\subsection{Method B: Supervised Learning Models for Covariance Prediction (LSTM, Transformer, Autoformer)}

In this approach, we leverage advanced supervised learning models to forecast the covariance structures of asset returns by predicting the Cholesky decomposition of realized covariance matrices. This method ensures the positive definiteness of the predicted covariance matrices and captures complex temporal dependencies inherent in financial time series data. We explore three primary architectures: Long Short-Term Memory (LSTM) networks, Transformers, and Autoformers. Each model's theoretical foundations, core equations, and architectural nuances are detailed below.

\subsubsection{Long Short-Term Memory (LSTM) Networks}

LSTM networks, introduced by Hochreiter and Schmidhuber (1997), are a class of recurrent neural networks (RNNs) designed to capture long-term dependencies in sequential data, effectively addressing the vanishing gradient problem prevalent in traditional RNNs \citep{Hochreiter1997}. The LSTM architecture incorporates memory cells and gating mechanisms to regulate information flow over time.

\paragraph{LSTM Architecture and Equations}

An LSTM unit consists of a cell state ($c\_t$) and three gates: input gate ($i\_t$), forget gate ($f\_t$), and output gate ($o\_t$). The cell state acts as a conveyor belt, allowing information to flow unchanged, while the gates control the addition or removal of information. The computations within an LSTM cell at time step $t$ are as follows:

\begin{align}

f\_t &= \sigma(W\_f x\_t + U\_f h\_{t-1} + b\_f) \quad \text{(Forget gate)} \\

i\_t &= \sigma(W\_i x\_t + U\_i h\_{t-1} + b\_i) \quad \text{(Input gate)} \\

o\_t &= \sigma(W\_o x\_t + U\_o h\_{t-1} + b\_o) \quad \text{(Output gate)} \\

\tilde{c}\_t &= \tanh(W\_c x\_t + U\_c h\_{t-1} + b\_c) \quad \text{(Candidate cell state)} \\

c\_t &= f\_t \odot c\_{t-1} + i\_t \odot \tilde{c}\_t \quad \text{(Cell state update)} \\

h\_t &= o\_t \odot \tanh(c\_t) \quad \text{(Hidden state update)}

\end{align}

Here, $x\_t$ represents the input vector, $h\_{t-1}$ is the previous hidden state, $\sigma$ denotes the sigmoid activation function, and $\odot$ signifies element-wise multiplication. The weight matrices $W$ and $U$, along with biases $b$, are learnable parameters.

\paragraph{Application to Covariance Prediction}

In the context of covariance forecasting, the LSTM network processes sequences of past realized covariance matrices' Cholesky decompositions, capturing temporal dependencies to predict future decompositions. This approach ensures the positive definiteness of the forecasted covariance matrices, crucial for subsequent portfolio optimization.

\begin{figure}[!ht]

\centering

\includegraphics[width=0.7\textwidth]{./figures/lstm\_architecture.png}

\caption{Illustration of an LSTM cell architecture. The diagram depicts the flow of information through the forget, input, and output gates, highlighting the cell state and hidden state interactions.}

\label{fig:lstm\_architecture}

\end{figure}

\subsubsection{Transformer Models}

Transformers, introduced by Vaswani et al. (2017), have revolutionized sequence modeling by employing self-attention mechanisms to capture dependencies without relying on recurrent connections \citep{Vaswani2017}. This architecture allows for modeling relationships across all positions in a sequence simultaneously, facilitating parallelization and capturing long-range dependencies effectively.

\paragraph{Transformer Architecture and Equations}

A standard Transformer model comprises an encoder-decoder structure. Each encoder layer consists of a multi-head self-attention mechanism and position-wise feed-forward networks. The self-attention mechanism computes attention weights to aggregate information from different positions in the input sequence.

The scaled dot-product attention is computed as:

\begin{align}

\text{Attention}(Q, K, V) = \text{softmax}\left( \frac{QK^\top}{\sqrt{d\_k}} \right) V

\end{align}

where $Q$, $K$, and $V$ are the query, key, and value matrices, respectively, and $d\_k$ is the dimension of the key vectors.

Multi-head attention extends this by allowing the model to jointly attend to information from different representation subspaces:

\begin{align}

\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}\_1, \dots, \text{head}\_h) W^O

\end{align}

where each head is computed as:

\begin{align}

\text{head}\_i = \text{Attention}(Q W\_i^Q, K W\_i^K, V W\_i^V)

\end{align}

with learnable projection matrices $W\_i^Q$, $W\_i^K$, $W\_i^V$, and $W^O$.

\paragraph{Application to Covariance Prediction}

For covariance forecasting, the Transformer model processes sequences of Cholesky decompositions of realized covariance matrices, capturing complex temporal and cross-sectional relationships among assets. The self-attention mechanism enables the model to focus on relevant time steps and asset interactions, enhancing prediction accuracy.

\begin{figure}[!ht]

\centering

\includegraphics[width=0.9\textwidth]{./figures/transformer\_architecture.png}

\caption{Transformer model architecture. The diagram illustrates the encoder-decoder structure, highlighting the multi-head self-attention mechanisms and position-wise feed-forward networks.}

\label{fig:transformer\_architecture}

\end{figure}

\subsubsection{Autoformer}

Autoformer, proposed by Wu et al. (2021), is a Transformer-based architecture tailored for long-term time series forecasting. It introduces an auto-correlation mechanism and series decomposition blocks to effectively capture trend and seasonal components in time series data \citep{Wu2021}.

\paragraph{Autoformer Architecture and Equations}

The Autoformer model incorporates two key innovations:

1. \*\*Series Decomposition Block\*\*: This component decomposes the input time series into trend and seasonal components, facilitating the modeling of each component separately.

2. \*\*Auto-Correlation Mechanism\*\*: Replacing traditional self-attention, the auto-correlation mechanism captures period-based dependencies by identifying and utilizing the most relevant sub-series within the time series.

The auto-correlation is computed as:

\begin{align}

\text{AutoCorrelation}(X) = \mathcal{F}^{-1} \left( \text{Softmax}(\mathcal{F}(

::contentReference[oaicite:0]{index=0}

\subsection{Method B: Supervised Learning Models for Covariance Prediction (LSTM, Transformer, Autoformer)}

In contrast to the two-stage econometric modeling in Method A, Method B employs advanced supervised machine learning techniques directly on historical market data to forecast the covariance structure of asset returns. Specifically, we target the Cholesky decomposition of daily realized covariance matrices. The key motivation for this approach is to directly capture complex and potentially nonlinear relationships embedded in high-frequency data, while ensuring that the resulting covariance predictions remain valid and positive definite.

To construct the target matrices, we begin by computing daily realized covariance matrices $\Sigma^{realized}\_{t}$ from intraday (minute-frequency) returns. Formally, the realized covariance matrix for day $t$ is defined as:

\[

\Sigma^{realized}\_{t} = \sum\_{m=1}^{M} r\_{t,m} r\_{t,m}^\top,

\]

where $r\_{t,m}$ denotes the vector of asset returns for minute $m$ within day $t$, and $M$ represents the total number of minute observations during that day. To ensure the positive definiteness and stability of covariance predictions, we employ the Cholesky decomposition:

\[

\Sigma^{realized}\_{t} = L\_{t} L\_{t}^\top,

\]

where $L\_{t}$ is a unique lower-triangular matrix with strictly positive diagonal entries. The supervised models thus directly predict elements of $L\_{t+1}$ rather than the covariance itself.

We explore three sophisticated sequence modeling architectures extensively used in modern financial forecasting:

\begin{itemize}

\item \textbf{Long Short-Term Memory (LSTM) Networks:} The LSTM \citep{Hochreiter1997} architecture is particularly well-suited for capturing complex, nonlinear, and dynamic temporal relationships inherent in financial volatility and covariance. An LSTM network processes a rolling sequence of historical daily Cholesky matrices ($L\_{t-k},\dots,L\_{t}$), alongside relevant supplementary market information such as past returns, technical indicators, and macroeconomic variables. Each daily observation constitutes an input vector sequentially fed into the LSTM layers, which maintain internal states through gating mechanisms. These gates selectively retain, forget, or pass forward information, allowing the model to effectively capture long-range dependencies and volatility persistence—key characteristics observed in cryptocurrency markets. The LSTM outputs the predicted lower-triangular elements of $L\_{t+1}$ through a dense output layer. To enforce the triangular structure of $L\_{t+1}$, the output vector from the LSTM is reshaped accordingly. The final layer's activation functions ensure positivity for diagonal elements, thereby guaranteeing valid Cholesky matrices.

\item \textbf{Transformer-Based Models:} Transformers \citep{Vaswani2017} revolutionized sequence modeling through their self-attention mechanisms, capturing dependencies between all time steps simultaneously, without the sequential restrictions of recurrent architectures. In the context of covariance prediction, Transformers handle sequences of Cholesky decompositions and auxiliary market features as input tokens. At each attention head, self-attention mechanisms dynamically weigh historical observations based on their relevance to predicting future covariance structures. This enables the model to identify subtle regime shifts, periods of heightened volatility clustering, or specific market shocks that significantly influence covariance dynamics. Given their inherent parallelization capabilities, Transformers efficiently process lengthy sequences, capturing both short- and long-term temporal dependencies across multiple assets simultaneously. To mitigate overfitting risks typical in financial contexts with limited historical data, we apply strong regularization techniques, including dropout, attention dropout, and careful tuning of network hyperparameters through rigorous validation.

\item \textbf{Autoformer:} The Autoformer model, introduced by \citet{Wu2021}, is specifically engineered to address the challenges of long-term financial time series forecasting. By integrating auto-correlation mechanisms and series decomposition into the standard Transformer architecture, Autoformer explicitly models intrinsic temporal patterns such as trends, seasonality, and cyclical behaviors. For covariance forecasting, historical sequences of Cholesky decomposition matrices ($L\_{t-k},\dots,L\_{t}$) serve as input sequences. Autoformer initially decomposes these sequences into their underlying trend and seasonal components, effectively isolating stable long-term patterns from transient fluctuations. Subsequently, the model applies auto-correlation within the attention mechanism to identify dominant periodicities and temporal structures, enabling accurate extrapolation beyond immediate historical windows. The encoder-decoder architecture of Autoformer systematically learns representations of historical covariance dynamics (encoder) and then predicts future Cholesky decomposition matrices (decoder). The auto-correlation-enhanced attention mechanism significantly improves the model’s ability to generalize covariance patterns, particularly during turbulent market regimes commonly encountered in cryptocurrency markets. Figure~\ref{fig:autoformer\_arch\_cov} provides a visual representation of the Autoformer architecture adapted specifically to covariance forecasting tasks.

\end{itemize}

\begin{figure}[!ht]

\centering

\includegraphics[width=0.9\textwidth]{./figures/autoformer\_architecture.png}

\caption{Detailed architecture of the Autoformer model adapted for covariance prediction tasks. Input sequences of historical Cholesky decomposition matrices undergo trend-seasonality decomposition. The auto-correlation mechanism within the encoder-decoder captures essential periodicities and temporal dependencies, forecasting subsequent Cholesky decomposition matrices robustly.}

\label{fig:autoformer\_arch\_cov}

\end{figure}

Model training involves minimizing the mean squared error (MSE) between the predicted and realized Cholesky decomposition matrices:

\[

L\_{MSE} = \frac{1}{T}\sum\_{t=1}^{T}\|\hat{L}\_{t+1}-L\_{t+1}\|^{2}\_{F},

\]

where $\|\cdot\|\_{F}$ denotes the Frobenius norm, ensuring that forecasts closely replicate true realized covariance structures. We also introduce regularization terms in the loss function to encourage temporal smoothness in predictions, mitigating erratic shifts in predicted covariance elements from one period to the next. Specifically, we penalize large deviations between successive predicted Cholesky matrices, indirectly controlling transaction-induced portfolio turnover when these covariance matrices subsequently inform portfolio optimization decisions.

To ensure robustness and generalizability, we implement rigorous hyperparameter tuning procedures, selecting optimal model structures (number of layers, hidden units, attention heads, dropout rates) based on validation performance. Training also employs early stopping criteria to guard against overfitting, ensuring the learned covariance forecasting policy remains stable and reliable out-of-sample.

By directly modeling Cholesky decompositions of realized covariance matrices, these supervised learning methods potentially uncover sophisticated temporal and cross-sectional interactions between assets. They thus offer a complementary, data-driven alternative to traditional econometric approaches, and serve as a robust foundation for subsequent portfolio optimization strategies.

Having introduced two distinct forecasting methodologies within Approach 1—econometric machine learning (Method A: XGBoost + ADCC) and supervised deep learning (Method B: LSTM, Transformer, Autoformer)—we now move to Approach 2. Approach 2 diverges fundamentally by bypassing explicit volatility and covariance forecasts, instead directly modeling optimal portfolio allocations from historical data, leveraging the supervised learning architectures previously detailed.