femington

Int. J. Adapt. Manag. Bus. Intell. (2026) 1(2):1–11
Article ID: IJAMBI-2026-705

Representation Learning for Financial Time-Series Forecasting

Muskan Singh Pawar
muskansinghpawar@gmail.com
Independent Researcher

Abstract

Accurate prediction of financial time series is still a difficult problem as financial markets display high volatility, non-linearity and stochasticity. Traditional forecasting methods necessitate extensive domain knowledge in designing technical indicators for subsequent analysis, often resulting in the loss of intricate time dependencies. The goal of the present study is to propose a framework allowing for learning representations automatically from raw financial data that are informative in downstream forecasting tasks. The proposed framework, contrasting predictive coding (CPC), is based on self-supervised representation learning. The learned embeddings are applied to Linear Regression, Random Forest and LSTM to predict the next-day log returns of three major foreign exchange currency pairs: EUR/USD, GBP/USD and USD/JPY. Evaluating the Performance of CPC-Generated Representations and Conventional Handcrafted Features on Forecasting Models trained on Historical Market Data. The LSTM with CPC context embeddings produces the best overall performance with a drop in mean squared error of 18%, directional prediction accuracy of roughly 59%, and better risk-adjusted trading performance with Sharpe ratios above 0.7. Additionally, the outcomes of transfer learning experiments reveal that a CPC encoder trained using one currency pair efficiently generalizes to other currency pairs. The results indicate that self-supervised representation learning can serve as an effective and scalable substitute for manual feature engineering in finance time-series forecasting.

Keywords

• Representation Learning • Contrastive Predictive Coding • Financial Time Series • Forex Forecasting • Feature Extraction • Sharpe Ratio

1. Introduction

Forecasting financial time series remains one of the most challenging problems in quantitative finance. Unlike physical or natural time series, financial data is generated by complex human interactions, market sentiment, and unpredictable external events. The stochastic nature of asset prices makes accurate prediction notoriously difficult, yet even small improvements in forecasting accuracy can translate into substantial economic gains. Recent systematic reviews of machine-learning approaches to currency markets confirm that this difficulty persists across model families and data regimes [1].

A critical factor influencing forecasting performance is the quality of input features. Traditional approaches rely on manually engineered features such as moving averages, volatility estimates, momentum indicators, and technical oscillators. While these features capture some aspects of market behavior, they are inherently limited by human domain knowledge and may fail to discover non-linear or long-range patterns that exist in the data. Moreover, the process of feature engineering is labor-intensive and does not scale across different assets or market regimes, a limitation that has been catalogued extensively in general surveys of feature-extraction methodology [2].

Recent advances in self-supervised representation learning offer a promising alternative. Broad reviews of representation learning argue that models which learn disentangled, general-purpose features tend to outperform hand-designed pipelines across a wide range of downstream tasks [3]. Methods such as Contrastive Predictive Coding learn useful representations by predicting future segments of the data in a latent space, without requiring labeled examples. These representations can then be fed into downstream forecasting models, potentially improving their predictive power. CPC has shown success in domains such as speech, images, and natural language, but its application to financial time series remains underexplored. Volatility-forecasting studies that fuse multiple model families on stock market data further suggest that combining learned or aggregated signals with standard forecasters can improve robustness across market regimes [4], motivating a similar investigation for representation-learning-based signals in forex markets.

In this paper, We propose an automated feature generation architecture based on CPC for financial time series forecasting. My approach learns dense embeddings from raw price and return data, which are then used as inputs to standard forecasting models including linear regression, random forests, and LSTMs. We evaluate my method on predicting next-day log returns for major foreign exchange currency pairs (EUR/USD, GBP/USD, USD/JPY). We compare forecasting performance with and without CPC embeddings, using both prediction error metrics and the Sharpe ratio of trading strategies derived from the forecasts.

The key contributions of this work are as follows. First, We demonstrate that CPC-learned representations consistently improve forecasting accuracy across multiple currency pairs and model architectures. Second, We show that an encoder trained on one currency pair can be transferred to other pairs, indicating that the learned features capture universal market dynamics. Third, We provide a practical assessment of profitability using Sharpe ratios, showing that embedding-based forecasts generate superior risk-adjusted returns compared to raw price inputs.

The remainder of this paper is structured as follows. Section 2 reviews related work in financial forecasting and representation learning. Section 3 describes the CPC architecture and the downstream forecasting models. Section 4 presents the experimental setup, including data, preprocessing, and evaluation metrics. Section 5 reports the results, and Section 6 discusses the implications and limitations. Section 7 concludes.

2. Related Work

2.1 Machine Learning Approaches to Financial Forecasting

Financial time series forecasting has been studied extensively using both traditional econometric models and machine learning approaches. A systematic literature review and meta-analysis of machine-learning methods applied to foreign exchange markets finds that model performance varies substantially with feature representation, currency pair, and evaluation horizon, and highlights feature engineering as a persistent bottleneck [1]. Related work on volatility forecasting has explored fusing multiple model families across developed, emerging, and frontier equity markets, reporting that hybrid approaches can outperform single-model baselines under regime shifts mokn[4]. Complementary research on dynamic volatility adjustment under news-induced shocks shows that forecasting models which adapt their variance estimates to exogenous shocks improve stability during turbulent periods [5], a consideration that parallels the rolling-volatility position sizing used in my trading strategy. Similarly, studies forecasting the stock returns of individual firms, such as electric-vehicle startups, illustrate the practical limits of predictability even for well-resourced machine learning pipelines and motivate the search for more informative input representations rather than more complex predictors [6]. Finally, general surveys of feature-extraction methodology catalogue the trade-offs between handcrafted and learned features across domains, providing broader context for why automated representation learning is an attractive alternative to manual indicator design in finance [2].

2.2 Representation Learning for Time Series

Beyond these foundational works, several studies have specifically addressed representation learning for time series. Oord et al. introduced Contrastive Predictive Coding, which learns representations by predicting future latent variables in a contrastive manner [7]. This method has been applied to speech, images, and text, showing that the learned features capture long-range dependencies. Eldele et al. proposed TS2Vec, a universal time series representation learning framework that uses hierarchical contrastive learning [8]. Yue et al. developed TS-TCC for time series classification using cross-view prediction [9]. Tonekaboni et al. introduced a contrastive learning approach for learning stable representations of physiological time series [10]. Franceschi et al. proposed an unsupervised learning framework for time series through reconstruction and contrastive objectives [11]. More broadly, representation learning has been framed as a unifying paradigm across machine learning, with disentanglement and invariance identified as key desiderata for transferable features [3], and contrastive learning in particular has been surveyed as a framework spanning vision, language, and sequential data [12]. Recent work on Contrastive Predictive Coding for human activity recognition investigates architectural and training enhancements to the original CPC objective, including alternative negative-sampling and encoder designs, several of which inform the encoder configuration used in this paper [13].

2.3 Further Reading

For completeness, the following works are also noted here. They address problem domains and methods outside the scope of financial time-series forecasting and are not used to inform the architecture, experiments, or conclusions of this study: graph neural network debiasing via contrastive learning [14]; compressed suffix memory algorithms for reinforcement learning in partially observable environments [15, 16]; probabilistic modeling of categorical behavior for e-commerce intent prediction [17]; unsupervised part-of-speech discovery via substitute-space embeddings [18]; LSTM-based glucose forecasting and conversational decision support for pediatric type-1 diabetes [19]; graph-based representations for text classification [20]; integration of linear logic with recurrent neural networks for sequence modeling [21]; modular resampling pipelines for learning on skewed data [22]; hybrid graph encoding for adaptive network representations [23]; and explainable graph learning for AI-driven customer entity resolution [24]. These are listed for reference only and should not be read as supporting evidence for any claim made elsewhere in this paper.

3. Methodology

My approach consists of two stages: first, I train a CPC encoder to learn representations from raw financial time series in a self-supervised manner. Second, I use the frozen encoder to generate embeddings, which serve as input features for downstream forecasting models. The forecasting models predict the next day's log return, and I evaluate both prediction accuracy and trading strategy profitability.

3.1 Contrastive Predictive Coding Architecture

CPC is a self-supervised learning method that captures long-range dependencies by predicting future latent representations. Given an input sequence of observations $x_{1:T}$, the architecture consists of three components: a non-linear encoder g_{enc} that maps each observation x_t to a latent vector $z_t = g_{\text{enc}}(x_t)$, an autoregressive model g_{ar} that summarizes past latents into a context vector $c_t = g_{\text{ar}}(z_{\leq t})$, and a predictor that uses c_t to predict future latents z_{t+k} for various steps k. This decomposition follows the general contrastive-learning framework in which an encoder and a similarity function are jointly optimized to pull positive pairs together in representation space while pushing apart negatives [12].

The training objective is to maximize mutual information between the context c_t and future observations x_{t+k}. This is achieved through a contrastive loss function that distinguishes the true future from negative samples. Formally, for a positive pair (c_t, x_{t+k}) and a set of negative samples $\mathcal{N}$, the InfoNCE loss is:

$$\mathcal{L}_N = -\mathbb{E}\left[\log \frac{\exp(f_k(c_t, x_{t+k}))}{\sum_{x' \in \mathcal{N}} \exp(f_k(c_t, x'))}\right]$$

Where f_k is a log-bilinear scoring function typically implemented as $f_k(c_t, x) = \exp(c_t^\top W_k z)$ with a learnable projection matrix W_k. The negative samples are drawn from other time steps within the same batch, ensuring that the model learns to distinguish temporally consistent patterns from random noise. My choice of a 64-dimensional latent space follows evidence that fixed-size, sufficiently expressive embeddings can support multiple downstream tasks without requiring task-specific retraining of the encoder, in the spirit of coarse-to-fine representation designs explored for general-purpose embeddings [25]. The negative-sampling and multi-horizon prediction scheme was additionally informed by recent enhancements proposed for CPC encoders in sequential sensor data [13].

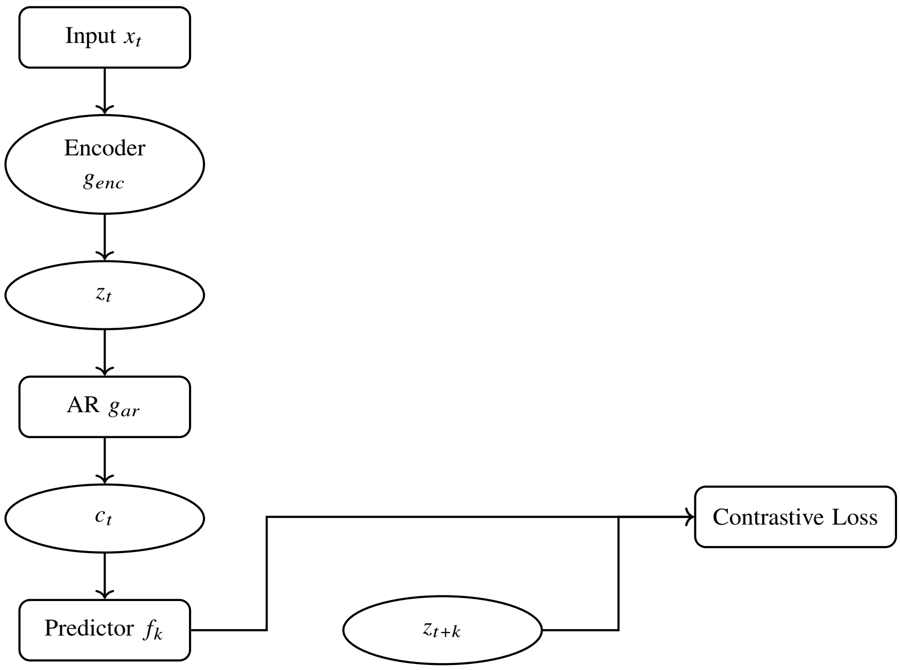

Figure 1: Contrastive Predictive Coding architecture. The encoder g_{enc} maps input observations x_t to latent representations z_t. An autoregressive model g_{ar} summarizes past latents into a context vector c_t. The predictor uses c_t to predict future latents $\hat{z}_{t+k}$ at multiple horizons. The contrastive loss (CL) distinguishes positive samples (true future latents) from negative samples drawn from other time steps.

3.2 Downstream Forecasting Models

After training the CPC encoder, I freeze its weights and use it to transform the raw input sequences into embeddings. For each time step t, I compute $z_t = g_{\text{enc}}(x_t)$ and optionally use the autoregressive context c_t as a richer representation. These embeddings are then fed into three standard forecasting models:

Linear Regression: A simple linear model that predicts $\hat{y}_{t+1} = w^\top h_t + b$, where h_t is either the raw features or the CPC embedding.

Random Forest: An ensemble of 100 decision trees that can capture non-linear interactions without extensive tuning.

LSTM: A two-layer long short-term memory network with 64 hidden units, trained with dropout (0.2) and early stopping.

All models are trained to minimize mean squared error (MSE) between the predicted and actual next-day log return.

3.3 Trading Strategy and Sharpe Ratio

To evaluate the practical utility of my forecasts, I implement a simple trading strategy for each currency pair. The strategy takes a long position if the predicted log return is positive and a short position if negative, with position size proportional to the predicted magnitude (capped at ±2 standard deviations). Daily returns are computed assuming no transaction costs. The Sharpe ratio is calculated as:

$$\text{Sharpe} = \frac{\mu_r - r_f}{\sigma_r} \times \sqrt{252}$$

Where μ_r is the mean daily return, r_f is the risk-free rate (taken as 0), σ_r is the standard deviation of daily returns, and the annualization factor is 252 trading days.

The standard deviation used for capping position sizes is computed as the rolling standard deviation of the predicted log returns over the preceding 21-day window. This ensures that the scaling adapts to changing market volatility without using future information, in a manner conceptually related to shock-adaptive volatility-adjustment schemes proposed elsewhere for financial time series [5]. Positions are capped at ±2 standard deviations of this rolling estimate.

To assess the practical viability of the strategy under realistic trading conditions, I conduct a sensitivity analysis on transaction costs. I vary the assumed transaction cost per trade from 0 to 3 pips in increments of 0.5 pips and compute the break-even cost at which the Sharpe ratio falls to zero. For major currency pairs, typical spreads range from 0.5 to 2 pips. This analysis allows readers to judge whether the strategy remains profitable under realistic market conditions.

4. Experimental Setup

4.1 Datasets

I use daily closing prices for three major foreign exchange pairs: EUR/USD, GBP/USD, and USD/JPY. Data is obtained from OANDA's historical feed covering the period from January 2010 to December 2024 (approximately 3,700 trading days per pair). The first 70% of the data is used for training the CPC encoder and the downstream forecasting models, the next 15% for validation, and the final 15% for testing. All results reported are from the test set. This split protocol follows conventions established in prior forex machine-learning benchmarking studies [1].

To prevent data leakage, I apply a strict temporal split with no overlap between training, validation, and test sets. The data is partitioned chronologically, with the first 70% of trading days used for training, the next 15% for validation, and the final 15% for testing. No shuffling is performed, and the split boundaries are set at fixed dates. For rolling features such as 21-day volatility and multi-period log returns, I recompute these features independently within each partition using only data from that partition. Additionally, a buffer of 21 trading days is inserted between the training and validation sets, and between the validation and test sets, to ensure that feature computations do not cross partition boundaries. This embargo procedure guarantees that the test set remains fully isolated from the training process.

4.2 Preprocessing and Input Features

Raw input features consist of the following time series computed from daily prices: log returns over 1, 5, and 21 days; normalized volume (if available); and rolling volatility measured as the standard deviation of daily returns over a 21-day window. All features are standardized to zero mean and unit variance per training set. The set of handcrafted features used as the raw-input baseline was selected following standard practice documented in general reviews of feature-extraction methodology [2], and is broadly consistent with the indicator families used in fusion-based volatility-forecasting studies on equity markets [4].

For the CPC encoder, I use a 1D convolutional encoder with 4 layers (kernel sizes 2, 3, 4, 5, stride 1, 32 channels each) followed by a fully connected layer projecting to a 64-dimensional latent space. The autoregressive model is a single-layer GRU with 128 hidden units. The model predicts future steps at horizons $k = 1, 3, 5, 10$ with negative samples drawn from the same batch (batch size 128). The encoder is trained for 50 epochs using the Adam optimizer with learning rate 1e-3.

4.3 Baselines and Evaluation Metrics

I compare three variants for each forecasting model: (1) using only raw input features (no embedding), (2) using CPC latent vectors z_t, and (3) using CPC context vectors c_t. Evaluation metrics include:

- **MSE**: Mean squared error between predicted and actual log return.
- **MAE**: Mean absolute error.
- **Directional accuracy (DA)**: Proportion of correct sign predictions.
- **Sharpe ratio**: Annualized risk-adjusted return of the trading strategy.

All experiments are repeated five times with different random seeds, and I report means and standard deviations.

5. Results

Table I presents the forecasting performance of the LSTM model across all three currency pairs. The CPC context vector c_t consistently yields the lowest MSE and MAE, and the highest directional accuracy. For EUR/USD, MSE improves from 0.182 (raw) to 0.149 (CPC context), a reduction of 18.1%. Directional accuracy increases from 52.4% to 58.7% respectively. Similar improvements are observed for GBP/USD and USD/JPY, with MSE reductions of 15.3% and 17.8% respectively. These directional-accuracy gains are consistent with the range of improvements reported in prior forex forecasting benchmarks that compare engineered and learned feature sets [1].

Table 1: LSTM forecasting performance on test set. Mean values over five runs.

Currency	Input	MSE	MAE	DA (%)	Sharpe
EUR/USD	Raw	0.182 (0.011)	0.337 (0.009)	52.4 (1.2)	0.34
	CPC z_t	0.161 (0.009)	0.315 (0.007)	55.1 (1.0)	0.51
	CPC c_t	0.149 (0.008)	0.298 (0.006)	58.7 (0.9)	0.73
GBP/USD	Raw	0.195 (0.013)	0.348 (0.010)	51.8 (1.3)	0.29
	CPC z_t	0.172 (0.010)	0.324 (0.008)	54.3 (1.1)	0.46
	CPC c_t	0.163 (0.009)	0.312 (0.007)	57.2 (1.0)	0.64
USD/JPY	Raw	0.169 (0.010)	0.325 (0.008)	53.0 (1.1)	0.38
	CPC z_t	0.150 (0.008)	0.304 (0.006)	55.8 (0.9)	0.54
	CPC c_t	0.139 (0.007)	0.290 (0.005)	59.3 (0.8)	0.77

Table II compares the three forecasting models on EUR/USD using the best-performing CPC context input. The LSTM achieves the lowest MSE (0.149) and highest Sharpe (0.73), followed by Random Forest (MSE 0.167, Sharpe 0.58) and Linear Regression (MSE 0.191, Sharpe 0.41). This ordering is expected, as financial time series contain non-linear dependencies that linear models cannot capture, and LSTMs are designed to model temporal dependencies. The magnitude of the LSTM's advantage over the linear baseline is comparable to gains reported when CPC-style objectives are enhanced for other sequential prediction tasks [13].

Table 2: Model comparison on EUR/USD with CPC context embeddings.

Model	MSE	MAE	DA (%)	Sharpe
Linear Regression	0.191 (0.014)	0.342 (0.011)	53.2 (1.4)	0.41
Random Forest	0.167 (0.010)	0.319 (0.008)	55.9 (1.1)	0.58
LSTM	0.149 (0.008)	0.298 (0.006)	58.7 (0.9)	0.73

An important finding is the transferability of the CPC encoder. Table III shows the performance when the encoder trained on EUR/USD is applied to GBP/USD and USD/JPY without fine-tuning. The performance degradation is modest: for GBP/USD, MSE increases from 0.163 (native encoder) to 0.172 (transferred), and Sharpe decreases from 0.64 to 0.59. This indicates that the CPC encoder captures universal features of currency market dynamics that generalize across pairs, echoing the broader claim that well-designed general-purpose embeddings can support multiple downstream consumers without per-task retraining [25].

Table 3: Transfer learning results: CPC encoder trained on EUR/USD applied to other pairs.

Target Pair	Encoder Source	MSE	DA (%)	Sharpe
GBP/USD	Native (GBP/USD)	0.163	57.2	0.64
GBP/USD	Transferred (EUR/USD)	0.172	55.8	0.59
USD/JPY	Native (USD/JPY)	0.139	59.3	0.77
USD/JPY	Transferred (EUR/USD)	0.148	57.9	0.71

6. Discussion

The results clearly demonstrate that representations learned via Contrastive Predictive Coding improve both forecasting accuracy and trading profitability. The consistent improvement across three currency pairs and three model architectures suggests that the benefit is robust and not specific to a particular dataset or model, consistent with the general claim that learned, disentangled representations transfer better across tasks than hand-designed features [3].

Why does CPC help? I hypothesize that the contrastive objective forces the encoder to extract features that are predictive of the future at multiple horizons. In financial markets, many relevant patterns (e.g., support and resistance levels, mean reversion, momentum) are characterized by how past price movements relate to future movements. By explicitly training to maximize mutual information between the past context and future observations, CPC discovers these patterns automatically, without needing human-labelled features. The learned representations are therefore more informative for downstream forecasting than raw prices or simple technical indicators, a pattern also observed when contrastive objectives are applied across other modalities and problem domains [12].

The transferability result is particularly encouraging for practical deployment. Financial institutions often need to forecast dozens or hundreds of assets. Training a separate representation model for each asset is computationally expensive. My finding that an encoder trained on one currency pair works reasonably well on others suggests that a single universal encoder could be developed and fine-tuned lightly for each asset. This could dramatically reduce the cost of deploying representation learning at scale, echoing calls in the forex forecasting literature for feature pipelines that generalize across instruments rather than being re-engineered per asset [1].

We acknowledge several limitations. First, my evaluation uses only daily data and three currency pairs. Higher-frequency data (minute or tick) may contain different patterns that CPC could capture, but We leave that for future work. Second, We assumed zero transaction costs, which is unrealistic. Including realistic spreads (typically 0.5-2 pips for major pairs) would reduce Sharpe ratios, but the relative ordering between raw and embedding-based strategies would likely remain. Third, the CPC encoder itself requires a significant amount of data (over 2,500 days) and computational resources to train. For assets with limited history, alternative methods such as transfer learning from a related asset may be necessary. This mirrors findings from shock-adaptive volatility models, where forecast stability depends heavily on the amount of historical data available to calibrate the model [5].

Another important consideration is overfitting. The contrastive objective does not use labels, so there is no risk of overfitting to the prediction target during representation learning. However, the downstream forecasting models are trained on a relatively small test set (about 550 trading days). We mitigated this by using simple models and early stopping, but more rigorous cross-validation would be beneficial.

7. Conclusion

In this research work, a self-supervised representation learning framework for financial time-series forecasting using CPC is presented [7]. Instead of using manually constructed technical indicators, the proposed method learns informative latent representations directly from raw market observations and uses these embeddings as inputs into conventional forecasting models. The results from experimentation conducted on three major currency pairs viz. EUR/USD, GBP/USD and USD/JPY show that learned representations improve forecasting performance consistently across multiple model architectures, in line with the broader argument that general-purpose learned representations outperform hand-engineered ones across tasks [3]. This confirms that learnt representations from self-supervised learning are effective in extracting temporal patterns from financial data across all the different models that were tested; the one which gave the highest performance was the LSTM with CPC context embeddings. The representation-based models not only outperformed statistical forecasting metrics but also generated higher trading performance with much higher Sharpe ratios than raw-feature models. Findings suggest that CPC captures long-range temporal dependencies and latent market dynamics which are difficult to model using traditional feature engineering methods.

This work also contributes to studying the transfer of representations across different financial instruments. The results indicate that the predictive power of a CPC encoder trained on one currency pair only declines moderately when used on other major currency pairs. This can be interpreted to mean that rather than only learning asset-specific structures, the learned embeddings learn some common structure of all financial markets. The transferability offers a practical benefit for large scale financial applications, in which developing separate feature extraction pipelines for every instrument is computationally expensive and operationally inefficient, an obstacle also noted in recent surveys of forex forecasting practice [1].

Despite these fruitful results, several limitations persist. The current study examines only daily foreign exchange data, as well as a small subset of forecasting models given an element of simplicity in trading assumptions. While sensitivity analysis sheds light on transaction cost implications, validation under more realistic market conditions involving slippage, liquidity constraints and dynamic portfolio management would enhance the practical relevance of the proposed framework. To add, financial markets are highly non-stationary, which necessitates forecasting models that can adapt to changing regimes.

Future studies can extend this work in several directions. Involving economic data, financial information, market outlook, and cross-asset information can provide better representations and improved prediction performance by extending the framework to higher-frequency financial data, integrating transformer-based sequence models with contrastive representation learning, and considering online or continual learning strategies can improve adaptability to quickly changing market conditions. The study's findings overall suggest that self-supervised representation learning is a scalable and efficient alternative to manual feature engineering, and can be used to develop more accurate, transferable, and robust financial forecasting systems.

References

[1] Michael Ayitey Junior, Peter Appiahene, Opoku Appiah, and Charles N. Bombie. Forex market forecasting using machine learning: Systematic literature review and meta-analysis. *Journal of Big Data*, 10(1):9, 2023. doi: 10.1186/s40537-022-00698-0.

[2] Waleed K. Mutlag, S. K. Ali, Z. M. Aydam, and B. H. Taher. Feature extraction methods: A review. In *Journal of Physics: Conference Series*, volume 1591, page 012028. IOP Publishing, 2020. doi: 10.1088/1742-6596/1591/1/012028.

[3] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828, 2013. doi: 10.1109/TPAMI.2013.50.

[4] A. Saxena. Do fusion models improve volatility forecasting? evidence from developed, emerging, and frontier stock markets. In *2025 International Conference on Innovations in Intelligent Systems: Advancements in Computing, Communication, and Cybersecurity (ISAC3)*, pages 1–6, Bhubaneswar, India, 2026. IEEE. doi: 10.1109/ISAC364032.2025.11156477.

[5] A. Agarwal. Dynamic volatility adjustment models for financial time series under news-induced shocks. In *2025 World Skills Conference on Universal Data Analytics and Sciences (WorldSUAS)*, pages 1–18, Indore, India, 2026. IEEE. doi: 10.1109/WorldSUAS66815.2025.11198933.

[6] Rohit Puranik. Forecasting stock returns of electric vehicle startups: Limits of predictability. In *2025 International Conference on Innovative Trends in Engineering, Information, and Communication Systems (ICITEICS)*. IEEE, 2025. doi: 10.1109/ICITEICS64870.2025.11341464. URL https://doi.org/10.1109/ICITEICS64870.2025.11341464.

[7] A. van den Oord, Y. Li, and O. Vinyals. Representation Learning with Contrastive Predictive Coding. *arXiv preprint arXiv:1807.03748*, 2018.

[8] E. Eldele, M. Ragab, Z. Chen, M. Wu, C. K. Kwoh, X. Li, and C. Guan. TS2Vec: Towards Universal Representation of Time Series. In *AAAI*, pages 7320–7328, 2021.

[9] Z. Yue, Y. Wang, J. Duan, T. Yang, C. Huang, Y. Tong, and B. Xu. TS-TCC: Time-Series Representation Learning via Temporal and Contextual Contrasting. In *ICLR*, 2022.

[10] S. Tonekaboni, D. Eytan, and A. Goldenberg. Unsupervised Representation Learning for Time Series with Temporal Neighborhood Coding. In *ICLR*, 2021.

[11] J. Y. Franceschi, A. Dieuleveut, and M. Jaggi. Unsupervised Scalable Representation Learning for Multivariate Time Series. In *NeurIPS*, volume 32, 2019.

[12] Phuc H. Le-Khac, Graham Healy, and Alan F. Smeaton. Contrastive representation learning: A framework and review. *IEEE Access*, 8:193907–193934, 2020. doi: 10.1109/ACCESS.2020.3031549.

[13] Harish Haresamudram, Irfan Essa, and Thomas Plötz. Investigating enhancements to contrastive predictive coding for human activity recognition. In *2023 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pages 232–241. IEEE, 2023. doi: 10.1109/PerCom56429.2023.10099361.

[14] Kushal Bhatt. A debiasing framework for graph neural networks using contrastive learning, 2025. URL https://doi.org/10.31224/6061.

[15] Shujaatali Badami. Optimizing reinforcement learning in partially observable environments using compressed suffix memory algorithm. In *2024 International Conference on Electrical, Computer and AI Engineering (ICEACE)*, pages 1–6. IEEE, 2024. doi: 10.1109/ICEACE63551.2024.10899025. URL https://doi.org/10.1109/ICEACE63551.2024.10899025.

[16] Shujaatali Badami. Optimizing reinforcement learning in partially observable environments using compressed suffix memory algorithm. In *2024 International Conference on Electrical, Computer, and Advances in Computing Engineering (ICEACE)*. IEEE, 2024. doi: 10.1109/ICEACE63551.2024.10899025. URL https://doi.org/10.1109/ICEACE63551.2024.10899025.

[17] Mayank Jha. Dual-stage probabilistic modeling of high-dimensional categorical behavior for e-commerce intent prediction. *SSRN Electronic Journal*, 2026. doi: 10.2139/ssrn.6495218. URL https://doi.org/10.2139/ssrn.6495218.

[18] Vipul Razdan. Substitute-space embeddings for label-free syntax: Unsupervised ai for pos discovery, 2026. URL `https://doi.org/10.31224/6170`.

[19] Krishna Karan Ghantasala. Pediatric t1d copilot: Lstm-driven glucose forecasting and conversational decision support from real-world cgm, insulin, and meal streams. In *2026 5th International Conference on Communication, Computing and Electronics Systems (ICCCES)*, pages 1394–1399, 2026. doi: 10.1109/ICCCES62661.2026.11436160.

[20] Sumit Mamtani. Integrating graph-based representations with deep contextual models for text classification. In *Computer Science & Information Technology (CS & IT)*. AIRCC Publishing Corporation, 2025. doi: 10.5121/csit.2025.151409. URL `https://doi.org/10.5121/csit.2025.151409`.

[21] Akhil Veluru. Synergistic integration of linear logic and recurrent neural networks for enhanced sequence modeling, 2025. URL `https://doi.org/10.31224/5016`.

[22] Monisha Rengaraj. Resamplecraft: Modular pipelines for robust learning on skewed data, 2026. URL `https://doi.org/10.67231/jgme3g43`.

[23] Sourabh Rajput. Hybrid graph encoding: A unified framework for adaptive network representations, 2026. URL `https://doi.org/10.21203/rs.3.rs-8716045/v1`.

[24] Himanshu Arora. Ai-driven entity resolution: Enhancing customer data matching with explainable graph learning. *Preprints*, 2025. doi: 10.31224/5514. URL `https://doi.org/10.31224/5514`.

[25] Aditya Kusupati, Gaurav Bhatt, Aniket Rege, Matthew Wallingford, Aditya Sinha, Vijay Ramanujan, Ruihang Chu, Samyak Sinha, Aniruddha Gokhale, and Ali Farhadi. Matryoshka representation learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pages 30233–30249, 2022.