Received: 28 October 2022 | Revised: 31 January 2023 | Accepted: 5 February 2023

DOI: 10.1002/for.2955

RESEARCH ARTICLE

WILEY

Bayesian bilinear neural network for predicting the mid-price dynamics in limit-order book markets

Martin Magris | **Mostafa Shabani** | **Alexandros Iosifidis**

Department of Electrical and Computer Engineering, Aarhus University, Aarhus, Denmark

Correspondence
Martin Magris, Aarhus University, Department of Electrical and Computer Engineering, Finlandsgade 22, Aarhus 8200, Denmark.
Email: magris@ece.au.dk

Funding information
Marie Skłodowska-Curie project BNNmetrics, Grant/Award Number: 890690; Independent Research Fund Denmark project DISPA, Grant/Award Number: 9041-00004

Abstract

The prediction of financial markets is a challenging yet important task. In modern electronically driven markets, traditional time-series econometric methods often appear incapable of capturing the true complexity of the multilevel interactions driving the price dynamics. While recent research has established the effectiveness of traditional machine learning (ML) models in financial applications, their intrinsic inability to deal with uncertainties, which is a great concern in econometrics research and real business applications, constitutes a major drawback. Bayesian methods naturally appear as a suitable remedy conveying the predictive ability of ML methods with the probabilistically oriented practice of econometric research. By adopting a state-of-the-art second-order optimization algorithm, we train a Bayesian bilinear neural network with temporal attention, suitable for the challenging time-series task of predicting mid-price movements in ultra-high-frequency limit-order book markets. We thoroughly compare our Bayesian model with traditional ML alternatives by addressing the use of predictive distributions to analyze errors and uncertainties associated with the estimated parameters and model forecasts. Our results underline the feasibility of the Bayesian deep-learning approach and its predictive and decisional advantages in complex econometric tasks, prompting future research in this direction.

KEYWORDS

Bayesian neural networks, bilinear neural network, financial time-series classification, limit-order book

1 | INTRODUCTION

Bayesian inference is known to be a difficult task outside a relatively small class of well-studied models, generally involving conjugate priors for the likelihood. The analytical Bayesian treatment of general, even small-dimensional, problems is widely unfeasible. The increased computational capacity available these days, as much as the availability of powerful algorithms such as Markov chain Monte Carlo (MCMC) or Metropolis–Hastings, opened the possibility for a simulation-based approach to Bayesian inference. However, Bayesian methods in typical large-scale complex machine learning (ML) problems have long been impractical. Though ML

generally operates under a frequentist perspective, the first steps into a probabilistic approach to deep learning (DL) are relatively recent; see, for example, Gal and Ghahramani (2016) and Murphy (2012) and references therein. Only recently, we are witnessing a growing interest in Bayesian DL, boosted by its demand across multiple disciplines. Indeed, even in a simulation-aided setting, Bayesian inference on the potentially thousands of parameters over highly nonlinear models like neural networks (NNs) is certainly not a simple task.

Yet the interest in probabilistic modeling and Bayesian methods in other disciplines has a much longer history. Especially in econometrics and finance, the probabilistic dimension is an innate and essential element in modeling. Indeed econometric research is at the cross-edge between applied statistics, probability theory, stochastics, and the study of economic phenomena (Ragnar, 1933). As such, the econometric practice is that of developing well-reasoned and economically motivated, essential, mostly parametric, probabilistic models that are thoroughly tested, validated, and back-tested following the principles of statistical inference. For example, the concepts of significance testing, confidence intervals, asymptotic analysis, and stationarity are typical in the econometric literature. On the other hand, such an approach in DL is currently inapplicable.

At the same time, researchers in economics and practitioners in finance acknowledge the flexibility, scalability, and gains in predictive tasks that ML can bring when applied to economic problems; for example, Gal and Ghahramani (2016) and Mullainathan and Spiess (2017). Especially in modern, electronically driven financial markets, operating at ultra-high frequencies and generating massive complex and multidimensional datasets underlying the complex dynamics of market variables arising from the interactions of multiple players and forces at different levels, ML methods have gained much attention; see, for example, Varian (2014). Business and financial applications are part of those high-risk domains where quantifying the uncertainty underlying models' estimates and predictions is of utmost importance (Salinas et al., 2020). A probabilistic dimension reflecting uncertainties related to model estimation and perhaps accounting for the typical elements of business activity that are difficult to predict (Makridakis et al., 2009) would be beneficial.

The recent advances in Bayesian DL have the potential of bringing this element into play, narrowing the gap between the highly probabilistic yet parsimonious modeling of the econometric practice and the flexible nonlinear and nonparametric ML rationale. Bayesian inference for NNs has recently been shown to be challenging yet feasible (Blundell et al., 2015; Kingma & Welling, 2014; Osawa et al., 2019). Bayesian neural networks (BNNs) are engaged with the typical elements of Bayesian inference, in particular with a trainable distribution over its parameters and a consequent predictive distribution that enables classical statistical tools, econometric methods, and relevant risk-related and uncertainly related analyses, for example, based on predictive distributions (Geweke & Amisano, 2010). Nevertheless, much research in this direction is still needed.

This paper aims to introduce the use of BNNs in economic problems in light of the above discussion, boosting further research and interest in this research direction. We propose a Bayesian version of the temporal attention-augmented bilinear network (TABL) as a lightweight DL model for a financial times-series classification task. We propose a first Bayesian DL econometric application in the challenging task of predicting mid-price movements in limit-order book (LOB) markets. Our results explore the feasibility of such an approach, compare its forecasting performance against non-Bayesian specifications based on different optimization algorithms, and address the advantages of adopting BNNs in financial applications.

2 | LITERATURE REVIEW

NNs have been successfully applied in several ML problems, such as image classification (Krizhevsky et al., 2012; Simonyan & Zisserman, 2015), computer vision (Girshick et al., 2014; Ren et al., 2015), natural language processing (Collobert & Weston, 2008; Goldberg, 2017), or speech recognition (Dahl et al., 2012; Mohamed et al., 2012). Despite their undeniable use and performance in delivering leading results on different predictive tasks, the decisions are achieved in a rather uninterpretable manner.

NNs correspond to statistical black-box models that achieve feasible point estimate predictions by adapting their high-dimensional parameters on a try-and-error basis. Based on the nature of the problem, the user defines a cost function and a network architecture that allows to approximate complex nonlinear functions to tackle a prediction or classification task. By allowing for a sufficiently large number of trainable parameters, under mild assumptions, NNs can approximate any arbitrary function (Cybenko, 1989; Hanin, 2019; Lu et al., 2017). It is with little surprise that NNs found significant use in financial and econometrics applications, where complex and interacting latent structures in the data drive the behavior of different economic variables. A review of different NN applications in finance is provided in McNelis (2005), an early discussion on econometric applications can be found in Kuan and White (1994) and

within time-series analysis in Hewamalage et al. (2021), Qi and Zhang (2008), and Teräsvirta et al. (2005). Cenesizoglu et al. (2022) analyzed the relationship between LOB variables and mid-price movements showing that it is possible to obtain economical gain from these variables and the mid-price return. Further, their causality analysis supports the use of lagged LOB variables for forecasting purposes. The design of a set of features, extending the LOB feature set in the high-frequency forecasting application of Kercheval and Zhang (2015), can be found in Ntakaris et al. (2019), while Ntakaris et al. (2018, 2019) tackle the mid-price movement prediction with the rich LOB data under different ML perspectives, including NNs. For a similar prediction task, Dixon (2018) addressed the use of recurrent NN. The use of long–short-term memory (LSTM) networks and convolutional neural networks (CNNs) is discussed in Tsantekidis et al. (2017), Tsantekidis et al. (2020), Passalis et al. (2019), and Zhang et al. (2019) and the use of Neural Bag of Features in Passalis et al. (2017, 2018). Taking advantage of the spatial structure in the LOB, Sirignano (2019) provides an extensive analysis of over 500 stocks for prediction price movements, while Tran et al. (2017) encode the LOB data as two-order tensors. An attention mechanism capable of exploiting and retaining the temporal mode of the order flow is introduced in Tran et al. (2019) and extended to accommodate multiple attentions in Shabani et al. (2022).

By fine-tuning the network and increasing the number of parameters, one possibly achieves functions with higher complexity and improved forecasting ability. The lack of interpretation and the impossibility of condensing the decision process to a simple decision rule, along with overfitting issues and their native nonprobabilistic setup, create challenges in the use of NNs in high-risk domains and for all those applications where uncertainties in predictions are of relevance (Goan & Fookes, 2020). Moreover, the lack of a well-defined building protocol (e.g., the absence of an Akaike information criterion-like statistic for features' relevance determination) makes their adoption by experts and practitioners from such domains difficult (Caruana et al., 2015; Holzinger et al., 2017, 2019; Vu et al., 2018).

A Bayesian perspective on NNs provides a natural way to reason around uncertainties. At the same time, it provides tools for model regularization and offers insights into how decisions are made. Indeed, the Bayesian paradigm offers a perspective on NNs that can address many of the issues currently faced by NNs. Recent research investigated how Bayesian principles can adapt to large NNs. To this end, a learnable distribution is placed over the parameters, resulting in BNNs. A survey on early developments in BNNs can be found in Mackay (1995), and recent introductions to BNNs are those of, for example, Goan and Fookes (2020), Jospin et al. (2020), and Lampinen and Vehtari (2001). For a specialized survey on algorithms for training BNNs, see Magris and Iosifidis (2022). In BNNs, parameters are treated as random variables, and the learning focuses on the distribution of these parameters conditional on the observed training data sample. In the learning phase, the latent distribution of the parameters is inferred based on the current knowledge and the observed data by use of the Bayes theorem resulting in a distribution of the model parameters conditional on the data, the posterior distribution. Further details on BNNs and their training appear in Section 3.

Financial applications involving BNNs are somewhat limited. An application for automatic relevance determination in option pricing is that of Mbuvha et al. (2019). A recent example of stock-price prediction is found in Chandra and He (2021), where exploitative MCMC-based learning is used to forecast daily closing prices of four stocks, showing that in terms of RMSE performance metric, their BNN outperforms non-Bayesian counterparts. A forecasting study based on electricity prices is provided in Ghayekhloo et al. (2019) and Vahidinasab and Jadid (2008), while Bitcoin data are used in Jang and Lee (2018). Sign changes in returns have been analyzed under a multilayer perceptron (MLP) BNN in Skabar (2009). This study uses low-frequency daily closing prices and lagged moving averages as features, showing a slight 52% accuracy over a random classifier and no gains with respect to a standard MLP. There are, however, no applications involving tick-by-tick data generated from typical modern financial markets running over the LOB systems. Nevertheless, the bridging potential that BNNs could provide between the fields of econometrics and ML has not been recognized.

3 | METHODS

3.1 | BNNs

A BNN is any stochastic artificial neural network (ANN) trained using Bayesian inference. ANNs aim at approximating arbitrary functions $\boldsymbol{y} = NN_\theta(\boldsymbol{x})$, whose parameters are denoted by θ. Over a training dataset $\mathcal{D}$, the standard estimation approach is to determine a minimal-cost point estimate $\hat{\theta}$ using backpropagation. In BNNs, parameters are treated as latent random variables, and the goal is to learn the distribution of the parameters conditional on $\mathcal{D}$. The first step is that of defining the joint distribution of the data and the parameters $p(\theta, \mathcal{D}) = p(\mathcal{D}|\theta)p(\theta)$, which depends on our prior belief over the latent variables $p(\theta)$ and the chosen form of

likelihood $p(\mathcal{D}|\theta)$. Under independence between the model parameters and the inputs, the Bayesian posterior is written as $p(\theta|\mathcal{D}) = p(\mathcal{D},\theta)p(\theta)/p(\mathcal{D})$. Computing the weight-independent term known as marginal likelihood (or evidence) is perhaps one of the most difficult tasks in Bayesian inference: The prior-to-posterior update is usually intractable. From the posterior distribution, the model uncertainty is quantified as the marginal probability distribution of the output $\boldsymbol{y}_i$ for a certain input $\boldsymbol{x}_i$, through the predictive distribution

$$p(\boldsymbol{y}_i|\boldsymbol{x}_i,\mathcal{D}) = \int p(\boldsymbol{y}_i|\boldsymbol{x}_i,\theta)p(\theta|\mathcal{D})d\theta. \tag{1}$$

When performing classification, the average model prediction approximates the relative probability of each class,

$$\hat{p}_{ic} \approx 1/N_s \sum\nolimits_{n=1}^{N_s} p(y_i = c|\boldsymbol{x}_i,\theta^{(n)}), \tag{2}$$

where $\theta^{(n)} \sim p(\theta|\mathcal{D})$. If the cost of giving a false positive is equal across all classes, the final classification is taken according to the most likely class, that is,

$$\hat{y}_i = \max_c \hat{p}_{ic}. \tag{3}$$

3.2 | TABL

The TABL architecture (Tran et al., 2019) is a lightweight DL model which has been shown to be particularly suited for multidimensional time-series forecasting. It augments the bilinear projection with an attention mechanism exploiting the temporal dimension across the features. This enables it to compare favorably with alternative architectures such as bilinear networks, CNNs, LSTM networks, and several other ML algorithms (Tran et al., 2019).

Figure 1 illustrates the architecture of the TABL layer. It maps a $D \times T$-dimensional input matrix $\mathbf{X}$ onto a $D' \times T'$-dimensional output $\mathbf{Y}$, where D and D' correspond to the number of features and T and T' correspond to the number of temporal instances. The network initially operates a projection of the temporal dimension of the input matrix to a $D' \times T$-dimensional feature space modeling the dependence on the first mode while preserving the temporal order of the features. It further learns the relative importance of the temporal instances with respect to each other, producing an attention mask where only the most relevant instances are preserved. A learnable scalar drives the mixture of the temporal and nontemporal features passed to a final mapping that returns the final representation adjusted for bias. This is achieved by

$$\overline{\mathbf{X}} = \mathbf{W}_1\mathbf{X}, \tag{4}$$

$$\mathbf{E} = \overline{\mathbf{X}}\mathbf{W}, \tag{5}$$

$$a_{ij} = \exp(e_{ij}) \Big/ \sum\nolimits_{k=1}^{T} \exp(e_{ik}), \tag{6}$$

$$\tilde{\mathbf{X}} = \lambda(\overline{\mathbf{X}} \odot \mathbf{A}) + (1-\lambda)\overline{\mathbf{X}}, \tag{7}$$

$$\mathbf{Y} = \phi(\tilde{\mathbf{X}}\mathbf{W}_2 + \mathbf{B}), \tag{8}$$

where a_{ij} and e_{ij} denote the element (i,j) of $\mathbf{A}$ and $\mathbf{E}$, respectively; $\phi(\cdot)$ is a predefined activation function; $\mathbf{W}_1 \in \mathbb{R}^{D\prime\times D}$, $\mathbf{W} \in \mathbb{R}^{T\times T}$, $\mathbf{W}_2 \in \mathbb{R}^{T\times T\prime}$, and $\mathbf{B} \in \mathbb{R}^{D\prime\times T\prime}$ are the parameters of the layer; and $0 \le \lambda \le 1$ is a learnable mixing coefficient which determines the importance of using the temporal attention in the mapping. Experiments in Shabani et al. (2022) further show that the inclusion of additional temporal attention heads opens toward richer structures in the temporal dependence across lagged features, relevant for forecasting purposes.

3.3 | Bayesian TABL

To formulate the Bayesian network formed by one TABL layer (B-TABL), we define the parameter vector $\boldsymbol{\theta}$ formed by the parameters of TABL, that is,

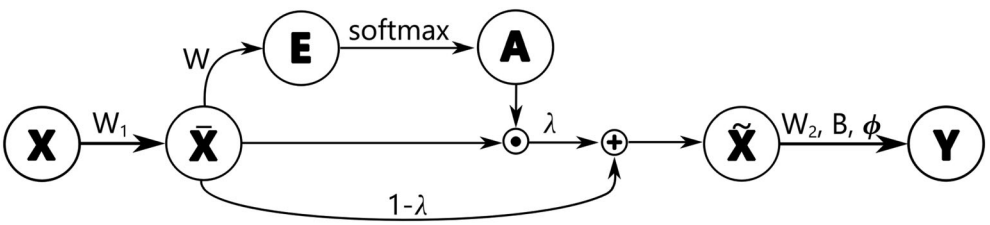

FIGURE 1 Illustration of the TABL architecture.

$\boldsymbol{\theta} = \{\mathbf{W}, \mathbf{W}_1, \mathbf{W}_2, \mathbf{B}, \lambda\}$. Variational inference (VI) is a well-established methodology for complex statistical inference circumventing the typical intractable integration problem arising in Bayesian inference by approximating the true posterior $p(\boldsymbol{\theta}|\mathcal{D})$ with a distribution $q(\boldsymbol{\theta})$ whose normalization constant is easier to compute (variational distribution). A review on VI from a statistician's perspective is that of Blei et al. (2017), from the ML perspective, that of Tran et al. (2021), while very recent applications in multidimensional econometric models are, for example, those of Gefang et al. (2023) and Gunawan et al. (2021). Fixed form variational Bayes assumes a fixed parametric form for the density in some class of distributions $\mathcal{Q}$, indexed by a variational parameter vector. A perspective on the problem with general nonconjugate likelihoods for priors in the exponential family can be found in Khan and Nielsen (2018). We chose both $p(\boldsymbol{\theta})$ and $q(\boldsymbol{\theta})$ to be Gaussian distributions with diagonal covariance matrices:

$$p(\boldsymbol{\theta}) = \mathcal{N}(\theta|\mathbf{0}, \boldsymbol{I}/\alpha), \qquad q(\boldsymbol{\theta}) = \mathcal{N}(\theta|\boldsymbol{\mu}, \mathrm{diag}(\boldsymbol{\sigma}^2)),$$

where $\alpha > 0$ is a known precision parameter and $\boldsymbol{\mu} \in \mathbb{R}^P$, $\boldsymbol{\sigma}^2 \in \mathbb{R}^P$. P is the number of the parameters in the network, that is, the number of parameters in $\boldsymbol{\theta}$.

$q(\boldsymbol{\theta})$ implies a factorization of the join in the product of the marginals, known as mean-field approximation. In VI, the variational parameters $(\boldsymbol{\mu}, \boldsymbol{\sigma}^2)$ are obtained by maximizing the following objective:

$$\mathcal{L}(\boldsymbol{\mu}, \boldsymbol{\sigma}^2) = \sum_{i=1}^{N} \mathbb{E}_q[\log p(\mathcal{D}|\boldsymbol{\theta})] + \mathbb{E}_q\left[\log \frac{p(\boldsymbol{\theta})}{q(\boldsymbol{\theta})}\right]. \tag{9}$$

Equation (9) can be maximized with the gradient-based optimization, that is, with the following update:

$$\boldsymbol{\mu}_{t+1} = \boldsymbol{\mu}_t + \rho_t \hat{\nabla}_{\boldsymbol{\mu}} \mathcal{L}_t \quad \text{and} \quad \boldsymbol{\sigma}_{t+1} = \boldsymbol{\sigma}_t + \delta_t \hat{\nabla}_{\boldsymbol{\sigma}} \mathcal{L}_t, \tag{10}$$

where t is the iteration index, $\hat{\nabla}_x \mathcal{L}_t$ denotes an unbiased estimate of the gradient of $\mathcal{L}$ at $(\boldsymbol{\mu}_t, \boldsymbol{\sigma}_t^2)$ with respect to x, and ρ_t, δ_t are adaptable learning rates. The natural-gradient VI method of Khan and Lin (2017) tackles the update (10) in terms of the natural parameter $\boldsymbol{\alpha}$ of $q(\boldsymbol{\theta})$, rather than its mean and covariance matrix, and scales the gradient of the corresponding SGD update for $\boldsymbol{\alpha}_t$ with the inverse of the Fisher information matrix (FIM) of $q(\boldsymbol{\theta})$. Khan and Lin (2017) show that the direct computation of the FIM can be avoided by computing *natural* gradients in the natural parameter space using the gradient with respect to the expectation parameters of the exponential-family posterior. For the Gaussian mean-field VI under consideration, this leads to the natural-gradient variational inference (NGVI) update:

$$\boldsymbol{\mu}_{t+1} = \boldsymbol{\mu}_t + \beta_t \boldsymbol{\sigma}_{t+1}^2 \odot \hat{\nabla}_{\boldsymbol{\mu}} \mathcal{L}_t, \tag{11}$$

$$\boldsymbol{\sigma}_{t+1}^{-2} = \boldsymbol{\sigma}_t^{-2} - 2\beta_t \hat{\nabla}_{\boldsymbol{\sigma}^2} \mathcal{L}_t, \tag{12}$$

with $\beta_t > 0$ being a scalar learning rate. By expressing (9) in terms of the standard MLE objective $f(\boldsymbol{\theta}) = -1/N \sum_{i=1}^{N} \log p(\mathcal{D}_i|\boldsymbol{\theta})$ and expressing the gradients of its expectation with respect to $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ in terms of the gradient $g(\boldsymbol{\theta})$ and Hessian $\mathbf{H}(\boldsymbol{\theta})$ of $f(\boldsymbol{\theta})$, the update results in

$$\begin{aligned} \boldsymbol{\mu}_{t+1} &= \boldsymbol{\mu}_t - \beta_t (g(\boldsymbol{\theta}_t) + \tilde{\alpha}\boldsymbol{\mu}_t)/(\mathbf{s}_{t+1} + \tilde{\alpha}), \\ \mathbf{s}_{t+1} &= (1-\beta_t)\mathbf{s}_t + \beta_t \mathrm{diag}(\mathbf{H}(\boldsymbol{\theta}_t)), \end{aligned} \tag{13}$$

where the division is element-wise, $\tilde{\alpha} = \alpha/N$ and $\boldsymbol{\theta}_t \sim \mathcal{N}(\boldsymbol{\theta}|\boldsymbol{\mu}_t, \mathrm{diag}(\boldsymbol{\sigma}_t^2))$, with $\boldsymbol{\sigma}_t^2 = [N(\mathbf{s}_t + \tilde{\alpha})]^{-1}$. The scaling vector $\mathbf{s}_t$ involves the gradient and the diagonal of the Hessian, which can be replaced by their stochastic estimates $\hat{g}(\boldsymbol{\theta})$ and $\hat{\nabla}^2_{\boldsymbol{\theta}\boldsymbol{\theta}} f(\boldsymbol{\theta})$. The former can be computed using backpropagation. Due to the general nonconvexity of f, the latter can be negative, which might lead to negative variances. Nonnegativity is granted by the following approximation:

$$\hat{\nabla}^2_{\theta_j \theta_j} f(\boldsymbol{\theta}) \approx \frac{1}{M} \sum_{i \in \mathcal{M}} \left[\hat{\nabla}_{\theta_j} f_i(\theta)\right]^2 := \hat{h}_j(\boldsymbol{\theta}), \tag{14}$$

with i denoting the ith data sample in the mini-batch $\mathcal{M}$ of size M and θ_j the jth element of $\boldsymbol{\theta}$. By writing for $\hat{\mathbf{h}}(\boldsymbol{\theta}_t)$ the vector of all $\hat{h}_j$, under this approximation, the update for $\mathbf{s}_t$ reads:

$$\mathbf{s}_{t+1} = (1-\beta_t)\mathbf{s}_t + \beta_t \hat{\mathbf{h}}(_t). \tag{15}$$

The algorithm involving updates (13) and (15) is referred to as the variational online Gauss–Newton (VOGN) (Khan & Nielsen, 2018). Opposed to SGD and related algorithms such as RMSprop, Adam, and AdaGrad, which use the gradient magnitude $\left[\frac{1}{M}\sum_{i \in \mathcal{M}} \hat{\nabla}_{\theta_j} f_i(\theta)\right]^2$ for approximating the jth entry of the diagonal Hessian, in (14), VOGN uses averages of the squared gradients, avoiding explicit constraints on $\boldsymbol{\sigma}^2$. As shown in Osawa et al. (2019), it leads to good empirical performance and practical feasibility of the updates (13) and (15) on large datasets compared with alternatives, for example, Bayes by Backprop (BBB) (Blundell et al., 2015).

Existing automatic-differentiation libraries can be used to retrieve the gradients; however, current codebases directly return sums of the gradients over mini-batches, whereas individual gradients are required in (14). Thus, via chain rule, we derive the individual gradients for a TABL layer and adapt current second-order optimization routines (Osawa, 2019) to accommodate them. Our B-TABL implementation adopts a log-softmax activation function at the last layer such that the output vector of the network interprets as logs of class probabilities, with a one-to-one mapping between classes and indexes of vectors' elements. That is, for an input time-series $\mathbf{X}_i$, the corresponding output of the network is a vector

$$\log \boldsymbol{p}_i = [\log p_{i1}, \log p_{i2}, ..., \log p_{iC}],$$

where $p_{ic} = \exp(\boldsymbol{l}_i[c]) / \sum_c \exp(\boldsymbol{l}_i[c])$, c being an index running along the number of network outputs (the number of classes C). $\boldsymbol{l}_i$ denotes the output of the last layer, corresponding to the input $\mathbf{X}_i$, passed to the softmax activation, and $\boldsymbol{l}_i[c]$ its cth element. The loss used is the negative log likelihood, that is, for a sample in class c the loss is computed as $-\log \boldsymbol{p}_i[c]$. We shall refer to

$\boldsymbol{p}_i$ as class probabilities or scores. Losses are averaged across samples for each mini-batch. For a trained model and for each input vector, predictions are provided by the class of the maximum log score (or maximum class probability), that is, $\max_c \log \boldsymbol{p}_i[c]$. We shall refer to this criterion as the classification rule. Therefore, due to posterior sampling, log scores are stochastic (as for the class probabilities and the index of their maximum element), leading to stochastic class labels.

4 | EXPERIMENTS

4.1 | High-frequency LOB data

Trading in modern financial markets is organized through an order-driven mechanism that collects and matches inflowing limit and market orders through a time-priority rule. Trades participate in the market by submitting orders or cancellations over previously submitted orders. Each message (order or cancellation) submitted to the exchange comes with an associated timestamp, price, and quantity (along with a unique identifier). By submitting a limit order, a trader expresses his/her willingness to buy or sell a certain amount of the security at a specified price, that is, the trader specifies the buy/sell price and the number (or fractions) of stocks he/she wants to trade. Limit orders are collected and stored in what is known as the LOB. At a time instance t, the cross-section of the LOB provides a snapshot of the number of outstanding limit orders, their prices, and quantities.

In particular, buy (sell) limit orders define the bid (ask) side of the book. The highest buy and lowest ask prices represent the best prices to sell or buy a certain amount of a security. These best prices are known respectively as bid and ask prices (p_t^B, p_t^A). Market orders are immediately executed on the bid or ask side at the current best price, leading to trades. Limit orders at the current bid/ask prices are filled according to a time-priority rule (first submitted, first traded). A market order decreases the quantity available at the best price, and if the market order quantity is equal to or greater than the outstanding quantity of the limit orders at the current best price, it reduces the total depth of the market (i.e., the number of different price levels on which the limit orders are arranged). As the limit orders on the top of the book are filled, the actual best price moves to that of the next LOB level until a new incoming limit order (on the same side of the book) refills the gap between the bid and ask prices, or a new market order erodes the top of the book causing a further update in the best bid or ask price. We refer to, for example, Ntakaris et al. (2018) for further details on the LOB mechanism.

It is clear that the order inflow (along with order cancellations) is governed by a highly stochastic mechanism that leads to a rich, multidimensional dataset consisting of order types, prices, and quantities, whose instances reflect the dynamics of the bid and ask prices as well as of deeper LOB levels. Although broad stylized facts on the LOB dynamics lead to some analytic tools for modeling the LOB, for example, Cont et al. (2010), tackling its dynamics is very challenging, and ML methods can provide a useful alternative for a number of forecasting goals. While the first level of the LOB has been commonly used in econometric research, Tran et al. (2021) showed that the information in multiple levels increases the performance of ML models. Both ML and econometrics research focused on the dynamics of the synthetic price measure across the two sides of the book known as mid-price: $p_t = \frac{1}{2}(p_t^A - p_t^B)$.

We focus on the task of forecasting mid-price changes at the future (tick-by-tick) updates of the LOB. This implies a complex classification problem over three classes: mid-price increases, mid-price decreases, or remains stationary. We used the publicly available FI-2010 dataset (Ntakaris et al., 2018), which collects the LOB states for five stocks traded at the NASDAQ Nordic Helsinki exchange from June 1 to June 14, 2010 (collecting approximately 4.5 million events across 10 trading days). At each epoch (i.e., LOB update), the data consists of 144-dimensional feature vectors. In total, there are

453,975 features extracted over nonoverlapping blocks of 10 events and normalized using the z score. The dataset provides labels corresponding to the direction of the price movement on five different horizons, corresponding to the price movements in the next 10, 20, 30, 50, and 100 events. In our experiments, we utilize the 10-event horizon and adopt the experimental setup of (Tsantekidis et al., 2017, 2020) where the last 3 days are taken as the test set (corresponding to 150,418 samples). For the first 7 days, the initial 75% of instances constitute the training set, and the last 15% the validation set.

4.2 | Experiment setting

As in Tran et al. (2019), we use the first 40 dimensions consisting of raw prices and quantities. For the BNN implementation, we employ the VOGN algorithm. Networks' weights are initialized under a multivariate Gaussian prior with parameters $\boldsymbol{\mu} = \boldsymbol{0}$ and $\boldsymbol{\Sigma} = \boldsymbol{I}$. The learning rate, momentum factor, and decay rate of the L2 norm regularization are, respectively, set to 0.01, 0.999, and 0.85. Its performance on the validation set is evaluated over 10 MC draws from the posterior at each epoch, while the predictive distribution for each input in the test set is approximated by the collection of $N_s = 50$ forecasts following N_s feed-forward passes for N_s independent samples from the variational posterior. Prior's means and variances, respectively set to one and zero, are initially warmed up following the method described in Osawa et al. (2019).

The Bayesian training of the network is evaluated with respect to two non-Bayesian alternatives: the ADAM (Kingma & Ba, 2015) optimizer and stochastic gradient descent (SGD). For SGD, the momentum is set to 0.99; for ADAM, the first and second moments are fixed to 0.9 and 0.999. For both algorithms, the initial learning rate is set to 0.01 and dynamically updated until the validation loss reaches a plateau. For all the optimizers, the training is set to 1000 epochs with a mini-batch of size 256. When training with ADAM, we also employ MC dropout (Gal & Ghahramani, 2016) in the testing phase. MC dropout is not a Bayesian method but has a connection with Bayesian theory and serves as an approach to predictive distribution approximation (Gal & Ghahramani, 2016). A random deletion of the NN connections allows for posterior sampling. Sampling from the approximate posterior enables MC integration of the likelihood, uncovering an approximation to the predictive distribution. By repeated forward passes for the same input sample, the randomized dropout yields samples from the predictive distribution. Gal and Ghahramani (2016) find that even a small number of forward passes can suffice. Similar to the B-TABL, we apply $N_s = 50$ and set the dropout rate to 10%. As in Osawa et al. (2019), we do not compare VOGN with BBB (Blundell et al., 2015) because it is very slow to converge for larger scale experiments like the one targeted in this paper.

5 | RESULTS

5.1 | Model calibration and learning curves

In Figure 2, we compare F1 scores and accuracy metrics across training epochs for both training and validation sets. For both VOGN and ADAM, it takes as little as 15 epochs to stabilize and smooth the learning rate of the curves. The initial values of the parameters are randomly

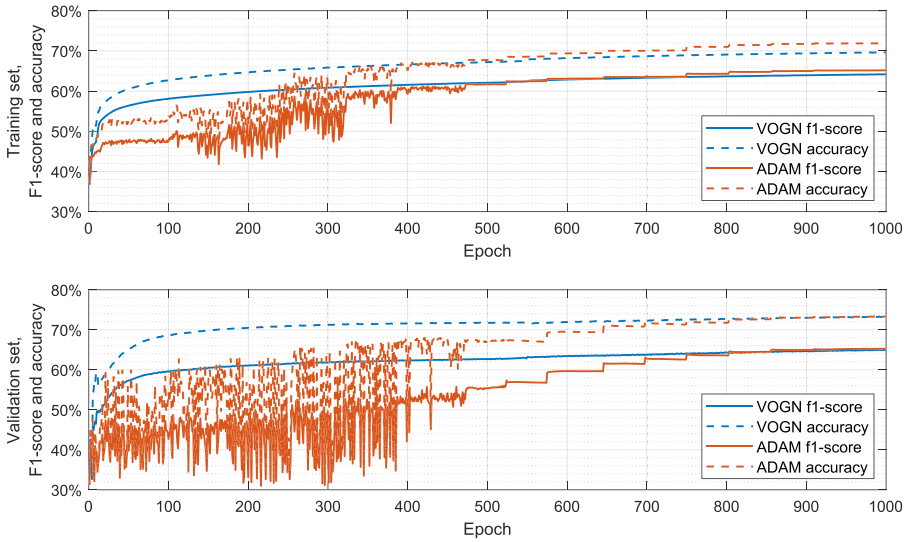

FIGURE 2 Learning curves for VOGN and ADAM for the training set (upper panel) and validation set (lower panel)

initialized. To avoid local minima and to boost the search, the learning rate in ADAM is regularly perturbed, resulting in the step-wise behavior observed of the curves observed across the panels.

For VOGN, curves referring to the training set show a steeper rate at initial epochs and up to about epoch 500, reporting a remarkably higher F1 score and accuracy than for ADAM. This stands for a general superiority in the performance of VOGN with respect to ADAM at early epochs up to moderate ones, indicating that despite the random initialization and the stochastic components embedded in VOGN, perhaps due to its higher number of parameters due to the existence of the parameters' variance, the algorithm converges rather quickly. Only around epoch 500 ADAM metrics are comparable with those of VOGN. At higher epochs, we do not observe a relevant difference in F1 scores, while, in terms of accuracy, ADAM slightly outperforms VOGN on the training set. That is, the learning in ADAM is shown to be slower but, on average, steady, with certainly lower rates compared with VOGN, but constantly improving across the epochs. On the other hand, after steep improvements in initial phases, VOGN's training is quite achieved already at epoch 500, leaving only a slight 5% improvement of the metrics in the following 500 epochs.

Also on the validation set, we observe that at initial epochs, metrics for VOGN greatly outperform those for ADAM; perhaps the prior variance adds a randomization effect that allows a wider sampling of the space around the local parameters to access large gradients that readily adjust the step direction towards the minimum. It is only around epoch 800 that we observe a comparable performance. This could be interpreted as a better generalization ability of VOGN on unseen data, especially if noticing that for VOGN the F1 score and accuracy curves on the validation set are slightly higher than for training. The homogeneity of the data and the same complexity across the two sets, further motivate the conclusion that VOGN embeds a more general classification rule, along with the metrics provided in Appendix A1. Also the different levels in curves' smoothness underline that while VOGN quickly approaches the minimization objective, ADAM appears to repeatedly overshoot the objective, leading to segmented curves up to epoch 500. At higher training epochs, validation curves' rates of growth for VOGN and ADAM appear quite flat, indicating that the training is overflown and the performance metrics are comparable. In this light, we might expect a comparable performance of the two algorithms on the test set, perhaps without a strong winner. This is indeed the case, see Section 5.5 and Appendix B. As it also emerges from Section 5.5, the performances for MCD and SGD are quite poor compared with VOGN and ADAM, making the former two optimizers quite unsuitable for our classification task, and therefore omitted from Figure 2.

5.2 | Posterior distribution

Following the updates (13) and (15), VOGN learns variational posterior's mean and updates the prior precision to the posterior diagonal covariance matrix $\boldsymbol{\sigma}^2 I$, with $\boldsymbol{\sigma}_t^2 = 1/(N(\boldsymbol{s}_t + \tilde{\alpha}))$. For illustration purposes, Figure 3 depicts the learning of B-TABL's λ, a characterizing parameter for the network architecture.

In general, for all the parameters in TABL, we observe a similar pattern where both parameters' means and variances converge to certain (different) levels. The posterior distribution is representative of the parameters' uncertainty after observing the data, that in VOGN's variational setting is forced to be a Gaussian distribution. We make use of the posterior's valuable information on the parameters' relevance by conducting individual t-tests on their significance. Very low p-values support the relevance of all the TABL parameters and implicitly that the network architecture is well scaled for the problem under consideration. Following (1), the posterior distribution builds the predictive one, a major focus in this paper.

5.3 | Predictive distribution

5.3.1 | Interpreting predictive probabilities

We approximate VOGN's predictive distribution with $N_s = 50$ draws from the posterior distribution. That is, according to (2), we approximate the posterior distribution in (1) for a given input sample by 50 samples drawn from it to capture the uncertainly associated with a forecast $\hat{y}_{ic}$ given an unseen input $\boldsymbol{x}_i$ and the data used in the training phase. According to the decision criterion in (3),

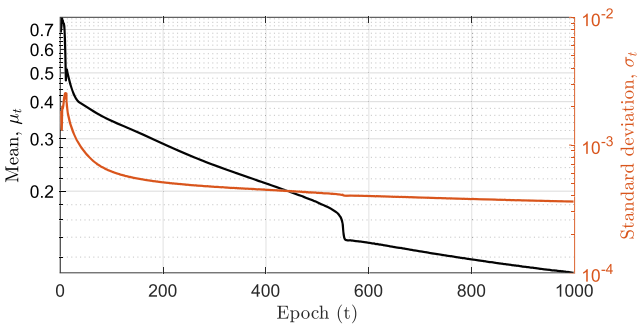

FIGURE 3 Learning of the variational parameters for TABL's mixing coefficient λ

the forecast's class is given by the predicted class of maximum class probability. Note that, aligned with (1), the predictive distribution is a distribution on class probabilities $p(\boldsymbol{y}_i|\boldsymbol{x}_i,\mathcal{D})$ and not on the forecasts $\hat{y}_{ic}$. Figure 4 provides insights into interpreting predictive probabilities and explains pitfalls in uncertainty interpretation.

The bottom-right panel in Figure 4 depicts the box plot of the class probabilities corresponding to N_s forward passes for a certain input in the test set. For this example, the distribution of the predicted classes is unarguably leaning towards Class 3, while across the 50 samples, the class probabilities for Classes 1 and 2 are similar. The bottom-left panel depicts the corresponding kernel probability density functions of the class probabilities, along with their per-class average. That is, predicted probabilities for the sample input in Figure 4 are $p(\boldsymbol{y}_i|\boldsymbol{x}_i,\mathcal{D}) = [0.323,0.324,0.352]$ for Classes 1, 2, and 3, respectively. A complete analysis of the joint distribution of $p(\boldsymbol{y}_i|\boldsymbol{x}_i,\mathcal{D})$, here out of scope as it would likely involve a dependence modeling step through copulas that perhaps is not general. Yet the extracted mean values are quite informative. In particular, we see that the probability of Class 3 is certainly higher than those of other classes, even though not overwhelming. That is, the predictive distribution underlines a scenario of rather high uncertainty. Indeed, an investor who, for example, invests in an asset expecting its price to increase would, on average, observe an actual increase in the asset value with a 35% probability. This is the correct interpretation upon the predictive probability $p(\boldsymbol{y}_i|\boldsymbol{x}_i,\mathcal{D})$. However, upon applying the classification criterion (3), the conclusions might be quite misleading. The top-right panel in Figure 4 depicts per-sample class probabilities, where big-sized points represent classes of maximum probability. For 43 samples out of 50, the class of maximum probability is 3. For three samples out of 50 is Class 1, and for three samples out of 50 is Class 2. By applying (3), we would classify 86% of the samples in Class 3, as depicted by the histogram in Panel 4 of Figure 4. The histogram misleadingly covers a situation of high uncertainty with a quite overwhelming frequency observed for Class 3: If we were to draw the forecasts according to the joint distribution in Panel 2, we would observe, on average, only about 35% of the samples in Class 3.

The availability of the predictive distribution in Bayesian DL frameworks is certainly the most remarkable aspect with respect to non-Bayesian approaches such as ADAM. Indeed, continuing with the above example, ADAM would not capture any uncertainty in the predicted label. An investor following ADAM's forecasts (whose performance is comparable with that of VOGN, see Section 5.5) would not be able to capture the high degree of uncertainty that VOGN unveils. Needless to say that the impact of uncertainties on whatever trading strategy an investor adopts is quite significant. For example, an investor might choose to trade based only on predictions associated with relatively low uncertainty or take well-informed actions to account for the actual possibility that the direction of the price movement is opposite to the predicted one. On the other hand, ADAM's forecasts are incapable of addressing the low 35% probability chances of a price increase, leaving the investor completely blind about the actual probability of a price increase and the perhaps adverse downward 32%-likely movement.

By inspecting a large number of examples and the overall statistics on class probabilities, we observe that the case studied in Figure 4 is somewhat atypical, in the sense that 35% of the predicted probabilities for the maximum-probability class are in the lowest quantiles of predicted probabilities for the prediction's class; see Section 5.3.2. Typical values are about 50%: This still

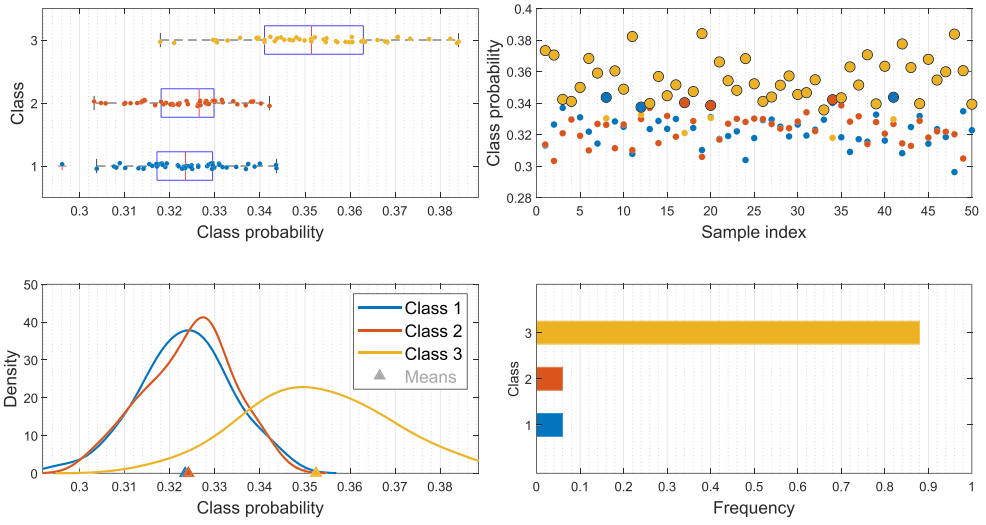

FIGURE 4 Class probabilities and forecasts for a typical test example. Top-left, Panel 1: box plots of class probabilities. Bottom-left, Panel 2: kernel density estimates and means of class probabilities. Top-right, Panel 3: class probabilities per class, highlighting those of maximum probability. Bottom-right, Panel 4: histogram of forecasts' labels.

results in a wildly uncertain general scenario. Such relevant uncertainty for operational scenarios and real-life decision processes is entirely left unaddressed by ADAM and non-Bayesian methods. Whereas the decision criterion (3) does provide a feasible and practical way to construct forecasts, average class probabilities on repeated forward passes (i.e., predictive distributions) are the truly informative element about forecasts' uncertainty, inaccessible to non-Bayesian DL approaches.

5.3.2 | Predictive probability for the maximum-probability class

For the sample input in Figure 4, the true label corresponds to Class 1. In general, for wrong and correct classifications, the score variation in the class of maximum-probability class can be large. Table 1 reports some statistics on the class of maximum probability (Rank 1, denoted by for an input $\boldsymbol{x}_i$ with $\hat{p}_i^{(1)}$), class of second-highest maximum probability (Rank 2, $\hat{p}_i^{(2)}$) and on the remaining class (Rank 3, $\hat{p}_i^{(3)}$), along with the difference between the first two ranks, for correctly and missclassified labels. For correct classifications, the average (median) predictive probability on Rank 1 classes is 55% (51%), while for the missclassified ones 51% (50%), the average distance between the predictive probabilities on classes of Ranks 1 and 2 is, respectively, 30% (25%) for correct classifications and 22% (20%) for missclassifications. In both cases, we do observe samples of probability Rank 1 with a corresponding predictive probability as high as 100% and as low as 33%. Ideally, we would like to observe (i) high Rank 1 predictive probabilities for correct classifications, (ii) quite lower values for missclassified samples, and (iii) a neat separation between $\hat{p}_i^{(1)}$ and $\hat{p}_i^{(2)}$ for correctly classified samples. This is the case, but the magnitudes of the differences are small. Table 1 underlines that the predictive uncertainty in the forecasts is consistent and homogeneous whether the labels are eventually correct or wrong, with $\hat{p}_i^{(1)}$ consistently being of about twice $\hat{p}_i^{(2)}$ and $\hat{p}_i^{(2)}$. The observed differences in Rank 1 and Rank 2 predictive probabilities are, on average, as little as 7.5% between correctly and misscorrectly classified labels, while $\hat{p}_i^{(1)}$ differs by only 3.4%.

Accordingly, the empirical survivor function (ESF) for correctly classified labels in Figure 5 slightly dominates the one for missclassified labels. Importantly, Figure 5 unveils that the probability of observing $\hat{p}_i^{(1)} > 0.6$ is only about 10% to 15%. This means that mild-to-low uncertainties on the maximum-probability class are quite rare, and high confidence is even rarer (7% to 9% for $\hat{p}_i^{(1)} > 0.9$). However, the ESFs do not cross each other, and the difference is positive. For the same (or greater) level of confidence, the number of correctly classified samples is, on average, 5% (but up to 9%) higher for the correctly classified samples than the missclassified ones (difference curve in Figure 5).

TABLE 1 Statistics on VOGN's predictive probabilities.

	Class probability			Difference
	$\hat{p}_i^{(1)}$	$\hat{p}_i^{(2)}$	$\hat{p}_i^{(3)}$	$\hat{p}_i^{(1)} - \hat{p}_i^{(2)}$
Correctly classified labels				
Mean	0.550	0.254	0.196	0.296
Median	0.515	0.262	0.220	0.250
Min	0.335	0.000	0.000	0.000
Max	1.000	0.478	0.330	1.000
Missclassified labels				
Mean	0.516	0.295	0.189	0.220
Median	0.500	0.296	0.204	0.196
Min	0.335	0.000	0.000	0.000
Max	1.000	0.479	0.330	1.000

5.3.3 | Distribution of the scores

To better understand the uncertainties arising from the predictive distribution, Figure 6 depicts the (kernel) density estimates of the scores across correctly and missclassified labels. The top row in Figure 6 refers to correctly classified samples (TPs). For TPs, scores are well separated in the sense that the distribution of class one is well

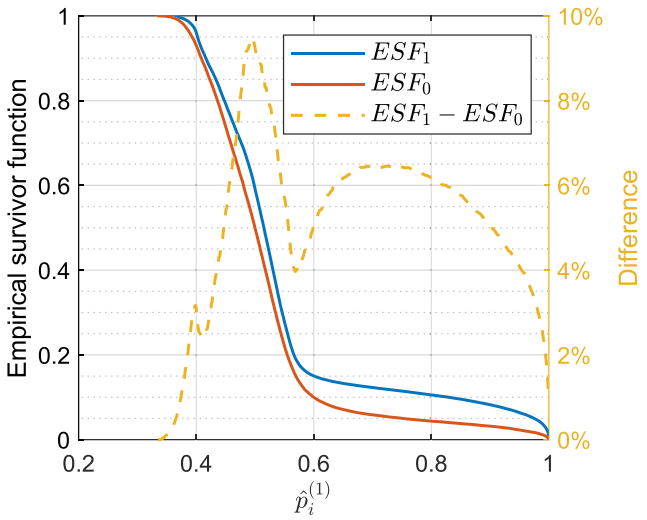

FIGURE 5 Empirical survivor function of $\hat{p}_i^{(1)}$ for correct classifications (ESF_1) and misscorrect (ESF_0), along with their difference.

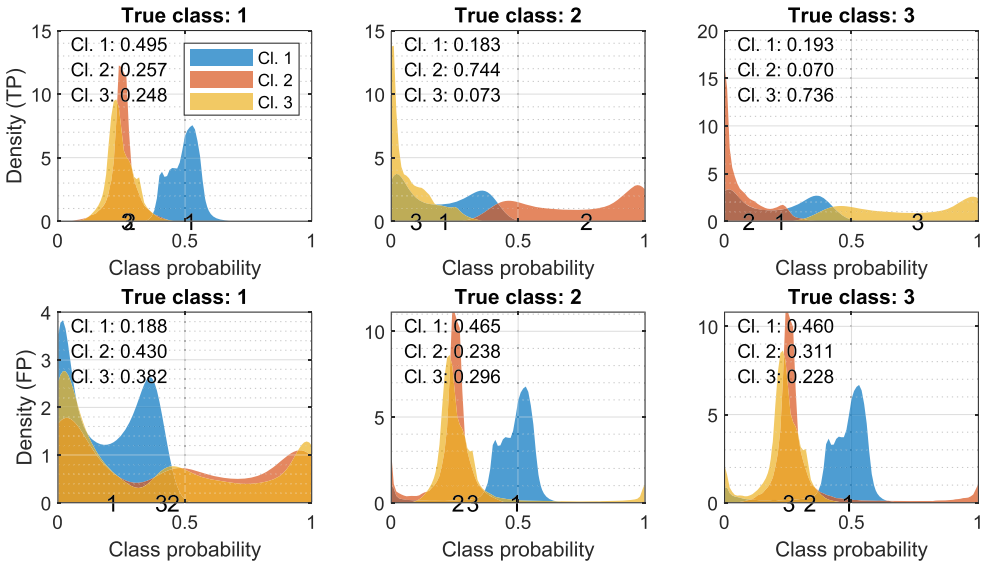

FIGURE 6 Distribution of VOGN's predictive probabilities. Top row: distribution of the class probabilities for correctly classified labels. Bottom row: distribution of the class probabilities for missclassified labels. Labels on the x axis are in correspondence of class-probability averages, also overprinted in the top-left corners.

detached and distinguishable from the others. Interestingly, when the model is correct, the uncertainty in Classes 2 and 3 is much lower than in the stationary-price case. This stands for the existence of clear patterns in the features that are truly indicative of the direction of the price movement, driving predictive probabilities close to 1 (i.e., uncertainty close to zero). When the model is correct about Class 1 assignments, its confidence is somewhat lower and the densities of the scores for whatever change in price direction generally overlap. Confidence in stationary prices is about 0.5, while the remaining 0.5 is equally spread across Classes 2 and 3.

Indeed, by only focusing on the missclassified labels (FPs) in the bottom row in Figure 6, we find further evidence that when the model does not correctly classify a stationary mid-price, its predictions are about equally spread among a price increase and a price decrease, showing that in this case, there is no intrinsic bias in the model parameters leaning towards a certain class; the model is simply wrong, and forecasts are flip coins on Classes 2 and 3. On the other hand, the bias towards the majority class is consistent for FPs in Class 2 or 3, and the scores for the true label are always those of lowest means. The same distribution on Class 1 TPs almost identically replicates on Class 2 and 3 FPs: The model interprets certain patterns in the features as remarkably nonindicative of the true Class 2 and 3 labels, causing an overflow of low scores for both of them. The relevant probability mass, excluded from Classes 2 and 3, is transferred to Class 1 following a distribution being very close to that observed on Class 1 TPs. This suggests that the model well distinguishes patterns indicative of Classes 2 and 3, and when these are absent, Class 1 classification is enforced. In this regard, see Section 5.3.4.

Tails in FPs for Classes 2 and 3 constitute interesting cases of very high Class 3 and Class 2 predictive probabilities corresponding to wrong assignments in Classes 2 and 3. Patterns indicative of Classes 3 and 2 are causing false positives in Classes 2 and 3: (Rarely) typical features for Classes 3 and 2 are observed for mid-prices, eventually moving in the opposite direction. These real-surprise elements in the order flow are perhaps aligned with its stochastic nature.

5.3.4 | Model learning

There is a conclusive important insight on the model's learning within Figure 6. In particular, the B-TABL (and TABL, as the following applies to the scores from all the optimizers) architecture is learning how to classify increases and decreases in mid-prices but not stationary prices. The distribution of Class 1 scores is the very same for the TPs in Class 1 and the FPs in Classes 2 and 3. That is, a distribution is placed over the scores in Class 1 and is updated only if relevant features for Class 2 or Class 3 decisions are detected. Indeed, when the model is not capable of correctly classifying Classes 2 and 3, the distribution of the scores on Classes 2 and 3 is roughly the same. In fact, the three plots in Figure 6 are almost identical at a visual inspection. That is, unless there is robust evidence of an upward or downward price movement very likely corresponding to correct classification, the distribution of Class 1 scores as FPs for Classes 2 and 3 is the same. When the model truly does not detect any relevant information to discern whether the price is moving, movements are classified as stationary, and the very same distribution observed in Class 1 TPs is adopted. This

means that the density for TPs in Class 1 is not actually learned from features that characterize this class. The density that is observed for Class 1 labels is to be interpreted as the one that best improves the validation loss when the model is not detecting clear signals of future price movements. This is further supported by the analyses in Sections 5.4 and 5.6 and Appendix B1.

5.4 | Labels' forecasts

Non-Bayesian methods such as ADAM and SGD provide single forecasts for a trained model. VOGN and MCD, due to the random sampling from the posterior and the dropout layer, lead to different foretasted labels at each forward pass. As noted in Section 5.3.1, the distribution of the scores provides a misleading interpretation of forecasts' uncertainties. However, scores embed rich information for understanding the behavior of the labels' forecast and the underlining label-assignment mechanism. Again, the distribution of labels' forecasts analyzed in this section is based on $N_s = 50$ draws. For VOGN, 96% of drawn labels are all assigned in the same class, 4% to two classes, 0.3% to three. Among the inputs whose labels' forecasts are across two classes, the predictive probability for the maximum class is 42%, and the difference between the predictive probability on the two classes is, on average, 3% (max. 36%, min. 0%). Forty-five percent of these inputs correspond to true labels in Class 1, and for 98%, Class 1 represents one of the two classes where the labels distribute, while the others are Classes 2 and 3 with 50% frequency. That is, the model appears inconsistent in labeling stationary prices over positive or negative movements but is very consistent in labeling the latest two. Sixty-one percent of the samples with forecasts across two labels show, however, at least 80% of the 50 draws in the same class, while very ambiguous inputs account for only 6%, with a difference in the number of samples in the two classes not exceeding three. We could provide analogous information for the 0.3% of the samples whose forecasts' labels are observed over three classes; rather, we point out that the predictive probability on the three classes is on average 31%, 35%, and 35%, corresponding indeed to the most uncertain classifications. The above numbers suggest that typically the consistency in the foretasted labels is remarkable, that is, that the modal value is overwhelming. This means that a single draw from the network would be very likely to equal the modal value. Indeed by randomly selecting with replacement 500 output vectors of labels from the 50 draws available for each test example, on average, 99.36% of the labels correspond to theirs distribution modal value.

Therefore, in the following analyses, we include performance metrics based on modal forecasts as representative of the typical performance observed over a single forward pass. For completeness, we also consider metrics based on means and medians of the N_s labels, rounded to the closest class' integer index. We omit the above details for MCD but underline that 72% of the labels' forecasts are observed over three classes: a remarkable difference. This is due to the fact that on a single TABL layer, the regularization usually provided by dropout causes a random-like assignment of the output classes.

5.5 | Performance measures

Table 2 reports the performance of different Bayesian and non-Bayesian optimizers for the (B)-TABL architecture. With respect to the test set, Table 2 includes microaverages, macroaverages, and weighted macroaverages as synthetic measures for evaluating the overall performance of the different classifiers across multiple classes. Microaverages are constructed by summing the true-/false-positive/-negative rates individually for each class, before applying the definition of the specific performance measure under consideration. On the other hand, macroaverages refer to simple averages of the individual performance measures computed for each class. By accounting for the relative sample frequency of each class in taking averages, we construct weighted macroaverages. Note that accuracy and microaverages for precision, recall, and F1 score are all equal and reported under a single column. Although macroaverages are the performance measures usually reported, as our sample is highly imbalanced (67% of the test samples in the stationary class and equally distributed across the remaining two classes), alternative multiclass statistics are here relevant. Macroaverages weight each class equally by computing the average of the metrics computed independently for each class. As a consequence, nondominant-class' metrics might mislead the conclusions on the overall performance of the classifier. By accounting for class weights, weighted macroaverages naturally alleviate this issue. On the other hand, microaverages, by summing the true-/false-positive/-negative rates individually for each class, aggregate the contributions of all classes to compute average metrics. By weighting each sample equally, microaverages apply well to imbalanced problems where, from a qualitative standpoint, there are no differences in the importance of each class. In our context of imbalanced classes and multiclass task, the preferred metrics are the F1 score, which embeds both precision and recall, and microaverages.

TABLE 2 Performance measures for the multiclass classification task.

	Any	Precision		Recall		F1 score	
	Micro	Macro	Weighted	Macro	Weighted	Macro	Weighted
VOGN sample-by-sample							
Mean	0.774	0.736	0.763	0.592	0.774	0.636	0.751
Median	0.774	0.736	0.763	0.592	0.774	0.636	0.751
Min	0.772	0.730	0.761	0.589	0.772	0.633	0.749
Max	0.776	0.743	0.766	0.596	0.776	0.638	0.752
VOGN based on forecasts' function							
Mean($\hat{Y}_i$)	0.772	0.731	0.761	0.591	0.772	0.633	0.749
Median($\hat{Y}_i$)	0.774	0.736	0.763	0.592	0.774	0.636	0.751
Mode($\hat{Y}_i$)	0.774	0.737	0.763	0.592	0.774	0.636	0.751
VOGN predictive distribution							
$\hat{Y}_{pred}$	**0.774**	0.737	0.763	**0.592**	**0.774**	**0.636**	**0.751**
$\hat{Y}_{pred}$ (med.)	0.774	0.737	0.763	0.592	0.774	0.636	0.751
Other optimizers							
ADAM	0.772	**0.767**	**0.770**	0.570	0.772	0.619	0.741
MCD (mea.)	0.581	0.450	0.598	0.460	0.581	0.454	0.588
MCD (pred.)	0.638	0.500	0.630	0.492	0.638	0.495	0.634
SGD	0.687	0.556	0.660	0.505	0.687	0.522	0.667
Differences							
Min—ADAM	0.0%	−3.8%	−0.9%	1.8%	0.0%	1.3%	0.7%
$\hat{Y}_{pred}$—ADAM	0.2%	−3.1%	−0.7%	2.2%	0.2%	1.6%	0.9%
$\hat{Y}_{pred}$—MCD (pred.)	13.6%	23.7%	13.3%	10.0%	13.6%	14.0%	11.7%
$\hat{Y}_{pred}$—SGD	8.7%	18.1%	10.4%	8.7%	8.7%	11.4%	8.3%

Note: Sample-by-sample refers to metrics computed for each of the N_s samples. $\hat{Y}_{pred}$ (med.) refers to forecasts obtained from the predictive distribution based on the sample median (instead of sample average (2), as for $\hat{Y}_{pred}$). MCD entries refer to sample-by-sample averages. MCD (pred.) refers to forecasts based on the predictive distribution. Top metrics are reported in bold, excluding the row referring to the sample-by-sample maximum (Max row).

For the VOGN optimizer, results are divided into three panels. The upper one reports summary statistics for individual metrics computed for each of the $N_s = 50$ simulated outputs, that is, in a sample-by-sample fashion. Following the discussion in Section 5.4, the second panel addresses the possibility of constructing labels' forecasts based on group statistics extracted from the N_s labels. These correspond to forecasts' labels sample mean, median, and mode. The former statistics requires rounding to the nearest integer to be feasible, yet in our sample, rounding applies to only 3.5% of the per-example labels' means, to 0.26% of medians, and never to modes. The third panel reports the metrics corresponding to predictive distribution, i.e., by considering the class of maximum predictive probability, computed under the standard Bayesian averaging approach and, alternatively, by considering median probabilities as a robust alternative to possible severe outliers.

For VOGN, predictive's distribution results are consistently the highest ones. However, up to three decimals, there are generally no differences between the three panels. Performance measures for median and modal forecasts largely overlap and equal predictive's distribution metrics. Slightly worse results are obtained by considering (rounded) forecasts' averages. The former aligns with the sample-by-sample centrality measures and predictive distributions' ones. This also suggests that for forecasting purposes, a single draw from the posteriors (whose corresponding label would approximate the forecasts' median label very closely) would lead to results perfectly aligned with the predictive's ones (implying a considerable computational advantage).

Among the other optimizers, ADAM stands out as the most valid alternative. Expect on precision, it does not perform better than any VOGN's metrics. Interestingly, metrics' minima in the top panel are always

higher than ADAM's metrics (except for precision, where neither the maximum reaches ADAM's performance). This provides significance to the results in favor of VOGN as even the most unfortunate posterior sampling shows superior performance than ADAM, up to 1.8%. Concerning VOGN's predictive distribution, the observed improvements in performance with respect to ADAM are slight yet significant: the Bayesian optimizer does not provide worse results than the widely adopted ADAM (except for precision), and it enables the predictive analysis of forecasts' uncertainty described in Section 5.3. Lastly, MCD and SGD do not seem to be competitive for the prediction task under analysis. In Appendix A1, we provide analogous stock-specific results.

Our following considerations concern the single-class problem classification. Despite the above multiclass task where each label is classified across three classes, by the single-class task, we mean a binary classification problem where the true class is specified in advance, and the other classes constitute the negative class. Remind that the model is calibrated for the multiclass task: single-class metrics could be improved by recalibrating the model specifically for forecasting a specific price-change direction.

A first useful analysis is that of inspecting the distribution of labels assigned to the true class; see Figure 7. The plot suggests a positive bias towards Class 1 and a negative bias in the labels frequencies in other classes. As confirmed later, the first is due to the large number of FPs for class one, the latter is due to low TP rates for Classes 2 and 3. Note that the differences between the frequencies based on VOGN's modal prediction and predictive distribution are irrelevant, while for MCD, these are minor and favor predictions based on the predictive density. In the following, we will focus only on results resulting from predictive probabilities. From the analyses in Appendix B1, we find that MCD alignment with the sample frequencies is not indicative of a genuine satisfactory performance: For Class 1 (Classes 2 or 3), this arises from a lower (comparable) true-positive rate (TPR) and comparable (lower) false-positive rate (FPR) with respect to the other optimizers. See, for example, Figure B1 therein.

From Table 3, it appears that VOGN and ADAM have quite heterogeneous performances based on the measure and class under consideration. In particular, a conclusion on whether it is advisable to use VOGN or ADAM, in general, cannot be made. Overall, we observe a tendency for ADAM to perform better in terms of precision and recall, thus on TPs therein involved. Yet when the two are considered jointly (harmonic mean), the F1 score favors VOGN. VOGN furthermore improves the detection of TNs involved in computing accuracy and of course enables the uncertainty analyses based on the predictive distribution.

5.6 | ROC and calibration curves

For our multiclass classification problem, we consider receiver operating characteristic (ROC) curves for the predicted classes. In cases where there are no disparities in the cost of false negatives as opposed to false positives, the ROC is a synthetic measure of the quality of models' prediction, irrespective of the chosen classification threshold. To construct ROC curves, we discard ambiguous examples by thresholding each validation input's softmax output and mark the remaining test examples as correctly or incorrectly classified, from which TRP and FPR rates are computed. We apply following thresholds $\{0.05,0.1,0.15,...,1\}$.

Figure 8 depicts ROC curves computed from micro and macro FPR and TRP rates for both VOGN and ADAM. For VOGN the figure includes the 95% interval extracted from the TRP variation across the forecast samples along with the main solid line based on the predictive distribution. The multiclass microaverages and macroaverages for VOGN's curves are dominating. This indicates that larger predicted scores are increasingly more tightly associated with TP than FP, for VOGN more than for ADAM, and that across the whole FPR domain

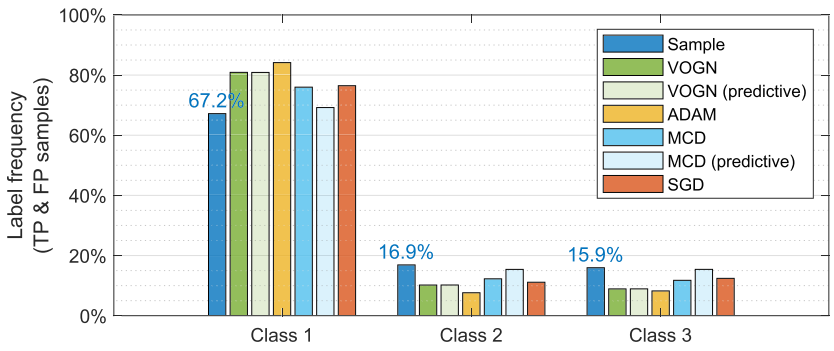

FIGURE 7 Overall distribution of labels' frequencies across the classes. The actual sample-data distribution is meant to be used as a benchmark. Bars display for each class and optimizer the fraction of labels correctly assigned to it (TPs)

TABLE 3 Performance measures for the single-class classification task.

	Precision			Accuracy		
	Class 1	**Class 2**	**Class 3**	**Class 1**	**Class 2**	**Class 3**
VOGN	**0.789**	0.721	0.699	0.795	0.876	0.876
VOGN (pred.)	0.789	0.722	0.700	**0.795**	**0.876**	0.876
ADAM	0.774	**0.740**	**0.789**	0.789	0.868	**0.888**
MCD	0.741	0.314	0.295	0.633	0.763	0.765
MCD (pred.)	0.757	0.381	0.361	0.684	0.794	0.798
SGD	0.759	0.476	0.433	0.725	0.826	0.824
	Recall			**F1 score**		
	Class 1	**Class 2**	**Class 3**	**Class 1**	**Class 2**	**Class 3**
VOGN	0.950	**0.436**	0.391	0.862	0.544	0.501
VOGN (pred.)	0.950	0.436	0.391	**0.862**	**0.544**	0.502
ADAM	**0.969**	0.334	**0.407**	0.861	0.460	**0.537**
MCD	0.698	0.340	0.342	0.719	0.327	0.316
MCD (pred.)	0.780	0.347	0.349	0.768	0.363	0.355
SGD	0.864	0.314	0.337	0.808	0.378	0.379

Note: Bold values denote the highest value in each column (to simplify the visualization of the results).

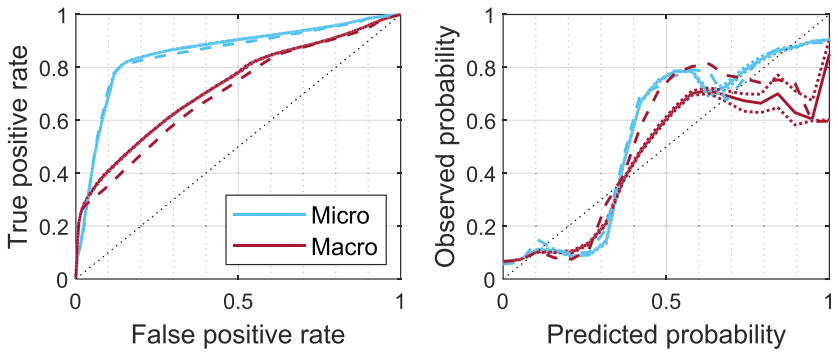

FIGURE 8 ROC curves (left panel) and calibration curves (right panel). Solid lines: VOGN (predictive), dashed lines: ADAM, dotted lines: 95% region for VOGN's Calibration curves (sample-by-sample). Legends in the ROC panel apply to CCs too.

scores implied by VOGN are more conclusive (in terms of TPs) for the true label.

A commonly reported measure is the FPR at 95% TPR, which can be interpreted as the probability that a negative example is misclassified as positive when the TPR is as high as 95%: for macroaverages we compute 88% and 90%, and for microaverages 76% and 77%, for VOGN's forecasts based on the predictive distribution and ADAM, respectively. To assess how well calibrated a model is, CC compares how well the true class frequency determined by a classifier is calibrated to the true frequency of the positive class, for binned predictions (we take 20 bins). The CC curve of a perfectly calibrated model would lie on the diagonal curve, while overconfident predictions would generally result in CC above the diagonal.

CCs in the left panel of Figure 8 underline a comparable performance on microaverages and macroaverages. A remarkable S-shape occurs at lower predicted probabilities. This means that the overall satisfactory statistics in Table 4 arise from a balance between the nonideal scenario where the models are either too much overconfident (predicted probabilities around 20%) and too much underconfident (around 50%). That is, the underlying scores shift from associating too little probability to the true label to way too much. At high scores, both VOGN's and ADAM's microaverage is quite aligned to the diagonal, yet macroaverages are overconfident suggesting high FPs for the dominant Class 1.

ROC and CC plots for the single-class task are found in Figure 9. From the ROC panel, we observe that VOGN outperforms ADAM in classifying labels of Classes 1 and 2, and it has a slightly lower performance on Class 3. As for Table 3, ADAM's metrics are higher than VOGN's for Class 3, determining improved TPRs. CCs for Classes 1 and 2 are quite satisfactory, and the same comment applies as for the CCs in Figure 8. Remarkable is however the U-shape of the curves for Class 1: high Class

	Single-class task			Multiclass task	
	Class 1	**Class 2**	**Class 3**	**Micro**	**Macro**
Area under the ROC curve					
VOGN (pred.)	**0.716**	**0.739**	0.722	**0.858**	**0.726**
ADAM	0.697	0.665	**0.742**	0.851	0.702
MCD (pred.)	0.672	0.649	0.657	0.770	0.659
SGD	0.691	0.660	0.656	0.790	0.669
Expected calibration error					
VOGN (pred.)	−0.107	−0.014	**−0.016**	0.035	−0.046
ADAM	−0.104	0.043	0.033	0.040	**−0.009**
MCD (pred.)	**−0.051**	**−0.016**	−0.044	**0.021**	−0.039
SGD	0.153	−0.081	−0.072	−0.021	−0.032
Expected calibration distance					
VOGN (pred.)	0.144	**0.008**	**0.009**	**0.018**	**0.018**
ADAM	0.146	0.021	0.018	0.018	0.030
MCD (pred.)	0.181	0.028	0.012	0.019	0.023
SGD	**0.039**	0.028	0.027	0.024	0.018

Note: Bold values denote the highest value in each column (to simplify the visualization of the results).

TABLE 4 Measures related to ROC curves and CCs.

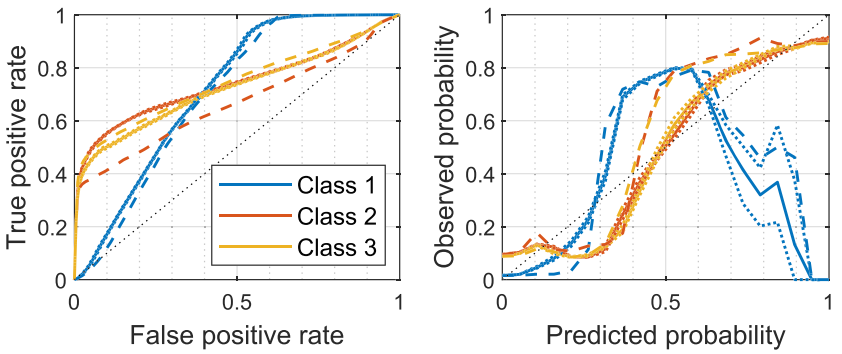

FIGURE 9 ROC curves (left panel) and calibration curves (right panel). Solid lines: VOGN, dashed lines: ADAM, dotted lines: 95% region for VOGN's calibration curves. Legends in the ROC panel apply to CCs too.

1 probabilities are overconfident and misleading as there are no samples in Class 1 at all when models' probabilities for Class 1 are about 1 (confirming the inference from micro-CC and macro-CC in Figure 8). Aligned with the discussion in Section 5.3.4, models are truly learning the classification of Classes 2 and 3. For samples in Classes 2 and 3 which however do not display typical Class 2 or 3 features, scores associated with Classes 2 and 3 are about zero, and all the probability mass is allocated on Class 1. In fact, out of the (only) 20 Class 1 probabilities higher than 0.75, the 75% of them correspond to FNs for Classes 2 or 3. This might be indicative of inadequacy in networks' architecture in uncovering deeper patterns in the data that could address Class 2 and 3 classification or nonstationarity elements of true and atypical surprise not observed in the training set or perhaps not learnable at all due to their randomness.

Table 4 reports the area under the ROC (AUROC), the expected calibration error (ECE), and the L2-norm distance (ECD) between the CCs and the diagonal line and the CCs. High AUROC, small ECD, and small (in absolute value) ECE are preferred. Results are aligned with the earlier plots and confirm the above comments. The low ECEs for MCD are not to be interpreted as evidence of improved calibration, as they arise from rather symmetric S-shaped CCs, that however largely deviate from the diagonal (see ECDs).

6 | CONCLUSION

We proposed a first econometric time-series application with BNNs. Our task focuses on predicting the direction of mid-price changes in modern LOB markets. By

utilizing a state-of-the-art optimizer for Bayesian learning and adopting the suitable TABL capable of fully exploiting the ultra-high-frequency and complex multidimensional nature of the data, we obtain promising results showing that Bayesian methods in DL are feasible, attractive, and valuable for economic applications.

With a number of detailed analyses, we compare several optimizers on the same forecasting exercise and unveil that the Bayesian VOGN optimizer provides, on a general level, the best performance metrics on both multiclass and single-class classification tasks. Yet VOGN's performance is comparable with the well-known and reliable optimization scheme provided by ADAM. At the same time, Monte Carlo dropout and SGD methods do not seem to be suitable for the task under analysis. Furthermore, we extensively interpret and discuss the results, grasping important insights into the model's learning and decision process. The unique feature of Bayesian methods is that of providing posterior and predictive distributions, leading to estimates of the uncertainties associated with the forecasts. The paper discusses how to use and interpret predictive probabilities, providing insights into their implication in the decision process. Following our analysis, and besides promoting further research and applications involving Bayesian DL methods, future research might explore to which extent posterior probabilities lead to better uncertainty-informed trades, for example, by applying and comparing Bayesian and non-Bayesian models for constructing actionable trading strategies, verified with robust back-testing procedures.

ACKNOWLEDGMENTS

This project has received funding from the European Union's Horizon 2020 research and innovation program under the Marie Skłodowska-Curie project BNNmetrics (Grant 890690) and the Independent Research Fund Denmark project DISPA (Project No. 9041-00004).

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are openly available in Fairdata at https://urn.fi/urn:nbn:fi:csc-kata20170601153214969115.

REFERENCES

Blei, D. M., Kucukelbir, A., & McAuliffe, J. D. (2017). Variational inference: A review for statisticians. *Journal of the American statistical Association*, *112*(518), 859–877.

Blundell, C., Cornebise, J., Kavikcuoglu, K., & Wierstra, D. (2015). Weight uncertainty in neural networks. In *Proceedings of the 32nd International Conference on Machine Learning, PMLR*, pp. 1613–1622. https://proceedings.mlr.press/v37/

Caruana, R., Lou, Y., Gehrke, J., Koch, P., Sturm, M., & Elhadad, N. (2015). Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In *Proceedings of the 21st International Conference on Knowledge Discovery and Data Mining*, Association for Computing Machinery, pp. 1721–1730.

Cenesizoglu, T., Dionne, G., & Zhou, X. (2022). Asymmetric effects of the limit order book on price dynamics. *Journal of Empirical Finance*, *65*, 77–98.

Chandra, R., & He, Y. (2021). Bayesian neural networks for stock price forecasting before and during COVID-19 pandemic. *PLOS One*, *16*(7), 1–32.

Collobert, R., & Weston, J. (2008). A unified architecture for natural language processing: Deep neural networks with multitask learning. In *25th International Conference on Machine Learning*, Association for Computing Machinery, pp. 160–167. https://dl.acm.org/doi/abs/10.1145/1390156.1390177

Cont, R., Stoikov, S., & Talreja, R. (2010). A stochastic model for the order book dynamics. *Operations Research*, *58*(3), 549–563.

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, *2*(4), 303–314.

Dahl, G. E., Yu, D., Deng, L., & Acero, A. (2012). Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *IEEE Transactions on Speech and Audio Processing*, *20*(1), 30–42.

Dixon, M. (2018). Sequence classification of the limit order book using recurrent neural networks. *Journal of Computational Science*, *24*, 277–286.

Dixon, M. F., Halperin, I., & Bilokon, P. (2020). *Machine learning in finance*, Vol. 1170: Springer.

Gal, Y., & Ghahramani, Z. (2016). Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of The 33rd International Conference on Machine Learning*, *48*, PMLR, pp. 1050–1059. https://proceedings.mlr.press/v48/

Gefang, D., Koop, G., & Poon, A. (2023). Forecasting using variational Bayesian inference in large vector autoregressions with hierarchical shrinkage. *International Journal of Forecasting*, *39*(1), 346–363.

Geweke, J., & Amisano, G. (2010). Comparing and evaluating Bayesian predictive distributions of asset returns. *International Journal of Forecasting*, *26*(2), 216–230.

Ghayekhloo, M., Azimi, R., Ghofrani, M., Menhaj, M. B., & Shekari, E. (2019). A combination approach based on a novel data clustering method and Bayesian recurrent neural network for day-ahead price forecasting of electricity markets. *Electric Power Systems Research*, *168*, 184–199.

Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, pp. 580–587.

Goan, E., & Fookes, C. (2020). Bayesian neural networks: An introduction and survey, *Case studies in applied Bayesian data science: CIRM Jean-Morlet Chair, Fall 2018*: Springer International Publishing, pp. 45–87.

Goldberg, Y. (2017). *Neural network methods for natural language processing*: Morgan & Claypool Publishers.

Gunawan, D., Kohn, R., & Nott, D. (2021). Variational Bayes approximation of factor stochastic volatility models. *International Journal of Forecasting*, *37*(4), 1355–1375.

Hanin, B. (2019). Universal function approximation by deep neural nets with bounded width and relu activations. *Mathematics*, *7*(10), 992.

Hewamalage, H., Bergmeir, C., & Bandara, K. (2021). Recurrent neural networks for time series forecasting: Current status and future directions. *International Journal of Forecasting*, *37*(1), 388–427.

Holzinger, A., Biemann, C., Pattichis, C. S., & Kell, D. B. (2017). What do we need to build explainable AI systems for the medical domain? arXiv:1712.09923.

Holzinger, A., Langs, G., Denk, H., Zatloukal, K., & Müller, H. (2019). Causability and explainability of artificial intelligence in medicine. *WIREs Data Mining and Knowledge Discovery*, *9*(4), e1312.

Jang, H., & Lee, J. (2018). An empirical study on modeling and prediction of bitcoin prices with Bayesian neural networks based on blockchain information. *IEEE Access*, *6*, 5427–5437.

Jospin, L. V., Buntine, W., Boussaid, F., Laga, H., & Bennamoun, M. (2020). Hands-on Bayesian neural networks—A tutorial for deep learning users. arXiv:2007.06823.

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

Khan, M. E., & Lin, W. (2017). Conjugate-computation variational inference: Converting variational inference in non-conjugate models to inferences in conjugate models. In *20th International Conference on Artificial Intelligence and Statistics*, PMLR, pp. 878–887. http://proceedings.mlr.press/v54/

Khan, M. E., & Nielsen, D. (2018). Fast yet simple natural-gradient descent for variational inference in complex models. In *2018 International Symposium on Information Theory and Its Applications (ISITA)*, IEEE, pp. 31–35. https://ieeexplore.ieee.org/document/8664326

Kingma, D. P., & Ba, J. L. (2015). Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations*, pp. 1–15. https://iclr.cc/archive/2014/conference-proceedings/

Kingma, D. P., & Welling, M. (2014). Auto-encoding variational Bayes. In *2nd International Conference on Learning Representations* (Bengio, Y., & LeCun, Y., Eds.), ICLR, pp. 1–14.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, *25*. https://proceedings.neurips.cc/paper/2012/hash/c399862d3b9d6b76c8436e924a68c45b-Abstract.html

Kuan, C.-M., & White, H. (1994). Artificial neural networks: An econometric perspective. *Econometric Reviews*, *13*(1), 1–91.

Lampinen, J., & Vehtari, A. (2001). Bayesian approach for neural networks—Review and case studies. *Neural Networks*, *14*(3), 257–274.

Lu, Z., Pu, H., Wang, F., Hu, Z., & Wang, L. (2017). The expressive power of neural networks: A view from the width. *Advances in Neural Information Processing Systems*, *30*, 1–9.

Mackay, D. J. C. (1995). Probable networks and plausible predictions—A review of practical Bayesian methods for supervised neural networks. *Network: Computation in Neural Systems*, *6*(3), 469–505.

Magris, M., & Iosifidis, A. (2022). Bayesian learning for neural networks: an algorithmic survey. arXiv:2211.11865.

Makridakis, S., Hogarth, R. M., & Gaba, A. (2009). Forecasting and uncertainty in the economic and business world. *International Journal of Forecasting*, *25*(4), 794–812.

Mbuvha, R., Boulkaibet, I., & Marwala, T. (2019). Automatic relevance determination Bayesian neural networks for credit card default modelling. arXiv:1906.06382.

McNelis, P. D. (2005). *Neural networks in finance: Gaining predictive edge in the market*, Academic Press Advanced Finance Series: Elsevier Academic Press.

Mohamed, A., Dahl, G. E., & Hinton, G. E. (2012). Acoustic modeling using deep belief networks. *IEEE Transactions on Speech and Audio Processing*, *20*(1), 14–22.

Mullainathan, S., & Spiess, J. (2017). Machine learning: An applied econometric approach. *Journal of Economic Perspectives*, *31*(2), 87–106.

Murphy, K. P. (2012). *Machine learning—A probabilistic perspective*, Adaptive Computation and Machine Learning Series: MIT Press.

Ntakaris, A., Magris, M., Kanniainen, J., Gabbouj, M., & Iosifidis, A. (2018). Benchmark dataset for mid-price forecasting of limit order book data with machine learning methods. *Journal of Forecasting*, *37*(8), 852–866.

Ntakaris, A., Mirone, G., Kanniainen, J., Gabbouj, M., & Iosifidis, A. (2019). Feature engineering for mid-price prediction with deep learning. *IEEE Access*, *7*, 82390–82412.

Osawa, K. (2019). *PyTorch-SSO: Scalable second-order methods in PyTorch*. GitHub Repository.

Osawa, K., Swaroop, S., Khan, M. E., Jain, A., Eschenhagen, R., Turner, R. E., & Yokota, R. (2019). Practical deep learning with Bayesian principles. *Advances in Neural Information Processing Systems*, *32*, 1–13.

Passalis, N., Tefas, A., Kanniainen, J., Gabbouj, M., & Iosifidis, A. (2018). Temporal bag-of-features learning for predicting mid price movements using high frequency limit order book data. *IEEE Transactions on Emerging Topics in Computational Intelligence*, *4*(6), 774–785.

Passalis, N., Tefas, A., Kanniainen, J., Gabbouj, M., & Iosifidis, A. (2019). Deep adaptive input normalization for time series forecasting. *IEEE Transactions on Neural Networks and Learning Systems*, *31*(9), 3760–3765.

Passalis, N., Tsantekidis, A., Tefas, A., Kanniainen, J., Gabbouj, M., & Iosifidis, A. (2017). Time-series classification using neural bag-of-features. In *2017 25th European Signal Processing Conference (EUSIPCO)*, IEEE, pp. 301–305. https://ieeexplore.ieee.org/document/8081217

Qi, M., & Zhang, G. P. (2008). Trend time–series modeling and forecasting with neural networks. *IEEE Transactions on Neural Networks*, *19*(5), 808–816.

Ragnar, F. (1933). Editor's note. *Econometrica*, *1*(1), 1–4.

Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: Towards real-time object detection with region proposal networks. *Advances in Neural Information Processing Systems*, *28*.

Salinas, D., Flunkert, V., Gasthaus, J., & Januschowski, T. (2020). DeepAR: Probabilistic forecasting with autoregressive recurrent

networks. *International Journal of Forecasting*, *36*(3), 1181–1191.

Shabani, M., Tran, D. T., Magris, M., Kanniainen, J., & Iosifidis, A. (2022). Multi-head temporal attention-augmented bilinear network for financial time series prediction. In *2022 30th European Signal Processing Conference (EUSIPCO)*, IEEE, pp. 1487–1491. https://ieeexplore.ieee.org/document/9909957

Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. In *3rd International Conference on Learning Representations*, pp. 1–14. https://iclr.cc/archive/www/doku.php%3Fid=iclr2015:accepted-main.html

Sirignano, J. A. (2019). Deep learning for limit order books. *Quantitative Finance*, *19*(4), 549–570.

Skabar, A. A. (2009). *Direction-of-change financial time series forecasting using neural networks: A Bayesian approach*: Springer.

Teräsvirta, T., van Dijk, D., & Medeiros, M. C. (2005). Linear models, smooth transition autoregressions, and neural networks for forecasting macroeconomic time series: A re-examination. *Nonlinearities, Business Cycles and Forecasting*, *21*(4), 755–774.

Tran, D. T., Iosifidis, A., Kanniainen, J., & Gabbouj, M. (2019). Temporal attention-augmented bilinear network for financial time-series data analysis. *IEEE Transactions on Neural Networks and Learning Systems*, *30*(5), 1407–1418.

Tran, D. T., Kanniainen, J., & Iosifidis, A. (2021). How informative is the order book beyond the best levels? Machine learning perspective. In *NeurIPS 2021 Workshop on Machine Learning Meets Econometrics*, pp. 1–12. https://nips.cc/Conferences/2021/ScheduleMultitrack?event=21847

Tran, D. T., Magris, M., Kanniainen, J., Gabbouj, M., & Iosifidis, A. (2017). Tensor representation in high-frequency financial data for price change prediction. In *IEEE Symposium Series on Computational Intelligence*, IEEE, pp. 1–7.

Tran, M.-N., Nguyen, T.-N., & Dao, V.-H. (2021). A practical tutorial on variational Bayes. arXiv:2103.01327.

Tsantekidis, A., Passalis, N., Tefas, A., Kanniainen, J., Gabbouj, M., & Iosifidis, A. (2017). Forecasting stock prices from the limit order book using convolutional neural networks. In *19th IEEE Conference on Business Informatics*, IEEE, pp. 7–12.

Tsantekidis, A., Passalis, N., Tefas, A., Kanniainen, J., Gabbouj, M., & Iosifidis, A. (2020). Using deep learning for price prediction by exploiting stationary limit order book features. *Applied Soft Computing*, *93*, 106401.

Vahidinasab, V., & Jadid, S. (2008). Bayesian neural network model to predict day-ahead electricity prices. *European Transactions on Electrical Power*, *20*, 231–246.

Varian, H. R. (2014). Big data: New tricks for econometrics. *Journal of Economic Perspectives*, *28*(2), 3–28.

Vu, M.-A. T., Adalı, T., Ba, D., Buzsáki, G., Carlson, D., Heller, K., Liston, C., Rudin, C., Sohal, V. S., Widge, A. S., Mayberg, H. S., Sapiro, G., & Dzirasa, K. (2018). A shared vision for machine learning in neuroscience. *Journal of Neuroscience*, *38*(7), 1601–1607.

Zhang, Z., Zohren, S., & Roberts, S. (2019). DeepLOB: Deep convolutional neural networks for limit order books. *IEEE Transactions on Signal Processing*, *67*(11), 3001–3012.

How to cite this article: Magris, M., Shabani, M., & Iosifidis, A. (2023). Bayesian bilinear neural network for predicting the mid-price dynamics in limit-order book markets. *Journal of Forecasting*, *42*(6), 1407–1428. https://doi.org/10.1002/for.2955

APPENDIX A: PERFORMANCE ON INDIVIDUAL STOCKS

All the models are trained in an end-to-end manner over stacked features and labels corresponding to five stocks. As a sanity check, we report in Table A1 the performance for the multiclass task for each of them. We observe metrics aligned in magnitudes with the overall ones in Table 2, confirming a qualitative consistency in the data across different stocks, the reliability of the results, and the robustness of the methods. Standard deviations in the metrics are lowest for VOGN, proving a firmer consistency in the results and perhaps a better generalization ability to unseen market data.

TABLE A1 Performance measures for the multiclass classification task on different stocks.

	Any	Precision		Recall		f1-score	
	Micro	Macro	Weighted	Macro	Weighted	Macro	Weighted
Stock: Kesko Oyj, ISIN: FI0009000202							
VOGN (pred.)	0.776	0.732	0.764	0.594	0.776	0.636	0.753
ADAM	0.776	**0.771**	**0.774**	0.574	0.776	0.624	0.746
MCD (pred.)	0.638	0.497	0.632	0.491	0.638	0.494	0.635
SGD	0.690	0.558	0.663	0.507	0.690	0.524	0.671
Stock: Outokumpu Oyj, ISIN: FI0009002422							
VOGN (pred.)	**0.743**	**0.663**	0.730	0.591	**0.743**	**0.616**	**0.730**
ADAM	0.667	0.656	**0.738**	**0.602**	0.667	0.595	0.685
MCD (pred)	0.607	0.469	0.600	0.447	0.607	0.451	0.600
SGD	0.659	0.527	0.652	0.518	0.659	0.522	0.655
Stock: Rautaruukki Oyj, ISIN: FI0009003552							
VOGN	**0.748**	**0.669**	0.735	0.599	**0.748**	**0.624**	**0.735**
VOGN (pred.)	0.747	0.669	0.735	0.599	0.747	0.624	0.735
ADAM	0.675	0.662	**0.744**	**0.613**	0.675	0.605	0.692
MCD (pred)	0.607	0.470	0.601	0.450	0.607	0.453	0.600
SGD	0.663	0.534	0.658	0.527	0.663	0.530	0.660
Stock: Sampo Oyj, ISIN: FI0009003305							
VOGN (pred.)	0.743	0.663	0.730	0.592	0.743	0.617	0.730
ADAM	0.669	0.658	**0.739**	**0.605**	0.669	0.598	0.686
MCD (pred)	0.608	0.467	0.599	0.446	0.608	0.451	0.600
SGD	0.659	0.526	0.653	0.519	0.659	0.522	0.655
Stock: Wärtsilä Oyj, ISIN: FI0009000727							
VOGN (pred.)	**0.747**	0.666	0.735	0.600	**0.747**	**0.625**	**0.736**
ADAM	0.675	0.661	**0.743**	**0.613**	0.675	0.604	0.692
MCD (pred)	0.615	0.476	0.608	0.457	0.615	0.461	0.609
SGD	0.663	0.532	0.659	0.527	0.663	0.529	0.661
Standard deviation							
VOGN (pred.)	**0.014**	**0.030**	**0.014**	**0.004**	**0.014**	**0.008**	**0.010**
ADAM	0.047	0.050	0.015	0.016	0.047	0.011	0.026

Note: Bold values denote the highest value in each column (to simplify the visualization of the results).

APPENDIX B: FURTHER DETAILS ON THE SINGLE-CLASS CLASSIFICATION TASK

A number of further considerations can be drawn by analyzing the details of correct and misscorrect assignments for the single-class classification task. The top-left panel in Figure B1 displays a slightly higher TPR rate for ADAM than for VOGN. For all the optimizers, FNs are equally distributed across Classes 2 and 3, suggesting that missclassifications of stationary-price movements are due to patterns in the features that are truly atypical, neither representative of Class 2 nor 3. Whereas TPRs for Class 1 are generally overwhelming with respect to FNRs, the opposite holds for Classes 2 and 3. For all the true labels in Class 2 or 3, only 35% of them are detected in such classes (TPs), while more than 50% are classified as Class 1 (FPs). The small fraction of FPs for Classes 3 and 2 under the true labels being 2 and 3 underlines that the model confuses price increases (decreases) with stationary prices but not with price decreases (increases). On the other hand, in the bottom row of Figure B1, we find that TNRs for Classes 2 and 3 are very high for all the models, indicating that the models unveil patterns in the features that are truly indicative of Classes 2 and 3, that when not detected lead to high TNRs. For Class 1, however, TNRs are skimpy, and FPs are equally distributed across Classes 2 and 3, underlying the difficulty the model has in detecting features and patterns in the data truly indicative of the stationary-price case. Along with the observations in Section 5.3.4, this provides further evidence that the models truly learn a classification rule for upward and downward price movements only.

A relevant metric for actionable trading decisions is the false discovery rate (FDR). FDR indicates the fraction of false discoveries (FP) over the positives (FP and TP), approximating the probability that a foretasted price direction is a FP. FDR quantifies the risk of undertaking a trading decision (e.g., placing a sell order based on a price-decrease forecast) based on a signal that turns out to be false (price increases). We observe that for all the optimizers the FDR is about 30% in all three classes and that for VOGN and ADAM the difference is always well-beyond 1%. This corresponds to a great achievement upon a random classifier (50% FDR), yet for business operations, it still represents a substantial risk. FDR is an aggregate measure: For a given example, labels' uncertainties are captured by predictive probabilities.

Lastly, we investigate whether VOGN's predictive distribution is capable of quantifying different uncertainty levels for correctly and missclassified labels. Indeed, a considerable difference in predictive probabilities

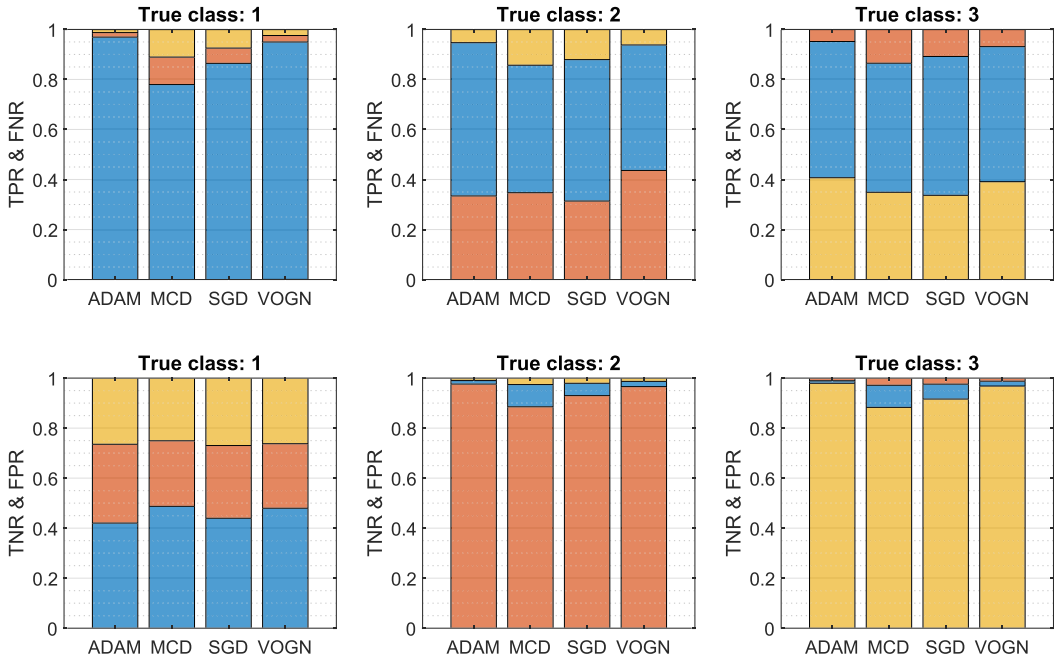

FIGURE B1 Top row: True-positive rates (TPRs) and distribution of the false negatives. TRPs correspond to the heights of the lowest bars, while false-negative rates (FNRs) are extracted as their complement to one. Bottom row: True-negative rates (TNR) and distribution of the false positives. FNRs correspond to the heights of the lowest bars, while false-positive rates (FPRs) are extracted as their complement to one. Forecasts for VOGN and MCD are based on the predictive distribution. Classes 1, 2, and 3 are, respectively, denoted by blue, red, and yellow colors.

	$\hat{p}_i^{(1)}$			$\hat{p}_{ic}$		
True class:	**Class 1**	**Class 2**	**Class 3**	**Class 1**	**Class 2**	**Class 3**
TP	0.495	**0.808**	**0.809**	0.495	**0.808**	**0.809**
FN	0.497	0.513	0.524	0.324	0.253	0.245
FP	0.497	0.579	0.566	**0.497**	0.579	0.566
TN	**0.793**	0.520	0.523	0.159	0.236	0.226

Note: Bold values denote the highest value in each column (to simplify the visualization of the results).

TABLE B1 VOGN's predictive probabilities across correctly and missclassified samples for the class of maximum probability and the true class, that is, $\hat{p}_i^{(1)}$ and $\hat{p}_{ic}$ with c being the true label class, respectively.

between TP and FP as much as TN and FN would be desirable. Low uncertainties associated with, for example, TPs or TNs, would certainly indicate that the predictive distribution is, in fact, well calibrated, being confident on the assignments that eventually turn out to be correct.

The first three columns in Table B1 refer to the predictive distribution of the class of maximum probability $\hat{p}_i^{(1)}$, that is, Class 1 would take as a forecast in an actual forecasting exercise. As desirable, TPs for Classes 2 and 3 correspond to the predictive probabilities, thus to the lowest uncertainties. However, predictive probabilities are comparable for FNs and TNs and slightly higher for FP. That is, low levels of uncertainties can be safely associated with TPs, yet no insight can be grasped on FN, FP, and TN. Enforcing the observations in Section 5.3.4, high scores in Class 1 are associated with TN, indicating that the uncertainty in Class 1 is low when actual forecasts are in Classes 2 or 3. The last three columns in Table B1 refer to the predictive distribution over the true class. This information is clearly unavailable in real settings but useful for model back-testing. Across all the classes, high probabilities are always associated with TPs (desirable), lowest probabilities with TN (would be desirable to observe high values), and about 50% of the predictive probabilities to FP (indicating noteworthy confidence in forecasts that are indeed missclassified).

AUTHOR BIOGRAPHIES

Martin Magris is a postdoctoral researcher at the Department of Electrical and Computer Engineering at Aarhus University (Denmark). His recent research focuses on Bayesian machine learning methods, with a focus on optimization algorithms and applications oriented toward financial and econometrical problems. Martin joined the Department in 2020 as a Marie-Curie fellow after completing his Ph.D. in Econometrics in 2019 at Tampere University (Finland) within the Marie Curie BigDataFinance training network. He received his B.Sc. in Statistics and Mathematics in 2013 and his M.Sc. in Statistical and Actuarial Sciences in 2015 from the University of Trieste (Italy). Before commencing his postgraduate studies, Martin worked as a nonlife actuarial analyst.

Mostafa Shabani earned a master's degree in Information Systems and is currently a Ph.D. candidate at Aarhus University, Denmark, specializing in deep learning for financial data analysis. His research interests lie in machine learning techniques for analyzing time-series data, particularly in the field of computational finance.

Alexandros Iosifidis is currently a professor at Aarhus University, Denmark. He leads the Machine Learning and Computational Intelligence Group, at the Department of Electrical and Computer Engineering, and the Machine Intelligence research area at the Centre for Digitalisation, Big Data, and Data Analytics (DIGIT). His work focuses on designing, analyzing, and understanding machine learning approaches finding applications in problems coming from computer/robot vision, financial modeling, and graph analysis. He is the associate editor-in-chief of the Neurocomputing journal, covering the research area of neural networks, and an associate editor of IEEE Transactions on Neural Networks and Learning Systems and IEEE Transactions on Artificial Intelligence.