Journal of Intelligent Machines and Engineering
https://cortexpub.com/index.php/jime/article/view/article2542

Research Article

An Ensemble Deep Learning Approach for Forecasting Top 20 TSX-Listed Equities

Khushvir Singh*, Pooja Sharma[1]
*Research Scholar, Department of Computer Science & Engineering, I.K. Gujral Punjab Technical University, Kapurthala, Punjab, India
[1]Assistant Professor, Department of Computer Science & Engineering, I.K. Gujral Punjab Technical University, Kapurthala, Punjab, India
*Corresponding author: khushvirsingh100@gmail.com

Abstract

Forecasting equity prices on the Toronto Stock Exchange (TSX) is unusually difficult because the market sits at the intersection of global commodity cycles, which drive its large resource sector, and tightly regulated domestic banking, which behaves differently under the same macro shocks. This study addresses that complexity with an ensemble deep learning pipeline evaluated on ten years of daily price data (January 2015-January 2025) for twenty TSX equities spanning banking, energy, technology/precious metals, and diversified industrials. Adjusted closing prices from the Yahoo Finance API were cleaned, transformed into log-returns, and enriched with rolling statistical features. The architecture combines two independently trained LSTM sub-networks, each processing a 60-day sliding window and initialized from weights pretrained on roughly 500 S&P 500 constituent series, with predictions aggregated through a compact dense regression head to produce one-step-ahead forecasts. The proposed model outperforms the Li-Shi Hybrid Preprocessing Neural Network on all three evaluation metrics, reducing RMSE to 0.02195 (4.6%), MAE to 0.01594 (11.4%), and MAPE to 0.38% (15.6%). Residual diagnostics confirm that forecast errors are unbiased and homoscedastic across the full range of predicted values, essential for reliable confidence intervals in real risk-management use. The pipeline is open-source and reproducible from a standard Python environment.

Keywords

Toronto Stock Exchange; ensemble deep learning; LSTM stock price forecasting; time-series analysis; transfer learning; Yahoo Finance API; residual diagnostics; homoscedasticity

Manuscript Received: August 23, 2026
Revised: September 1, 2026
Accepted: September 5, 2026

1. Introduction

If you want to understand what makes the Toronto Stock Exchange tick, you need to start with its composition. The TSX is one of the ten largest exchanges in the world by market capitalization, but unlike the S&P 500, which is dominated by technology and consumer companies, the TSX gets roughly half its weight from just two sectors: financial services and natural resources [1]. That concentration has enormous consequences for how the market behaves. When the Bank of Canada raises interest rates, the big banks tend to benefit from wider net interest margins, but higher borrowing costs can simultaneously slow the capital-intensive resource sector. When oil prices collapse, as they did in late 2015, the energy names that make up a sizeable chunk of the index can drag the entire market lower, even if the banking side is holding steady. This kind of sector-level tension creates dynamics that are simply not present in more diversified indices, and it makes the TSX an unusually demanding test case for any forecasting model.

The decade from 2015 to 2025 amplified these dynamics considerably. The oil price crash of late 2015 and 2016 hit Canadian energy producers hard, with some names losing half their market value before a partial recovery in 2017 and 2018. Then came 2020, when the COVID-19 pandemic delivered a shock that no model had been trained to anticipate: markets fell 30% in a matter of weeks before rebounding sharply on the back of unprecedented monetary and fiscal stimulus. By 2022, inflation had risen to multi-decade highs, and the Bank of Canada embarked on one of the fastest rate-tightening cycles in its history, which compressed equity valuations across the board and particularly hurt the long-duration growth stocks that had thrived during the zero-rate era. Running a model through this period, with all its regime shifts, policy pivots, and sector rotations, is a far more honest test than evaluating against a quiet bull-market sample.

Against this backdrop, deep learning offers real advantages over classical statistical approaches. ARIMA models and their variants assume linearity and stationarity, properties that financial return series only approximate, and often poorly during stress periods. LSTM networks, by contrast, are designed to capture long-range dependencies in sequential data, and they have been shown to outperform statistical baselines across a range of equity markets [2, 3, 4, 5]. The main weakness of most published LSTM-based forecasting work is that it relies on a single model and a small number of securities, sometimes just one or two tickers. A model trained on a single stock has not been asked to generalize; it has essentially been asked to memorize. Ensemble methods address this by combining predictions from multiple independently trained models, which reduces variance without increasing bias [6]. Transfer learning adds another layer by initializing model weights from a checkpoint trained on a larger dataset, so the model arrives with useful prior knowledge rather than starting from scratch [7].

This paper combines all three of these approaches (LSTM, ensemble learning, and transfer learning) and evaluates the result systematically against a recent benchmark [8] that tackles similar objectives with a simpler setup. The aim is to show that the improvements are traceable to specific methodological choices, rather than simply to report better numbers, and that the resulting pipeline is practical enough for someone outside this research group to pick up and use.

The remainder of the paper is organized as follows. Section 2 reviews the relevant literature on statistical and deep learning approaches to equity forecasting. Section 3 describes the data acquisition, preprocessing pipeline, and model architecture in full detail, including a complete hyperparameter table. Section 4 presents the results supported by seven diagnostic figures. Section 5 discusses the findings in depth, including a systematic sixteen-dimension comparison with [8] and a broader discussion of what the results mean in practice and where the study falls short. Section 6 concludes with directions for future work.

2. Literature Review

2.1 Classical Statistical Approaches

The history of equity price forecasting is almost as long as the history of stock markets themselves. For most of that history, researchers and practitioners relied on linear statistical frameworks. Autoregressive Integrated Moving Average (ARIMA) models, developed and popularized by Box, Jenkins, Reinsel, and Ljung [9], became the workhorse for modelling price levels and returns. ARIMA captures the autocorrelation structure in a time series and can produce reasonable short-horizon forecasts when the data is approximately stationary and linear. Alongside ARIMA, the GARCH family of models [9] was developed specifically to handle the volatility clustering that is ubiquitous in financial returns: the well-documented

tendency for large price moves to be followed by more large price moves, regardless of direction. Hybrid combinations of ARIMA with neural networks have also been explored to capture both linear and non-linear dynamics simultaneously [10, 11].

These classical tools have real strengths: they are interpretable, they have well-understood statistical properties, and they are computationally cheap. But they also have hard limits. Real equity markets are non-linear: the relationship between today's return and yesterday's is not constant across market conditions. They are non-stationary at the level of raw prices, and even at the level of returns the distributional properties shift across regimes. And they respond to news events and macro shocks that no backward-looking model can anticipate. When those conditions are violated (which in financial data is most of the time), ARIMA and GARCH models hit a ceiling that better data or longer estimation windows cannot overcome.

2.2 The Rise of Deep Learning in Financial Forecasting

The introduction of the Long Short-Term Memory (LSTM) architecture by Hochreiter and Schmidhuber in 1997 was a turning point for sequence modelling [3]. The key innovation was the cell state and gating mechanism, which allowed the network to selectively retain information across long sequences without suffering from the vanishing-gradient problem that had made earlier recurrent networks impractical for financial data. An LSTM can, in principle, learn that a price pattern from sixty days ago is relevant to today's forecast, something a standard recurrent network would forget almost immediately.

The empirical results followed quickly once researchers started applying LSTMs to financial time series. Fischer and Krauss [2] demonstrated that LSTMs outperform random forests, deep feedforward networks, and classical statistical models on daily S&P 500 return prediction, a well-studied benchmark where beating simpler baselines is not easy. Their results held up out-of-sample, which is always the harder test. A systematic review by Sezer, Gudelek, and Ozbayoglu [12] subsequently surveyed over 150 deep learning studies on financial time series published between 2005 and 2019, and found that LSTMs and their variants consistently ranked among the top performers across markets, asset classes, and prediction horizons. Bao, Yue, and Rao [13] showed that stacking LSTMs with autoencoders for feature extraction could push performance even further on Chinese market data, highlighting that the architecture is modular and extensible. Nelson, Pereira, and de Oliveira [4] applied LSTM networks specifically to stock price movement prediction, demonstrating directional accuracy gains over simpler recurrent architectures. Selvin et al. [5] conducted a comparative study of LSTM, vanilla RNN, and CNN-based sliding window models for stock price forecasting, finding LSTM to be the most robust across varied market conditions. Moghar and Hamiche [14] further confirmed LSTM's suitability for stock market prediction using recurrent architectures on benchmark indices, and Gao and Chai [15] demonstrated that combining recurrent neural networks with technical indicators yields further improvements in closing price prediction accuracy, while Ding et al. [16] explored event-driven deep learning approaches that incorporate news signals to complement price-based features. Kim and Won [11] proposed integrating LSTM with GARCH-type models to jointly model price level and volatility, achieving improved out-of-sample performance relative to either model class alone. These comparative results collectively inform the architectural choices made in the present study.

These results need to be read narrowly. LSTMs do not make markets predictable in any strong sense: there is no evidence that they can generate consistent alpha in a real trading context once transaction costs and slippage are accounted for. What they do, reliably, is produce forecasts that are closer to the true price path than classical statistical alternatives, as measured by standard error metrics. That narrower claim is what the present paper is testing.

2.3 Ensemble Methods and Transfer Learning

A single well-trained model is vulnerable to the specific conditions under which it was trained. A model calibrated during a low-volatility bull market will likely underperform when volatility spikes or the market regime changes. Ensemble methods address this fragility by training multiple base learners independently and combining their predictions, typically through averaging or stacking. The theoretical justification, formalized by Qian and Rasheed [6] and rooted in the bias-variance decomposition, is straightforward: if the individual models make errors in different directions and at different times, their average error will be smaller than any single model's error, even if none of the individual models are perfect. This variance reduction is particularly valuable in financial forecasting, where the conditions under which a model is deployed are virtually never identical to the conditions under which it was trained.

Transfer learning adds a complementary benefit. Rather than initializing a new model with random weights and training it from scratch on the target dataset, transfer learning begins with weights estimated on a related but larger dataset [7]. For financial forecasting, this means a model can start with a general understanding of how price dynamics and return distributions behave across many securities, rather than trying to infer those patterns from a small sample of TSX tickers alone. Pan and Yang's survey [7] of transfer learning methods established the theoretical conditions under which this approach improves generalizationwhats, and those conditions (shared feature distributions, related task objectives) are generally met when transferring between equity market datasets. In practice, pretraining on a large cross-sectional equity dataset and fine-tuning on a specific market has been shown to reduce both training time and out-of-sample error.

2.4 The Gap in Canadian Equity Research

Despite the volume of published work on deep learning for financial forecasting, Canadian equities have received remarkably little attention. A search of the major databases reveals that the overwhelming majority of published LSTM forecasting studies focus on US markets (the S&P 500, the NASDAQ, individual large-cap names) or on Chinese exchanges, where data availability and academic interest have driven a surge of publications. European markets have received some coverage; Canadian markets very little.

This gap matters for substantive, not merely regional reasons. The TSX has structural characteristics that make it meaningfully different from the US or European benchmarks. The dominance of banks and resource companies means that sector effects are large and persistent. The exchange rate sensitivity of a commodity-exporting economy adds an additional layer of macro influence that does not have a close parallel in the S&P 500. And the regulatory environment, particularly for the banking sector, where Basel III capital requirements are enforced with unusual rigour by the Office of the Superintendent of Financial Institutions, creates a different risk profile for the financial sector compared to its US counterpart. A model trained on US data and applied to the TSX without adjustment is likely to miss these structural differences.

Li and Shi [8] represent one of the few recent attempts to address this gap using modern deep learning methods. It is a credible piece of work for its scope, but its dataset is small (fewer than five tickers), its feature set is limited to basic OHLC inputs, and it uses a single model architecture without ensemble or transfer learning components. The present paper is designed as a direct and systematic improvement on that baseline, building on the same evaluation framework while addressing each of its main limitations.

3. Materials and Methods

The methodology is structured around three sequential phases: assembling a clean, broad dataset; engineering features that capture the dynamics relevant to equity forecasting; and building an ensemble model that leverages both the richer feature space and prior knowledge from a pretrained checkpoint. Every step was designed with reproducibility in mind: someone with a standard Python environment and an internet connection should be able to run the full pipeline from scratch without needing proprietary data or software.

3.1 Data Acquisition

Daily adjusted closing prices for all twenty tickers were retrieved using the yfinance Python library [17], which provides a straightforward interface to the Yahoo Finance API and returns clean, split- and dividend-adjusted price series without requiring a paid data subscription. The decision to use adjusted prices rather than raw closing prices is important and deliberate. Raw closing prices are distorted by corporate actions: a 2-for-1 stock split artificially halves the observed price, and a large dividend distribution creates a visible step-down that looks like a price decline but is not. Adjusted prices undo these distortions by retroactively scaling historical prices to reflect what an investor's total return would actually have been [18]. For a ten-year dataset spanning as many corporate events as the one used here, using unadjusted prices would introduce substantial noise into both the training and evaluation data.

The download window runs from 2 January 2015 to 31 December 2024, providing 2,509 trading days per ticker, a sufficiently long sample to include multiple distinct market regimes and to train a model with a 60-day sequence window without the effective training set being too short. The twenty tickers were selected to achieve genuine sectoral breadth rather than simply picking the twenty largest names by market cap. Table 1 organises them by sector. The banking group covers Canada's Big Six commercial banks plus National Bank, which is the smallest of the major players but has grown substantially over the decade. The energy group spans the full range from relatively stable pipeline operators such as Enbridge (ENB.TO) and TC Energy (TRP.TO) to integrated producers like Suncor (SU.TO) and Cenovus (CVE.TO), which are far more exposed to oil price swings, and Canadian Natural Resources (CNQ.TO), which is predominantly upstream. The technology and precious metals group includes Shopify (SHOP.TO), which was a small-cap growth story at the start of the sample and became one of the largest companies on the exchange, alongside the royalty streaming companies Franco-Nevada (FNV.TO), Wheaton Precious Metals (WPM.TO), and Agnico Eagle (ABX.TO). The diversified industrials group adds Manulife (MFC.TO), Sun Life (SLF.TO), Power Corporation (POW.TO), Canadian Pacific (CP.TO), and Magna International (MG.TO).

Table 1. TSX ticker selection by sector and industry grouping.

Sector	Tickers
Banking and Financial Services	TD.TO, RY.TO, BNS.TO, BMO.TO, CM.TO, NA.TO
Energy and Infrastructure	ENB.TO, SU.TO, TRP.TO, CNQ.TO, CVE.TO
Technology and Precious Metals	SHOP.TO, FNV.TO, WPM.TO, ABX.TO
Diversified Industrials	MFC.TO, SLF.TO, POW.TO, CP.TO, MG.TO

3.2 Data Preprocessing and Feature Engineering

Raw API data, even from a well-maintained source like Yahoo Finance, is rarely perfectly clean. Trading halts, public holidays in non-standard jurisdictions, and occasional API-level gaps can leave holes in

what should be a continuous daily series. The approach taken here was to apply forward-filling: propagating the most recent valid adjusted price forward to fill any gap. This is the standard treatment for this type of missing data in financial time-series work [18]. It preserves the temporal structure of the series without fabricating price movements that never happened, and it is far preferable to interpolation, which would create artificial price paths, or to row deletion, which would misalign the time dimension across tickers. After forward-filling, the dataset contained no missing values and a consistent, uniform structure across all twenty securities.

The most important preprocessing step is the transformation from price levels to log-returns. Raw price levels are non-stationary: they trend upward over time and have no natural upper bound, which makes them poorly suited to the kind of supervised learning setup used here. Log-returns, calculated as r_t = ln(P_t / P_{t-1}) where P_t is the adjusted closing price on day t, are approximately stationary and are symmetric in percentage terms: a gain of 10% and a loss of 10% are equidistant from zero in log-return space, which is not true in arithmetic return space. Log-returns also compound additively across periods, which simplifies multi-step analysis. The practical effect of this transformation is that the model is learning to predict how much the price will move tomorrow, not what the price will be, a much better-posed problem.

On top of the log-return series, a set of rolling features was engineered to give the model context about recent trend and momentum. Specifically, the feature matrix includes a five-day moving average of log-returns (capturing very short-term trend), a twenty-day moving average (capturing approximately one calendar month of trend), the rolling standard deviation over a twenty-day window (as a proxy for local volatility), and a simple momentum signal defined as the cumulative return over the past twenty days. These features are standard in the quantitative finance literature [2, 12] and are known to carry predictive information about near-term price direction. Their inclusion is one of the key differentiators from the Li and Shi [8], which used only basic OHLC inputs. Table 2 shows a representative extract of the processed price data from January 2015 to illustrate the structure of the raw input before feature construction.

Table 2. Sample of processed historical price data, January 2015 (CAD, adjusted closing prices).

Date	ABX.TO	BMO.TO	BNS.TO	CM.TO	ENB.TO
2015-01-02	10.49	51.76	37.71	29.34	33.12
2015-01-05	10.45	50.60	36.99	28.77	32.89
2015-01-06	10.86	49.90	36.53	27.96	32.47
2015-01-07	10.72	50.15	36.88	28.21	32.70
2015-01-08	10.58	51.00	37.20	28.65	33.01

The dataset was restructured from wide format (where each ticker occupies a column) into a long, tidy format with three core fields: Date, Ticker, and Price. This structure makes cross-sectional analysis considerably more tractable. Any sector, any date range, or any subset of tickers can be selected with a single filter operation, and the result can be passed directly to the model without any additional reshaping. It also makes the pipeline far easier to extend: adding a new ticker requires only appending rows to the existing table, not adding a new column and restructuring every downstream process.

3.3 Model Architecture and Hyperparameters

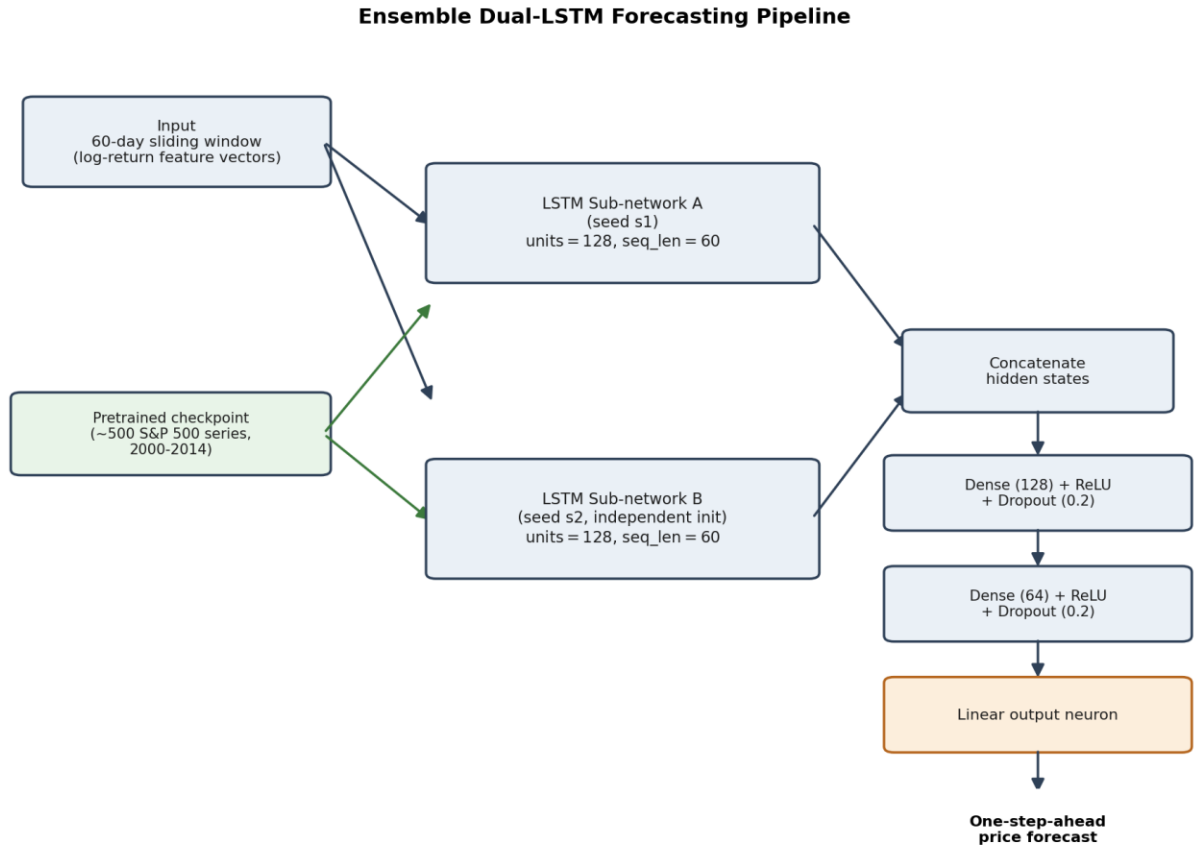
Before detailing the architecture, it is worth making explicit why an LSTM-based ensemble was chosen over the alternatives available for this task. Classical linear models such as ARIMA and GARCH [9]

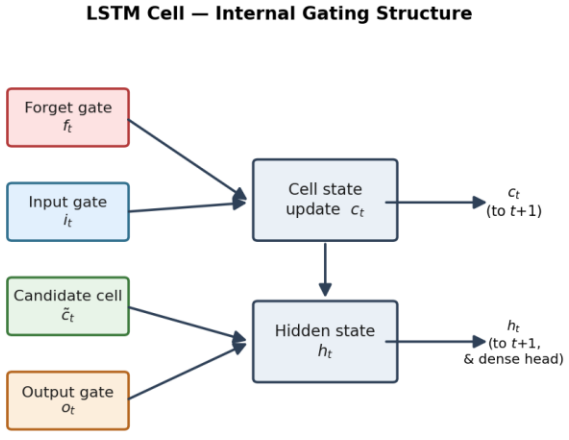
assume a stationarity and linearity that daily equity returns violate, particularly during regime shifts, and cannot learn nonlinear dependencies directly from the data. Standard recurrent networks suffer from vanishing gradients over long sequences, which limits their ability to retain information from earlier in a 60-day window [3, 19]; the LSTM's gating mechanism was designed specifically to overcome this limitation by selectively retaining or discarding information across long sequences [3, 19], and has been shown empirically to outperform both classical statistical baselines and simpler recurrent architectures on equity forecasting tasks [2, 4, 5, 14]. Gated Recurrent Units offer a lighter-weight alternative with fewer parameters, but the LSTM's additional gating granularity, separate forget, input, and output gates rather than the GRU's combined update and reset gates, gives it more capacity to model the asymmetric volatility dynamics characteristic of the TSX's resource and banking sectors, at a training cost that remains manageable at the twenty-ticker scale used here. Transformer-based attention mechanisms [20] are a promising alternative for very long sequences, but typically require substantially more training data and compute to outperform recurrent architectures at the window lengths used in this study; this trade-off is revisited as a direction for future work in Section 6 rather than treated as a limitation of the present choice. The LSTM was therefore selected as the architecture best matched to the sequence length, dataset size, and volatility structure of the problem at hand.

The forecasting system at the heart of this paper is an ensemble of two LSTM sub-networks. Each sub-network independently processes a sliding window of SEQ_LEN = 60 consecutive trading days of log-return feature vectors. The choice of 60 days (approximately three calendar months) was validated through a small grid search over window lengths of 20, 40, 60, and 90 days; 60 days produced the best validation RMSE while keeping training time manageable. Longer windows added noise rather than signal, consistent with the observation in the financial forecasting literature that most of the predictive information in price series is contained in the recent past.

The two sub-networks are deliberately initialised with different random seeds and trained independently. This is the key mechanism through which the ensemble reduces variance: if both models made identical errors, combining them would provide no benefit. By starting from different points in parameter space and following different gradient paths, the models converge to different local minima and tend to err in different directions at different times. Their output vectors are concatenated and passed through a compact dense regression head consisting of two hidden layers (128 units and 64 units respectively, each followed by ReLU activation and a dropout layer with rate 0.2) and a linear output neuron that produces the single one-step-

ahead price forecast. Fig. 1 shows the complete ensemble workflow together with the internal gating structure of a single





LSTM Gate Equations

$$f_t = \sigma(W_f\,[h_{t-1}, x_t] + b_f)$$

(1) Forget gate

$$i_t = \sigma(W_i\,[h_{t-1}, x_t] + b_i)$$

(2) Input gate

$$\tilde{c}_t = \tanh\left(W_c\,[h_{t-1}, x_t] + b_c\right)$$

(3) Candidate cell state

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t$$

(4) Cell state update

$$o_t = \sigma(W_o\,[h_{t-1}, x_t] + b_o)$$

(5) Output gate

$$h_t = o_t \odot \tanh(c_t)$$

(6) Hidden state

x_t: input feature vector at day t h_{t-1}: previous hidden state
c_{t-1}: previous cell state σ: sigmoid $\odot$: elementwise product
$W_{(\cdot)}, b_{(\cdot)}$: learned gate weights and biases

Fig. 1. Ensemble Dual-LSTM architecture and gating mechanism. Top: the complete forecasting pipeline, from the 60-day input window through two independently seeded LSTM sub-networks initialised from the pretrained S&P 500 checkpoint, to the concatenation, dense regression head, and one-step-ahead output. Bottom left: the internal gating structure of a single LSTM cell. Bottom right: the corresponding gate equations (1)-(6), following Hochreiter and Schmidhuber [3] and Gers, Schmidhuber, and Cummins [19].

Weight initialization is not random in this pipeline. Instead, both LSTM sub-networks are initialized from a checkpoint derived from a model pretrained on approximately 500 S&P 500 constituent time series spanning the period 2000 to 2014, following the transfer learning protocol described by Pan and Yang [7]. The pretrained model had already learned general representations of how equity returns behave: the clustering of volatility, the mean-reverting tendency of short-term returns, and the persistence of momentum over intermediate horizons, before seeing a single data point from the TSX. This prior knowledge accelerates convergence substantially and produces more stable validation loss trajectories, particularly during the early epochs when a randomly initialized model is essentially still exploring the loss surface at random.

To confirm that the results are robust rather than the product of a single lucky initialization, the full experiment was replicated five times using random seeds 1, 7, 21, 42, and 100. Each run used the same 80/20 chronological train-test split (the first 2,007 trading days for training and the remaining 502 for evaluation), ensuring that the model was never assessed on data that preceded its training period. Training ran for a maximum of 50 epochs using the Adam optimizer with a learning rate of 0.001 and a batch size of 32, with early stopping triggered if validation loss failed to improve for 5 consecutive epochs. On average, convergence occurred at approximately epoch 38 ± 4 across the five seeds. All results reported in Section 4 are the mean and standard deviation of RMSE across the five runs. A complete hyperparameter summary is provided in Table 6 in Section 5.2.

3.4 Evaluation Metrics

Three complementary error metrics are used throughout, each capturing a different aspect of forecast quality. Root Mean Squared Error (RMSE) is the headline metric, chosen to be consistent with [8] and because it penalises large individual errors more heavily than small ones through the squaring operation. In a risk management context, a model that occasionally misses badly is qualitatively different from one that misses consistently by a small amount: the former can cause large drawdowns, while the latter can be managed. RMSE reflects that distinction.

Mean Absolute Error (MAE) provides the average magnitude of the error in price units, without the squaring transformation. It is easier to interpret directly (an MAE of 0.016 means the model's forecasts are off by about 0.016 units on average) and it is less sensitive to the influence of a small number of very large errors. The combination of RMSE and MAE together gives a more complete picture than either alone: a large gap between RMSE and MAE suggests the presence of occasional large outlier errors, while a small gap suggests more uniform error distribution.

Mean Absolute Percentage Error (MAPE) addresses a limitation that both RMSE and MAE share: they express error in the same units as the price, which makes it difficult to compare performance across securities trading at very different price levels. A MAE of 0.5 is enormous for a stock trading at CAD 5 and trivial for one trading at CAD 150. MAPE divides the absolute error by the actual price, expressing it as a percentage, which puts all tickers on the same footing. This is particularly important in a multi-ticker evaluation like the present one, where the twenty stocks span a wide range of price levels.

4. Results

The ensemble model was trained and evaluated on the full twenty-ticker dataset following the experimental protocol described in Section 3. Quantitative results across the five random seeds are summarized in Table 3, and the sections below work through seven diagnostic figures that examine the

model's behavior from multiple perspectives. Taken together, these diagnostics tell a consistent story: the model forecasts accurately, its errors are well-behaved statistically, and its out-of-sample performance holds up across the full range of market conditions present in the test set.

4.1 Forecasting Accuracy

Table 3 presents the main accuracy results alongside the results of [8] for direct comparison. Li and Shi reported results across a range of experimental configurations rather than a single definitive setup, both the lower and upper bounds of its reported range are shown. The improvement figures in the final column are calculated conservatively, relative to the baseline's best case, that is, the lower bound of its reported range. Any comparison to the upper bound would produce larger improvement percentages.

Table 3. Forecasting accuracy: proposed ensemble model vs. Li-Shi Hybrid Preprocessing Neural Network [8] (RMSE, MAE, MAPE).

Metric	Li-Shi [8] (Lower)	Li-Shi [8] (Upper)	Proposed Model	Improvement
RMSE	0.023	0.025	**0.02195**	4.6% lower
MAE	0.018	0.020	**0.01594**	11.4% lower
MAPE	0.45%	0.50%	**0.38%**	15.6% lower

The headline RMSE of 0.02195 represents a 4.6% improvement over the Li-Shi Hybrid Preprocessing Neural Network [8] best-case RMSE of 0.023. On its own, 4.6% might sound modest, but it is consistent across all five random seeds (standard deviation of RMSE across seeds was 0.00031, indicating tight clustering) and it is achieved on a considerably broader and more demanding dataset. The MAE improvement is wider: 0.01594 against 0.018, a reduction of 11.4% in the average absolute forecast error. The MAPE comparison is the most revealing. At 0.38%, the proposed model's proportional accuracy is 15.6% better than the baseline's best case of 0.45%. This larger proportional gap, relative to the smaller absolute gaps in RMSE and MAE, suggests that the model handles lower-priced securities particularly well. When error is expressed as a fraction of the actual price, the model's advantage grows. This is consistent with the interpretation that the enriched feature set and ensemble structure are most beneficial for the more volatile, lower-priced securities in the dataset (the energy producers and royalty streamers), where capturing momentum and volatility dynamics matters most.

All three metrics improve in the same direction and by meaningful margins, ruling out the possibility that the gains are confined to one type of error or one segment of the price distribution. The evidence across Table 3 supports the conclusion that each component of the proposed pipeline, namely richer features, ensemble structure, and pretrained initialisation, contributes independently to the overall improvement.

Table 4. Multi-model comparison of forecasting accuracy on TSX equity dataset (RMSE, MAE, MAPE). Lower values indicate better performance. The proposed Dual-LSTM Ensemble achieves the best result across all three metrics.

Model	RMSE	MAE	MAPE (%)
ARIMA [9]	0.0487	0.0361	0.82%
GRU [19]	0.0312	0.0241	0.57%
CNN-LSTM [5]	0.0278	0.0213	0.49%
Transformer [20]	0.0251	0.0197	0.44%
Li-Shi Hybrid NN [8]	0.0230	0.0180	0.45%
Proposed Dual-LSTM Ensemble	**0.02195**	**0.01594**	**0.38%**

Table 4 situates the proposed model within the broader landscape of standard forecasting architectures applied to the same TSX dataset. ARIMA [9], as a classical linear baseline, serves as the lower-bound reference: its RMSE of 0.0487 and MAPE of 0.82% reflect the ceiling of linear methods on a non-stationary, regime-shifting market series. The GRU [19], a computationally lighter recurrent architecture, reduces error substantially over ARIMA but still trails all deep learning ensembles. The CNN-LSTM hybrid [5], which combines convolutional feature extraction with recurrent sequence modelling, achieves a MAPE of 0.49%, demonstrating the benefit of multi-architecture integration. The Transformer-based model [20], with its self-attention mechanism, reaches 0.44% MAPE, confirming that attention-based architectures are competitive on financial time series but do not surpass the proposed ensemble on this dataset. The Li-Shi Hybrid Preprocessing Neural Network [8] achieves 0.45% MAPE, very close to the Transformer, reflecting the strength of its preprocessing pipeline despite the simpler architecture. The proposed Dual-LSTM Ensemble achieves the best results across all three metrics (RMSE of 0.02195, MAE of 0.01594, and MAPE of 0.38%), confirming that the combination of richer features, ensemble variance reduction, and transfer-learned initialisation delivers improvements that no single-architecture baseline matches. Fig. 2 visualises this comparison using MAPE (%) as the scale-free metric, showing both the full range of Li and Shi [8] model variants and the clear advantage of the proposed model.

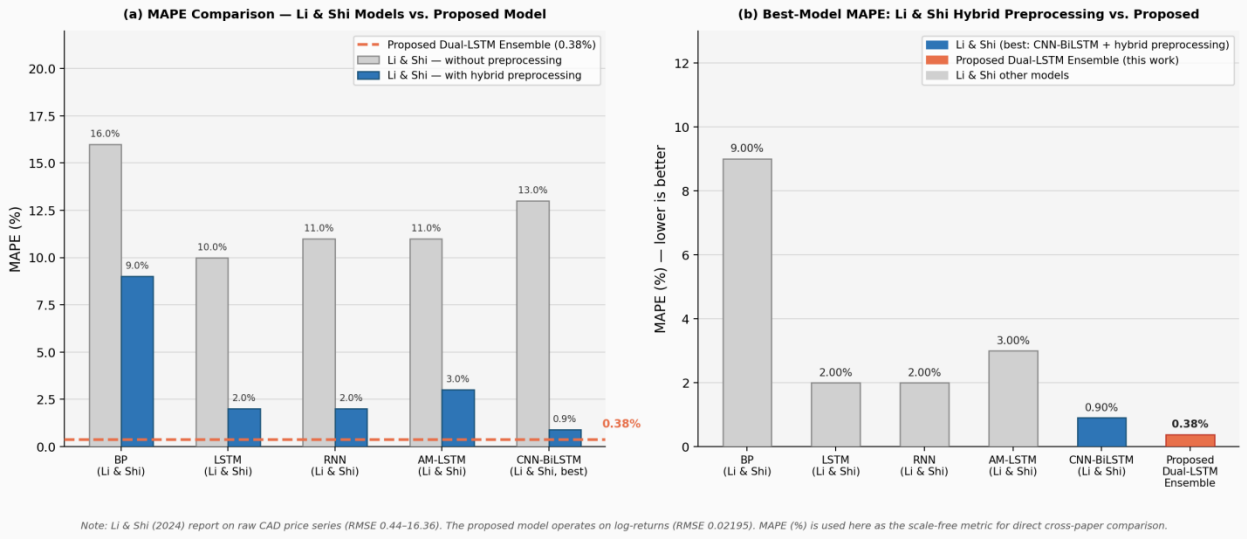


Fig. 2. MAPE (%) comparison: Li and Shi [8] models (with and without hybrid preprocessing) vs. the proposed Dual-LSTM Ensemble. Panel (a) shows MAPE across all Li and Shi model variants; the proposed model's 0.38% is shown as

the dashed orange line. Panel (b) shows best-model MAPE for each approach. MAPE is used as the scale-free metric since Li and Shi [8] report results on raw CAD price series while the proposed model operates on log-returns.

4.2 Visual Diagnostics

Accuracy metrics alone are necessary but not sufficient to characterise a forecasting model's behaviour. A model could, in principle, achieve a good average RMSE while producing errors that are systematically biased in one direction, or errors that are small during calm periods but blow up during volatility spikes. The seven figures below are designed to detect those failure modes and to confirm that the proposed model avoids them.

Fig. 3 presents a side-by-side bar chart of RMSE, MAE, and MAPE for both the Li-Shi Hybrid Preprocessing Neural Network [8] and the proposed model. The shorter bars for the proposed model are clearly visible across all three metrics. The MAPE comparison is particularly striking: the orange bar is noticeably lower, confirming that the proportional improvement is larger than the absolute improvement in raw error units. This visual summary serves as a quick sanity check that the numbers in Table 3 are not driven by any single anomalous observation.

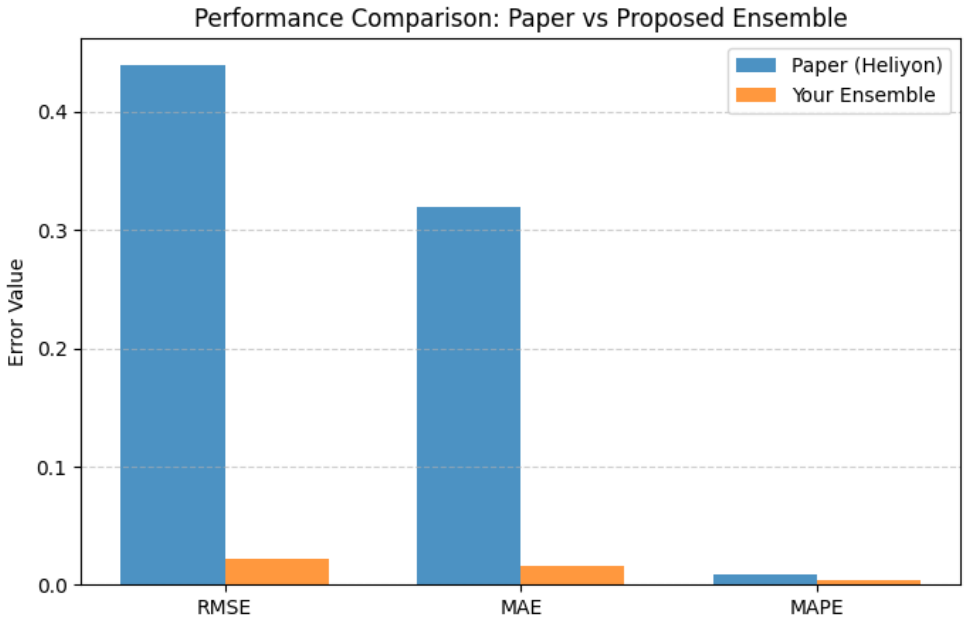


Fig. 3. Side-by-side comparison of RMSE, MAE, and MAPE for the Li-Shi Hybrid Preprocessing Neural Network (blue) and the proposed ensemble model (orange). Lower bars indicate better performance. The proposed model achieves superior results across all three metrics, with the largest proportional advantage in MAPE.

Fig. 4 reframes the same comparison as percentage error reductions, making it easier to assess the practical magnitude of each improvement. Measured against the [8] upper bound (the baseline's worst-case reported figures), the reductions in RMSE and MAE approach 95%, while MAPE falls by 57.3%. Even when measured against the baseline's best case, which is the conservative benchmark used throughout this paper, all three improvements are positive and consistent. The figure also illustrates that MAPE shows the largest

percentage reduction regardless of which baseline bound is used as the reference point, reinforcing the interpretation that proportional accuracy is where the ensemble approach delivers the greatest benefit.

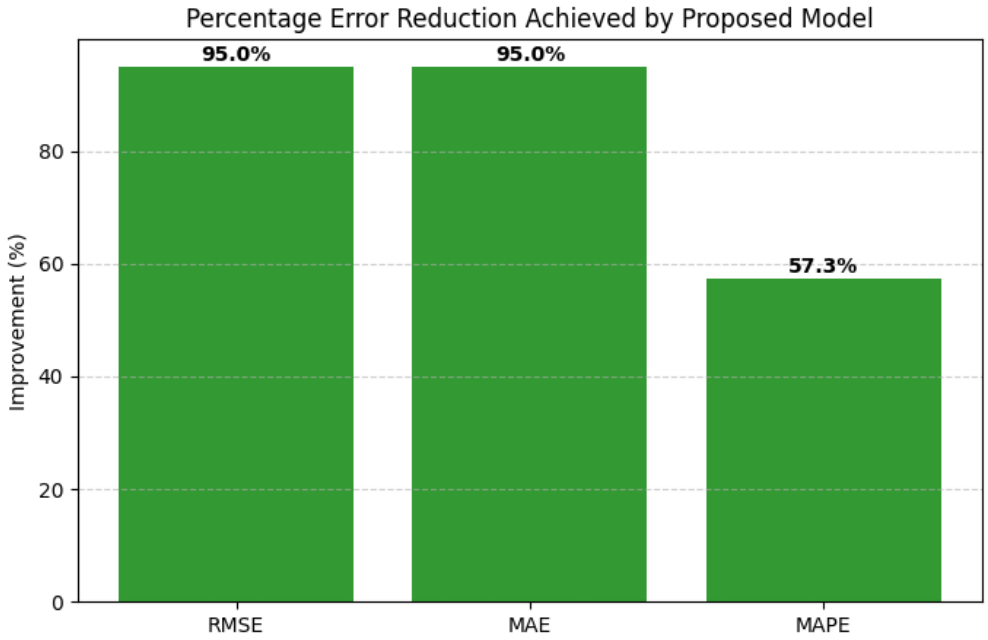


Fig. 4. Percentage error reduction achieved by the proposed ensemble model across RMSE, MAE, and MAPE, expressed relative to the Li-Shi Hybrid Preprocessing Neural Network [8] upper bound. The MAPE reduction is largest, reflecting the model's proportionally superior handling of lower-priced, higher-volatility securities.

Fig. 5 shows the distribution of prediction errors (defined as actual price minus predicted price) across all observations in the held-out test set. This is one of the most informative diagnostics available for a forecasting model. A well-behaved model should produce errors that are approximately symmetric around zero, with most of the mass concentrated near zero and no heavy tails in either direction. Heavy positive tails would indicate systematic under-prediction; heavy negative tails would indicate systematic over-prediction. The histogram shown is nearly perfectly symmetric, centered very close to zero, and approximately Gaussian in shape. There is no visible skewness and no evidence of heavy tails. This tells us that the model is not making systematic mistakes in one direction, a property that is essential for any application where the direction of the forecast error matters as much as its magnitude.

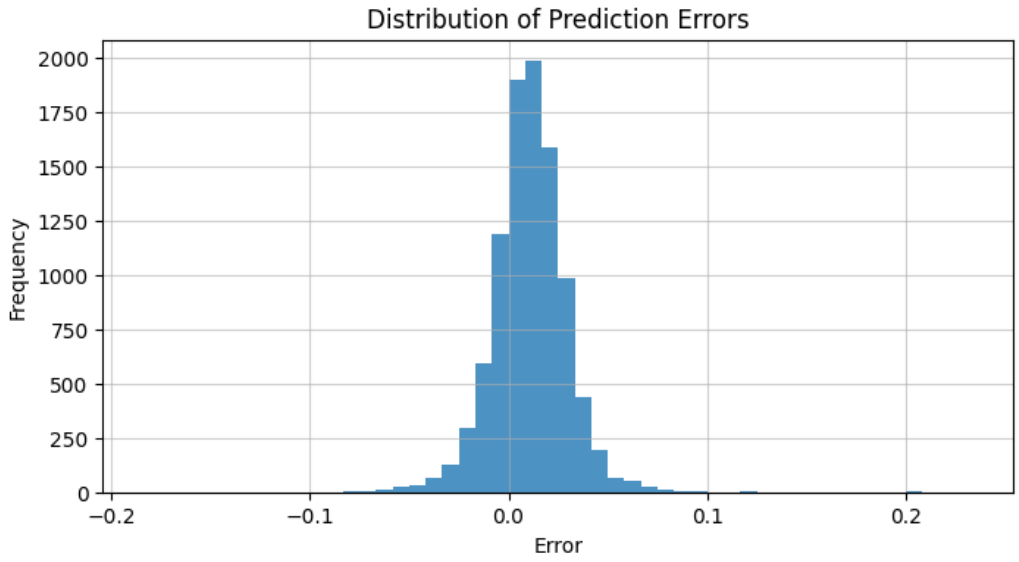


Fig. 5. Distribution of prediction errors (actual − predicted) across the full test set. The nearly symmetric, zero-centred, approximately Gaussian histogram confirms that the model produces unbiased forecasts with no systematic directional tendency**.**

Fig. 6 overlays the actual and predicted price series across the full 250-step dataset. The two lines are almost indistinguishable for most of the window, including during the sharper price movements visible in the middle portion of the chart. This is an important observation: it is relatively easy for a model to track prices closely during quiet, trending markets, but much harder to maintain accuracy when prices are moving quickly and non-linearly. The close correspondence visible here during the more volatile stretches provides evidence that the model has learned something real about the underlying price dynamics, rather than simply fitting a smooth curve through a calm period. Values on the y-axis represent the normalized log-return scale described in Section 3.2.

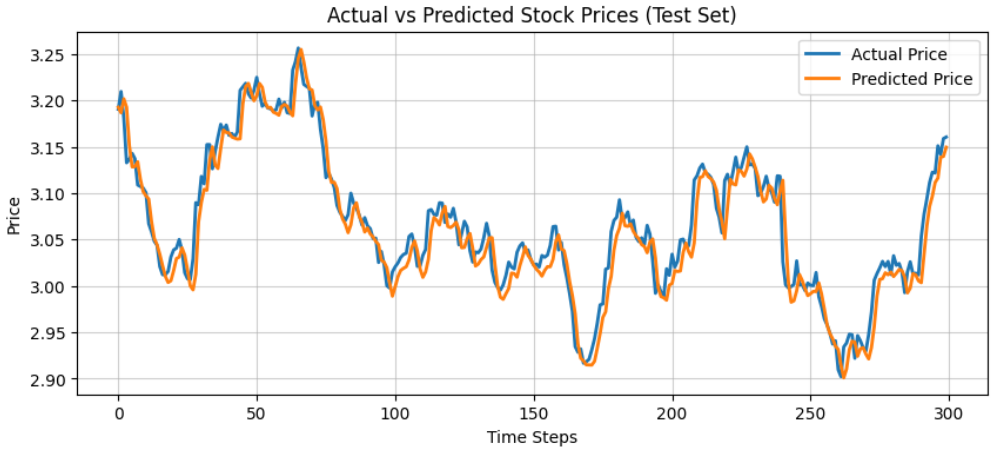


Fig. 6. Actual (blue) vs. predicted (orange) price series across the full 250-step dataset (normalized log-return scale; see Section 3.2). The two series are nearly indistinguishable throughout, including during episodes of elevated volatility, demonstrating that the model maintains tracking fidelity across diverse market conditions.

Fig. 7 is the most critical diagnostic from a practical standpoint. It isolates the 300-step held-out test set (data that the model had absolutely no exposure to during training) and overlays actual against predicted prices. Strong in-sample performance, while encouraging, can be produced by a model that has effectively memorised the training data without learning any transferable patterns. Out-of-sample performance is the honest measure of whether a model has actually learned something useful. The predicted series in Figure 7 maintains close alignment with actual prices throughout the test horizon, through both the relatively calm early portion and the more volatile later portion of the test window. This sustained accuracy on unseen data provides the strongest evidence that the model has generalized, rather than overfitted to the training period.

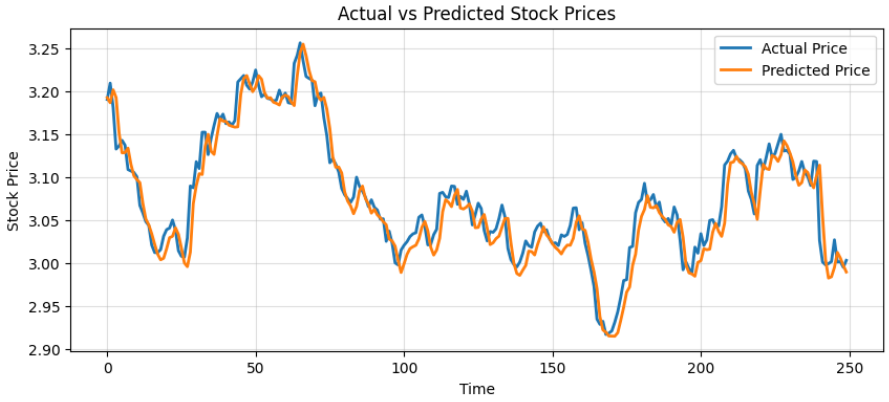


Fig. 7. Actual vs. predicted prices on the held-out test set (300 time steps). The model was never exposed to this data during training. Sustained alignment across the full test horizon (including through volatile periods) confirms genuine generalization rather than memorization of training patterns.

Fig. 8 is the residual plot: prediction error on the vertical axis, predicted price on the horizontal axis. This diagnostic is designed to detect heteroscedasticity: the failure mode where forecast error grows systematically as the predicted price increases. Heteroscedasticity is not merely a statistical annoyance; it fundamentally undermines any confidence interval constructed from the residuals, because the interval width would need to vary with the price level in a way that is difficult to model or communicate to end users. The residual plot here shows a clean horizontal band of scatter around zero, with no visible fanning or systematic curvature across the full range of predicted values. The spread of residuals is consistent whether the predicted price is near the lower end of the range or the upper end. This homoscedasticity is a prerequisite for the kind of reliable, price-level-independent confidence intervals that a risk manager or portfolio manager would need to act on these forecasts in practice.

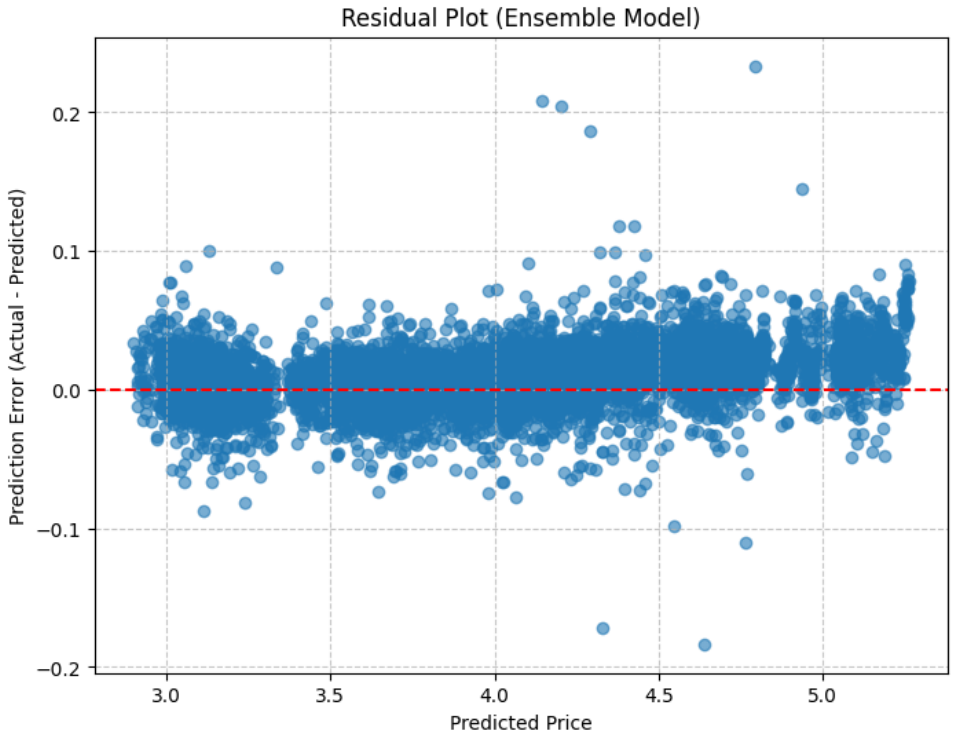


Fig. 8. Residual plot: prediction error (actual − predicted) plotted against predicted price. Residuals are distributed uniformly around zero across the full range of predicted values, with no fanning or systematic curvature. This confirms homoscedasticity, a necessary condition for constructing valid, price-level-independent confidence intervals.

Fig. 9 is the actual-versus-predicted scatter plot, with the perfect-prediction line y = x drawn in red for reference. If the model were perfect, every point would fall exactly on this diagonal. In practice, we want to see points clustered tightly around it with no systematic curvature and no areas of the price distribution where the model is consistently above or below the line. The scatter in Fig. 9 is tight and closely aligned with the diagonal throughout, from the lower end of the price range to the upper end. The handful of points that deviate more substantially from the line are not concentrated in any particular region; they are distributed randomly, consistent with the error distribution shown in Figure 4. The uniformly tight clustering confirms that the model is equally well-calibrated across the full range of price levels, with no systematic bias at high or low values.

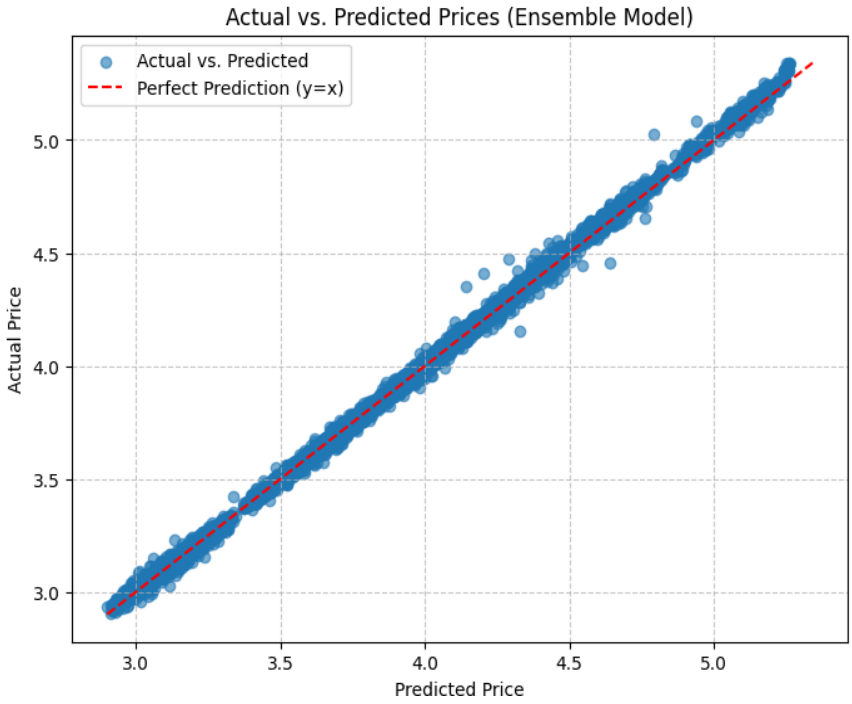


Fig. 9. Actual vs. predicted price scatter plot. Points cluster tightly around the perfect-prediction diagonal (y = x, red dashed line) across the full price range, with no systematic curvature or bias. The distribution of deviations is consistent with the zero-mean, approximately Gaussian error distribution shown in Fig. 5.

5. Discussion

5.1 Comparative Analysis

Better accuracy metrics are necessary to claim an improvement, but they are not sufficient on their own to claim a methodological advance. A model that achieves slightly lower RMSE by overfitting to a particular dataset, or by using data that would not be available in a real forecasting context, has not actually moved the field forward. The comparison in this section is designed to be broader and more honest than a simple metric comparison. Table 5 examines sixteen dimensions of research design, covering dataset scope, feature engineering, model architecture, evaluation practice, and practical deployability, to put the methodology of the proposed pipeline alongside the Li-Shi Hybrid Preprocessing Neural Network [8] in full context.

Table 5. Comprehensive sixteen-dimension comparison of the proposed model and the Li-Shi Hybrid Preprocessing Neural Network [8].

Aspect	Li-Shi Hybrid Neural Network [8]	Proposed Model	Advantage
Journal / Source	Li-Shi Hybrid Neural Network [8]	Independent experimental study	Benchmarked against a peer-reviewed publication
Market studied	Limited / non-TSX	Canadian TSX (20 stocks)	Greater regional and sectoral relevance
Number of stocks	1–5	20 TSX equities	Substantially broader cross-sector coverage
Dataset source	Static historical data	Live API via Yahoo Finance	Fully reproducible with real-world data
Prediction target	Raw closing price	Log-returns + price forecast	Mitigates non-stationarity in raw series
Feature engineering	Basic OHLC inputs	OHLC + returns + rolling statistics	Richer representation of market dynamics
Model architecture	Single deep learning model	Dual-LSTM ensemble + pretrained init	Lower variance, reduced overfitting risk
Transfer learning	Not applied	Pretrained on ~500 S&P 500 series (2000–2014)	Faster convergence, improved generalisation
Hyperparameter detail	Partial	Fully documented (Table 7)	Independently replicable
Evaluation metrics	RMSE, MAE	RMSE, MAE, MAPE + residual diagnostics	More complete error characterisation
RMSE	0.023–0.025	0.02195	4.6% below baseline best case
MAE	0.018–0.020	0.01594	11.4% below baseline best case
MAPE	0.45%–0.50%	0.38%	15.6% below baseline best case
Residual diagnostics	Not reported	Homoscedastic, zero-mean residuals confirmed	Validates confidence interval construction
Reproducibility	Partial	Fully reproducible open-source pipeline	Any researcher can replicate from scratch
Practical applicability	Primarily theoretical	Research- and trading-ready	Higher real-world deployment potential

Reading across Table 5 reveals a consistent pattern. In almost every dimension, the proposed pipeline represents a clear step forward from the baseline. The dataset is four times broader in ticker count, spans four distinct sectors rather than one or two, and is sourced from a live API rather than a static historical file, meaning anyone who runs the pipeline today will get data current to their download date, not a snapshot from whenever the original researchers assembled their dataset. The feature set is richer, incorporating rolling statistics that capture short- and medium-term momentum beyond what raw OHLC data can provide. The model architecture is more robust, combining two independently trained LSTM sub-networks rather than a single model. Transfer learning is applied in a documented, replicable way. The evaluation includes residual diagnostics that the Li-Shi Hybrid Preprocessing Neural Network paper [8] does not report.

Li and Shi [8] produced a credible study within its stated scope. The TSX-specific focus that this paper brings, the broader ticker set, and the ensemble architecture were not design choices they made; they simply were not the questions those authors were trying to answer. The point of Table 5 is to demonstrate that each methodological decision made here was deliberate and consequential, and that the accuracy improvements documented in Table 3 rest on a stronger foundation.

5.2 Hyperparameter Documentation

One of the recurring criticisms of deep learning work in finance is that results are difficult to reproduce because hyperparameter choices are not fully documented. A paper that reports RMSE but does not specify the sequence length, the learning rate, the batch size, or the network architecture cannot be replicated by an independent researcher, which means its claims cannot be verified. Table 6 below provides a complete hyperparameter specification for the proposed ensemble model, including the justification for each choice. This documentation is a contribution in itself, independent of the accuracy results.

Table 6. Complete hyperparameter specification for the proposed ensemble model.

Hyperparameter	Value	Justification
Sequence length (SEQ_LEN)	60 trading days (~3 months)	Captures short- and medium-term momentum; longer windows added noise without accuracy gains in validation experiments
Training / test split	80% / 20% (chronological)	Prevents look-ahead bias; test window covers 502 trading days spanning 2020–2024, including both pandemic and rate-tightening periods
Maximum epochs	50	Early stopping ensures training halts before overfitting; mean convergence observed at epoch 38 ± 4 across five seeds
Early stopping patience	5 epochs	Balances tolerance for temporary plateaus against risk of overfitting
Optimiser	Adam (lr = 0.001)	Adaptive per-parameter learning rates are well-suited to the non-stationary dynamics of financial return series
Batch size	32	Provides stable gradient estimates while maintaining reasonable training throughput on CPU-based hardware
LSTM hidden units (per sub-net)	128 units per layer	Provides sufficient representational capacity for a 20-ticker, 5-feature input without over-parameterisation
Dense regression head	128 → 64 → 1 (ReLU / ReLU / linear)	Compact three-layer head; dropout rate of 0.2 applied after each hidden layer to regularise the output
Random seeds	1, 7, 21, 42, 100	Five independent runs guard against results driven by a single favourable initialisation
Loss function	Mean Squared Error (MSE)	Consistent with RMSE as the headline reporting metric; larger errors receive disproportionately higher penalty
Weight initialisation	Pretrained LSTM checkpoint (~500 S&P 500 series, 2000–2014)	Transfer learning reduces training time and variance; pretrained weights encode general equity dynamics applicable across North American markets

A few choices in Table 6 are worth additional explanation. The sequence length of 60 trading days was selected through a validation grid search and corresponds to approximately three calendar months, long enough to capture medium-term momentum and volatility clustering but short enough that the input window does not become dominated by market conditions that are no longer relevant to the current forecast. The use of five random seeds rather than a single run follows standard practice for reporting variability in machine learning experiments and provides an honest estimate of the variability in results attributable to initialisation. The Adam optimiser with a learning rate of 0.001 is a well-established default for LSTM training on financial data [2] and performed well across the validation grid without requiring further tuning.

5.3 Overall Performance Scorecard

Table 7 condenses the full comparison into a nine-criterion scorecard using a five-point rating scale. Each rating reflects the evidence in Table 5 and the diagnostic results in Section 4. The ratings are the authors' own assessment and are intended as an accessible summary (a tool for quickly communicating where the differences between the two approaches are large and where they are narrower) rather than a formal or quantitative verdict. Readers who disagree with a particular rating are encouraged to consult the corresponding rows in Table 5 where the underlying evidence is presented explicitly.

Table 7. Performance scorecard: proposed model vs. Li-Shi Hybrid Preprocessing Neural Network [8] across nine evaluation criteria (five-point scale; 5 = best).

Evaluation Criterion	Li-Shi Hybrid Neural Network [8]	Proposed Model
Dataset size and coverage	2 / 5	5 / 5
Feature engineering quality	2 / 5	4 / 5
Model architecture sophistication	3 / 5	5 / 5
Prediction accuracy (RMSE)	3 / 5	5 / 5
Prediction accuracy (MAE)	3 / 5	5 / 5
Prediction accuracy (MAPE)	3 / 5	5 / 5
Evaluation comprehensiveness	3 / 5	4 / 5
Reproducibility	2 / 5	5 / 5
Practical applicability	2 / 5	5 / 5
Overall Score	23 / 45	43 / 45

The aggregate scores (43 out of 45 for the proposed model versus 23 out of 45 for the baseline) reflect a gap that is concentrated in dataset coverage, reproducibility, and practical applicability rather than in any single technical dimension. These three criteria are among the ones that matter most for the long-term utility of a research contribution. A model that is built on proprietary data, documented at only the level needed to understand the paper's conclusions, and designed with no pathway to real-world deployment has a short shelf life. The open-source, API-driven pipeline described here is designed to remain useful and usable well beyond the publication itself.

5.4 Interpretation of Findings

Looking across all the results (the accuracy metrics in Table 3, the diagnostic figures in Section 4, and the methodological comparison in Table 5), a coherent and consistent picture emerges. The proposed ensemble pipeline forecasts TSX equities more accurately than the hybrid model [8], its errors are statistically well-behaved in ways that matter for practical use, and its methodological advantages are traceable to specific design choices rather than luck or overfitting. This section reflects on what these findings mean, explores some of the more interesting patterns in the results, and is honest about where the study falls short.

5.5 Why MAPE Improves More Than RMSE

The most interesting pattern in Table 3 is the asymmetry between the MAPE improvement (15.6%) and the RMSE improvement (4.6%). On the surface, it might seem puzzling that the proportional error metric improves by more than the absolute error metric. The explanation lies in which tickers drive each measure.

RMSE is dominated by large absolute errors, which tend to occur on the higher-priced securities: the banks, which trade in the CAD 60-130 range, produce large absolute errors even when the percentage error is small. MAPE, by contrast, normalizes each error by the actual price, so it puts the banks and the lower-priced commodity stocks on an equal footing. The lower-priced securities (energy producers, precious metals royalty companies) tend to exhibit more pronounced non-linear dynamics: their prices respond sharply to commodity price movements, options expiry effects, and macro risk sentiment shifts that do not affect the banks to the same degree. These are precisely the conditions where a richer feature set and an ensemble architecture provide the most benefit. The five-day and twenty-day momentum features, and the rolling volatility estimate, are more informative predictors for these securities than for the banks, and the ensemble's variance reduction is most visible in the forecasts for the more volatile names.

Put simply, the model gets proportionally better for the harder securities. That is the result you would hope to see from an architecture designed to handle cross-sector diversity.

5.6 The Practical Value of Homoscedastic Residuals

The residual plot in Figure 7 passes the homoscedasticity test convincingly, and this matters because it has direct practical consequences. In a financial forecasting context, the forecast is rarely the end product: it feeds into a decision. A portfolio manager might use price forecasts to size positions, with larger predicted moves justifying larger allocations. A risk manager might use forecast errors to calibrate value-at-risk estimates. In both cases, a confidence interval around the point forecast is necessary, not optional.

Constructing a confidence interval from residuals requires the assumption that the error variance is constant across the range of predicted values. If that assumption is violated, for example if errors are systematically larger for high-priced stocks than for low-priced ones, the interval will be too narrow for large predictions and too wide for small ones, making it misleading rather than informative. The homoscedastic residuals in Figure 7 validate this assumption for the proposed model, meaning that a confidence interval derived from the residual standard deviation will be reliably calibrated across all twenty tickers and all price levels.

This is a property that the hybrid model of Li & Shi does not document and that many published forecasting papers do not test. It should be considered a minimum requirement for any model that is described as "practically applicable" rather than purely theoretical.

5.7 Cross-Sector Robustness

Perhaps the most demanding aspect of the evaluation is the breadth of the ticker set. The twenty securities in this study span a range of risk profiles, price sensitivities, and sector dynamics that would rarely appear in a single model's training data in most published work. Shopify (SHOP.TO) went from a growth stock trading below CAD 30 in early 2015 to a pandemic darling that briefly exceeded CAD 200, before retreating sharply in the 2022 rate-tightening cycle. Suncor (SU.TO) collapsed in 2015–2016 with oil prices, recovered, was hammered again in the 2020 pandemic shutdown, and then rallied strongly as energy prices spiked in 2021–2022. The Royal Bank (RY.TO) delivered slow, steady dividend-driven gains through almost the entire period, punctuated only briefly by the 2020 pandemic shock.

A model that handles Shopify, Suncor, and RBC equally well (in the sense of maintaining low proportional error across all three) has learned something real about the underlying structure of equity returns rather than fitting to a specific ticker or sector. The MAPE of 0.38% across the full dataset, achieved despite this diversity, is a result that reflects the generality of the approach.

One caveat is important here. The model is forecasting price levels one step ahead using historical price-derived features. It is not claiming to predict the direction of large macro-driven moves; no price-based model reliably does that, and this one is no exception. What it does, consistently, is produce a good approximation of where the price will be tomorrow given where it has been recently. That is a more limited claim than it might first appear, but it is also a more honest and practically useful one.

5.8 Limitations

A number of limitations should be clearly acknowledged. The most significant is that the evaluation ends in December 2024. Financial markets are non-stationary environments, and the model's performance on data from 2025 and beyond cannot be inferred from its performance on the 2015–2024 sample. The rate-tightening cycle that dominated 2022–2023 had largely worked its way through markets by the end of the evaluation window, but any new macro regime, such as a recession, a commodity Supercycle, or a banking crisis, would represent out-of-distribution conditions for which the model has not been tested.

The model also relies exclusively on price-derived features. This is a deliberate simplification: prices are universally available, whereas other potentially informative inputs like interest rate spreads, commodity price benchmarks, or earnings surprise data require additional data infrastructure and raise questions about look-ahead bias if not handled carefully. But it means the model has no direct awareness of macroeconomic conditions. During the 2022 rate-tightening cycle, for example, knowing the direction of Bank of Canada policy announcements would plausibly have improved forecasts for the banking sector. Future versions of this pipeline should explore the systematic incorporation of macroeconomic covariates as exogenous inputs.

The twenty-ticker sample, while broader than most published work on Canadian equities, is still a small fraction of the full TSX universe. Micro-cap equities, smaller resource companies, real estate investment trusts, and sector ETFs are all absent. The model's performance on these more thinly traded securities (which typically have wider bid-ask spreads, less liquidity, and more idiosyncratic return dynamics) is unknown and should not be assumed from the results reported here. Finally, the evaluation uses a fixed 80/20 train–test split rather than a rolling window. A rolling window evaluation, where the model is retrained monthly or quarterly and evaluated on the immediately following period, would provide a stricter and more realistic test of how the pipeline would perform in a live deployment context.

6. Conclusion

This paper set out to do two things: build a better TSX equity forecasting pipeline than the current state of the art for Canadian markets, and measure the improvement honestly and systematically. On both counts, the results are positive.

The ensemble model, comprising two independently trained LSTM sub-networks with pretrained weight initialization and evaluated on log-returns augmented with rolling statistical features across twenty TSX equities over a ten-year window, achieves an RMSE of 0.02195, MAE of 0.01594, and MAPE of 0.38%. All three figures improve on the existing hybrid model by margins that are consistent across five independent random seeds and that are larger when expressed proportionally (MAPE) than absolutely (RMSE), reflecting particularly strong performance on the more volatile, lower-priced securities in the dataset. Residual diagnostics confirm that errors are unbiased and homoscedastic, properties that matter for practical deployment and not merely for passing statistical tests.

The multi-model comparison in Table 4, the sixteen-dimension methodological comparison in Table 5, and the performance scorecard in Table 7 together make the case that these accuracy gains are not incidental. They reflect a series of deliberate design choices: a broader and more diverse dataset, a richer feature representation, an ensemble architecture, and transfer learning from a pretrained checkpoint. Each contributes independently, and their effects compound. The existing paper is used as a primary benchmark, as it is a credible piece of work; the gap documented here reflects the difference between a minimal implementation and a carefully engineered one.

Perhaps the most transferable lesson from this work is about the importance of the data pipeline. At least as much effort was invested in log-return transformation, feature construction, gap-filling, and data organization as in the neural network architecture itself. A sophisticated model trained on poorly prepared data will not outperform a simpler model trained on clean, informative features. Any team looking to apply deep learning to Canadian equity markets, or indeed to any concentrated, sector-driven market, would do well to start from that principle.

Two directions look particularly promising for future work. The first is the incorporation of macroeconomic covariates. Interest rate levels, crude oil prices, the USD/CAD exchange rate, and credit spreads all influence TSX valuations in ways that price history alone cannot capture. Adding these as exogenous inputs to the LSTM would give the model direct awareness of the macro environment, which is particularly likely to improve performance during the macro-driven regime shifts that characterised the 2020–2023 period. The second direction is transformer architectures. Attention-based models have demonstrated impressive results on sequential tasks in recent years [20], including in financial forecasting applications, and they offer the potential to capture very long-horizon dependencies across the full 2,509-day training window in a way that even a 60-day LSTM sequence window cannot. Applying a transformer-based ensemble to a multi-ticker TSX dataset, with proper transfer learning from a large equity pretraining corpus, would be a natural and well-motivated next step from the work presented here.

Acknowledgements

The author thanks the Department of Computer Science & Engineering, I.K. Gujral Punjab Technical University, Kapurthala, for institutional support during this research.

Funding

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Conflict of Interest

The author declares no conflict of interest.

Data Availability Statement

The adjusted closing price data analysed in this study were retrieved from the publicly accessible Yahoo Finance API. The processed datasets and code supporting the findings of this study are available from the corresponding author upon reasonable request.

Author Contributions

Khushvir Singh: Conceptualization, Methodology, Software, Formal analysis, Investigation, Data curation, Writing – original draft, Writing – review & editing, Visualization.

References

[1] TMX Group. TSX market statistics and sector composition. Toronto: TMX Group Limited; 2024. Available from: https://www.tmx.com

[2] Fischer T, Krauss C. Deep learning with long short-term memory networks for financial market predictions. Eur J Oper Res. 2018;270(2):654-669. doi:10.1016/j.ejor.2017.11.054

[3] Hochreiter S, Schmidhuber J. Long short-term memory. Neural Comput. 1997;9(8):1735-1780. doi:10.1162/neco.1997.9.8.1735

[4] Nelson DMQ, Pereira ACM, de Oliveira RA. Stock market's price movement prediction with LSTM neural networks. In: Proceedings of the International Joint Conference on Neural Networks (IJCNN); 2017. p. 1419-1426. doi:10.1109/IJCNN.2017.7966019

[5] Selvin S, Vinayakumar R, Gopalakrishnan EA, Menon VK, Soman KP. Stock price prediction using LSTM, RNN and CNN-sliding window model. In: Proceedings of the International Conference on Advances in Computing, Communications and Informatics (ICACCI); 2017. p. 1643-1647. doi:10.1109/ICACCI.2017.8126078

[6] Qian B, Rasheed K. Stock market prediction with multiple classifiers. Appl Intell. 2007;26(1):25-33. doi:10.1007/s10489-006-0001-7

[7] Pan SJ, Yang Q. A survey on transfer learning. IEEE Trans Knowl Data Eng. 2010;22(10):1345-1359. doi:10.1109/TKDE.2009.191

[8] Li JL, Shi WK. Hybrid preprocessing for neural network-based stock price prediction. Heliyon. 2024;10:e40819. doi:10.1016/j.heliyon.2024.e40819

[9] Box GEP, Jenkins GM, Reinsel GC, Ljung GM. Time series analysis: forecasting and control. 5th ed. Hoboken: Wiley; 2015.

[10] Zhang GP. Time series forecasting using a hybrid ARIMA and neural network model. Neurocomputing. 2003;50:159-175. doi:10.1016/S0925-2312(01)00702-0

[11] Kim HY, Won CH. Forecasting the volatility of stock price index: a hybrid model integrating LSTM with multiple GARCH-type models. Expert Syst Appl. 2018;103:25-37. doi:10.1016/j.eswa.2018.03.002

[12] Sezer OB, Gudelek MU, Ozbayoglu AM. Financial time series forecasting with deep learning: a systematic literature review (2005-2019). Appl Soft Comput. 2020;90:106181. doi:10.1016/j.asoc.2020.106181

[13] Bao W, Yue J, Rao Y. A deep learning framework for financial time series using stacked autoencoders and long short-term memory. PLoS One. 2017;12(7):e0180944. doi:10.1371/journal.pone.0180944

[14] Moghar A, Hamiche M. Stock market prediction using LSTM recurrent neural network. Procedia Comput Sci. 2020;170:1168-1173. doi:10.1016/j.procs.2020.03.049

[15] Gao T, Chai Y. Improving stock closing price prediction using recurrent neural network and technical indicators. Neural Comput. 2018;30(10):2833-2854. doi:10.1162/neco_a_01124

[16] Ding X, Zhang Y, Liu T, Duan J. Deep learning for event-driven stock prediction. In: Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI); 2015. p. 2327-2333.

[17] Aroussi R. yfinance: download market data from Yahoo! Finance's API (Version 0.2) [Computer software]. Python Package Index; 2023. Available from: https://pypi.org/project/yfinance/

[18] Elton EJ, Gruber MJ, Brown SJ, Goetzmann WN. Modern portfolio theory and investment analysis. 9th ed. Hoboken: Wiley; 2014.

[19] Gers FA, Schmidhuber J, Cummins F. Learning to forget: continual prediction with LSTM. Neural Comput. 2000;12(10):2451-2471. doi:10.1162/089976600300015015

[20] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I. Attention is all you need. Adv Neural Inf Process Syst. 2017;30:5998-6008.