Further details on GTN-VF implementations*

*for paper: Multivariate Realized Volatility Forecasting with Graph Neural Network

Anonymous author(s)

A List of node features

We introduce the features we use in our experiments. Most of the features are from the literature (Kercheval and Zhang, 2015; Bissoondoyal-Bheenick et al., 2019; Mäkinen et al., 2019), we also add other intuitive features in our experiments.

In each bucket, we have $\Delta T'$ lines as training data. We first calculate an indicator for each line, we then aggregate these indicators in the same bucket with an aggregation function (aggregator). In such way, we have one value per indicator per aggregator as one feature for the bucket. The full list of indicators and aggregators are listed in Table 1.

For some important indicators, we also calculate their progressive features. It means that instead of applying an aggregator on all the lines, we apply it on the lines between 0 and $\Delta T'/6$, $\Delta T'/3$, $\Delta T'/2$, $2\Delta T'/3$, $5\Delta T'/6$, $\Delta T'$. In such way, we have 6 features per indicator per aggregator for a progressive feature. This is shown in the column Progressive in Table 1.

We define our aggregation functions as follows. We use a_i to denote the i-th line in the bucket and N represents the total number of lines.

- Gini coefficient (Gini, 1921)

$$\frac{\sum_i^N \sum_j^N |a_i - a_j|}{2N^2 \overline{a}}$$

- percentage difference

$$\frac{\sum_i^N \mathbf{1}_{a_i \neq a_{i-1}}}{N}$$

- realized volatility (Eq. (2))

$$\sqrt{\frac{1}{N}\sum_i^N a_i^2}$$

- percentage greater than mean

$$\frac{\sum_i^N \mathbf{1}_{a_i > \overline{a}}}{N}$$

- percentage greater than zero

$$\frac{\sum_i^N \mathbf{1}_{a_i > 0}}{N}$$

- median deviation

$$median(|a_i - \overline{a}|)$$

- energy

$$\frac{1}{N} \sum_i^N a_i^2$$

- InterQuartile Range (IQR)

$$Q_{75}(a) - Q_{25}(a)$$

  where $Q_i(a)$ denotes the i-th percentile value for the series a.

Table 1: The list of features built from LOB data.

Type	Notation	Description	Aggregators	Progressive	Count
Quote	$\frac{P_b^1 V_a^1 + P_a^1 V_b^1}{V_a^1 + V_b^1}$	WAP	mean, std, gini mean of first 100, mean of last 100	N	5
	P_a^1	Ask Price	% difference	N	1
	P_b^1	Bid Price	% difference	N	1
	$\frac{P_a^1 - P_b^1}{P_a^1 + P_b^1}$	Price Relative Spread	mean, std, gini	N	3
	$WAP - P_b^1$	WAP bid difference	mean, std, gini	N	3
	$log(\frac{WAP_i}{WAP_{i-1}})$	Return	realized volatility	Y	6
	$log(\frac{WAP_i}{WAP_{i-1}})^2$	Squared Return	std, gini	N	2
	$\frac{V_a^1 - V_b^1}{V_a^1 + V_b^1}$	Size Relative Spread	mean, std, gini	N	3
	V_a^1	Ask Size	% difference	N	1
	V_b^1	Bid Size	% difference	N	1
	$V_a^1 / \overline{V_a^1}$	Normalized Ask Size	mean, std, gini	N	3
	$V_a^1 + V_b^1$	Total Size	sum, max	N	2
	$\lvert V_a^1 - V_b^1 \rvert$	Size Imbalance	sum, max	N	2
Trade	P_t	Price	% greater than mean, % less than mean median diviation, energy, IQR	N	5
	$log(\frac{P_{t,i}}{P_{t,i-1}})$	Return	realized volatility	Y	6
			% greater than 0, % less than 0	N	2
	$log(\frac{P_{t,i}}{P_{t,i-1}})^2$	Squared Return	std, gini	N	2
	V_t	Size	sum	Y	6
			max, median diviation, energy, IQR	N	4
	t	Seconds	count	Y	6
	N_t	Order Count	sum	Y	6
			max	N	1
	$P_t \times V_t$	Amount	sum, max	N	2
	Total				73

B Graph Building Pseudocode

To help better understand our graph building process, we give the pseudocode we used in each type of graph. Temporal feature correlation relationship is built with Algorithm 1, cross-sectional feature correlation relationship is built with Algorithm 2, cross-sectional activity sector relationship is built with Algorithm 3 and cross-sectional supply chain relationship is built with Algorithm 4.

Algorithm 1 Graph building algorithm for temporal feature correlation relationship

Input:

Selected features: I

Selected features for all stocks and all times: $\{f^i_{s,t} \mid \forall s \in [1, ..., n], \forall t \in [1, ..., m], \forall i \in I\}$

Number of pairs selected for each t: K

Output:

Temporal feature correlation among the buckets: $\mathcal{R}_t$

for $i \in I$ **do**
  for $t = 1, ..., m$ **do**
    $Q^i_t \leftarrow ([f^i_{1,t}, ..., f^i_{n,t}])^\top$
  end for
end for
$\mathcal{R}_t \leftarrow \{\}$ ▷ Initialize an empty list for all relations
for $i \in I$ **do**
  for $t_0 = 1, ..., m$ **do**
    for $t = 1, ..., t_0$ **do**
      $L_{t_0,t} \leftarrow RMSPE(Q^i_{t_0}, Q^i_t)$
    end for
    $\mathcal{G}_{t_0} \leftarrow K$ pairs of (t_0, t) with the smallest $L_{t_0,t}$
    $\mathcal{R}_{t_0} \leftarrow \{\}$
    for $s = 1, ..., n$ **do**
      $\mathcal{R}_{t_0} \leftarrow \mathcal{R}_{t_0} \cup \{(node_{s,t_0}, node_{s,t})\}$
    end for
    $\mathcal{R}_t \leftarrow \mathcal{R}_t \cup \mathcal{R}_{t_0}$
  end for
end for

Algorithm 2 Graph building algorithm for cross-sectional feature correlation relationship

Input:

Selected features: I

Selected features for all stocks and all times: $\{f_{s,t}^i \mid \forall s \in [1, ..., n], \forall t \in [1, ..., m], \forall i \in I\}$

Number of pairs selected for each s: K'

Output:

Cross-sectional feature correlation among the buckets: $\mathcal{R}_s$

for $i \in I$ **do**
  for $s = 1, ..., n$ **do**
    $Q_s'^i \leftarrow ([f_{s,1}^i, ..., f_{s,m}^i])^\top$
  end for
end for
$\mathcal{R}_s \leftarrow \{\}$ ▷ Initialize an empty list for all relations
for $i \in I$ **do**
  for $s_0 = 1, ..., n$ **do**
    for $s = 1, ..., s_0$ **do**
      $L_{s_0,s} \leftarrow RMSPE(Q_{s_0}^i, Q_s^i)$
    end for
    $\mathcal{G}_{s_0} \leftarrow K'$ pairs of (s_0, s) with the smallest $L_{s_0,s}$
    $\mathcal{R}_{s_0} \leftarrow \{\}$
    for $t = 1, ..., m$ **do**
      $\mathcal{R}_{s_0} \leftarrow \mathcal{R}_{s_0} \cup \{(node_{s_0,t}, node_{s,t})\}$
    end for
    $\mathcal{R}_s \leftarrow \mathcal{R}_s \cup \mathcal{R}_{s_0}$
  end for
end for

Algorithm 3 Graph building algorithm for cross-sectional activity sector relationship

Input:
Stock sectors: $\mathcal{S} = \{\mathcal{S}_1, ..., \mathcal{S}_{N_s}\}$
Output:
Sector relationship among the buckets: $\mathcal{R}_{sector}$

$\mathcal{R}_{sector} \leftarrow \{\}$ ▷ Initialize an empty list for all relations
for $i = 1, ..., N_s$ **do**
  for $s_j \in \mathcal{S}_i$ **do**
    for $s_k \in \mathcal{S}_i$ **do**
      if $s_j \neq s_k$ **then**
        for $t = 1, ..., m$ **do**
          $\mathcal{R}_{sector} \leftarrow \mathcal{R}_{sector} \cup \{(node_{s_j,t}, node_{s_k,t})\}$
        end for
      end if
    end for
  end for
end for

Algorithm 4 Graph building algorithm for cross-sectional supply chain relationship

Input:
Supply chain relations among stocks: $\mathcal{G}_{sc} = \{(s_i, s_j) \mid s_i \text{ and } s_j \text{ have supplier-customer relationship }\}$
Output:
Supply chain relationship among the buckets: $\mathcal{R}_{sc}$

$\mathcal{R}_{sc} \leftarrow \{\}$ ▷ Initialize an empty list for all relations
for $(s_i, s_j) \in \mathcal{R}_{sc}$ **do**
  for $t = 1, ..., m$ **do**
    $\mathcal{R}_{sc} \leftarrow \mathcal{R}_{sc} \cup \{(node_{s_i,t}, node_{s_j,t})\}$
  end for
end for

C Details on Experiment Setup

C.1 Model Details

In all baseline models, except for Naïve guess and HAR-RV that do not need features, the input features ($F_{s,t}$) are the same as GTN-VF, including 73 numerical features and 1 categorical feature. The categorical feature embedding is of size 32 whenever applicable. All loss functions are set to RMSPE. The hyperparameters are chosen based on validation set performance and the results presented are based on the best set of hyperparameters.

C.1.1 HAR-RV

The model is written as

$$\widehat{RV_{s,t+1d,\Delta T}} = c + \beta^{(d)} RV_{s,t}^{(d)} + \beta^{(w)} RV_{s,t}^{(w)} + \beta^{(m)} RV_{s,t}^{(m)} \tag{1}$$

where $RV_{s,t}^{(d)}$, $RV_{s,t}^{(w)}$ and $RV_{s,t}^{(m)}$ are respectively the daily, weekly and monthly average realized volatilities before t. $\beta^{(d)}$, $\beta^{(w)}$, $\beta^{(m)}$ and c are coefficients determined by linear regression. The weekly and monthly average realized volatilities are calculated with $RV_{s,t}^{(w)} = \frac{1}{5}(RV_{s,t} + ... + RV_{s,t-4d})$ and $RV_{s,t}^{(w)} = \frac{1}{21}(RV_{s,t} + ... + RV_{s,t-20d})$.

In our implementation, we calibrate one HAR-RV model for each sampling time since we find a better result than mixing all the sampling times and calibrate only one model. Therefore, we have 6 different HAR-RV models for 10:00, 11:00, 12:00, 13:00, 14:00 and 15:00.

C.1.2 LightGBM

We use the LightGBM package[1] to implement this baseline model. We use gradient boosting decision tree (GBDT) algorithm and set its learning to 0.1. The model is trained for a maximum of 1,000 iterations and the training process is stopped earlier if there is no improvement on the validation set for 50 iterations.

C.1.3 MLP

There are three hidden layers in this fully-connected neural network. Their sizes are 128, 64 and 32 respectively. The learning rate of Adam optimizer is set to 0.01, and the batch size is set to 2048. The model is trained for a maximum of 200 iterations and the training process is stopped earlier if there is no improvement on the validation set for 20 iterations.

It is worth noting that we normalize our features to values between -1 and 1 before training. This is to avoid overflow in the computation process.

C.1.4 TabNet

We set the width of both prediction layer and attention embedding to 16. All other settings, including learning rate, batch size, training epochs, early stopping and feature normalization are the same as the MLP model.

C.1.5 GTN-VF and its variants

In our experiments, the GTN-VF and its variants share the same set of hyperparameter and model dimension. The model dimension is already introduced in Section 5.3.

During training, we set the batch size to 2048 and the initial learning rate to 0.001. The model is trained for a maximum of 100 iterations and the training

[1] https://github.com/microsoft/LightGBM

process is stopped earlier if there is no improvement on the validation set for 20 iterations. We also reduce the learning rate by half if there is no improvement for 5 epochs.

In the neighbor sampling process, we sample all the connected neighbors without a maximum limit.

C.2 Hardware

Except for Naïve Guess, HAR-RV and LightGBM which can be quickly trained without GPU, we ran the experiments on a single machine with one NVIDIA Tesla V100 GPU (16GB RAM and 32GB/s bandwidth), 8 cores of Intel Xeon CPU (Broadwell E5-2686 v4) and 61GB of RAM. In average, a GTN-VF with all four relationships takes one hour to train. For comparison, the training time for MLP is 20 minutes and 3 hours for TabNet.

D Further Analysis on Sector Relationship

As introduced in Section 5.2, there are four granularities in the GICS sector data. We run different experiments to evaluate the performance of each type of sector relationship. The result is shown in Table 2.

Table 2: Test RMSPE with different granularities of GICS sector. The result for Sector is not available as the number of edges is too big to fit in the memory.

Granularity	Number of edges	RMSPE
Sector	178.7M	N.A.
IndustryGroup	83.39M	0.2578
Industry	47.86M	0.2422
SubIndustry	19.88M	0.2441

We observe that Industry shows the best performance with a modest number of edges. If we add more edges, such as IndustryGroup, the RMSPE decreases since the relations among stocks are less meaningful. For the Sector granularity, we are not even able to obtain a result since the number of edges exceeds the memory limit. Hence, in our final model combining multiple sources of relations, we choose Industry as our sector relationship.

References

Bissoondoyal-Bheenick, E., Brooks, R., and Do, H. X. (2019). Asymmetric relationship between order imbalance and realized volatility: Evidence from the australian market. *International Review of Economics & Finance*, 62:309–320.

Gini, C. (1921). Measurement of inequality of incomes. *The economic journal*, 31(121):124–126.

Kercheval, A. N. and Zhang, Y. (2015). Modelling high-frequency limit order book dynamics with support vector machines. *Quantitative Finance*, 15(8):1315–1329.

Mäkinen, Y., Kanniainen, J., Gabbouj, M., and Iosifidis, A. (2019). Forecasting jump arrivals in stock prices: new attention-based network architecture using limit order book data. *Quantitative Finance*, 19(12):2033–2050.