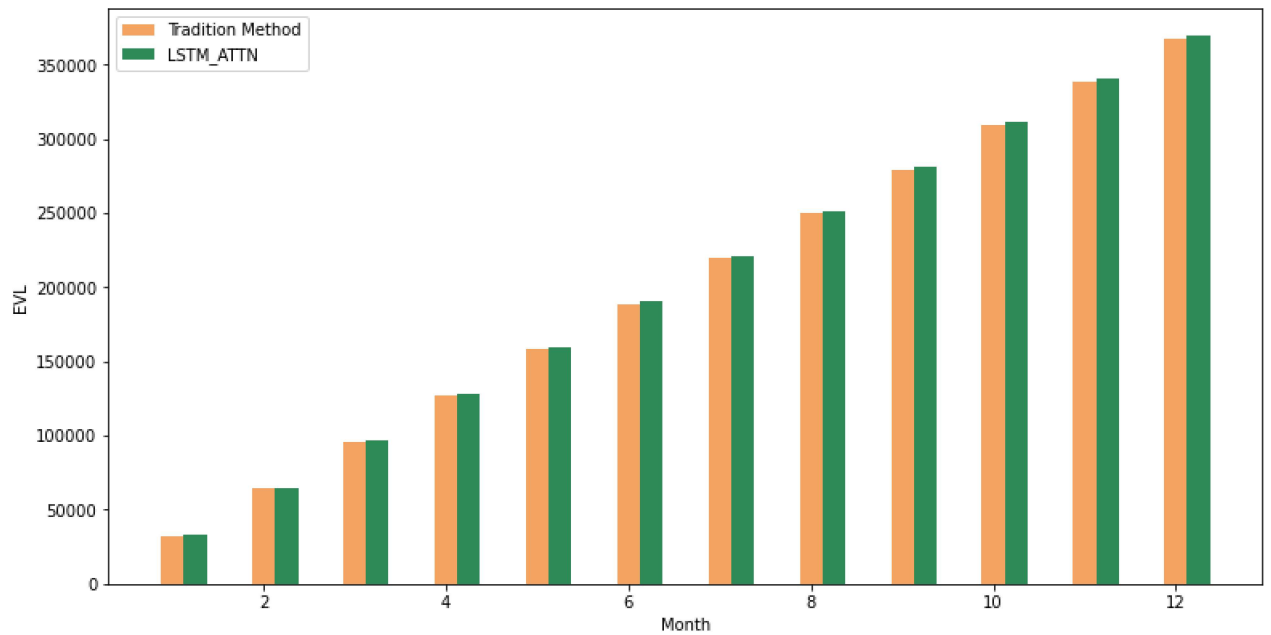
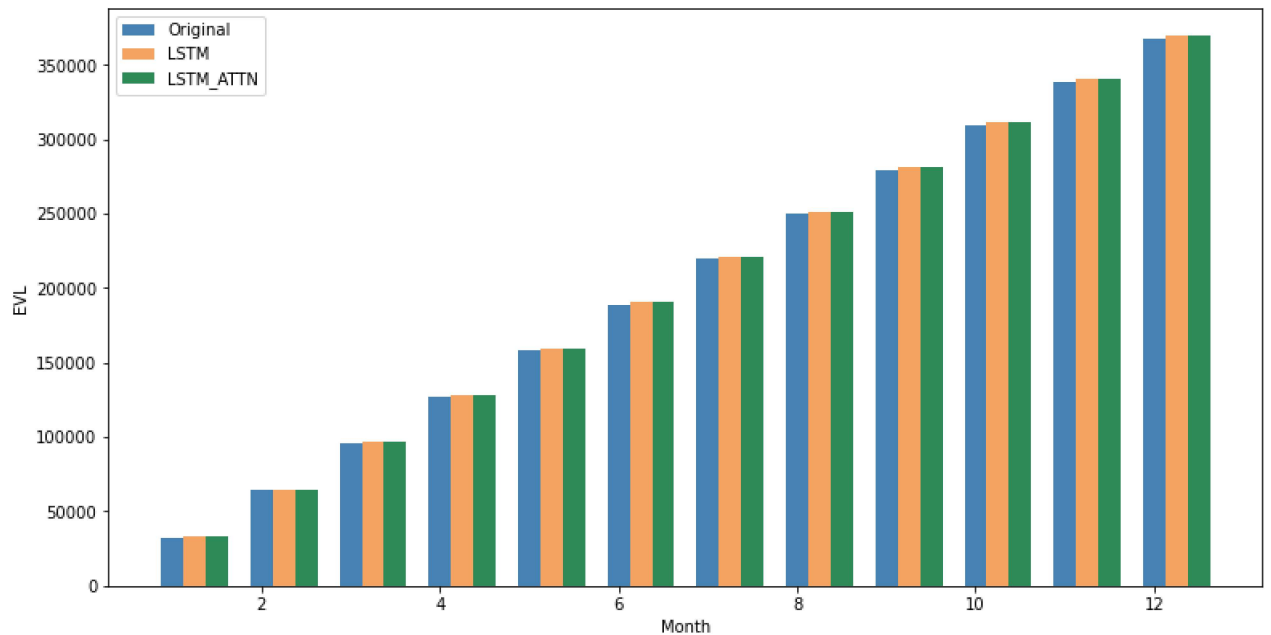


```
plt.legend()
plt.show()
```



```
In [37]: fig = plt.figure(figsize=(10,5))
ax = fig.add_axes([0,0,1,1])
ax.bar(x, Orig_CashFlow, color='steelblue',width=0.25, label='Original')
ax.bar(x+0.25,LSTM_Pred, color='sandybrown', width=0.25,label='LSTM' )
ax.bar(x+0.5,LSTM_Pred, color='seagreen', width=0.25,label='LSTM_ATTN' )
plt.xlabel('Month')
plt.ylabel('EVL')
plt.legend()
plt.show()
```



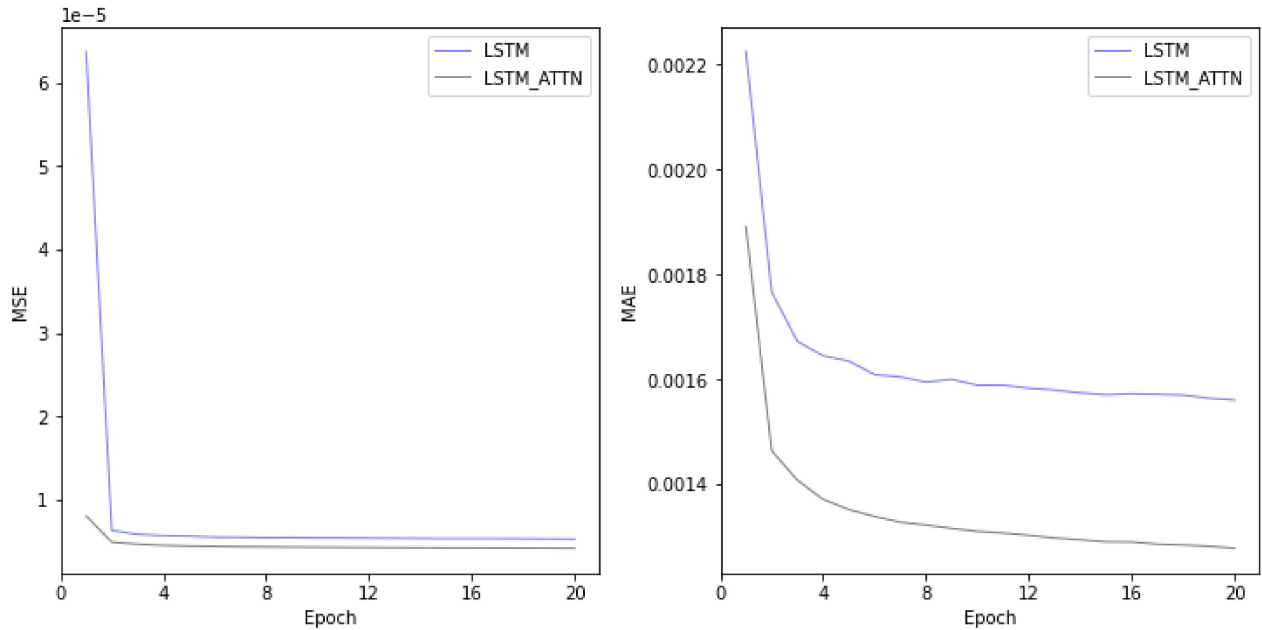
```
In [ ]:
```

```
In [38]: lstm_train_err = pd.read_csv('lstm_train_err.csv', usecols=range(1,3))
```

```

ax[1].plot(x,lstm_train_err['mae'], color='b', linewidth=1, alpha=0.5, label='LSTM')
ax[1].plot(x,lstm_attn_train_err['mae'], color='black', linewidth=1, alpha=0.5, label='LSTM_ATTN')
ax[1].set_xlabel('Epoch')
ax[1].set_ylabel('MAE')
ax[1].set_xticks(major_ticks)
ax[1].legend()
plt.show()

```



In []:

In []:

In []:

In []:

```

criterion = torch.nn.MSELoss()
optimizer = torch.optim.Adam(mv_net.parameters(), lr=0.001)

x_batch = torch.tensor(inpt, dtype=torch.float32)
y_batch = torch.tensor(target, dtype=torch.float32)

LSTM_Attn.init_hidden(x_batch.size(0))
output = LSTM_Attn(x_batch)
mse = criterion(output, y_batch)
mse.backward()
optimizer.step()
optimizer.zero_grad()

```

In []: