HW4 Deep Learning and Industrial Applications

1.

window sizes	steps	MSE
10	15	158.1097
20	5	17.2704
15	10	143.6928

The (20, 5) configuration achieved the lowest MSE because its larger window captures long-term trends while its smaller step retains ample sequence overlap, enabling richer feature learning. The (15, 10) setup yielded a moderate MSE, reflecting a balance between capturing short- and mid-term patterns. In contrast, the (10, 15) configuration produced the highest MSE (358.66); its small window misses long-range dependencies, and its large step reduces overlap, hindering the model's ability to learn crucial temporal patterns. Therefore, larger windows combined with smaller steps significantly improve prediction accuracy.

2. (i)

Input feature	MSE
'Open', 'High', 'Low', 'Close'	158.1097
'Open', 'High', 'Low', 'Close', 'Volume'	1359.9478

With only the four price features (Open, High, Low, Close), the Test MSE is 158.1097; adding Volume raises it dramatically to 1359.9478. This indicates that Volume does not provide effective predictive signals but instead introduces scale mismatches and market noise, hindering the model's fit. If Volume must be included, proper normalization or transformation (e.g., logarithmic scaling, relative volume indicators) is recommended to mitigate noise.

(ii)

Among all subsets of the five features, the combination {Open, High, Close} achieves the lowest Test MSE of approximately 99.7999, outperforming any subset including Low or Volume. These three features capture the core information—opening price, peak, and closing price—while excluding noise from low prices and trading volume. Removing redundant features improves the LSTM's ability to learn temporal dependencies, making {Open, High, Close} the optimal input feature set.

3.

model	MSE
without normalized inputs	138.4237
with normalized inputs	115.0572

Based on experimental results, the LSTM model using the raw four features (Open, High, Low, Close) achieved a Test MSE of 138.42. When we applied Z-score normalization fitted only on the training set and then used that scaling on the test set, the Test MSE dropped to 115.06, a reduction of about 16.9%. This demonstrates that proper input normalization mitigates scale disparities among features, stabilizes the training process, and accelerates convergence, thereby improving predictive accuracy.

4. According to Dehghani et al. (2019), under subject-independent cross-validation, non-overlapping windows (**step = window size**) achieve equivalent recognition performance while saving nearly 10× storage and 4× training time. If step exceeds the window size (window < step), gaps emerge and critical signals are missed, degrading accuracy.
   Reference: Dehghani, A., Sarbishei, O., Glatard, T., & Shihab, E. *A Quantitative Comparison of Overlapping and Non-Overlapping Sliding Windows for Human Activity Recognition Using Inertial Sensors.* Sensors 2019, 19(22), 5026.

5. A time-series-specific augmentation technique leverages **Dynamic Time Warping (DTW)** to synthesize new sequences. It first identifies pairs of similar series via DTW distance, then aligns and combines them—often through weighted averaging—to generate artificial samples. Fawaz et al. (2018) demonstrated that applying this DTW-based augmentation to deep residual networks on the UCR benchmark drastically improves classification accuracy, particularly for small datasets prone to overfitting
   Reference: Fawaz, H. I., Forestier, G., Weber, J., Idoumghar, L., & Muller, P.-A. (2018). *Data augmentation using synthetic data for time series classification with deep residual networks*.

6. **(i) Convolution-based models:**

During inference, a fixed-size sliding window is applied along the time dimension, with convolution kernels producing feature maps or predictions at each step. Thanks to the translation invariance of convolutions, all subwindows can be processed in parallel. Choosing an appropriate stride balances prediction accuracy and throughput.

(ii) Recurrent-based models:

RNNs/LSTMs consume one timestep at a time. At inference, a stateful approach can be used: the hidden state from the previous window is carried into the next, enabling seamless continuous predictions and avoiding redundant computation across overlapping windows.

(iii) Transformer-based models:

Transformers encode entire sequences at once. For long inputs, streaming inference or chunking is employed: the sequence is split into overlapping chunks, and a fixed-size memory of past key/value pairs is kept across chunks to capture long-term dependencies while maintaining computational efficiency.