Methodology and Computing in Applied Probability (2025) 27:36
https://doi.org/10.1007/s11009-025-10148-8

Generative-Discriminative Machine Learning Models for High-Frequency Financial Regime Classification

Andreas Koukorinis[1,2] · **Gareth W. Peters**[3,4,5] · **Guido Germano**[1,4]

Received: 29 May 2024 / Revised: 11 January 2025 / Accepted: 19 February 2025

Abstract

We combine a hidden Markov model (HMM) and a kernel machine (SVM/MKL) into a hybrid HMM-SVM/MKL generative-discriminative learning approach to accurately classify high-frequency financial regimes and predict the direction of trades. We capture temporal dependencies and key stylized facts in high-frequency financial time series by integrating the HMM to produce model-based generative feature embeddings from microstructure time series data. These generative embeddings then serve as inputs to a SVM with single- and multi-kernel (MKL) formulations for predictive discrimination. Our methodology, which does not require manual feature engineering, improves classification accuracy compared to single-kernel SVMs and kernel target alignment methods. It also outperforms both logistic classifier and feed-forward networks. This hybrid HMM-SVM-MKL approach shows high-frequency time-series classification improvements that can significantly benefit applications in finance.

Keywords Kernel methods · Fisher information kernel · Hidden Markov model · Support vector machine

✉ Andreas Koukorinis
andreas.koukorinis.12@ucl.ac.uk

Gareth W. Peters
garethpeters@ucsb.edu

Guido Germano
g.germano@ucl.ac.uk

1 Department of Computer Science, University College London, 66–72 Gower Street, London WC1E 6EA, UK

2 GSR Markets, Trading, 65 Curzon Street, London W1J 8PE, UK

3 Department of Statistics and Applied Probability, University of California, South Hall, Santa Barbara, California, USA

4 Systemic Risk Centre, London School of Economics and Political Science, Houghton Street, London WC2A 2AE, UK

5 Oxford Man Institute, University of Oxford, Walton Well Road, Oxford, UK

Published online: 16 April 2025

1 Introduction

Intraday price volatility spikes in financial markets pose challenges for accurate intraday regime classification. Classification of regimes is important for trading strategies that exploit short-term anomalies, especially when considering their importance for algorithmic execution and market surveillance (Hamilton 1990; Phillips and Xiao 1998; Cont 2001; Ang and Timmermann 2012). However, many methods overlook key aspects such as the information clock (Bauwens and Veredas 2004), high-frequency dynamics, model adaptability (Vanden 2005; Barndorff-Nielsen et al. 2009; Aldrich et al. 2016; Wang 2021) and volatility spikes that cause regime changes. This paper presents a new hybrid HMM-SVM-MKL learning approach to address these limitations.

Our methodology combines a hidden Markov model (HMM) and a kernel machine machine (SVM and MKL) into a two-stage HMM-SVM-MKL approach. Using a HMM on microstructure time series data, we capture regimes which are overlooked by conventional time series models, Cartea and Jaimungal (2013). Although a HMM can model complex dynamics well, it often faces challenges when used in a discriminative capacity (Jaakkola and Haussler 1999; Bicego 2013). Kernel machines, however, achieve good performance in predictive tasks, but that depends on the quality of their set of features (Schölkopf and Smola 2001; Breneman 2005). We extract descriptive features from the HMM (Bicego 2013; Bicego et al. 2009) and use the different kernel classifiers to improve precision and accuracy (Lasserre et al. 2006; Valstar and Pantic 2007; Fletcher and Shawe-Taylor 2013). These are known as *generative embeddings*. A generative embedding transforms objects into a fixed-dimensional feature space through a generative model learned from the data. This removes the need for hand-crafting features and extends the work of Kozhan and Salon on the exacting information from the order book (Kozhan and Salmon 2012).

In algorithmic trading and, generally, in computational finance applications, it is important to strike a balance between accuracy and computational efficiency. Our methodology combines the strengths of both approaches, with a balance of interpretability and computational efficiency (Lasserre et al. 2006; Wang et al. 2012). Our work is motivated by Zhang et al. (2013), where the graph neural network encoder acts on the input features and their relationships, while contrast learning helps to learn representations of multivariate time series data. This is a form of hybrid learning, since it combines a graph neural network architecture with a loss function for unsupervised representation learning.

We classify six distinct intraday regimes using high-frequency data derived from 40 major stocks of the FTSE100. Our model substantially improves accuracy over alternatives, advancing the modeling of regime identification (Cont 2001; Andersen 1996; Dacorogna et al. 2001). However, we also acknowledge challenges, especially those related to model validation and complexity. For the purposes of promoting explainability, we explore feature visualizations such as t-stochastic neighbor embedding (t-SNE) and RadViz to gain insight and increase the interpretability of the feature set.

This paper presents three contributions that advance research in market microstructure and high-frequency trading. First, we propose a hybrid learning paradigm that combines probabilistic generative modeling (Cartea and Jaimungal 2015; Hong and Sutardja 2015) with discriminative machine learning to predict trade direction (Mainali 2021; Li and Liu 2022), extending previous work (Fletcher and Shawe-Taylor 2013; Coates and Ng 2012). While Shu et al. (2024) also propose hybrid approaches for financial regime detection, our generalized HMM-SVM/MKL framework for high-frequency trading applications differs from their specialized methodology combining statistical jump models for portfolio allocation.

Second, we establish new benchmarks for ultra-high-frequency financial time series classification, demonstrating improvements in intraday regime prediction accuracy compared to existing methods. Third, although hybrid models exist in other domains, this application is novel in a high-frequency setting and effectively captures temporal patterns through HMM-generated feature embeddings from market microstructure data. It represents a novel extension of applied probability theory.

Our approach does not require manual feature engineering, improves classification accuracy compared to single-kernel SVMs and kernel target alignment methods, and outperforms both logistic classifier and feed-forward networks on market-based features.

1.1 Organisation of the Paper

Section 2 presents the modeling context. The experiment workflow is presented in Section 2.3 presents, followed by the data splitting strategy in Section 2.3.1. Section 3 describes the HMM market microstructure model, and Section 4 presents the generative feature embeddings generated by the model. Section 5 presents multiple kernel learning and support vector machine methods. Section 6 details the automatic class label construction methodology. In Subsection 7.3 we also evaluate the performance of generative features versus market features in a simple model setting. The benchmark experiments and descriptions of their results are presented in Subsections 8.1.1 and 8.1.2. Section 8 presents the results, and Section 9 discusses our conclusions. The material in the Supplement contains further technical details, feature vizualisations, the algorithm for the data splitting strategy, the algorithms for the generative embeddings and the Fisher score, and additional material on multiple kernel algorithms including EasyMKL.

2 Modeling Methodology

This work builds on recent advances in hybrid learning for sequence modeling (Li and Liu 2022), integrating modern machine learning techniques for the discriminative element (Wang 2021). We use a two-stage modeling approach.

2.1 First Stage: Choosing a Probabilistic Model of Market Microstructure

In the first stage, building on Cartea and Jaimungal (2013) and Kearns and Nevmyvaka (2013), we propose a probabilistic model that integrates transaction data with nonlinearity and heterogeneity. The hidden dynamic structure of executed trade prices and inter-trade arrival times is inferred, motivated by Cartea and Jaimungal (2013), who focused on joint dynamics and their simultaneous interdependence.

A central element of this approach is a hidden Markov model (henceforth, HMM) that identifies intraday trading states and their persistence, focusing on trade price revisions and arrival times as highlighted by Nasir and Ezeife (2023). This model can incorporate expert prior knowledge about the underlying variables; for example, setting the number of hidden states to three — representing bull, bear, and neutral market regimes — appeals to market intuition, Wang (2021).

This type of HMM is flexible for handling cross-sectional and temporal dependencies in financial data, which makes it a natural choice for modeling these dynamics. However, it will lead to fewer states than the maximum likelihood approach introduced by Pohle et al. (2017).

The first stage involves fitting a hidden Markov model (HMM) to each trading day's data, using the previous day's observations to identify distinct market regimes. The HMM captures underlying market dynamics by modeling the sequential relationships between trades, with a particular focus on trade price revisions and arrival times (Nasir and Ezeife 2023).

We obtain the probability distribution of trades with nonzero-price and zero-price revisions after fitting the model in the first stage. One of the most relevant characteristics in this model is the duration between trades, as it can identify trading behavior over short periods given the information contained (Aldrich et al. 2016; Cartea and Jaimungal 2013; Engle and Russell 1998). The transitions between states in the HMM are learned from observing sequences of purchases to build the transition probability matrix (Nasir and Ezeife 2023).

Next, we introduce a principled approach to extract features by leveraging the sequential modeling strength of an HMM for flexible representation of data sequences in conjunction with kernel-based discriminative learning, which is inspired by Li and Liu (2022) and Bicego (2013); Bicego et al. (2008). Market conditions and dynamics are represented by the generative embeddings, which update continuously based on incoming data ticks.

2.1.1 Generative Embeddings

Generative embeddings, such as the Fisher score and information matrix, are derived from the HMM to capture the sensitivity and information content of the model structure (Jaakkola and Haussler 1999; Bicego et al. 2009; Carli et al. 2010). These embeddings provide a principled approach to extract informative features from trade and quote (TAQ) data, serving as a foundation for creating the discriminative classifiers (Bicego 2013; Mainali 2021; Haiying et al. 2012). We use a dynamic approach to handle the high-frequency nature of trading data. Each trading day starts with fitting a new HMM-model based on the previous day's data to generate initial embeddings. These embeddings are continuously updated throughout the trading day via a real-time adjustment mechanism that responds to incoming data ticks.

These type of feature embeddings offer a bridge between traditional market microstructure models and machine learning methods, that we explore in the proposed HMM-SVM-MKL hybrid framework. As these embeddings can extract meaningful information from latent variable models, they serve as a foundation for creating discriminative classifiers (Bicego 2013; Mainali 2021; Haiying et al. 2012). Similarly to the information-theoretic approach of Mainali (2021), the use of HMM-induced generative embeddings provides the mechanism to quantify and select informative features from trade and quotes (TAQ) data. Embeddings such as the Fisher score allow us to extract gradients and curvature information from the model parameters. This captures the underlying sensitivity and information content related to the model structure.

2.2 Second Stage: Kernel Classification

In the second stage, the HMM-derived embeddings are used as input features for the kernel classifier. Firstly, an SVM is trained to classify market movements or regimes. In a second set of experiments, and to capture the complexities of financial data, a multiple kernel learning (MKL) approach is employed, allowing for the combination of different kernel functions (Fletcher et al. 2010; Kercheval and Zhang 2015). We use discriminative learning as it is known to improve generative modeling (Jaakkola and Haussler 1999; Lasserre et al. 2006; Ghahramani 2001), as an HMM has limited discriminative ability (Valstar and Pantic 2007).

Each discriminative classifier models the boundaries between various classes representing intraday market regimes while indicating potential future trade direction.

2.2.1 Multiple Kernel Learning

The reasons for focusing on multiple kernel learning (MKL) methods are threefold: the MKL approach is essentially a dynamic ensemble method that constructs a mixture kernel, thereby encoding complementary information. Second, it is known that MKL methods are appropriate for small but wide datasets (Gönen and Alpaydin 2011; Gong 2020), as is the case with our high-dimensional generative embeddings extracted from the HMM. The MKL approach involves integration of different kernels, such as polynomials and RBF. This integration helps capture nonlinear decision boundaries even with limited data, while the kernel-based regularization reduces the risk of overfitting (Vapnik 1999). Finally, MKL identifies the optimal combination of information sources encoded across kernels. This aligns well with our objective of finding the right balance between the generative embedding features (Bach et al. 2004; Hussain and Shawe-Taylor 2011; Aiolli and Donini 2015).

The proposed HMM-SVM-MKL learning process is set up in such a manner that it will be inherently tied to the price distribution and how prices adjust in response to information flow. This is based on characterizing market regimes with a state-dependent distribution of logarithmic returns, adjusting for any identifiable activity (Cartea and Jaimungal 2013). These states highlight the balance between informed traders and noise traders in the price discovery process. Moreover, a significant implication of this approach to studying microstructure is the potential to use order flow data to better distinguish the return profiles associated with different trading activities (Easley et al. 2012; O'Hara 2014). This not only offers insights into high-frequency trading dynamics, but also emphasizes the potential of exploiting limit order book dynamics for algorithmic strategies.

Although the use of single kernel approaches (Tay and Cao 2001; Van Gestel 2001; Yang et al. 2002; Huang et al. 2005) provided foundational groundwork, multi-kernel approaches align more closely with the intricacies of financial data and have shown success in similar applications (Fletcher et al. 2010; Kercheval and Zhang 2015). Our proposed learning approach combines the benefits of generative and discriminative models. Generative models can capture the underlying structure of the data and learn the relationships between different variables (Lasserre et al. 2006). Order book embeddings improve classification (Fletcher and Shawe-Taylor 2013). Therefore, we derive generative embeddings, similar to the Fisher score and information matrix, that incorporate information about the microstructure of the market (Jaakkola and Haussler 1999; Bicego et al. 2009; Carli et al. 2010).

We note that we opted to consider these methods in favour of alternative approaches involving deep learning methods for a few reasons. The first is they are open more readily to direct interpretation. As highlighted by Kearns and Nevmyvaka (Kearns and Nevmyvaka 2013), two-stage models strike a balance between interpretability and predictive power.

The HMM component captures underlying market dynamics and regime shifts, while the kernel component focuses on classification performance. In contrast, single-stage models like neural networks may lack interpretability due to their black-box nature (Sirignano 2019; Sirignano and Cont 2019).

The proposed generative approach offers robust statistical estimation and calibration capabilities. This framework demonstrates consistent performance across both liquid and illiquid markets when applied to high-frequency trading and quote data. While deep learning methods have shown limitations with high-frequency financial data, particularly regarding overfitting, our method provides reliable daily predictions across diverse market conditions.

2.2.2 Core Methodological Ideas

In summary, the core ideas from a modeling perspective are:

1. A principled hybrid approach (generative embeddings used as input into a discriminative algorithm), which removes the ad hoc choice of handcrafted features.
2. An HMM-induced feature construction method automates the building of features in a way that is efficient. This can be adapted to a wide range of time-series structures.
3. Specialised time series kernel structures are considered based on Fisher kernels and other generative embeddings. Explicit and efficient algorithm are developed for the computation of the Fisher score, the information matrix and other embeddings. The Fisher score algorithm is the gradient of the log-likelihood function with respect to the model parameters.
4. In addition, an automated process for making labels for the classifier training is developed.
5. Last, we utilise the methodology proposed by Bertolini and Finch (2022) to evaluate feature stability across various market conditions and analyze convergence properties for generative features.

The performance of the framework is presented in several variants. The objective is to validate the algorithm's performance and to compare the algorithm's accuracy and execution time in a manner that is applicable in a practical financial application. Specifically, to validate the effectiveness of the proposed two-stage model, experiments are conducted that compare its performance with single-stage models.

The results indicate that the HMM-SVM-MKL hybrid model performs better than single-stage models in classification accuracy and profitability. The literature supports that two-stage models outperform on complex tasks involving heterogeneity or minority classes. Carranza-García et al. (2021) showed that two-stage detectors continue to offer the most robust performance, despite the increasing popularity of one-stage detectors. Similarly, Kontopantelis (2018) found that for interaction effects, variations between models were more pronounced with the two-stage model consistently surpassed by the two fully specified one-stage models.

Comparing performance across different label mechanisms, the hybrid model improves accuracy by over 30% and f_1 score by over 15% over the other models, confirming its ability to better capture market complexity.

2.3 Design of the experimental flow

A stepwise procedure of the proposed methodology is described next and presented in Fig. 1.

1. Accumulate training data: from TAQ data compute returns and inter-arrival durations based on trade clock.
2. Fit an HMM with 3 states for each day in the sample of length T.
3. Compute the HMM-based features for each day $i = 1, \ldots, T$.
4. Fit SVM/MKL models for each day i, and combination of labels 1 to 6 (6 models per day) using grid search for parameters C and γ. For our purposes, we use polynomial kernels for MKL and RBF for the single-kernel cases.
5. Evaluate out-of-sample for each day $j : j > i$, for each combination 1 to 6 (6 models per day, one for each parametrization of the label mechanism).

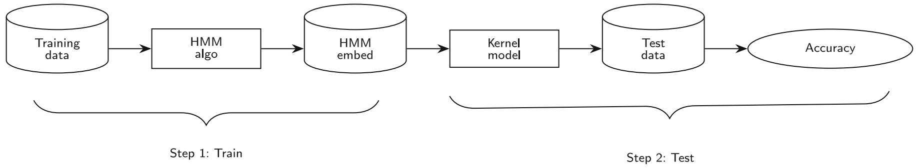

Fig. 1 Workflow of the experimental evaluation. In Step 1, TAQ data are used to train the HMM algorithm and derive features, which are then used to train the various kernel-based classifiers. In Step 2, the classifier is applied to out-of-sample data and the performance is evaluated

2.3.1 Windowed Waterfall Data Split

We use the windowed waterfall methodology that combines the temporal preservation characteristics of serial waterfall distributions with the computational efficiency of window-based sampling for time series data partitioning (Raykar and Saha 2015).

Let $\mathcal{W}_{\text{train}}, \mathcal{W}_{\text{val}}, \mathcal{W}_{\text{test}}$ be fixed-size windows,

$$
\begin{aligned}
&|\mathcal{W}_{\text{train}}| > |\mathcal{W}_{\text{val}}| > |\mathcal{W}_{\text{test}}| \\
&\forall t \in \mathbb{T} : t_{\text{test}} > t_{\text{val}} > t_{\text{train}} \\
&\forall x \in \mathcal{W}_{\text{test}} : \text{time}(x) > \max_{y \in \mathcal{W}_{\text{val}}} \text{time}(y) \\
&\forall x \in \mathcal{W}_{\text{val}} : \text{time}(x) > \max_{y \in \mathcal{W}_{\text{train}}} \text{time}(y),
\end{aligned}
\tag{1}
$$

where $\text{time}(x)$ represents the timestamp of data point x, $\mathbb{T}$ is the time domain, and $|\mathcal{W}|$ denotes the size of window $\mathcal{W}$. The approach maintains strict temporal ordering through fixed-size windows where the training window size exceeds validation which exceeds testing, while ensuring that every data point in each window is strictly more recent than all points in the preceding window, with timestamps satisfying $t_{\text{test}} > t_{\text{val}} > t_{\text{train}}$.

In this cascading window mechanism, the data systematically transitions through splits, maintaining the statistical validity of validation and test sets while employing finite windows to optimize computational resources. The methodology inherits the advantages of waterfall's temporal coherence and reduced validation bias, while leveraging windowing to address non-stationarity and concept drift in temporal distributions. Empirical evidence suggests that this combined approach can sustain model quality while significantly reducing computational complexity. The method demonstrates effectiveness for high-frequency temporal sequences where both long-term patterns and recent distributions must be effectively balanced in the training regime. This approach addresses the fundamental challenge of temporal data partitioning while ensuring statistical robustness in model evaluation frameworks (Raykar and Saha 2015; Derakhshan et al. 2019).

The method is graphically presented in Fig. 2 and the detailed algorithm is provided in the Supplement, Section 2.1.

Financial Markets Hypotheses The stochastic nature of information flow, trader behavior, and price discovery in financial markets forms the core of market microstructure probability models. While building on prior literature (Easley et al. 2012; O'Hara 2014), this work combines features from a probabilistic market microstructure framework with machine learning techniques, focusing on the statistical properties of high-frequency order flow data to characterize return distributions and predictability in a trading environment. We test three hypotheses:

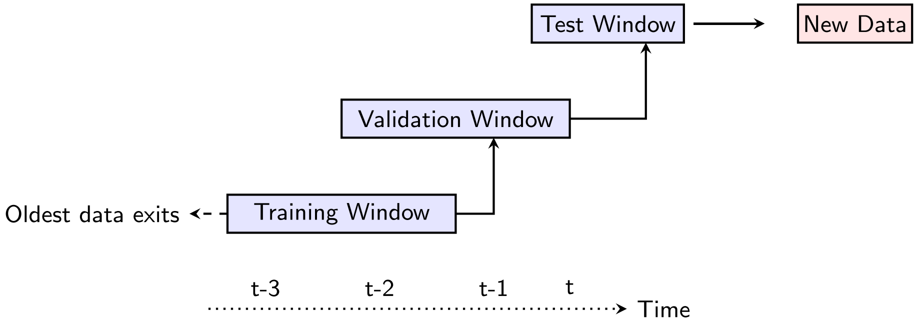

Fig. 2 Windowed waterfall: the figure shows the division of data into training, validation, and test windows, each with fixed sizes, while maintaining temporal order. As new data becomes available, the test window slides forward to incorporate the latest observations. The previous test data cascades into the validation window, the validation data cascades into the training window, and the oldest training data is removed, ensuring a dynamic and temporally consistent split for model development and evaluation

1. Market participant behavior follows regime-switching stochastic processes influenced by volatility and price changes, detectable through high-frequency generative embeddings within the HMM-SVM-MKL framework;
2. State-dependent trading patterns reveal probabilistic market regimes linked to participant behavior, such as market makers or noise traders, characterized by distinct statistical signatures in strategies like rebate trading; and
3. Features from the generative HMM embeddings can effectively represent the probabilistic nature of participant behaviors, validated via statistical visualization methods and characterized through stochastic financial interpretability.

We position microstructure models as probabilistic learning systems capturing the empirical distributions of prices and state-dependent returns, emphasizing the role of trades as stochastic information signals in decision-making (Cartea and Jaimungal 2013).

3 Market Structure Captured by Hidden Markov Models

With the prevalence of high-frequency trading, insight into market dynamics can be gained by examining the tick-by-tick dynamics of trade durations and price revisions (Andersen 1996). However, the joint distribution of these variables exhibit different statistical properties, as the trading varies between regimes dominated by distinct participants. Participants with different information and payoff functions dominate different regimes of market activity (Andersen 1996; Kirilenko et al. 2017; Rengifo and Trendafilov 2015; Rothschild and Sethi 2016).

HMMs model stochastic dynamic systems whose states comprise observable and unobservable components. An HMM consists of two processes: a hidden state process $\mathbf{Z}$ that encompasses the general state of the system (i.e.,the market) and an observation process $\mathbf{O}$ that follows a parametric distribution and models the observations, label sequence, providing an information flow about the market studied. The first fundamental assumption underlying a discrete-state HMM is that the latent process is a finite-state Markov chain. The second fundamental assumption is that the distribution of the observation process at any given time depends only on the current state on which the hidden process finds itself, not on the path of the observation process or the history of the latent states, see Definition 1.

Definition 1 Hidden Markov model. A hidden Markov model is a pair of discrete-time stochastic processes $(Z_i, \mathbf{O}_i)$, $i = 0, \dots, T$, where Z is a Markov chain, i.e. a stationary and homogeneous process that takes values in the finite set of states $\mathcal{Z} = \{\zeta_0, \dots, \zeta_K\}$ and such that

$$\mathbb{P}\left[Z_{i+1} = \zeta | Z_0, \dots, Z_i\right] = \mathbb{P}\left[Z_{i+1} = \zeta | Z_i\right], \quad \text{for } i = 0, \dots, T, \ \zeta \in \mathcal{Z}; \tag{2}$$

$\mathbf{O}$ is vector-valued and such that at any time t_i, $i = 0, \dots, T$, the distribution of the random vector $\mathbf{O}_i$ conditional on all the information collected up to time t_i depends only on the state of the latent process,

$$\mathbb{P}\left(\mathbf{O}_i | \mathbf{O}_0, \dots, \mathbf{O}_{i-1}, \ Z_0, \dots, Z_i\right) = \mathbb{P}\left(\mathbf{O}_i | Z_i\right). \tag{3}$$

We model the relation of trade durations and logarithmic returns as two independent, continuously distributed random variables (Cartea and Jaimungal 2013). An observation process is a two-dimensional object with a parametric distribution, the exact parameters of which are governed by the states of a latent Markov chain. We assume that we have a finite number of trades (observations) T, indexed by $i = \{0, \dots, T\}$. The latent process is a finite-state Markov chain and, as such, its dynamics is governed by a $K \times K$ transition probability matrix $\mathbf{A}$ with elements $a_{jk} \in [0, 1]$,

$$a_{jk} \stackrel{\text{def}}{=} \mathbb{P}\left(Z_{i+1} = \zeta_k | Z_i = \zeta_j\right), \quad i = 0, \dots, T-1, \ \zeta_j, \zeta_k \in \mathcal{Z}. \tag{4}$$

The distribution of the initial state at $t = 0$ is $\boldsymbol{\pi} = (\pi_1, \dots, \pi_k)$,

$$\pi_k \stackrel{\text{def}}{=} \mathbb{P}(Z_0 = \zeta_k), \ \zeta_k \in \mathcal{Z}. \tag{5}$$

In this work, log denotes the natural logarithm, i.e., the logarithm with base e, where e is Euler's number. We denote by $\mathbb{P} : \mathbb{R} \times \mathbb{R}_+ \times \mathcal{Z} \to \mathbb{R}$ the probability function of the complete data at any time t_i. The marginal probability of the observation data (trade durations and trade log-price revisions, which is known as the emission probability) is, of course, dependent on the latent process and will be denoted by $\mathbb{P}_{\mathbf{O}_i|Z_i=\zeta}$, $\zeta \in \mathcal{Z}$ for compactness, i.e.,

$$\mathbb{P}_{\mathbf{O}_i|Z_i=\zeta}(o) \stackrel{\text{def}}{=} \mathbb{P}\left(O_i \in o | Z_i = \zeta\right), \quad o \stackrel{\text{def}}{=} (\xi, \tau) \in \mathbb{R} \times \mathbb{R}_+, \ i = 0, \dots, T. \tag{6}$$

However, given our earlier assumption on the independence of the distribution of duration and price revision conditional on a state, we may write

$$\mathbb{P}_{\mathbf{O}_i|Z_i=\zeta}(o) = \mathbb{P}_{\Xi_i|Z_i=\zeta}(\xi)\mathbb{P}_{\tau_i|Z_i=\zeta}(\tau). \tag{7}$$

We assume that the marginal distribution of trade durations follows an exponential density,

$$\mathbb{P}_{\tau_i|Z_i=\zeta}(\tau) = \lambda_\zeta \exp(-\lambda_\zeta \tau), \ \lambda_\zeta \in \mathbb{R}_+. \tag{8}$$

In this study, we model the trades for a selection of stocks listed on the FTSE100, where on any given calendar date we treat each day as a separate data set comprised of intra-daily data, producing one fitted HMM per day per symbol. We denote the event times for these trades by $\{t_1, \dots, t_T\}$. These times are recorded as the trades occur and need not be equally spaced; they form event-spaced time series data. The *duration* between two consecutive trades is denoted $\tau_i \stackrel{\text{def}}{=} \Delta t_i = t_i - t_{i-1}$, $\tau_i \in \mathbb{R}_+$. The price process of the symbol we are studying, sampled at the time of the symbol's trades, that is, the spot price of the i-th trade, is S_i, $i = 0, \ldots, T$. Logarithmic returns are $\xi_i \stackrel{\text{def}}{=} \log \frac{S_i}{S_{i-1}}$, $i = 1, \ldots, T$. For these observation processes, we will interchangeably use the notation O_i or (ξ_i, τ_i). Given a discrete-time stochastic process, for example, the hidden state process $\mathbf{Z}$, we will use the compact notation $\mathbf{Z}_{k:l}$ to denote the collection of values $\{Z_k, Z_{k+1}, \ldots, Z_l\}$, $k < l$. A similar notation applies to any other process. Lastly, we will denote the realization of a stochastic process along a given path with lowercase letters, e.g. $z_{k:l} \stackrel{\text{def}}{=} \{z_k, z_{k+1}, \ldots, z_l\}$.

Remark 1 Notice that the assumption of independence in the distribution of duration and price changes conditional on the latent process, within the context of an HMM, does not contradict the intuition offered to us by inspecting the data in Fig. 3, that is, a dependence structure between durations and price changes, since from the perspective of our model, this is only the marginal density, once the latent process has been integrated out.

Remark 2 The choice of the probability density for trade durations differs from that used by Cartea and Jaimungal (2013), as the exact format of the exponential distribution without quantisation is utilised. There are well-established degeneracies in any high-frequency financial data set that would justify a quantisation approach, particularly when one is concerned with modeling the limit order book; see Filimonov and Sornette (2015, Section 4) for a detailed discussion on the 'bundling' effect of high-frequency data and plausible explanations. However, as we are only concerned with the trades data on any given symbol and having access to data with microsecond accuracy, we found no significant differences when using quantisation for the trades duration density, and hence chose to work with the exact density.

Log-price revisions are modeled according to a mixture distribution that arises naturally from the trade data. We often see a significant number of trades that occur at the same price, i.e., at zero price revision. In contrast, we model the nonzero part of the price revision as a normal random variable with zero mean and standard deviation that again depends on

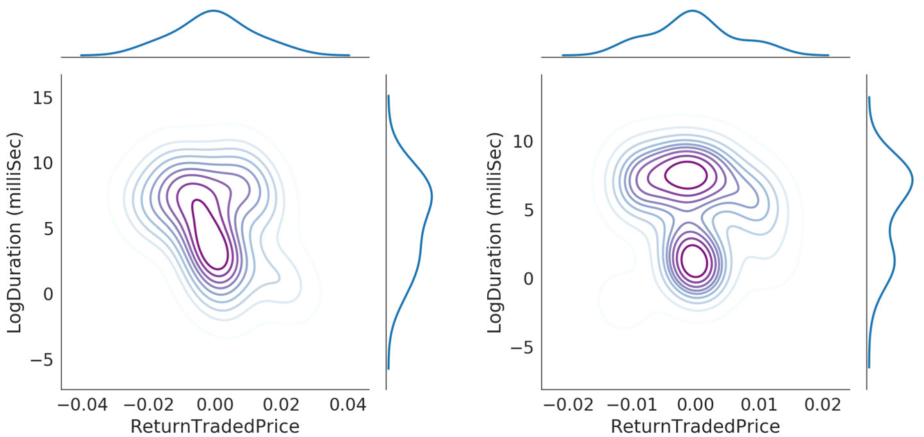

Fig. 3 Logarithmic duration in milliseconds (on the y axis) versus the relative traded price return (on the x axis) for Anglo Pacific Group, observed over two trading days, 4 August 2017 (left) and 26 September 2017 (right). There is a cluster around trades that arrive quickly and cause either a very small or a very large perturbation of price. As the duration increases, the price impact is more meaningful

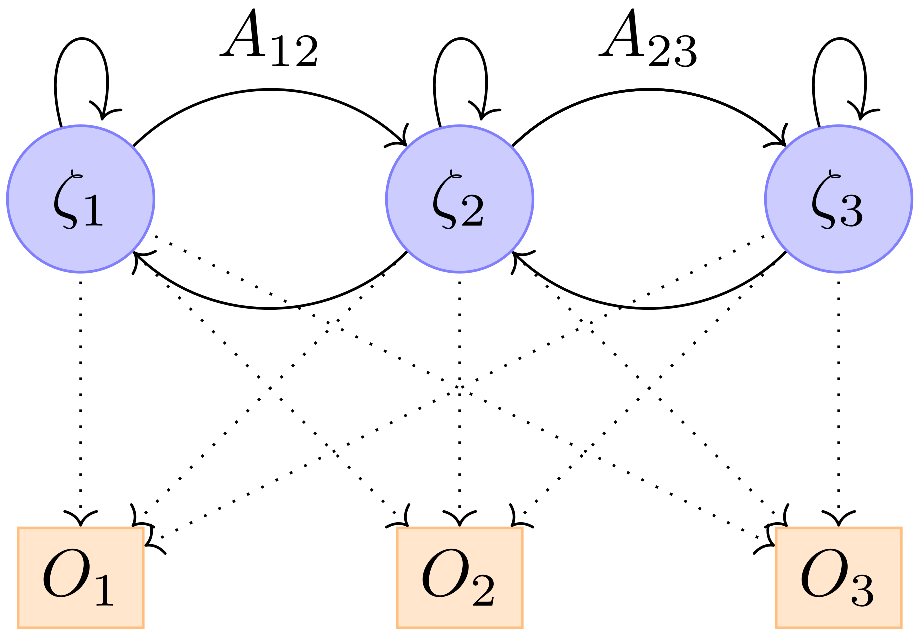

Fig. 4 An HMM with states ζ_i that emit variables O_i, $i = 1, 2, 3$. The transition probability between two states ζ_i and ζ_j is A_{ij}, Eq. 4. This example shows a one-step process in which states can only reach the adjacent state or themselves

the state of the latent process. Specifically, the probability density function of the log-price revision for our trades data set is

$$f_{\Xi_i|Z_i=\zeta}(\xi) \stackrel{\text{def}}{=} w_\zeta \delta(\xi) + (1 - w_\zeta) f_N(\xi; 0, \sigma_\zeta), \tag{9}$$

where

1. $w_\zeta \stackrel{\text{def}}{=} \mathbb{P}(\xi_i = 0|Z_i = \zeta)$ is the conditional probability of observing a zero price revision trade given the event that the hidden process is at ζ.
2. $\delta(\xi)$ is the Dirac delta function centered on zero.
3. $f_N(\xi; 0, \sigma_\zeta)$ is the probability density of a normal random variable with mean 0 and standard deviation σ_ζ.

To be able to do any meaningful work with our parametric model, one needs to learn/calibrate all parameters to the data. Therefore, to facilitate discussion, we will denote the set of all parameters that define our discrete-state HMM (Fig. 4) as $\boldsymbol{\theta} \stackrel{\text{def}}{=} \{\boldsymbol{\pi}, \mathbf{A}, \bar{\boldsymbol{\lambda}}, \bar{\mathbf{w}}, \bar{\boldsymbol{\sigma}}\}$, $\bar{\boldsymbol{\lambda}}, \bar{\mathbf{w}}, \bar{\boldsymbol{\sigma}} \in \mathbb{R}^K$, where, for example, $\bar{\boldsymbol{\lambda}}$ is the vector of trade arrival parameters with entries K, one for each hidden state and similarly for the other two vectors. The density of the complete data and the marginal density of the observation data will depend on the vector $\boldsymbol{\theta}$. When we need to be explicit about the density dependence of this vector, we will use the notation $\mathbb{P}(o; \boldsymbol{\theta})$.

The parameter estimation for the HMM is performed using expectation maximization (EM), which iteratively maximizes the log-likelihood of the observation data through expectation and maximization steps. To avoid local maxima, we initialize using k-means clustering and employ multiple parallel solutions with randomized initial conditions, converging when parameters achieve a relative precision of 10^{-6}. The complete derivation of the EM algorithm, including closed-form solutions for the model parameters and the algorithm, is provided in the Supplement.

4 HMM Induced Generative Feature Embeddings

In applying HMMs to financial data, our goal was leveraging model-induced features for their discriminative power, an approach initially proposed by Jaakkola, Jaakkola et al. (1999) and applied in other domains (Bicego et al. 2008; Jebara et al. 2004; Brodersen 2011) but not yet with high-frequency financial data. This section outlines our structured process for deriving the generative-model embeddings.

We employ HMM-induced feature spaces, often referred to as the generative embedding feature space. This approach is twofold: model calibration (HMM parameter estimation) and construction of generative features based on the calibrated model (Bicego 2013; Jaakkola et al. 1999; Bicego et al. 2008). This calibration and feature generation will be performed intra-daily on a day-by-day basis. Note that these embeddings constitute a model-driven dimensionality reduction of the data. We focus on four classes of feature space embeddings derived from the calibrated HMM: state-space embeddings; transition-space embeddings; emission-space embeddings; and information-flow embeddings. These HMM-derived feature embeddings can be interpreted using machine learning based visualization methods. We explore visualizations and the characterization of the features in the Supplement, in Section 4.

For a detailed overview of the algorithms and analysis of their computational complexity of each component (SSE: $\mathcal{O}(T \times K^2)$, TSE: $\mathcal{O}(T \times K^2)$, ESE: $\mathcal{O}(T \times K)$), see Section 8 of the Supplement.

4.1 State-Space Embedding

The state-space embedding (SSE) measures individual states' contribution to the most probable generation of the observation sequence. Specifically, this feature vector describes the frequency (and associated probability) that the state process will pass through a particular state when a specific observation sequence is seen from the model (Bicego et al. 2009). The components of the SSE feature vector are defined as

$$\gamma_{i,\zeta_j} \stackrel{\text{def}}{=} \mathbb{P}\left(Z_i = \zeta_j | O_{0:T}; \boldsymbol{\theta}\right), \tag{10}$$

and are the probability of being in state ζ_j given the observation sequence $O_{0:T}$ (Bicego et al. 2009). Consequently, we define the quantity (Bicego et al. 2008)

$$\widetilde{\gamma}_{\zeta_j} \stackrel{\text{def}}{=} \sum_{i=0}^{T} \gamma_{i,\zeta_j}. \tag{11}$$

Therefore, the quantity $\widetilde{\gamma}_{\zeta_j}$ is the sum over time of (probability) γ_{i,ζ_j} and can be viewed as the predicted number of transitions from the state ζ_j. As a result, it is a natural indicator of the importance of the state in the estimation process $\mathbb{P}\left(O_{0:T}|\boldsymbol{\theta}\right)$. Thus, the HMM is in the state ζ_j, while observing $O_{0:T}$ in the model $\boldsymbol{\theta}$. The state-space embedding can now formally be defined.

Definition 2 State-space embedding:

$$\gamma_{\mathrm{SS}}(O_{0:T};\boldsymbol{\theta}) \stackrel{\mathrm{def}}{=} \left(\sum_{i=0}^{T}\widetilde{\gamma}_{i,\zeta_1},\ldots,\sum_{i=0}^{T}\widetilde{\gamma}_{i,\zeta_K}\right)^{\mathsf{T}} \in \mathbb{R}^K. \tag{12}$$

4.2 Transition-Space Embedding

The transition-space embedding (TSE) evaluates the importance of individual transitions of our chosen generative model, and is therefore similar to Definition 2. Specifically, the characteristic property of the model, with which each transition is utilized in the process of generating the realization of the observation process $o_{0:T}$. The basic variable is a byproduct of the EM algorithm, which is the probability of transitioning from one state to another. The symbol ξ corresponds to the original reference (Bicego et al. 2008).

Definition 3 Transition-space embedding. The transition-space embedding feature $\xi_{\mathrm{TS}} \in \mathbb{R}^{K^2}$ is

$$\xi_{\mathrm{TS}}\,(o_{0:T};\boldsymbol{\theta}) \stackrel{\mathrm{def}}{=} \begin{bmatrix} \sum_{i=0}^{T}\mathbb{P}(Z_i=\zeta_1, Z_{i+1}=\zeta_1|o_{0:T},\boldsymbol{\theta}) \\ \sum_{i=0}^{T}\mathbb{P}(Z_i=\zeta_1, Z_{i+1}=\zeta_2|o_{0:T},\boldsymbol{\theta}) \\ \vdots \\ \sum_{i=0}^{T}\mathbb{P}(Z_i=\zeta_k, Z_{i+1}=\zeta_k|o_{0:T},\boldsymbol{\theta}) \end{bmatrix}. \tag{13}$$

The vector has K^2 components, which can be interpreted as the number of transitions between all pairs of states, conditional on an observed sequence (Martins et al. 2010). Therefore, this is a vector representation of the probabilities of transitioning between different hidden states based on the observed data. Each element of the vector corresponds to the probability of transitioning between a particular pair of states. By analysing the transition-state embedding, one can gain insight into the temporal dynamics of the data and how the different hidden states interact with each other over time.

4.3 Emission-Space Embedding

The emission-space embedding (ESE) is characterised by the sum of the emission probabilities of the model in a particular state. We define such a characteristic property of the model as the sum of emission probabilities in a given state, $\sum_{i=0}^{T}\mathbb{P}(O_i|Z_i=\zeta)$.

Definition 4 Emission-space embedding (Bicego et al. 2009):

$$\boldsymbol{\varepsilon}_{\mathrm{ES}}(O_{i=0}^{T};\boldsymbol{\theta}) \stackrel{\mathrm{def}}{=} \left(\sum_{i=0}^{T}\mathbb{P}(O_i|Z_i=\zeta_1),\ldots,\ \sum_{i=0}^{T}\mathbb{P}(O_i|Z_i=\zeta_K)\right)^{\mathsf{T}}. \tag{14}$$

4.4 Fisher Score and Information Matrix Embeddings

Two additional embeddings popular in dynamic settings, such as state-space models, are the Fisher score and Fisher Information matrix embeddings, Perronnin et al. (2010); Van Der Maaten et al. (2011). The algorithm to calculate these quantities is presented in the Supplement (Section 9.5).

Definition 5 Fisher score and information matrix (Perronnin et al. 2010; Van Der Maaten et al. 2011): We would like to perform the calculations of the following quantities up to time $n = 1, \ldots, T$ on a set of data (a sequence of observations and parameters),

$$\text{Fisher score} \quad \nabla_{\boldsymbol{\theta}} \log \mathbb{P}(o_{1:n}; \boldsymbol{\theta}), \tag{15}$$

$$\text{Information matrix} \quad -\nabla^2_{\boldsymbol{\theta}} \log \mathbb{P}(o_{1:n}; \boldsymbol{\theta}), \tag{16}$$

where $o_{1:n}$ is a sequence of observations.

5 Multi-kernel learning and support vector machines

The proposed HMM-SVM-MKL framework leverages support vector machines (SVMs) and multiple kernel learning (MKL) to perform the classification task. SVMs optimize a decision boundary by minimizing the empirical risk regularized by a penalty term

$$\min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, f(x_i)) + \lambda \|f\|_{\mathcal{H}}, \tag{17}$$

where ℓ is the hinge loss function and λ controls the trade-off between margin maximization and misclassification error. The solution involves identifying the optimal hyperplane defined by

$$d(\mathbf{x}_i, \mathbf{w}, b) = \mathbf{w} \cdot \mathbf{x}_i + b = 0, \tag{18}$$

where $\mathbf{w}$ is the weight vector and b is the bias term. The dual formulation of the optimization problem is given by

$$\max_{\boldsymbol{\alpha}} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i y_i k(\mathbf{x}_i, \mathbf{x}_j) y_j \alpha_j, \tag{19}$$

subject to $\sum_{i=1}^{n} \alpha_i y_i = 0$ and $0 \leq \alpha_i \leq \frac{1}{2n\lambda}$. Here, $k(\mathbf{x}_i, \mathbf{x}_j)$ is the kernel function, allowing nonlinear classification through the kernel trick.

5.1 Kernel Trick

The kernel trick projects data into a higher-dimensional space where it becomes linearly separable. The kernel function computes the inner product in this space,

$$k(\mathbf{x}_i, \mathbf{x}_j) = \langle \boldsymbol{\phi}(\mathbf{x}_i), \boldsymbol{\phi}(\mathbf{x}_j) \rangle_{\mathcal{H}}, \tag{20}$$

where $\boldsymbol{\phi}$ is the feature map. Common kernel functions include radial basis functions (RBF), polynomial kernels, and linear kernels, as detailed in Table 1 in the Supplement.

5.2 Multi-kernel Learning (MKL)

We leverage MKL which enables the automated combination of multiple kernels, rather than relying on a single kernel. Two key challenges are addressed: determining optimal similarity measures across different kernel functions, and effectively integrating information from diverse data sources or modalities. The latter point is of particular relevance for our approach. Careful consideration is needed for the functional form of kernel combination and the optimization framework for learning combination weights. This usually happens through structural risk minimization or similarity-based optimization. We can express the combined kernel as

$$k_\eta(\mathbf{x}_i, \mathbf{x}_j) = \sum_{m=1}^{M} \eta_m k_m(\mathbf{x}_{mi}, \mathbf{x}_{mj}), \tag{21}$$

where $\eta_m \geq 0$, $\sum_{m=1}^{M} \eta_m = 1$ represent the weights.

5.2.1 Computation of the Weights

Optimization-based MKL methods set kernel weights by solving optimization problems that minimize risk or maximize margins. In contrast, kernel alignment approaches use simpler methods based on measuring similarity between kernel matrices and the ideal kernel from label information. We will use both, as the KTA approach is a useful benchmark. The weights η_m represent the relative importance of each kernel, computed using performance metrics or kernel target alignment (KTA). KTA is defined as follows,

Definition 6 Kernel target alignment (Cristianini et al. 2006).

Consider a kernel matrix $\mathbf{K}$ with elements $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$ for a Mercer kernel $k(\langle \cdot, \cdot \rangle)$; then the kernel target alignment (KTA) is

$$F(\mathbf{K}, \mathbf{Y}) = \frac{\langle \mathbf{K}, \mathbf{Y} \rangle}{\sqrt{\langle \mathbf{K}, \mathbf{K} \rangle \langle \mathbf{Y}, \mathbf{Y} \rangle}}, \tag{22}$$

where the covariance matrix $\mathbf{Y} = \mathbf{y}\mathbf{y}^\top$ has element $y_{ij} = y_i y_j = 1$ if both examples belong to the same class, otherwise -1, as the class labels $y_i = \pm 1$. As a baseline mixture weight for the multi-kernel method, the mixture weights for the kernel j are

$$\eta_m = \frac{F(\mathbf{K}_m, \mathbf{Y})}{\sum_{m=1}^{M} F(\mathbf{K}_m, \mathbf{Y})}. \tag{23}$$

While not guaranteeing global optimality of the mixture weights for kernel $k_\eta(\mathbf{x}_i, \mathbf{x}_j)$, this approach provides a computationally efficient baseline that improves prediction accuracy over single kernels. We refer the reader to Section 2 in the Supplement, and to this (partial) list of excellent references on the topic: Cristianini et al. (2006); Lanckriet et al. (2004); Smola and Schölkopf (2004); Gonen and Alpaydin (2011).

6 Automatic Class Label Construction

Supervised learning in high-frequency markets requires constructing meaningful labels as targets. For our second stage, we develop an automated labeling framework using dynamic thresholds based on empirical quantile estimates, building on López de Prado (2018)'s multiple-barrier approach. Our proposed labeling method builds on prior work in state identification and segmentation of time-series data, including persistent state modeling (by penalizing transitions) and probabilistic regime identification using statistical jump models (Nystrup et al. 2020; Aydınhan et al. 2024). While these methods do emphasize latent state modeling and probabilistic segmentation, they rely on either penalizing state jumps for temporal consistency or estimating regime probabilities. We complement this by employing dynamic empirical quantile thresholds that adapt to local volatility dynamics and market conditions, operating on trade time rather than calendar time. We combine multiple time frames through moving averages and exponential weighting to capture both short-term fluctuations and longer-term trends. Unlike the existing methods that focus on latent states, our method generates both binary and multiclass labels using range-based volatility estimators, making it specifically tailored for supervised learning. By grounding regime detection in observable market behavior while maintaining computational efficiency, we offer a method that integrates into machine learning pipelines and offers a scalable solution for high-frequency data. Adaptability in volatile, data-intensive environments is improved while avoiding the need for assumptions about market states. The framework generates labels using smoothed return trends,

$$s_{\mathrm{EWMA}}(t_i) = w x_{t_i} + (1-w) s_{\mathrm{EWMA}}(t_{i-1}), \tag{24}$$

where x_{t_i} is the log return and $w \in [0, 1]$ is the smoothing parameter. Labels $o_{t_i} = -1, 0, 1$ are assigned based on deviations from this trend using a threshold cv_{t_i}, with v_{t_i} as the realized volatility estimator,

$$v_t = \sqrt{\frac{1}{4N \log 2} \sum_{i=1}^{N} \left(\log \frac{p_i^H}{p_i^L} \right)^2}. \tag{25}$$

Two primary labeling strategies are implemented:

- *Point-in-time*: Labels based on threshold crossings relative to previous trade prices
- *Moving-average-based*: Labels derived from deviations from price moving averages

The labeling mechanism generates six label sets through different parameterizations of these strategies, detailed in the Supplement. This approach ensures adaptability across various asset classes and trading frequencies while maintaining robustness to market noise and volatility dynamics (Sirignano and Cont 2019; Krauss et al. 2017; Dixon 2018). Furthermore, we provide an illustrative example of the algorithm in Fig. 5.

We use a range of options for capturing different aspects of market behavior, from short-term fluctuations to longer-term trends, and from binary to multiclass categorizations. The

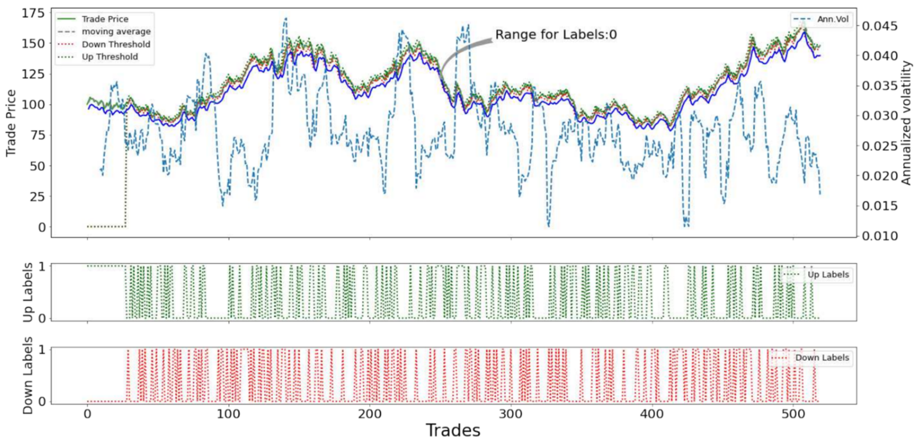

Fig. 5 Illustrative time series with the visualization of some stylized boundaries for the labeling mechanism framework. At every point in the trade clock, we observe what the moving average trade price was, and as long as the current trade price exceeds the empirical quantile threshold, computed using historical volatility, an event that constitutes a label is constructed. The binary label indicator function can be seen with the green and red labels

Table 1 Parameter settings for each label in the experiments

Label	Type	Lookback window	Updown threshold
One	Binary point in time	40	0.05%
Two	Binary point in time	8	0.09%
Three	Multiclass point in time	24	0.05%
Four	Binary point in time	9	0.41%
Five	Multiclass point in time	15	0.10%
Six	Binary point in time	15	0.03%

The updown threshold is used to trigger the labeling mechanism. This table presents the experimental parameter settings for six different labels (one to six) in the context of financial time-series analysis. Each row represents a distinct label, and the columns provide information on the label's type, lookback window, and updown threshold expressed as a percentage

choice of parameterization depends on the specific trading objectives, risk tolerance, and market conditions of interest. These are provided in Table 1.

7 Data and Features

This section presents an overview of the raw high-frequency trading data from 40 FTSE 100 stocks and a set of analyses for the features, establishing the empirical foundation for our experiments. Building on the raw data, we demonstrate that HMM-based generative embeddings effectively capture market regimes and outperform traditional market features, particularly during high-volatility periods.

7.1 Data Description

The data consists of millisecond resolution quotes and trades for 40 FTSE 100 stocks over the 2017 calendar year, with time stamps featuring 10^{-7} precision sourced from Reuters. The raw data includes trades and (top of the order book) quotes (TAQ), OHLC information, with key variables comprising time stamps, trade prices, bid/ask, and quoted volumes, enabling intra-day microstructure modeling of inter-trade durations and price changes. To construct the features, models were fitted separately for each symbol and day to generate embedding features, with the kernel classifiers built on top. Table 2 compiles an aggregate summary of the

Table 2 Summary of data for all 40 symbols for all trading days of 2017

Metric	Median
Duration/milliseconds	0.71
Standard deviation of the log return	0.46
Number of trades in a day	2115
Number of traded shares	717.25

Duration is the difference between the calendar time of sequential trades. The table summarizes key trading statistics, such as the duration in milliseconds, the standard deviation of logarithmic return, the number of trades per day, and number of traded shares

40 assets throughout 2017. The data exhibit significant cross-stock variability in key descriptors, highlighting diverse market microstructure dynamics. This spans multiple dimensions: mean durations (0.41 seconds for CCL.L to 1.29 for AV.L), daily trades (789.5 for APF.L to 5,493.0 for BLT.L), return volatility (0.40 for SDR.L to 0.55 for AZN.L), and traded volume (predominantly 900.00, with exceptions like BLT.L at 509.00 and CCL.L at 500.00).

7.1.1 Choice of Symbols to Illustrate the Methodology

Our analysis centers on five representative FTSE 100 stocks (APF.L, CCL.L, CPG.L, RBS.L, and KGF.L) chosen across basic materials, consumer goods, banking, and retail sectors. These stocks exhibit diverse microstructure properties, exemplified by CCL.L's fast trading dynamics (0.41-second durations) versus APF.L's slower patterns (0.56 seconds), enabling evaluation of our framework across varying market regimes. This difference in liquidity, volatility, and trading characteristics, supplemented by features from additional FTSE 100 symbols, provides insights into sector-specific influences on high-frequency strategies, feature engineering, and trading performance (Chaboud et al. 2014).

7.1.2 Performance Metrics

In the machine learning literature, a classifier is evaluated with precision (positive predictive value) and recall (sensitivity), which we define below (Takahashi et al. 2022; Santafe et al. 2015). No single metric fully captures the efficacy of the model (Manning et al. 2008), but together these metrics are well-suited for evaluating multi-class imbalance problems. Our goal is to create a pragmatic and practical framework, the experimental assessment becomes an integral part of our efforts. However, there is no precise recipe for the evaluation of classification algorithms (Takahashi et al. 2022). The relevant details and formulation can be found in Manning, Manning et al. (2008).

1. *Precision* is the ratio of true positives to the sum of true positives and false positives. Similarly, the *recall* gauges the model's ability to correctly identify true positives.
2. Beyond these individual metrics, the f_1-*score* offers a harmonized evaluation criterion, taking into account the trade-offs between precision and recall. To accommodate the complexity of multiclassification problems, variations such as the *macro*-f_1, *micro*-f_1, and the *weighted* f_1 score are employed. The latter, in particular, modifies the *macro* version to weight out label imbalance.
3. In multilabel classification scenarios, the *Hamming loss* emerges as an important metric. It accounts for the number of instances-label pairs that are misclassified, providing a single comprehensive measure of performance across multiple labels (Gao and Zhou 2013). The experimental results and performance metrics are presented in Tables 3, 4, 5, 6, 7, 8 and 9.

7.2 Feature Extraction and Analysis

Our feature analysis methodology consists of three complementary approaches: empirical feature characterization, stability validation and comparative analysis.

The empirical analysis employs statistical and visual techniques to understand feature relationships and discriminative power. We use Pearson correlation coefficients to quantify linear dependencies between features, while Shapiro-Wilk tests assess their distributional

Table 3 Mean duration, number of trades, mean return standard deviation of logarithmic price change, and mean traded volume for each stock

Stock ticker	Mean duration	Mean nr. of trades	Mean stdev of returns	Mean traded volume
AAL.L	0.50	845	0.53	900.00
APF.L	0.56	790	0.55	900.00
AV.L	1.29	845	0.53	900.00
AZN.L	0.58	1487	0.55	900.00
BARC.L	0.75	1778	0.54	900.00
BATS.L	0.54	4488	0.48	900.00
BLT.L	0.48	5493	0.43	509.00
CCL.L	0.41	4576	0.46	500.00
CEY.L	0.49	2371	0.49	510.00
CPG.L	0.46	1768	0.46	600.00
CPI.L	0.49	1688	0.49	679.25
ITV.L	0.53	1653	0.50	717.25
KGF.L	0.50	1713	0.48	750.00
LAND.L	0.50	1652	0.48	720.50
LGEN.L	0.54	1713	0.47	750.00
LLOY.L	0.60	1780	0.46	900.00
MAB.L	0.70	1717	0.47	750.00
MKS.L	0.76	1780	0.47	900.00
NG.L	0.78	1934	0.46	753.50
PRU.L	0.75	2153	0.45	750.00
PSON.L	0.71	2082	0.46	752.62
RB.L	0.71	2163	0.45	750.00
RBS.L	0.72	2193	0.45	753.50
RDSa.L	0.71	2260	0.44	748.50
RDSb.L	0.73	2314	0.43	695.75
REL.L	0.71	2284	0.41	674.62
RR.L	0.69	2209	0.42	700.00
RTO.L	0.71	2188	0.41	706.50
RSA.L	0.51	811	0.50	900.00
SDR.L	0.72	2165	0.40	700.00
SGE.L	0.71	2139	0.41	706.50
SHP.L	0.69	2165	0.42	700.00
SMIN.L	0.71	2115	0.41	674.62
SPT.L	0.73	1883	0.42	677.00
STAN.L	0.75	2115	0.41	663.00
TSCO.L	0.76	2165	0.41	676.00
ULVR.L	0.74	2198	0.40	659.75
UU.L	0.77	2165	0.41	641.00
VOD.L	0.74	2198	0.40	661.00

All data are from 2017. All symbols were used in the experiments, but only a subset of the results are presented in this main body

Table 4 The table lists 12 symbols

Symbol	Accuracy	Precision	Recall	f_1-weighted	f_1-micro	f_1-macro	Hamming loss
AAL.L	0.03	0.21	0.07	0.08	0.03	0.12	-0.03
APF.L	0.02	0.06	0.03	0.04	0.02	0.04	-0.02
AV.L	0.01	0.20	0.04	0.06	0.01	0.08	-0.01
AZN.L	0.03	0.27	0.09	0.07	0.03	0.13	-0.03
BARC.L	0.00	0.18	0.02	0.03	0.00	0.05	0.00
BATS.L	0.06	0.26	0.10	0.09	0.06	0.14	-0.06
BLT.L	0.05	0.25	0.11	0.09	0.05	0.15	-0.05
CCL.L	0.09	0.36	0.23	0.14	0.09	0.28	-0.09
CEY.L	0.02	0.18	0.07	0.08	0.02	0.11	-0.02
CPG.L	0.05	0.28	0.15	0.12	0.05	0.20	-0.05
ITV.L	0.01	0.15	0.04	0.05	0.01	0.07	-0.01
KGF.L	0.02	0.23	0.09	0.06	0.02	0.14	-0.02

Differences between generative embeddings and market features across key metrics (generative-market) are presented. Positive values indicate that generative embeddings outperform market features, while negative values indicate the opposite. For Hamming loss, negative values are favourable since a lower Hamming loss indicates fewer prediction errors

properties. To visualize high-dimensional feature interactions, we employ RadViz for revealing feature-label relationships through radial coordinate mapping, and t-SNE for analyzing local feature similarity structures through nonlinear dimensionality reduction.

To validate feature stability, we conduct extensive synthetic experiments using an HMM calibrated to match typical market characteristics. The synthetic data generation process employs a two-state HMM with transition probabilities and emission distributions estimated from historical market data. We generate multiple price series of increasing lengths (1,000 to 100,000 observations) to assess convergence properties. For each series, we compute the full set of generative embedding features and analyze their statistical properties across different sample sizes and market regimes. The experiments demonstrate that our features exhibit strong stability properties: mean values converge within 5,000 observations, while variance estimates stabilize within 10,000 observations - equivalent to approximately one trading day

Table 5 Summary of methods used, their roles, and performance characteristics

Method	Role	Performance description
Logistic regression	Baseline	Standard binary and multiclass classification with one-vs-rest extension
Feed-forward network (FFN)	Baseline	Deep learning model capturing non-linear relationships, adaptive learning rate
Single-kernel SVM	Benchmark	Traditional SVM trained with all chosen features
Kernel target alignment (KTA)	Benchmark	Utilizes kernel target alignment
Multi-kernel learning (MKL)	Best-performing method	Deploys 11 distinct polynomial kernels

Table 6 Logistic regression performance across market regime labels

Label	Accuracy	Precision	Recall	f_1-weighted	f_1-micro	Hamming loss
One	0.45	0.42	0.45	0.43	0.45	0.55
Two	0.42	0.39	0.42	0.40	0.42	0.58
Three	0.31	0.28	0.31	0.29	0.31	0.69
Four	0.48	0.45	0.48	0.46	0.48	0.52
Five	0.35	0.32	0.35	0.33	0.35	0.65
Six	0.31	0.29	0.31	0.30	0.31	0.69

of tick data. The data study validates that our generative embeddings exhibit rapid asymptotic convergence to consistent estimates, typically within single trading day timescales. This stability in feature estimation supports their applicability for capturing market microstructure dynamics in real financial time series applications. The experimental protocol and detailed convergence analysis are provided in the Supplement.

Finally, we evaluate the predictive value of our generative embedding features against traditional market indicators through a comparative analysis using logistic regression models across multiple stock symbols.

7.2.1 Analysis of the HMM-Based Generative Feature Embeddings

We analyze the generative features extracted from real market data to identify and remove potential collinear structures that could impair out-of-sample classification accuracy. Our analysis examines feature separability (Wang and Suter 2008; Zhang et al. 2017) through two approaches: correlation analysis of feature embeddings computed day-by-day for each asset, and multi-dimensional visualization techniques to explore relationships between features and target labels (Liu and Wang 2016).

Our objectives include ranking individual features and feature pairs by covariance and applying RadViz and t-SNE for separability analysis. The same features and labels are used as in the hybrid model, without kernelization for RadViz and t-SNE. This reveals links between classification accuracy and separability, justifying our multi-kernel approach, as linear or single kernel classifiers cannot sufficiently separate all class pairs based on their geometry (Makowiec 2017).

Table 7 Feed-forward neural network performance across market regime labels.

Label	Accuracy	Precision	Recall	f_1-weighted	f_1-micro	Hamming loss
One	0.97	0.49	0.97	0.96	0.97	0.03
Two	0.85	0.45	0.85	0.82	0.85	0.15
Three	0.45	0.22	0.45	0.41	0.45	0.55
Four	0.65	0.31	0.65	0.62	0.65	0.35
Five	0.35	0.18	0.35	0.32	0.35	0.65
Six	0.10	0.08	0.10	0.09	0.10	0.90

Table 8 Kernel target alignment (KTA) performance metrics across six market regime labels

Label	Accuracy	Precision	Recall	f_1-weighted	f_1-micro	f_1-macro	Hamming loss
1	0.81	0.96	0.81	0.85	0.81	0.47	0.19
2	0.71	0.74	0.71	0.72	0.72	0.47	0.29
3	0.33	0.40	0.33	0.32	0.33	0.27	0.68
4	0.71	0.74	0.71	0.72	0.72	0.47	0.29
5	0.33	0.40	0.33	0.32	0.33	0.27	0.68
6	0.42	0.47	0.42	0.40	0.42	0.30	0.58

Results show strong performance for Labels 1 and 2 (high volatility regimes) but significant degradation for more complex market states (Labels/Regimes 3-6). Performance metrics include accuracy, precision, recall, various f_1 scores, and Hamming loss, averaged across all stocks and out-of-sample test periods

RadViz RadViz projects multidimensional data onto a 2D plane through a unit circle representation (Hoffman et al. 1997), where variables are anchored equidistantly and instances are positioned by spring forces proportional to their normalized [0, 1] coordinate values. This visualization reveals feature correlations and dependencies, with central points indicating similar values across contrasting dimensions and peripheral points showing dominance of specific features (Patrick et al. 1999). The RadViz analysis reveals distinct class/regime separation patterns when combining volatility-based the HMM features with transition probability embeddings. State transition features ($\xi_{0\to1}$, $\xi_{2\to0}$) demonstrate strong discriminative power through distinct radial clustering, while volatility indicators (Fischer score of δ weight and of $\delta\lambda$) exhibit broader dispersion, capturing complementary instantaneous volatility information. The opposition between transition probability features and emission distribution parameters (γ_0, λ_{weight}) characterizes market regime changes, with compact clusters in specific market conditions demonstrating shared feature interactions. Features like $\xi_{1\to0}$ bridge multiple market regimes, while dispersed features excel at outlier identification. A detailed feature-wise selection of RadViz plots is available in the Supplement.

t-SNE

t-SNE is a technique for embedding high-dimensional data in lower dimensions for visualization (Van Der Maaten and Hinton 2008). It models similarities between data points in both high and low-dimensional spaces, minimizing KL divergence between distributions with gradient descent. This non-linear transformation reveals structure while preserving local

Table 9 Multi-kernel learning (MKL) classification performance across market regime labels

Label	Accuracy	Precision	Recall	f_1-weighted	f_1-micro	f_1-macro	Hamming loss
1	0.98	0.96	0.98	0.97	0.98	0.49	0.02
2	0.91	0.84	0.91	0.87	0.91	0.48	0.09
3	0.50	0.50	0.50	0.46	0.50	0.33	0.50
4	0.84	0.70	0.84	0.77	0.84	0.46	0.16
5	0.79	0.62	0.79	0.69	0.79	0.44	0.21
6	0.73	0.55	0.73	0.61	0.73	0.39	0.27

The approach demonstrates robust performance across all labels, with particularly strong results for labels 1 and 2 (0.98 and 0.91 accuracy respectively). Results are averaged across all 40 FTSE stocks and out-of-sample test periods, showing consistent performance even in complex market states (labels 5 and 6) relationships. We apply t-SNE to the raw features and kernelized features, exposing the value of our kernelization approach.

The t-SNE visualizations of the kernelized features reveal three primary clustering structures: high volatility states form distinct clusters in the outer regions of the projection space, while moderate and low volatility states show more nuanced separation in the central regions. In the three-dimensional t-SNE space, we observe that the first component (t-SNE 1) primarily captures the separation between extreme market states, while the second and third components (t-SNE 2, t-SNE 3) reveal finer granularity in the feature relationships, particularly in transition periods between volatility regimes. These nonlinear projections, provided in the Supplement, demonstrate how our kernel transformations improve the separability of market states while preserving the continuous nature of regime transitions, with notable clustering occurring at scales between -400 and 600 in the primary components.

7.2.2 Market Features for the Benchmarking

We construct sets of features based on the indicators moving average spread, on-balance volume (OBV) and Chaikin money flow (CMF). The set of moving average spreads identifies trends based on price momentum over different time frames. It can signal potential reversals when short-term price movements deviate significantly from a more long term trend. The OBV adds a volume dimension to these set of features, helping confirm the strength of the detected trend. For instance, if the price is trending upward and OBV is also rising, it suggests that there is significant buying pressure supporting the trend. CMF confirms trend strength by measuring volume-weighted accumulation and distribution over a specified period. A positive CMF during an uptrend indicates buying pressure, while a negative CMF during a downtrend signals selling pressure. All formulas and implementation are detailed in the Supplemental material.

7.3 Do the Generative Features Add Value?

The initial comparison involves testing on an unknown testing subset after evaluating the performance of generative embeddings and market features in a logistic regression model using a training subset. This comparison aims to determine which type of feature (generative embeddings or market features) performs better in predicting the outcome variable in a logistic regression model. We find that generative embeddings outperform market features in predicting the outcome variable in a logistic regression model based on their higher accuracy and lower error rates. We present the results for 13 symbols. We calculate various performance metrics (such as the classification report, which provides precision, recall, and f_1 scores and the overall accuracy).

This specific experiment includes splitting the dataset into training and testing subsets for model training and evaluation. After training on the training subset, the logistic regression model undergoes performance evaluation on the testing subset. Similar, to the rest of the experiments, this assessment comprises calculating various performance metrics such as precision, recall, f_1 scores, and overall accuracy in the classification report. In this comparison of two types of features, the first type is generative embeddings, and the second type is market features.

Specifically, generative embeddings demonstrate higher accuracy (e.g., AAL.L: 0.71 vs. 0.68), f_1 macro (e.g., APF.L: 0.63 vs. 0.59), f_1 micro (e.g., CPG.L: 0.72 vs. 0.67), and f_1 weighted scores (e.g., KGF.L: 0.75 vs. 0.69), along with superior precision and recall (e.g., AZN.L: precision 0.60 vs. 0.34, recall 0.51 vs. 0.42) and lower Hamming loss (e.g., BATS.L: 0.19 vs. 0.25). These findings suggest that the generative embeddings are more reliable and effective, providing a more balanced prediction performance and fewer errors.

8 Model Performance Results and Discussion

We evaluate our HMM-based generative embeddings through three increasingly sophisticated approaches: baseline models (logistic regression, FFN) previously validated in financial prediction tasks (Dixon et al. 2017; Dixon 2018; Gu et al. 2020), single-kernel method (SVM) (The results of the single method can be found in the Supplement, Tables 1 & 1.), and multi-kernel approaches (KTA as in Eq. 22, and our proposed MKL algorithm) utilizing 11 distinct polynomial kernels. The experimental framework employs cross-validation for robust out-of-sample testing, with particular attention to performance during varying market conditions.

All experiments are conducted on the FTSE 100 stock data (described in Section 7.1), with summaries of the performance metrics provided.

8.1 Baseline Models Analysis

We establish baseline performance through two widely-used approaches in financial prediction tasks: logistic regression and feed-forward neural networks (FFN). These methods serve as fundamental benchmarks against which we evaluate our kernel-based approaches.

8.1.1 Logistic Classification

For binary classification tasks, we implement logistic regression with a standard 0.5 decision threshold, balancing sensitivity and specificity. The multiclass extension employs a one-vs-rest (OvR) approach, where separate models are fitted for each class against all others. This approach maintains consistent decision criteria while addressing the inherent multi-regime nature of market states. Performance varies significantly across the six market regime labels, with accuracy ranging from 31.32% to 48.01%. Label/Regime Four achieves the highest accuracy, while Label/Regime Six shows the poorest performance. The large variation in performance metrics (detailed in Table 6) reveals the challenges of capturing complex market dynamics through linear classification (Tables 7, 8 and 9).

8.1.2 Feed-Forward Neural Network

Following validated architectures (Dixon et al. 2017; Dixon 2018), we implement a sequential FFN model with an input layer matched to feature dimensionality, two dense hidden layers (64 and 32 neurons) with ReLU activation, a sigmoid output layer, and Adam optimizer using binary cross-entropy loss.

Despite achieving high accuracy, the FFN model demonstrates key limitations: frequent false positives (48.62% precision), substantial performance deterioration for minority classes, and poor generalization to complex market states (evidenced by Label Six's low 9.54% accuracy and high 90.46% Hamming loss).

These baseline experiments uncover fundamental challenges in market regime classification. While linear methods struggle to handle complex market dynamics, neural architectures have shown promise but lack consistency, and traditional approaches often fail to maintain performance across all regimes, particularly when contending with class imbalances.

8.2 Kernel-Based Methods

Building on the baseline results, we evaluate three kernel approaches (progressively): single-kernel SVM, kernel target alignment (KTA), and multi-kernel learning (MKL). This progression allows assessment of how generative embeddings perform with increasingly complex learning frameworks.

8.2.1 Single Kernel Experiments

For the single-kernel implementation, we apply SVMs using both RBF and polynomial kernels to predict trade direction. The experimental framework employs 5-fold cross-validation to select optimal parameters, with C values ranging from 10^{-1} to 10^{2} for RBF kernels and degrees $\nu \in 1, \ldots, 11$ for polynomial kernels. This cross-validation search exhaustively iterates over parameter combinations, with RBF kernels exploring α values from 10^{-5} to 10^{-1}.

8.2.2 Kernel Target Alignment and Multi-Kernel Approach

The KTA approach demonstrates improved classification capability over single-kernel methods, achieving 55% accuracy and 62% precision across all labels. However, performance varies significantly across market regimes, particularly during high-volatility periods. This variability suggests that while KTA effectively aligns kernel features with target labels, it may not fully capture the complexity of market state transitions. The MKL framework, employing 11 distinct polynomial kernels, consistently delivers superior performance (79% accuracy, 70% precision). This improvement stems from MKL's ability to adaptively combine multiple kernel representations, effectively capturing both linear and nonlinear relationships in the HMM-derived features. The framework's strength lies in its ability to maintain performance across different market conditions, making it particularly valuable for real-world applications.

8.2.3 Multi-Kernel Learning Framework

The MKL implementation utilizes 11 distinct polynomial kernels, optimizing their combination through individual kernel performance weighting. This approach demonstrates consistently superior performance (across all metrics and market regimes).

The MKL framework shows particular strength in handling complex market regimes, maintaining robust performance even in challenging conditions. For Label 1, it achieves 98% accuracy with 0.96 precision, while maintaining above 70% accuracy even for the more challenging Labels 5 and 6. This consistent performance across different market conditions highlights the framework's ability to capture and adapt to varying market dynamics. Each kernel method demonstrates progressive improvement in handling market microstructure complexities, with MKL showing the most robust and consistent performance. The results validate our approach of combining multiple kernels to capture different aspects of market behavior, particularly evident in the handling of complex market regimes.

8.2.4 Implementation Considerations

Each symbol exhibits distinct characteristics in terms of volume traded, average trade arrival times, and price volatility. Our approach of fitting and training models daily, then applying the learned classifiers out-of-sample, offers two key advantages: it requires relatively small training sets and emphasizes kernel quality over the classifier choice in determining final model accuracy. This methodology proves especially valuable in high-frequency trading environments where rapid adaptation to changing market conditions is crucial.

8.3 Comparative Analysis Across Methods

To systematically evaluate the relative performance of all methods, we present a unified comparison across baseline and kernel approaches. Table 10 provides this comprehensive comparison.

This comparison reveals key insights. First, there is a clear progression in performance from simple to more sophisticated methods, with each kernel approach showing improvement over baseline methods. Second, while FFN shows strong initial improvement over logistic regression (accuracy increase from 41% to 64%), it struggles with precision (32%), indicating potential overfitting. Finally, MKL demonstrates superior performance across all metrics, with statistically significant improvements over both baseline and other kernel methods. This comparison naturally leads us to examine performance variations across different market regimes-labels in more detail. The following section shows how each method performs under varying market conditions, with particular attention to the stability and consistency of predictions in complex market states.

8.4 Label-Specific Analysis Across Methods

The relative performance of each method varies significantly across different market regimes/labels. We analyze performance patterns across all six, with particular attention to how methods handle increasing regime complexity (Fig. 6).

MKL demonstrates notably strong performance for Labels One and Two (accuracies of 0.98 and 0.91 respectively), indicating robust classification of clear market regimes. Even for more complex states (Labels Three through Six), MKL maintains superior performance, with accuracy never falling below 0.50. In contrast, KTA shows strong initial performance (0.81 accuracy for Label One) but degrades more rapidly for complex market states, falling to 0.33 accuracy for Labels Three and Five. The baseline methods show a more severe performance

Table 10 Performance comparison across all classification methods

Model	Accuracy	Precision	Recall	f_1-weighted	f_1-micro	f_1-macro	Hamming loss
Logistic	0.41	0.49	0.42	0.47	0.41	0.35	0.59
FFN	0.64	0.32	0.42	0.55	0.64	0.32	0.36
KTA	0.55	0.62	0.55	0.56	0.55	0.38	0.45
MKL	**0.79**	**0.70**	**0.79**	**0.73**	**0.79**	**0.43**	**0.21**

Results demonstrate progressive improvement from baseline to kernel methods, with MKL showing statistical superiority ($p < 0.05$ compared to logistic baseline) across all metrics. Metrics are averaged across all labels and symbols for out-of-sample predictions, with best results highlighted in bold. Results shown are averages; for detailed statistics including median, standard deviation, and min/max values, see Supplement Section 13

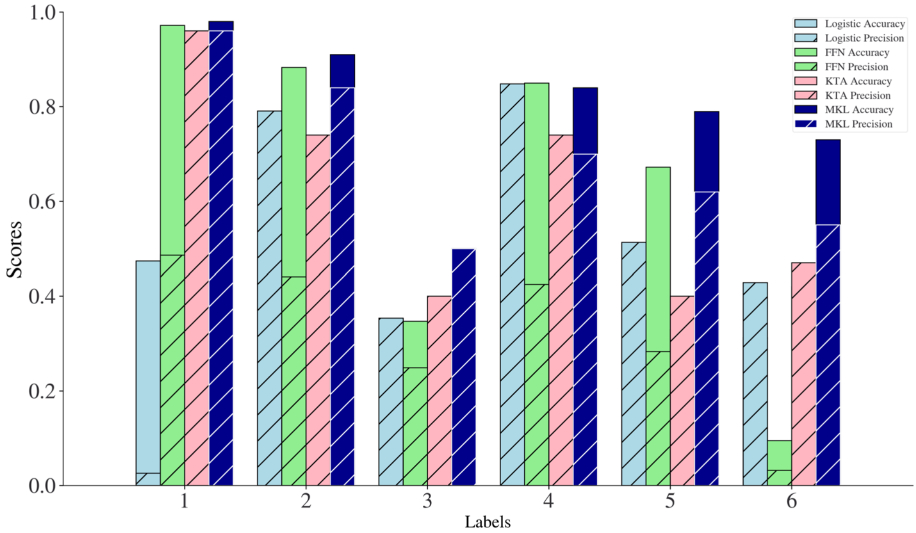

Fig. 6 Comparative classification performance across all methods and labels. Solid bars represent accuracy while diagonally-hatched bars show precision. Methods are color-coded: Logistic regression (light blue), FFN (green), KTA (pink), and MKL (dark blue). Performance metrics range from 0 to 1 on the y-axis. Results demonstrate MKL's superior and more consistent performance across all labels, particularly in complex market regimes (labels 3-6)

deterioration across labels. While FFN achieves competitive accuracy for Labels One and Two (0.97 and 0.85), its performance drops dramatically for complex regimes, reaching just 0.10 accuracy for Label Six. Logistic regression shows consistent but poor performance across all labels, never exceeding 0.48 accuracy. This label-specific analysis reveals that method performance correlates strongly with market regime complexity, with MKL maintaining the most stable performance across all regime types while traditional approaches struggle with market state complexity. Though simpler methods may suffice for basic market states, our findings demonstrate that robust classification of complex market regimes requires multi-kernel-based approaches.

9 Conclusion

Our work presents a hybrid machine learning methodology (HMM-SVM-MKL) that combines multiple kernel learning (MKL), the HMM-induced features, and an automated flexible labeling algorithm for advanced financial time series classification. We demonstrate its effectiveness in classifying trade directionality and short-term high-frequency regimes for FTSE 100 stocks. This combination is unique and marks a significant improvement for algorithmic trading and high-frequency time series applications by enhancing prediction accuracy and capturing complex market dynamics.

A key advantage of our approach is the ability to handle various types of financial data, such as traded prices, volumes, and intertrade durations, which can be beneficial for modeling state-dependent trading. The HMM-derived generative embeddings effectively leverage distributional changes in prices and durations to identify regimes associated with different market participants' behaviors. This type feature extraction provides valuable information for developing adaptive trading strategies. Empirical comparisons with deep learning alternatives reveal two important advantages. First, our methodology achieves comparable or superior performance with substantially lower computational requirements, making it particularly suitable for real-time applications. Second, the framework requires significantly smaller training datasets, addressing a key limitation of deep learning approaches in practical trading applications. These benefits are demonstrated through extensive evaluations using unbalanced FTSE data sets, where the MKL model shows consistent advantages in both binary and multiclass classification problems.

The framework's effectiveness stems from its ability to fuse information from multiple kernels, addressing the inherent complexity of high-frequency financial data. Our hybrid approach delivers robust performance while requiring far less training data compared to deep learning methods, making it more practical for real-world implementations. This efficiency in handling market microstructure analysis enables trading systems to recognize and adapt to changing market conditions dynamically. Although the methodology provides significant practical value to traders through well-informed decision-making capabilities amid changing regimes, it faces limitations in model complexity and interpretability. These challenges, however, are outweighed by the substantial contributions to algorithmic trading and the framework's importance in understanding market dynamics. Future research directions should focus on three key areas: enhancing model transparency to address interpretability challenges, evaluating additional data types beyond traditional market metrics, and exploring alternative kernel learning approaches for specific market conditions. This work represents an essential step toward more responsive and insightful modeling of complex financial markets, while acknowledging the ongoing need for continued refinement and extension of these techniques.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s11009-025-10148-8.

Acknowledgements We are grateful to Konstantinos Manolarakis for his contribution in an intermediate phase of the project.

Author Contributions A.K. had the original idea, did the coding, figures, tables and most of the writing. All authors contributed to the overall research project.

Funding The authors did not receive specific financial support for this work from any organization.

Data Availability No datasets were analysed during the current study.

Declarations

Competing interests The authors declare no competing interests.

References

Aiolli F, Donini M (2015) EasyMKL: a scalable multiple kernel learning algorithm. Neurocomputing 169:215–224

Aldrich E, Heckenbach I, Laughlin G (2016) A compound duration model for high-frequency asset returns. J Empir Financ 39:105–128

Andersen TG (1996) Return volatility and trading volume: an information flow interpretation of stochastic volatility. J Finance 51:169–204

Ang A, Timmermann A (2012) Regime changes and financial markets. Annu Rev Financ Econ 4:313–337

Aydınhan AO, Kolm PN, Mulvey JM et al (2024) Identifying patterns in financial markets: extending the statistical jump model for regime identification. Annal Oper Res 1:1

Bach FR, Lanckriet GRG, Jordan MI (2004) Multiple kernel learning, conic duality, and the SMO algorithm, ICML '04, 6–14. Association for Computing Machinery, New York, NY, USA

Barndorff-Nielsen OE, Hansen PR, Lunde A, Shephard N (2009) Realized kernels in practice: trades and quotes. Econ J 12:C1–C32

Bauwens L, Veredas D (2004) The stochastic conditional duration model: A latent variable model for the analysis of financial durations. J Econ 119:381–412

Bertolini R, Finch SJ (2022) Stability of filter feature selection methods in data pipelines: a simulation study. Int J Data Sci Anal 17:225–248

Bicego M et al (2013) Combining information theoretic kernels with generative embeddings for classification. Neurocomputing 101:161–169

Bicego M, Pekalska E, Tax DM, Duin RP (2009) Component-based discriminative classification for hidden Markov models. Pattern Recogn 42:2637–2648

Bicego M, Grosso E, Otranto E (2008) A hidden Markov model approach to classify and predict the sign of financial local trends. In: da Vitoria Lobo N, Georgiopoulos K, Roli M, Kwok F, Anagnostopoulos J (eds) Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol 5342, pp 852–861. Springer, Berlin

Bicego M, Grosso E, Otranto E (2008) Recognizing and forecasting the sign of financial local trends using hidden markov models. Working paper, CRENoS, University of Cagliari

Breneman J (2005) Kernel methods for pattern analysis. Technometrics 47:237–237

Brodersen KH et al (2011) Generative embedding for model-based classification of FMRI data. PLoS Comput Biol 7:1–19

Carli A, Bicego M, Baldo S, Murinoy V (2010) Nonlinear mappings for generative kernels on latent variable models, 2134–2137. IEEE Computer Society, Los Alamitos

Carranza-García M, Torres-Mateo J, Lara-Benítez P, García-Gutiérrez J (2021) On the performance of one-stage and two-stage object detectors in autonomous vehicles using camera data. Remote Sens 13:89

Cartea Á, Jaimungal S (2013) Modelling asset prices for algorithmic and high-frequency trading. Appl Math Finance 20:512–547

Cartea Á, Jaimungal S (2015) Risk metrics and fine tuning of high-frequency trading strategies. Math Financ 25:576–611

Chaboud AP, Chiquoine B, Hjalmarsson E, Vega C (2014) Rise of the machines: algorithmic trading in the foreign exchange market. J Finance 69:2045–2084

Coates A, Ng AY (2012) Learning feature representations with K-means. In: Montavon G, Orr GB, Müller K-R (eds) Neural Networks: tricks of the Trade, vol 7700. Lecture Notes in Computer Science, pp 561–580. Springer, Berlin, Heidelberg

Cont R (2001) Empirical properties of asset returns: Stylized facts and statistical issues. Quantitative Finance 1:223–236

Cristianini N, Kandola J, Elisseeff A, Shawe-Taylor J (2006) On kernel target alignment. In: Dietterich T, Becker S, Ghahramani Z (eds) Studies in Fuzziness and Soft Computing, vol 194, pp 205–256. Springer, Berlin

Dacorogna MM, Gençay R, Müller UA, Olsen RB, Pictet OV (2001) An Introduction to High-Frequency Finance. Academic Press, San Diego

Derakhshan B, Mahdiraji AR, Rabl T, Markl V (2019) Continuous deployment of machine learning pipelines. In: Herschel M, et al (eds), International Conference on Extending Database Technology (EDBT-2019). (OpenProceedings, Konstanz)

Dixon M (2018) Sequence classification of the limit order book using recurrent neural networks. J Comput Sci 24:277–286

Dixon M, Klabjan D, Bang JH (2017) Classification-based financial markets prediction using deep neural networks. Algorithmic Finance 6:67–77

Easley D, De Prado ML, O'Hara M (2012) The volume clock: insights into the high-frequency paradigm. J Portf Manag 39:19–29
Engle RF, Russell JR (1998) Autoregressive conditional duration: a new model for irregularly spaced transaction data. Econometrica 66:1127
Filimonov V, Sornette D (2015) Apparent criticality and calibration issues in the Hawkes self-excited point process model: application to high-frequency financial data. Quantitative Finance 15:1293–1314
Fletcher T, Shawe-Taylor J (2013) Multiple kernel learning with Fisher kernels for high frequency currency prediction. Comput Econ 42:217–240
Fletcher T, Hussain Z, Shawe-Taylor J (2010) Multiple kernel learning on the limit order book, vol 11, pp 167–174. JMLR Workshop and Conference Proceedings (Microtome, Brookline, Mass, USA)
Gao W, Zhou Z-H (2013) On the consistency of multi-label learning. Artif Intell 199–200:22–44
Ghahramani Z (2001) An introduction to hidden Markov models and Bayesian networks. Int J Pattern Recognit Artif Intell 15:9–42
Gönen M, Alpaydin E (2011) Multiple kernel learning algorithms. J Mach Learn Res 12:2211–2268
Gonen M, Alpaydin E (2011) Multiple kernel learning algorithms. J Mach Learn Res 12:2211–2268
Gong X et al (2020) Multiple kernel SVM based on two-stage learning. IEEE Access 8:101133–101144
Gu S, Kelly B, Xiu D (2020) Empirical asset pricing via machine learning. Rev Financial Studies 33:2223–2273
Haiying X, Jiao H, Chaocheng X, Liu X (2012) Face recognition based on multi-class Fisher scores, vol 38, pp 1221–1224 (Elsevier)
Hamilton JD (1990) Analysis of time series subject to changes in regime. J Econ 45:39–70
Han J, Hong J, Sutardja N (2015) Machine learning techniques for price change forecast using the limit order book data. Technical report, University of California, Berkeley
Hoffman P, Grinstein G, Marx K, Grosse I, Stanley E (1997) DNA visual and analytic data mining. In: Ebert DS, Shaw CD (eds), Proceedings of the 8th Conference on Visualization '97, pp 437–441 (IEEE Computer Society Press, Washington, DC)
Huang W, Nakamori Y, Wang SY (2005) Forecasting stock market movement direction with support vector machine. Comput Oper Res 32:2513–2522
Hussain Z, Shawe-Taylor J (2011) Improved loss bounds for multiple kernel learning. J Mach Learn Res 15:370–377
Jaakkola T, Diekhans M, Haussler D (1999) Using the Fisher kernel method to detect remote protein homologies. In: Lengauer T et al (eds) Proceedings, Seventh International Conference on Intelligent Systems for Molecular Biology Proceedings, pp 149–158. AAAI Press, Washington, DC
Jaakkola T, Haussler D (1999) Exploiting generative models in discriminative classifiers. In: Jordan MI, Kearns MJ, Solla SA (eds), Proceedings of the 1998 conference, vol 11 of Advances in Neural Information Processing Systems, pp 487–493 (MIT Press, Cambridge, Mass)
Jebara T, Kondor R, Howard A (2004) Probability product kernels. J Mach Learn Res 5:819–844
Kearns M, Nevmyvaka Y (2013) Machine learning for market microstructure and high frequency trading. In: Easley DA, de Prado ML, O'Hara M (eds) High-frequency trading: New realities for traders, markets and regulators, pp 91–124. Risk Books, London
Kercheval AN, Zhang Y (2015) Modelling high-frequency limit order book dynamics with support vector machines. Quantitative Finance 15:1315–1329
Kirilenko A, Kyle AS, Samadi M, Tuzun T (2017) The flash crash: high-frequency trading in an electronic market. J Finance 72:967–998
Kontopantelis E (2018) A comparison of one-stage vs two-stage individual patient data meta-analysis methods: a simulation study. Res Synthesis Method 9:417–430
Kozhan R, Salmon M (2012) The information content of a limit order book: the case of an FX market. J Financial Markets 15:1–28
Krauss C, Do XA, Huck N (2017) Deep neural networks, gradient-boosted trees, random forests: statistical arbitrage on the S&P 500. Eur J Oper Res 259:689–702
Lanckriet GR, Cristianini N, Bartlett P, El Ghaoui L, Jordan MI (2004) Learning the kernel matrix with semidefinite programming. J Mach Learn Res 5:27–72
Lasserre JA, Bishop CM, Minka TP (2006) Principled hybrids of generative and discriminative models, vol 1. IEEE, Cambridge, UK, pp 87–94
Li C, Liu K (2022) Path signature-based phase space reconstruction for stock trend prediction. Int J Data Sci Anal 14:293–304
Liu L, Wang Z (2016) Encoding temporal markov dynamics in graph for visualizing and mining time series. arXiv:1610.07273
López de Prado M (2018) Advances in Financial Machine Learning. Wiley, Hoboken, NJ
Mainali S et al (2021) An information-theoretic approach to dimensionality reduction in data science. Int J Data Sci Anal 12:185–203

Makowiec D et al (2017) Visualization of short-term heart period variability with network tools as a method for quantifying autonomic drive. In: Jelinek HF, Cornforth DJ, Khandoker AH (eds) ECG Time Series Variability Analysis: Engineering and Medicine. CRC Press, Boca Raton

Manning CD, Raghavan P, Schütze H (2008) Introduction to Information Retrieval. Cambridge University Press, Cambridge, UK

Martins A, Bicego M, Murino V, Aguiar P, Figueiredo M (2010) Information theoretical kernels for generative embeddings based on hidden Markov models. In: Hancock ER, Wilson RC, Windeatt T, Ulusoy I (eds) Lecture Notes in Computer Science (including subseries in Artificial Intelligence and Bioinformatics), vol 6218 of Joint IAPR International Workshop, SSPR 2010, pp 463–472. Springer, Berlin

Nasir M, Ezeife C, I, (2023) Semantic enhanced markov model for sequential e-commerce product recommendation. Int J Data Sci Anal 15:67–91

Nystrup P, Lindström E, Madsen H (2020) Learning hidden Markov models with persistent states by penalizing jumps. Expert Syst Appl 150:113307

O'Hara M (2014) High-frequency trading and its impact on markets. Financ Anal J 70:18–27

Patrick H, Georges G, David P (1999) Dimensional anchors: a graphic primitive for multidimensional multivariate information visualizations. In: Shaw CD, Ebert DS (eds), Proceedings of the 1999 Workshop on New Paradigms in Information Visualization and Manipulation in Conjunction with the Eighth ACM International Conference on Information and Knowledge Management, pp 9–16 (Association for Computing Machinery, New York, NY)

Perronnin F, Sánchez J, Mensink T (2010) Improving the Fisher kernel for large-scale image classification, vol 6314. Lecture Notes in Computer Science. Springer-Verlag, Berlin, Germany, pp 143–156

Phillips PC, Xiao Z (1998) A primer on unit root testing. J Econ Surveys 12:423–470

Pohle J, Langrock R, van Beest FM, Schmidt NM (2017) Selecting the number of states in hidden Markov models: pragmatic solutions illustrated using animal movement. J Agric Biol Environ Stat 22:270–293

Raykar VC, Saha A (2015) Data split strategies for evolving predictive models. In: Appice A, et al (eds), Machine Learning and Knowledge Discovery in Databases. ECML PKDD 2015, vol 9284 of Lecture Notes in Computer Science, pp 3–19 (Springer, Cham)

Rengifo E, Trendafilov R (2015) High-frequency trading under information regimes. In: Gregoriou GN (ed) Handbook of High Frequency Trading, pp 279–303. Academic Press, London, UK

Rothschild D, Sethi R (2016) Trading strategies and market microstructure: evidence from a prediction market. J Prediction Markets 10:1–29

Santafe G, Inza I, Lozano JA (2015) Dealing with the evaluation of supervised classification algorithms. Artif Intell Rev 44:467–508

Schölkopf B, Smola AJ (2001) Learning with kernels: support vector machines, regularization, optimization, and beyond Adaptive Computation and Machine Learning. MIT Press, Cambridge, MA

Shu Y, Yu C, Mulvey JM (2024) Dynamic asset allocation with asset-specific regime forecasts. Annal Oper Res

Sirignano JA (2019) Deep learning for limit order books. Quantitative Finance 19:549–570

Sirignano J, Cont R (2019) Universal features of price formation in financial markets: perspectives from deep learning. Quantitative Finance 19:1449–1459

Smola AJ, Schölkopf B (2004) A tutorial on support vector regression. Stat Comput 14:199–222

Takahashi K, Yamamoto K, Kuchiba A, Koyama T (2022) Confidence interval for micro-averaged and macro-averaged F_1 scores. Appl Intell 52:4961–4972

Tay FE, Cao L (2001) Application of support vector machines in financial time series forecasting. Omega 29:309–317

Valstar MF, Pantic M (2007) Combined support vector machines and hidden Markov models for modeling facial action temporal dynamics. In: Lew M, Sebe N, Huang TS, Bakker EM (eds) Human-Computer Interaction, vol 4796. Lecture Notes in Computer Science, pp 118–127. Springer, Berlin

Van Der Maaten L, Hinton G (2008) Visualizing data using t-SNE. J Mach Learn Res 9:2579–2625

Van Der Maaten L, Lise Getoor T (eds) (2011) Learning discriminative Fisher kernels. In: Lise Getoor T (ed), 28th International Conference on Machine Learning, pp 217–224. ICML (Omnipress, Madison, WI)

Van Gestel T et al (2001) Financial time series prediction using least squares support vector machines within the evidence framework. IEEE Trans Neural Networks 12:809–821

Vanden JM (2005) Equilibrium analysis of volatility clustering. J Empir Financ 12:374–417

Vapnik VN (1999) An overview of statistical learning theory. IEEE Trans Neural Networks 10:988–999

Wang Q (2021) Cryptocurrencies asset pricing via machine learning. Int J Data Sci Anal 12:175–183

Wang L, Suter D (2008) Visual learning and recognition of sequential data manifolds with applications to human movement analysis. Comput Vis Image Underst 110:153–172

Wang N, Li D, Wang Q (2012) Visibility graph analysis on quarterly macroeconomic series of China based on complex network theory. Physica A 391:6543–6555

Yang H, Chan L, King I (2002) Support vector machine regression for volatile stock market prediction. In: Yin H, Allinson N, Freeman R, Keane J, Hubbard S (eds), Lecture Notes in Computer Science, vol 2412 of IDEAL '02, pp 391–396 (Springer, London, UK)

Zhang XW, Jin ND, Gao ZK, Zhai LS (2013) Local property of recurrence network for investigating gas-liquid two-phase flow characteristics. Chinese Phys Lett 30:50501-1–050501-4

Zhang R, Zou Y, Zhou J, Gao ZK, Guan S (2017) Visibility graph analysis for re-sampled time series from auto-regressive stochastic processes. Commun Nonlinear Sci Numer Simul 42:396–403