Conditional Forecasting of Margin Calls Using Dynamic Graph Neural Networks

Gabriele Visentin (Joint work with: Matteo Citterio, Marco D'Errico)

July 10, 2026
Swissquote Workshop on Graph Learning in Financial Networks
EPFL

Plan of the talk

1. Motivation
2. Dynamic Graph Neural Networks (DGNNs)
3. Problem formulation
4. Proposed method
5. Results

Motivation - Policy

- Despite regulatory efforts[1], **derivatives markets** have seen recurrent episodes of **instability** in recent years (e.g. the Nasdaq Clearing member failure, the COVID-19 "dash for cash" in March 2020, nickel market volatility in March 2022, the European energy crisis, and the UK gilt market turmoil in October 2022).

[1] See the Financial Stability Report report to G20 Leaders (2014) and Implementation and Effects of the G20 Financial Regulatory Reforms (2014).
[2] See the BCBS-CPMIIOSCO Review of margining practices (2022).
[3] See Liquidity risks arising from margin calls, ESRB (2020).

Motivation - Policy

- Despite regulatory efforts[1], **derivatives markets** have seen recurrent episodes of **instability** in recent years (e.g. the Nasdaq Clearing member failure, the COVID-19 “dash for cash” in March 2020, nickel market volatility in March 2022, the European energy crisis, and the UK gilt market turmoil in October 2022).
- Margin requirements mitigate counterparty credit risk, but also introduce liquidity pressures and procyclicality[2].

[1] See the Financial Stability Report report to G20 Leaders (2014) and Implementation and Effects of the G20 Financial Regulatory Reforms (2014).
[2] See the BCBS-CPMIIOSCO Review of margining practices (2022).
[3] See Liquidity risks arising from margin calls, ESRB (2020).

Motivation - Policy

- Despite regulatory efforts[1], **derivatives markets** have seen recurrent episodes of **instability** in recent years (e.g. the Nasdaq Clearing member failure, the COVID-19 "dash for cash" in March 2020, nickel market volatility in March 2022, the European energy crisis, and the UK gilt market turmoil in October 2022).
- Margin requirements mitigate counterparty credit risk, but also introduce liquidity pressures and procyclicality[2].
- **Recent advances in systemic risk data collection** have enabled real-time mapping of margin calls[3], but current approaches remain largely *ex post*, underscoring the need for a **network forecasting** framework.

[1] See the Financial Stability Report report to G20 Leaders (2014) and Implementation and Effects of the G20 Financial Regulatory Reforms (2014).
[2] See the BCBS-CPMIIOSCO Review of margining practices (2022).
[3] See Liquidity risks arising from margin calls, ESRB (2020).

Motivation - Policy

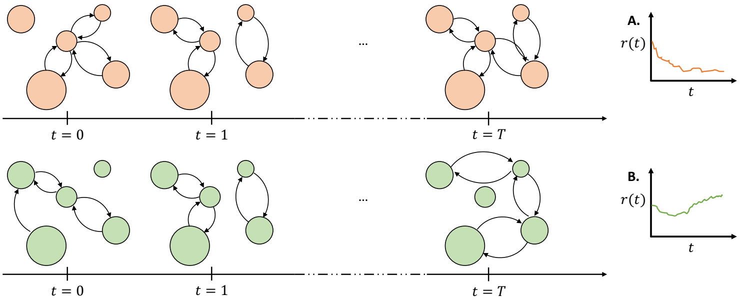
- Despite regulatory efforts[1], **derivatives markets** have seen recurrent episodes of **instability** in recent years (e.g. the Nasdaq Clearing member failure, the COVID-19 "dash for cash" in March 2020, nickel market volatility in March 2022, the European energy crisis, and the UK gilt market turmoil in October 2022).
- Margin requirements mitigate counterparty credit risk, but also introduce liquidity pressures and procyclicality[2].
- **Recent advances in systemic risk data collection** have enabled real-time mapping of margin calls[3], but current approaches remain largely *ex post*, underscoring the need for a **network forecasting** framework.

Research question: Can we forecast margin calls in stressed derivatives markets, by leveraging available transaction-level historical data?

[1] See the Financial Stability Report report to G20 Leaders (2014) and Implementation and Effects of the G20 Financial Regulatory Reforms (2014).
[2] See the BCBS-CPMIIOSCO Review of margining practices (2022).
[3] See Liquidity risks arising from margin calls, ESRB (2020).

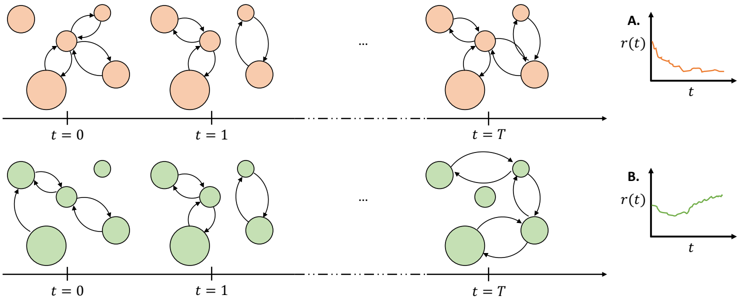
Motivation - Methodology

- Recent systemic risk events in the derivatives market are due to **feedback loops** arising from the **dynamic** and **adaptive** behavior of the market participants.



Motivation - Methodology

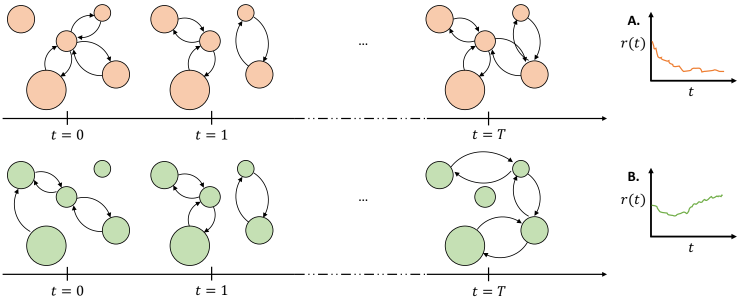
- Recent systemic risk events in the derivatives market are due to **feedback loops** arising from the **dynamic** and **adaptive** behavior of the market participants.



- But research on insolvency and illiquidity cascades has focused so far only on *static* financial networks.

Motivation - Methodology

- Recent systemic risk events in the derivatives market are due to **feedback loops** arising from the **dynamic** and **adaptive** behavior of the market participants.



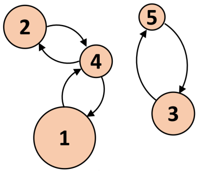
- But research on insolvency and illiquidity cascades has focused so far only on *static* financial networks.
- **Research question:** Can we develop a **Dynamic Graph Neural Network (DGNN)** method to **solve** m**-steps ahead forecasting problems** in **dynamic** financial networks,
conditionally on macro-economic stress test scenarios?

Dynamic Graph Neural Networks (DGNNs)

Graph Neural Networks (GNN)

Let $\mathcal{G} = (V, X, E)$ be a graph:

- $V = \{1, \dots, N\}$ set of vertices,
- $X \in \mathbb{R}^{N \times C}$ node features,
- $E = \{(i, j)$ if there is an edge from i to $j\}$.

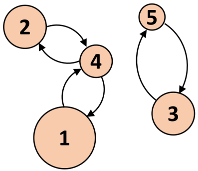


Goal: solve a non-linear **node regression** task, $Y = f(X) \in \mathbb{R}^N$, by exploiting the graph structure (because we know that the nodes' behavior can impact their neighbors).

Graph Neural Networks (GNN)

Let $\mathcal{G} = (V, X, E)$ be a graph:

- $V = \{1, \ldots, N\}$ set of vertices,
- $X \in \mathbb{R}^{N \times C}$ node features,
- $E = \{(i, j)$ if there is an edge from i to $j\}$.



Goal: solve a non-linear **node regression** task, $Y = f(X) \in \mathbb{R}^N$, by exploiting the graph structure (because we know that the nodes' behavior can impact their neighbors).

A **Graph Convolutional Network (GCN)** by [Kipf and Welling, 2017] consists of ℓ layers that recursively aggregate information from all nodes that are at most ℓ edges away ($\ell \leq$ diameter(G)):

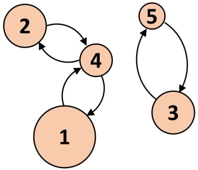
$$\begin{cases} h^{(1)} &= \sigma(A\, X\, W^{(1)}) \\ h^{(2)} &= \sigma(A\, h^{(1)}\, W^{(2)}) \\ \vdots & \\ h^{(\ell)} &= \sigma(A\, h^{(\ell-1)}\, W^{(\ell)}) =: \mathrm{GCN}^{(\ell)}(A, X) \in \mathbb{R}^{N \times P} \end{cases}$$

where $A \in \mathbb{R}^{N \times N}$ is the adjacency matrix and σ is an element-wise non-linearity.

Graph Neural Networks (GNN)

Let $\mathcal{G} = (V, X, E)$ be a graph:

- $V = \{1, \ldots, N\}$ set of vertices,
- $X \in \mathbb{R}^{N \times C}$ node features,
- $E = \{(i, j)$ if there is an edge from i to $j\}$.



Goal: solve a non-linear **node regression** task, $Y = f(X) \in \mathbb{R}^N$, by exploiting the graph structure (because we know that the nodes' behavior can impact their neighbors).

A **Graph Convolutional Network (GCN)** by [Kipf and Welling, 2017] consists of ℓ layers that recursively aggregate information from all nodes that are at most ℓ edges away ($\ell \leq$ diameter(G)):

$$\begin{cases} h^{(1)} &= \sigma(A\, X\, W^{(1)}) \\ h^{(2)} &= \sigma(A\, h^{(1)}\, W^{(2)}) \\ \vdots & \\ h^{(\ell)} &= \sigma(A\, h^{(\ell-1)}\, W^{(\ell)}) =: \mathrm{GCN}^{(\ell)}(A, X) \in \mathbb{R}^{N \times P} \end{cases}$$

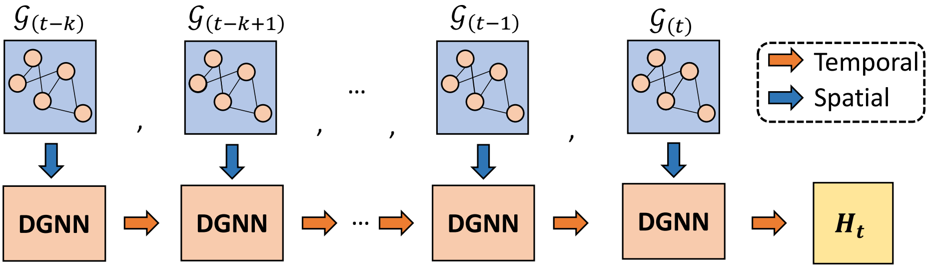
where $A \in \mathbb{R}^{N \times N}$ is the adjacency matrix and σ is an element-wise non-linearity.

The **hidden variables** $h^{(\ell)}$ extracted by the model can be used solve the regression task, i.e. $Y \approx \tilde{f}(h^{(\ell)})$, where $\tilde{f}$ may even be linear.

Dynamic Graph Neural Networks (DGNNs)

For dynamic graphs, $\mathcal{G}_t = (V_t, X_t, E_t)$, we must use a DGNN, for instance the **GC-LSTM** model[4]:

- extracts h_t from the graph $\mathcal{G}_t$ at time t using a Graph Convolutional Network (GCN) (spatial dimension),
- propagates h_t from time t to time $t+1$ using a Recurrent Neural Network (RNN) (temporal dimension).

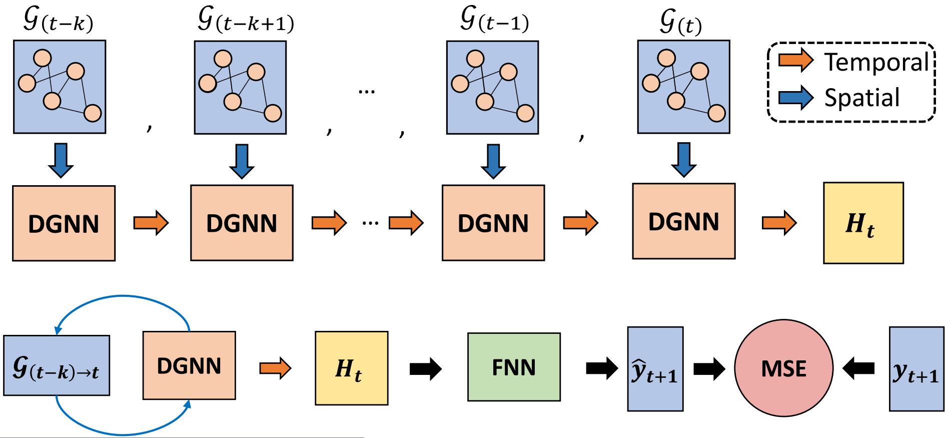


[4] See [Chen et al., 2022].

Dynamic Graph Neural Networks (DGNNs)

For dynamic graphs, $\mathcal{G}_t = (V_t, X_t, E_t)$, we must use a DGNN, for instance the **GC-LSTM** model[4]:

- extracts h_t from the graph $\mathcal{G}_t$ at time t using a Graph Convolutional Network (GCN) (spatial dimension),
- propagates h_t from time t to time $t+1$ using a Recurrent Neural Network (RNN) (temporal dimension).



[4] See [Chen et al., 2022].

Problem formulation

Problem formulation

Problem: we want to forecast margin calls in a network of heterogeneous market participants that dynamically exchange Interest Rate Swaps[5] (IRS).

[5]We focus on Overnight Indexed Swaps (IOS) on a reference overnight rate (e.g. €STR, SOFR or SONIA).

[6]Satisfying the usual conditions [Karatzas and Shreve, 2014, Definition 2.25], for $\mathbb{Q}$ a given martingale measure.

Problem formulation

Problem: we want to forecast margin calls in a network of heterogeneous market participants that dynamically exchange Interest Rate Swaps[5] (IRS).

- We develop an analytical dynamic network model (to simulate data and compute the exact ground truth), but **our DGNN method can be applied to any data generating process**.

[5]We focus on Overnight Indexed Swaps (IOS) on a reference overnight rate (e.g. €STR, SOFR or SONIA).
[6]Satisfying the usual conditions [Karatzas and Shreve, 2014, Definition 2.25], for $\mathbb{Q}$ a given martingale measure.

Problem formulation

Problem: we want to forecast margin calls in a network of heterogeneous market participants that dynamically exchange Interest Rate Swaps[5] (IRS).

- We develop an analytical dynamic network model (to simulate data and compute the exact ground truth), but **our DGNN method can be applied to any data generating process**.

Let $(\Omega, \mathcal{A}, (\mathcal{A}_t)_{t\geq 0}, \mathbb{Q})$ be a filtered probability space[6] and let $r(t)$ be a **reference rate** with Cox-Ingersoll-Ross (CIR) dynamics:

$$\begin{cases} dr(t) = \kappa(\theta - r(t))dt + \sigma\sqrt{r(t)}dW_t, \\ r(0) \in \mathbb{R}_+. \end{cases}$$

We denote the **bond price** by $p(t, T)$ and the corresponding **daily term structure** by $R(t, T)$.

[5]We focus on Overnight Indexed Swaps (IOS) on a reference overnight rate (e.g. €STR, SOFR or SONIA).

[6]Satisfying the usual conditions [Karatzas and Shreve, 2014, Definition 2.25], for $\mathbb{Q}$ a given martingale measure.

Problem formulation

Problem: we want to forecast margin calls in a network of heterogeneous market participants that dynamically exchange Interest Rate Swaps[5] (IRS).

- We develop an analytical dynamic network model (to simulate data and compute the exact ground truth), but **our DGNN method can be applied to any data generating process**.

Let $(\Omega, \mathcal{A}, (\mathcal{A}_t)_{t\geq 0}, \mathbb{Q})$ be a filtered probability space[6] and let $r(t)$ be a **reference rate** with Cox-Ingersoll-Ross (CIR) dynamics:

$$\begin{cases} dr(t) = \kappa(\theta - r(t))dt + \sigma\sqrt{r(t)}dW_t, \\ r(0) \in \mathbb{R}_+. \end{cases}$$

We denote the **bond price** by $p(t,T)$ and the corresponding **daily term structure** by $R(t,T)$.

Then the **value of an Interest Rate Swap** on the reference rate $r(t)$ at time t is:

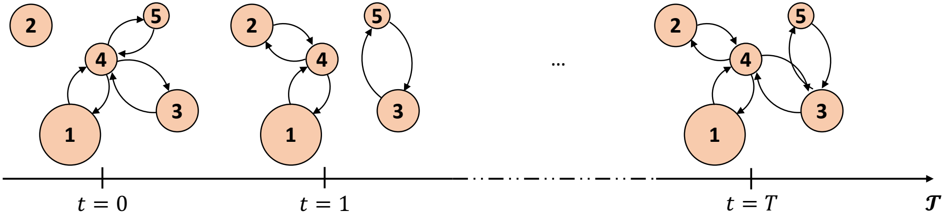
$$V(t) = \delta N \left(p(t,T)\left(1 + K(T - t_0)\right) - \prod_{t_i \in (t_0, t] \cap \mathcal{T}} \left(1 + r(t_i)(t_i - t_{i-1})\right) \right),$$

where N is the notional, t_0 the start date, T the maturity, K the fixed interest rate, and $\delta \in \{-1, +1\}$.

[5]We focus on Overnight Indexed Swaps (IOS) on a reference overnight rate (e.g. €STR, SOFR or SONIA).

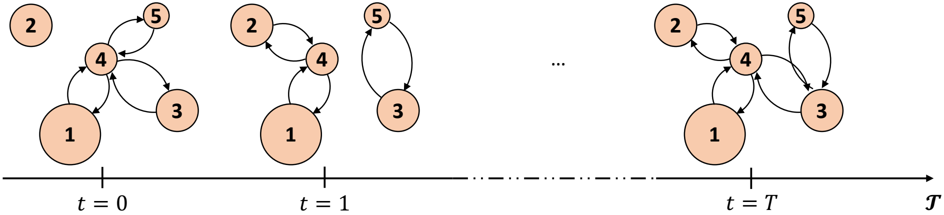
[6]Satisfying the usual conditions [Karatzas and Shreve, 2014, Definition 2.25], for $\mathbb{Q}$ a given martingale measure.

Problem formulation - Network dynamics



The financial network $\mathcal{G} = \{G_t = (V, X_t, E_t)\}_{t \in \mathcal{T}}$ evolves in time depending on the reference rate $r(t)$.

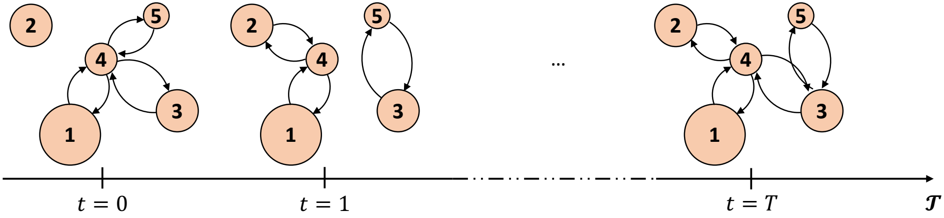
Problem formulation - Network dynamics



The financial network $\mathcal{G} = \{G_t = (V, X_t, E_t)\}_{t \in \mathcal{T}}$ evolves in time depending on the reference rate $r(t)$.

- $X_t \in \mathbb{R}^{N \times C}$ contains **market participant features** at time t (e.g. balance sheet data), in our case $x_i(t) = \{\text{hub, private}\}$,

Problem formulation - Network dynamics



The financial network $\mathcal{G} = \{G_t = (V, X_t, E_t)\}_{t\in\mathcal{T}}$ evolves in time depending on the reference rate $r(t)$.

- $X_t \in \mathbb{R}^{N\times C}$ contains **market participant features** at time t (e.g. balance sheet data), in our case $x_i(t) = \{\text{hub, private}\}$,
- **edges** $(i, j) \in E_t$ represent IRS contracts and follow a point process $\left(t_k^{(i,j)}, N_k^{(i,j)}, K_k^{(i,j)}, T_k^{(i,j)}, \delta_k^{(i,j)}\right)_{k\in\mathbb{N}}$ with intensity conditional on $r(t)$ and X_t:

$$\lambda_t^{(i,j)}(r(t), x_i(t), x_j(t)) = \gamma \cdot \exp\{\eta + (\theta + \beta \cdot g(x_i(t), x_j(t)))\, r(t)\}, \quad g(x_i(t), x_j(t)) = \begin{cases} +1 & \text{(hub-private)} \\ 1/3 & \text{(hub-hub)} \\ -1 & \text{(private-private)} \end{cases}$$

Problem formulation

Forecasting target:

At time t_l, for each counterparty i, we want to forecast its total m-**steps ahead variation margin**:

$$M^{(i)}(t_{l+m}) := \sum_{j \in \mathcal{N}_i(t)} \sum_{k \in \mathbb{N}} V_k^{(i,j)}(t_{l+m}) - (1 + r(t_{l+m})\Delta t) V_k^{(i,j)}(t_{l+m-1}),$$

<u>conditionally on a future trajectory of the reference rate $r(t)$</u>.

Problem formulation

Forecasting target:

At time t_l, for each counterparty i, we want to forecast its total m-**steps ahead variation margin**:

$$M^{(i)}(t_{l+m}) := \sum_{j \in \mathcal{N}_i(t)} \sum_{k \in \mathbb{N}} V_k^{(i,j)}(t_{l+m}) - (1 + r(t_{l+m})\Delta t) V_k^{(i,j)}(t_{l+m-1}),$$

conditionally on a future trajectory of the reference rate $r(t)$.

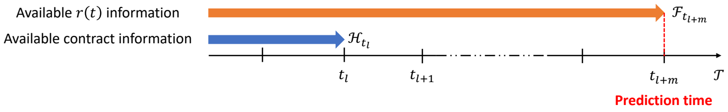
A successful forecasting method must take into account:

- the valuation of **existing contracts** (by time t_l):

$$\mathcal{H}_t = \sigma\left((t_k, N_k, K_k, T_k, \delta_k),\ k \in \mathbb{N} \text{ such that } t_k \leq t\right)$$

- the valuation of **contracts in the future** (i.e. appearing between t_l and t_{l+m}):

$$\mathcal{F}_t = \sigma\left(r(s), s \leq t\right)$$



Problem formulation

Forecasting target:

At time t_l, for each counterparty i, we want to forecast its total m-**steps ahead variation margin**:

$$M^{(i)}(t_{l+m}) := \sum_{j \in \mathcal{N}_i(t)} \sum_{k \in \mathbb{N}} V_k^{(i,j)}(t_{l+m}) - (1 + r(t_{l+m})\Delta t) V_k^{(i,j)}(t_{l+m-1}),$$

conditionally on a future trajectory of the reference rate $r(t)$.

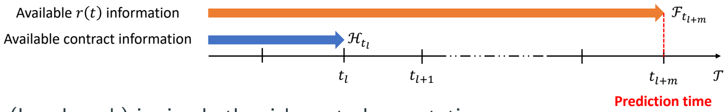
A successful forecasting method must take into account:

- the valuation of **existing contracts** (by time t_l):

$$\mathcal{H}_t = \sigma\left((t_k, N_k, K_k, T_k, \delta_k),\ k \in \mathbb{N} \text{ such that } t_k \leq t\right)$$

- the valuation of **contracts in the future** (i.e. appearing between t_l and t_{l+m}):

$$\mathcal{F}_t = \sigma\left(r(s), s \leq t\right)$$



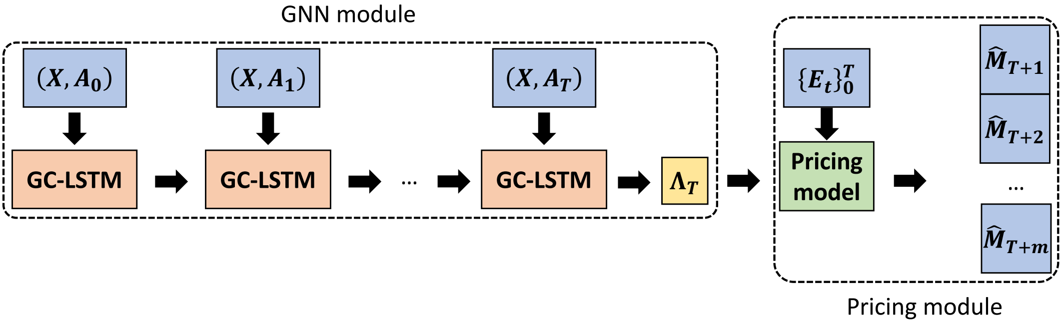
The **ground truth** (benchmark) is simply the risk-neutral expectation:

$$\prod_{t_i \in (t_l, t_{l+m}] \cap \mathcal{T}} (1 + r(t_i)\Delta t)^{-1} \mathbb{E}^{\mathbb{Q}}\left[M(t_{l+m}) \mid \mathcal{F}_{t_{l+m}} \vee \mathcal{H}_{t_l}\right].$$

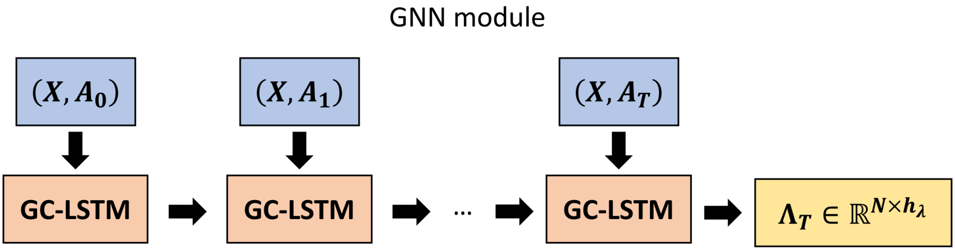
Proposed method

We propose a novel DGNN architecture to solve this conditional forecasting problem, which consists of:

- a **GNN module**,
- a **Pricing module**.

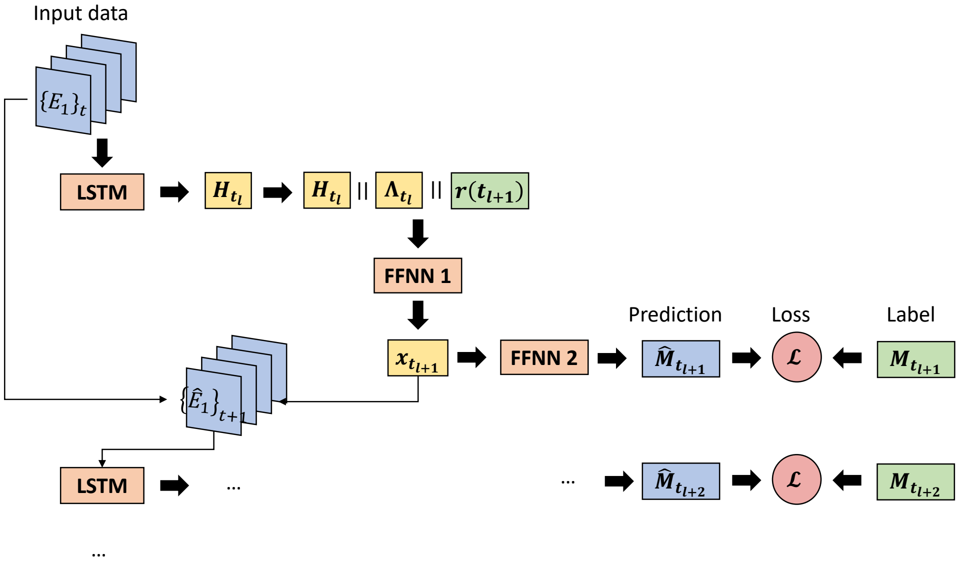


Proposed method - GNN module

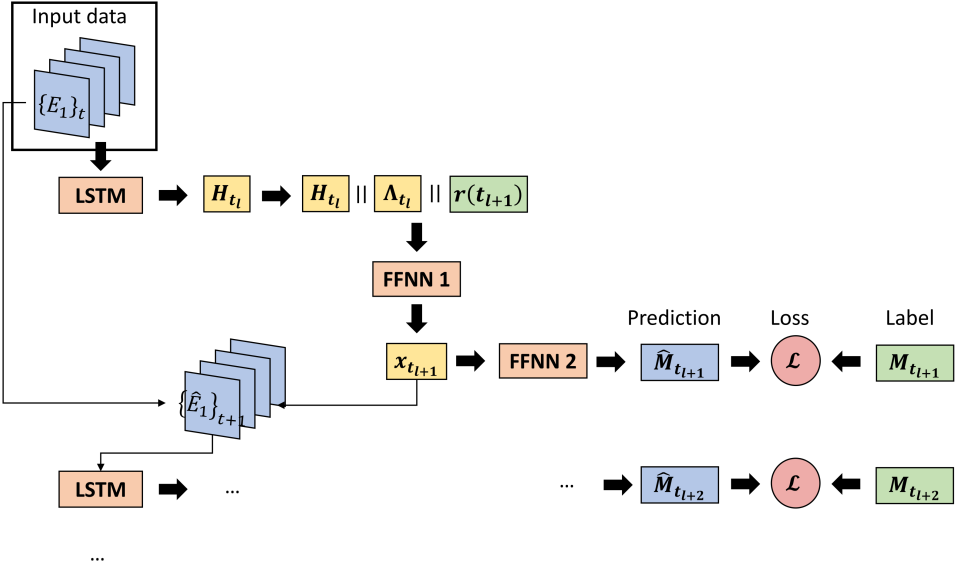


- The GNN module is a standard GC-LSTM to capture the network dynamics.
- The output Λ_T is a hidden state representation of the intensity process, to model contracts in the future.

Proposed method - Pricing module

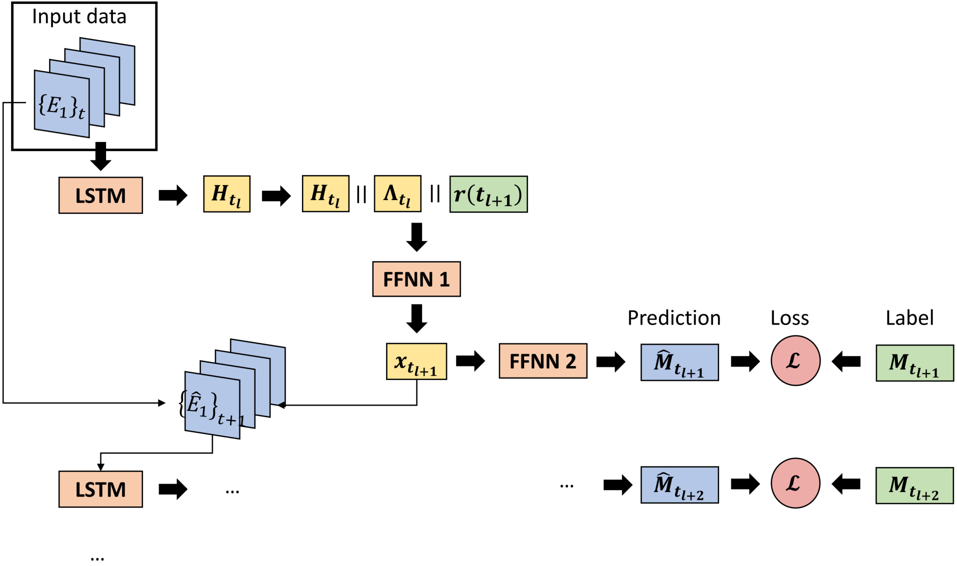


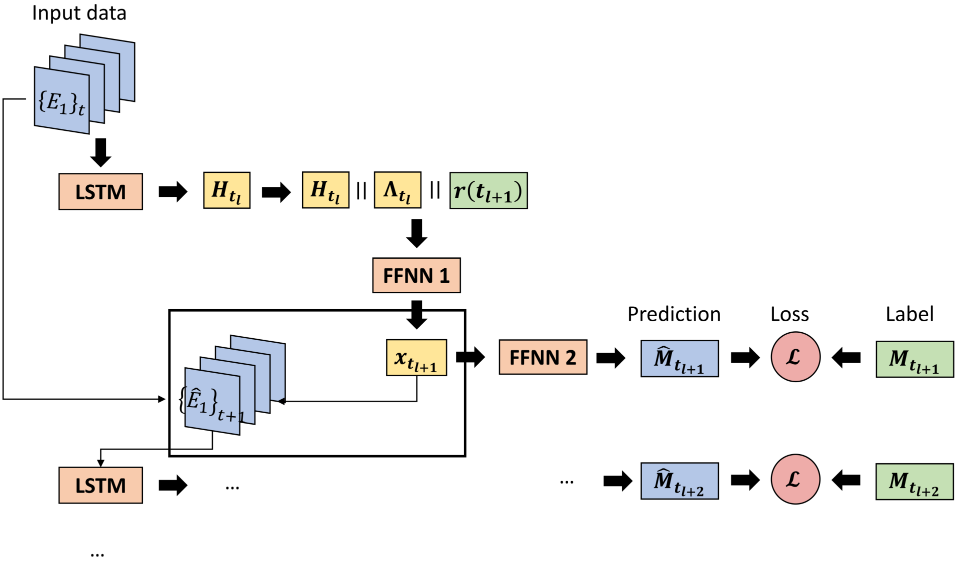
Pricing module

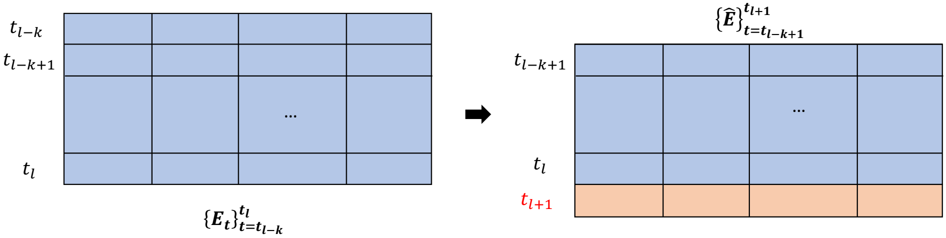


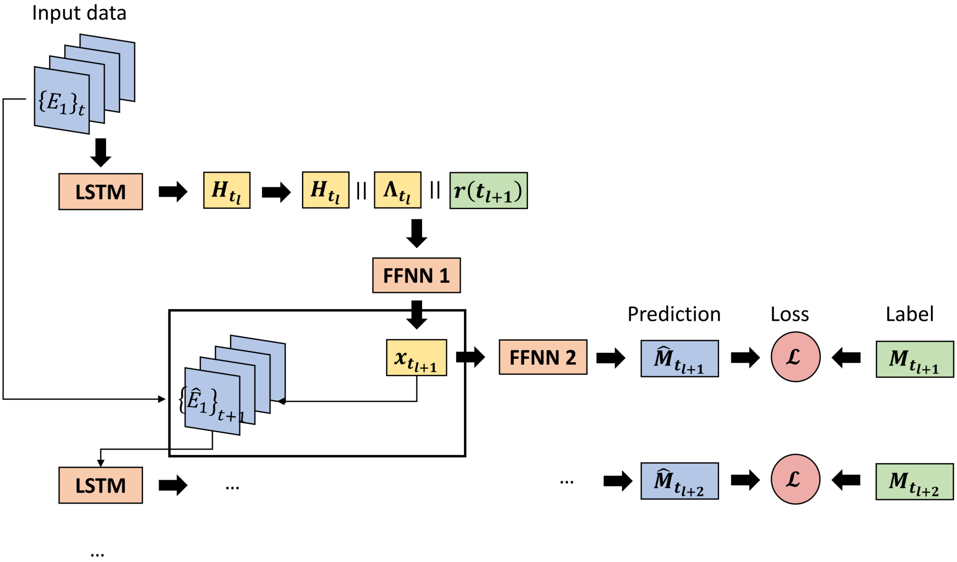
Proposed method - Pricing module - Contract matrix

$$\boxed{\{\boldsymbol{E}_t\}_{t=t_{l-k}}^{t_l}} = \underbrace{\left.\begin{array}{|c|c|c|c|}
\hline
\left[\frac{(\mathrm{T}-\mathrm{t})}{365}, p(t_0,T), p(t,T), B(t_0), B(t), \delta\right] & \dots & & \\
\hline
\left[\frac{(\mathrm{T}-\mathrm{t}-1)}{365}, p(t_0,T), p(t+1,T), B(t_0), B(t+1), \delta\right] & \dots & & \dots \\
\hline
\dots & & & \\
\hline
\left[\frac{(\mathrm{T}-\mathrm{t}-\mathrm{k})}{365}, p(t_0,T), p(t+k,T), B(t_0), B(t+k), \delta\right] & & \dots & \\
\hline
\end{array}\right\} k}_{\max_{t\in\mathcal{T}}\left(N_{active}(t)\right)\cdot 6}$$

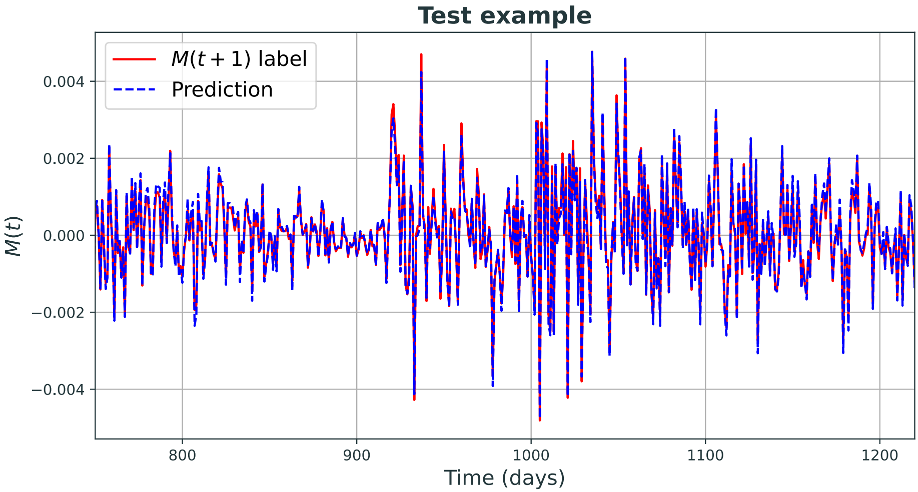

Input data
$\{E_1\}_t$
LSTM
H_{t_l}
$H_{t_l} \,||\, \Lambda_{t_l} \,||\, r(t_{l+1})$
FFNN 1
$x_{t_{l+1}}$
FFNN 2
Prediction
$\widehat{M}_{t_{l+1}}$
Loss
$\mathcal{L}$
Label
$M_{t_{l+1}}$
$\{\hat{E}_1\}_{t+1}$
LSTM
...
...
$\widehat{M}_{t_{l+2}}$
$\mathcal{L}$
$M_{t_{l+2}}$
...


Input data
$\{E_1\}_t$
LSTM
H_{t_l}
H_{t_l} || Λ_{t_l} || $r(t_{l+1})$
FFNN 1
$x_{t_{l+1}}$
$\{\hat{E}_1\}_{t+1}$
FFNN 2
Prediction
$\hat{M}_{t_{l+1}}$
Loss
$\mathcal{L}$
Label
$M_{t_{l+1}}$
LSTM
...
...
$\hat{M}_{t_{l+2}}$
$\mathcal{L}$
$M_{t_{l+2}}$
...


t_{l-k}
t_{l-k+1}
...
t_l
$\{\boldsymbol{E}_t\}_{t=t_{l-k}}^{t_l}$
$\{\widehat{\boldsymbol{E}}\}_{t=t_{l-k+1}}^{t_{l+1}}$
t_{l-k+1}
...
t_l
t_{l+1}

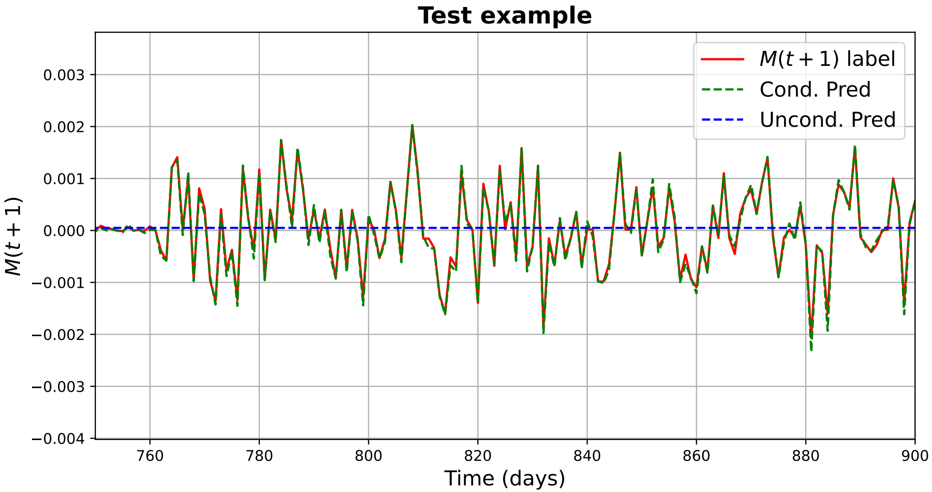

Input data
$\{E_1\}_t$
LSTM
H_{t_l}
$H_{t_l} \,||\, \Lambda_{t_l} \,||\, r(t_{l+1})$
FFNN 1
$x_{t_{l+1}}$
$\{\hat{E}_1\}_{t+1}$
FFNN 2
Prediction
Loss
Label
$\widehat{M}_{t_{l+1}}$
$\mathcal{L}$
$M_{t_{l+1}}$
LSTM
...
...
$\widehat{M}_{t_{l+2}}$
$\mathcal{L}$
$M_{t_{l+2}}$
...

Results



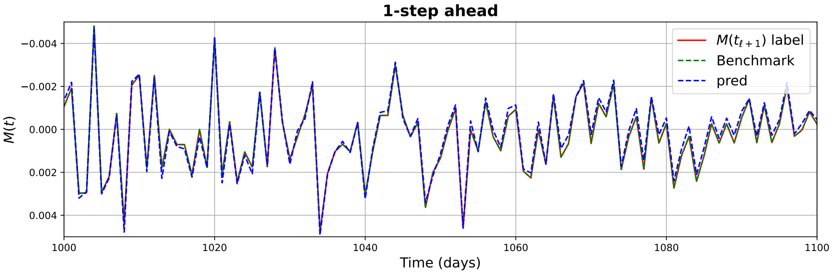
- One-step ahead stressed margin call forecast for one node in the network.

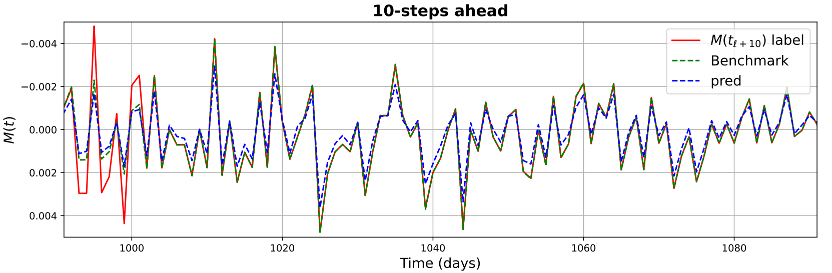
Results



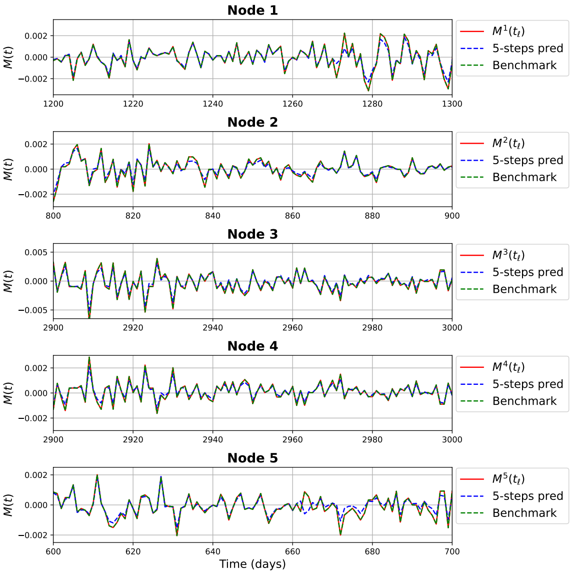
- The conditioning future interest rate trajectory (stress test scenario) is fundamental.

Results


1-step ahead
−0.004
−0.002
0.000
0.002
0.004
M(t)
1000
1020
1040
1060
1080
1100
Time (days)
$M(t_{\ell+1})$ label
Benchmark
pred


10-steps ahead
−0.004
−0.002
0.000
0.002
0.004
M(t)
1000
1020
1040
1060
1080
Time (days)
$M(t_{\ell+10})$ label
Benchmark
pred

Results



Node 1
M(t)
0.002
0.000
−0.002
1200
1220
1240
1260
1280
1300
$M^1(t_\ell)$
5-steps pred
Benchmark
Node 2
M(t)
0.002
0.000
−0.002
800
820
840
860
880
900
$M^2(t_\ell)$
5-steps pred
Benchmark
Node 3
M(t)
0.005
0.000
−0.005
2900
2920
2940
2960
2980
3000
$M^3(t_\ell)$
5-steps pred
Benchmark
Node 4
M(t)
0.002
0.000
−0.002
2900
2920
2940
2960
2980
3000
$M^4(t_\ell)$
5-steps pred
Benchmark
Node 5
M(t)
0.002
0.000
−0.002
600
620
640
660
680
700
Time (days)
$M^5(t_\ell)$
5-steps pred
Benchmark

Conclusions

Methodology:

- We introduced a **Dynamic Graph Neural Network (DGNN) model** for solving an m-steps ahead **margin call forecasting** problem in a fully dynamic network, **conditionally on stress test scenarios** (such as future trajectories of a reference rate).
- The model is modular, can be easily adapted to various derivatives markets, and produces **accurate forecasts up to a 21-day horizon**.

Conclusions

Methodology:

- We introduced a **Dynamic Graph Neural Network (DGNN) model** for solving an m-steps ahead **margin call forecasting** problem in a fully dynamic network, **conditionally on stress test scenarios** (such as future trajectories of a reference rate).
- The model is modular, can be easily adapted to various derivatives markets, and produces **accurate forecasts up to a 21-day horizon**.

Policy:

- This is a proof-of-concept work to show that **derivative transactions data could successfully be used to train machine learning models** that take into account **how the market reacts to shocks in reference interest rates** and other macro-economic variables.
- It could provide regulators and policymakers with a crucial tool to limit **pro-cyclicality** and complement **liquidity stress-testing** and **systemic risk monitoring**.

Conclusions

Methodology:

- We introduced a **Dynamic Graph Neural Network (DGNN) model** for solving an m-steps ahead **margin call forecasting** problem in a fully dynamic network, **conditionally on stress test scenarios** (such as future trajectories of a reference rate).
- The model is modular, can be easily adapted to various derivatives markets, and produces **accurate forecasts up to a 21-day horizon**.

Policy:

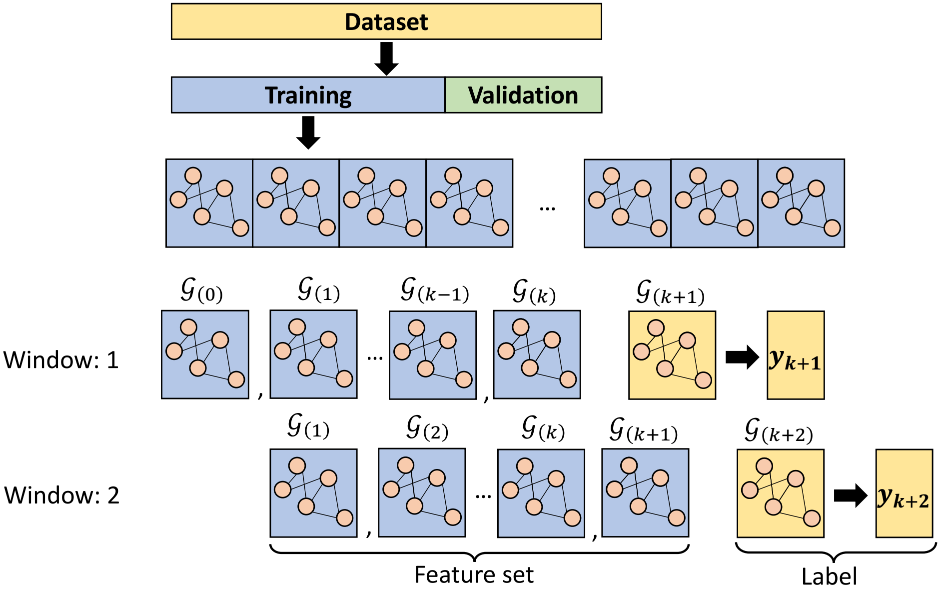
- This is a proof-of-concept work to show that **derivative transactions data could successfully be used to train machine learning models** that take into account **how the market reacts to shocks in reference interest rates** and other macro-economic variables.
- It could provide regulators and policymakers with a crucial tool to limit **pro-cyclicality** and complement **liquidity stress-testing** and **systemic risk monitoring**.

Thank you for your attention!

Chen, J., Wang, X., and Xu, X. (2022).
GC-LSTM: Graph convolution embedded LSTM for dynamic network link prediction.
Applied Intelligence, pages 1–16.

Karatzas, I. and Shreve, S. (2014).
Brownian Motion and Stochastic Calculus.
Springer.

Kipf, T. N. and Welling, M. (2017).
Semi-supervised classification with graph convolutional networks.
5th International Conference on Learning Representations.


Dataset
Training
Validation
...
$\mathcal{G}_{(0)}$
$\mathcal{G}_{(1)}$
$\mathcal{G}_{(k-1)}$
$\mathcal{G}_{(k)}$
$\mathcal{G}_{(k+1)}$
Window: 1
y_{k+1}
$\mathcal{G}_{(1)}$
$\mathcal{G}_{(2)}$
$\mathcal{G}_{(k)}$
$\mathcal{G}_{(k+1)}$
$\mathcal{G}_{(k+2)}$
Window: 2
y_{k+2}
Feature set
Label