The CNN-Transformer Time-Series (CTTS) Model, as inspired by the work done by JP. Morgan AI Research (Zhen Zeng, 2023), is designed to classify asset price movements (Up/Down) based on time-series data. It combines convolutional neural networks (CNNs) with Transformer encoders, effectively leveraging the power of both architectures: local pattern detection (from CNN) and global context awareness/capturing long-term dependencies (from Transformers). For the model, it receives an input sequence of shape (500,30), where 500 is the temporal length equating to roughly 2-years of trading period and 30 representing the features which includes the technical indicators and historical price attributes, as mentioned earlier in the data segment.

The model begins with a 1D convolutional layer that extracts local temporal patterns from the input time series. This layer applies multiple filters, equal to the Transformer embedding dimension, with a specified kernel size and stride, allowing the model to down sample the sequence while learning high-level representations. Batch normalisation is applied afterward to stabilise and speed up training. Next, positional embeddings are added to the convolutional output to inject information about the position of tokens within the sequence, which is essential for the Transformer to process order-dependent data.

Following the embedding stage, the core of the model consists of a stack of Transformer encoder layers. Each encoder layer incorporates multi-head self-attention to learn dependencies across different time steps in the sequence, followed by a position-wise feed-forward network (FFN). Residual connections and layer normalisation are used throughout to preserve information and ensure stable gradients. Dropout is applied in attention and FFN layers to mitigate overfitting. This structure enables the model to capture both short-term and long-term dependencies, which could be vital in financial time-series forecasting.

The output of the final Transformer layer is globally averaged across the time dimension, compressing the temporal information into a single fixed-size vector. This representation is passed through a dense layer with ReLU activation, followed by dropout for regularisation. Finally, a sigmoid-activated output layer predicts the binary class label—indicating the direction of stock movement. L2 regularisation is applied to the dense layers to further reduce the risk of overfitting.

During training, the model utilises callbacks like early stopping and learning rate reduction to enhance convergence and generalisation. It also accounts for class imbalance by computing and applying class weights dynamically. The CTTS architecture is particularly well-suited for financial applications due to its ability to model complex sequential dependencies and adapt to variable-length trends in time series data.

The CTTS model used in this study is a modified version of the architecture proposed in the JP. Morgan AI Research paper. Several adjustments were made to better suit the characteristics of the dataset and improve model performance. These include increasing the batch size from 64 to 128, adding L2 kernel regularisation to reduce overfitting, and incorporating early stopping and learning rate reduction callbacks to stabilise training and improve convergence. Additionally, the optimiser was changed from AdamW to the standard Adam optimiser, which provided comparable results with a simpler configuration.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Scaler** | **Target Variable** | **Train Acc** | **Validation Acc** | **Test AUC** |
| MinMax | R01 | 50.9% | 47.2% | 49% |
| MinMax | R05 | 57.7% | 50.9% | 56% |
| MinMax | R10 | 66.6% | 63.4% | 63% |

*(Training & Validation Accuracy based on final epochs from training)*

# Bibliography

Zhen Zeng, R. K. (2023). *Financial Time Series Forecasting using CNN and Transformer.* New York: JP. Morgan AI Research. Retrieved from https://arxiv.org/pdf/2304.04912