

# Hedonic Forecast: ML Housing Prediction in Japan

Team 90: Jianhuang Li, Shin Ying Chua, Wei Qi Thong, Ryan Kai Yan Seet

## 1 Introduction

Urban buyers, developers, and planners in Japan wait roughly three months for the Japan Residential Property Price Index (JRPPI), which also reports only broad regions [2, 3]. To close that gap, we build *Hedonic Forecast*, a reproducible pipeline that converts 485,093 Ministry of Land, Infrastructure, Transport and Tourism (MLIT) transactions [1] into ward- and 250 m mesh forecasts for Tokyo’s 23 wards and Sendai.

The pipeline pairs modelling with a Streamlit dashboard: choropleths, leaderboards, and SHapley Additive exPlanations (SHAP) give stakeholders both the numbers and the reasons behind them. In this paper the visuals are static, but every figure is generated from the same reproducible code. The system encompasses four main components: (1) quarterly mesh hedonic indices that keep rows with missing building years via indicators instead of dropping data, (2) hierarchical forecasting where mesh models ingest ward medians/lags to stabilise sparse geographies, (3) direct mapping from MLIT coordinates to JIS X 0410 meshes without external geocoding [11, 12], and (4) cold-start smoothing that follows state-space ETS (Error, Trend, Seasonality) initialization so early-quarter predictions do not spike [6]. These keep the experience interpretable while delivering ward MAE of 18,595 JPY/m<sup>2</sup>.

## 2 Problem Definition

We address three main tasks: (1) construct quarterly hedonic price indices at ward/mesh levels summarizing spatiotemporal dynamics from 2005–2025; (2) predict next-quarter median JPY/m<sup>2</sup> using only information available at prediction time; (3) provide interpretable visualiza-

tions of spatial patterns .

Inputs are MLIT real-estate transaction features (price, area, date, building year, location). Outputs include computed indices, one-step forecasts with uncertainty, and standard metrics such as mean absolute error (MAE), root mean squared error (RMSE), and the coefficient of determination ( $R^2$ ), also including SHAP explanations for each prediction [14, 18, 19].

Formally, letting  $p_{l,t}$  denote the median price-per-square-meter for location  $l$  at quarter  $t$  and  $x_{l,t}$  the engineered feature vector, we learn  $f_l$  such that  $\hat{p}_{l,t+1} = f_l(x_{l,t})$  by minimizing empirical mean squared error (MSE) subject to chronological splits (Train 2009–2019Q4, Val 2020Q1–2021Q4, Test 2022Q1–2025Q1). We additionally require  $f_l$  to supply additive feature attributions (SHAP) for downstream trust.

## 3 Literature Survey

Japanese real estate forecasting has multiple problems. Official JRPPI indices [2] lag 2–3 months, rendering them less useful for investors and stakeholders who need timely signals. Academic models are generally split between spatial and temporal factor models: spatial ML [4] ignores time, while temporal models such as LSTMs [5] ignore location. Transformers have the potential to handle both but have high computational costs and ultimately prove difficult to interpret [8].

Research also shows that nearby transactions influence local prices [9], but modeling full spatial covariance structures is computationally intensive. We use mesh aggregation [10]; a grid-based shortcut that captures neighborhood effects efficiently. For interpretability, we apply SHAP [14, 15], which quantifies the importance

of characteristics in models including LSTMs [16]. Some articles have also proven that multivariate LSTMs that incorporate multiple input features consistently outperform univariate approaches in financial and economic forecasting [17]. Operational research on Tokyo’s 23 wards also demonstrates practical MLIT data cleaning procedures and adaptive rolling windows [13] which we will utilize in our approach to work with the dataset. Peng and Inoue [20] apply eigenvector spatial filtering to show that Tokyo housing prices reflect both local factors, such as nearby schools or train stations, and regional effects across wards. This highlights the need for multi-scale spatial features in our forecasting models, though their study is mainly explanatory rather than predictive.

Overall, this review reveals multiple gaps in existing literature. Official pricing indices have delays, spatial-temporal effects are often separated, models lack transparency, and contemporary research focuses overwhelmingly on Tokyo. No work systematically compares multiple ML approaches with SHAP interpretability and interactive visualization while comparing Tokyo with regional cities.

## 4 Proposed Method

### 4.1 Data and Preprocessing

We cleaned a total of 485,093 MLIT transactions (2005-Q3 to 2025-Q1) across Tokyo’s 23 wards and Sendai’s main wards, covering ~3,000+ unique 250m meshes. Data cleaning addressed full-width numerals, Japanese era calendars, and missing building years (median-imputed by ward with `AgeUnknown` indicator). Spatial encoding derives *JIS X 0410* 250m mesh codes directly from MLIT transaction coordinates (lat/lon), then collapses to district-quarter medians. A ward-centroid fallback is then applied if coordinates are missing. This avoids separate geocoding and reduces processing overheads substantially [1, 11, 12]. Two panels are created as a result: ward×quarter (28 wards) and mesh×quarter, each with a set of median/dispersion price metrics, transaction

counts, building statistics, and temporal keys. The MLIT endpoint mixes structured and semi-structured values, so we normalize numerals, harmonize Japanese era dates to Gregorian quarters, and standardize areas to square meters. Transactions with blank coordinates trigger a ward-level centroid fallback so every row inherits a mesh code, while a secondary fallback copies the most recent valid mesh code when the MLIT dataset suppresses coordinates for privacy. These steps keep the spatial grid sufficiently dense even when lat/lon values are missing.

Table 1: Dataset summary and temporal splits

Metric	Value
Total transactions	485,093
Time horizon	2005-Q3 to 2025-Q1
Unique 250m meshes	~3,000+
Train/Val/Test (Ward)	1,507 / 336 / 252
Train/Val/Test (Mesh)	26,818 / 6,337 / 4,762

### 4.2 Intuition

Recent real-estate forecasting efforts in Japan tend to emphasize a single model family. Yoshida and Seya [4] benchmark linear regression against gradient boosted trees using macro/static predictors, while Otsuki [13] focuses on data-cleaning heuristics without combining interpretable and neural approaches. Existing LSTM work on housing [5] largely treats the task as univariate sequence prediction, and most studies lack feature-level explanations to justify deployment [14, 15].

Our approach attempts to blend complementary ideas so the user can trace why each choice improves stability or transparency:

- **Hierarchical hedonic context.** We rebuild ward and mesh hedonic indices every quarter and propagate ward-level scores down to meshes with explicit missingness flags. This mirrors the fixed-effects intuition of Otsuki [13] while keeping the higher spatial resolution that policy makers care about.

- **Model diversity.** Instead of betting on a single learner, we train linear, tree-based, and LSTM sequence models. This lets us quantify when simple momentum suffices (wards) versus when nonlinear interactions dominate (meshes), and gives stakeholders the option to trade accuracy for speed or transparency.
- **Built-in interpretability.** Every model, including the LSTMs, is paired with SHAP exports (CSV + PNG files) so analysts can see which features drive a forecast. This directly addresses the interpretability gap highlighted in SHAP literature [14, 15] and is largely absent from prior Japanese housing studies.
- **Software packaging.** Once datasets are scraped and processed from the MLIT API, All analyses run from the command line, which makes the project reproducible and attractive to other teams or agencies that wish to build on our work.

We keep the feature sets compact by design. Fifteen ward features capture price momentum (lags, moving averages), liquidity (transaction counts), and structural signals (age, area) without overfitting the  $\sim 1.7k$  ward samples, while the 13+ mesh features add hedonic indices and missingness flags to stabilise sparse grids. This balance is enough to outperform naive lag/MA baselines (evaluated in the workflow) while preserving interpretability.

### 4.3 Reproducibility and Workflow

All figures and tables come from the CLI pipeline in `CODE/`: `python -m CODE.run_workflow` rebuilds panels, trains models with the stated splits, exports SHAP, and writes outputs to `CODE/outputs/`. Efficiency tables use `python -m CODE.experiments --levels Ward Mesh --fractions 0.25 0.5 0.75 1.0`.

### 4.4 Detailed Description

**Hedonic price index.** We estimate municipality-level indices using two-way fixed

effects (PanelOLS):

$$\ln(P_{it}) = \beta_0 + \beta_1 \ln(\text{Area}_{it}) + \beta_2 \text{Age}_{it} + \beta_3 \text{AgeUnknown}_{it} + \alpha_i + \gamma_t + \varepsilon_{it} \quad (1)$$

where  $\alpha_i$  (mesh fixed effects) and  $\gamma_t$  (quarter fixed effects) capture spatial and temporal heterogeneity. The `AgeUnknown` dummy handles missing building year values without dropping observations. Quarterly granularity exceeds typical annual approaches. Fitted log prices are averaged by mesh-quarter, exponentiated, and normalized based on existing formulas [7] to construct 2005–2025 indices. We merge these tables into the ward/mesh panels, forward/back-fill missing meshes with ward-level scores, and record binary missing indicators so validation/test rows remain populated.

**Forecasting models.** We train separate ward and mesh models with strict chronological splits (Train  $\leq$  2019-Q4, Val 2020-Q1–2021-Q4, Test  $\geq$  2022-Q1). Feature sets mirror the production code:

Table 2: Ward feature set (15 columns).

Category	Features
Momentum	<code>MedianPriceSqM_lag1</code> , <code>MedianPriceSqM_lag4</code> , growth, <code>MedianPriceSqM_ma4q</code> , <code>MedianPriceSqM_std4q</code>
Liquidity	<code>TransactionCount</code> , <code>ActiveMeshes</code>
Structure	<code>AvgBuildingAge</code> , <code>AvgArea</code>
Time/ID	<code>TimeTrend</code> , <code>Ward_encoded</code> , <code>Q_2</code> , <code>Q_3</code> , <code>Q_4</code>

Table 3: Mesh feature set (13+ columns).

Category	Features
Momentum	<code>mesh_median_ppsqm_lag1</code> , <code>mesh_median_ppsqm_lag4</code> , growth, <code>mesh_median_ppsqm_ma4q</code> , <code>mesh_median_ppsqm_std4q</code>
Liquidity	<code>mesh_transaction_count</code> , <code>mesh_avg_age</code> , <code>mesh_avg_area</code>
Context	<code>TimeTrend</code> , <code>Ward_encoded</code> , <code>WardHedonicIndex</code> , <code>MeshHedonicIndex</code>
Missingness	<code>WardHedonicIndex_missing</code> , <code>MeshHedonicIndex_missing</code>

LSTM models use reduced feature sets focused on core momentum and liquidity signals:

Table 4: Ward LSTM feature set (7 columns).

Category	Features
Core momentum	MedianPriceSqM, MedianPriceSqM.lag1, MedianPriceSqM.ma4q, QoQ growth
Liquidity Structure	TransactionCount, ActiveMeshes AvgBuildingAge

Table 5: Mesh LSTM feature set (7 columns).

Category	Features
Core momentum	mesh_median_ppsqm, mesh_median_ppsqm.lag1, mesh_median_ppsqm.ma4q, QoQ growth
Liquidity Structure	mesh_transaction_count mesh_avg_age, mesh_avg_area

Models use a set of arbitrary SHAP caps in order to avoid long running computations: RF (300 trees), LightGBM (600 trees), LSTM window 8 quarters with hidden 96, dropout 0.3, and early stopping patience 8. SHAP uses the full validation set for metric computation and caps only the explainer sampling for speed (trees 500 rows, linear 200 rows, LSTM 120 sequences, plots 1,000 rows) so Sendai’s small-sample tails stay visible while keeping runtimes manageable. Models themselves train on all available data.

## 5 Evaluation

### Testbed Description

Ward and mesh panels cover Tokyo’s 23 wards and Sendai (2009–2025). Ward models use 15 momentum/liquidity/structure features; meshes add hedonic indices and missingness flags. Chronological splits are fixed (Train 2009–2019Q4, Val 2020Q1–2021Q4, Test 2022Q1–2025Q1) to prevent leakage. All models predict one quarter ahead; early lags are initialized with short moving averages (state-space ETS) so forecasts start in the first quarter [6]. SHAP is computed on validation slices and surfaced alongside metrics; LSTM timing is excluded from efficiency tables because sequence training and SHAP export are orders slower and hardware-dependent.

### Key Findings

1. **Feature importance.** Momentum dominates (Fig. 8). In both Tokyo and Sendai, four-quarter moving averages and one-quarter lags carry most SHAP mass; QoQ/YoY growth follows. Structural variables sit in the long tail. Mesh Random Forest plots echo this pattern, with hedonic indices and missingness flags mattering only when mesh history is thin. LSTM SHAP mirrors the same cues, which explains why it adds little lift on sparse meshes. The mild negative skew in Sendai (mesh RF, ward LightGBM/LSTM) reflects the designed fallbacks: forecasts damp when local momentum is weak, leaning on ward priors instead of overreacting.
2. **Accuracy across scales.** Tested on 2022–2025, ward Linear Regression achieves the lowest MAE (18.6k JPY/m<sup>2</sup>) with  $R^2 = 0.995$  (Table 6). Nonlinear models offer marginal gains at the ward scale but matter at the mesh scale: Random Forest slashes mesh MAE from 83.9k (linear) to 22.6k JPY/m<sup>2</sup> with  $R^2 = 0.95$  (Table 7).

3. **Granularity trade-offs.** Comparing ward and mesh metrics shows that higher resolution introduces heavier error tails but also reveals neighbourhood hotspots that ward medians smooth away. The dashboard exposes both scales simultaneously so analysts can choose the fidelity that matches their decision.

Table 6: Ward-level test accuracy (JPY/m<sup>2</sup>).

Model	Test MAE	Test RMSE	$R^2$
Linear Regression	18,595	29,651	0.995
Random Forest	44,302	112,689	0.933
LightGBM	48,039	116,653	0.928
Torch LSTM	41,101	61,214	0.980

Table 7: Mesh-level test accuracy (JPY/m<sup>2</sup>).

Model	Test MAE	Test RMSE	$R^2$
Linear Regression	83,896	197,137	0.876
Random Forest	22,620	123,618	0.951
LightGBM	27,702	126,969	0.948
Torch LSTM	103,143	324,578	0.664

Linear models remain strongest for wards because momentum alone explains nearly all the variance. Meshes benefit from nonlinear splits that capture interactions between hedonic indices, lags, and missingness flags; hence Random Forest leads and LightGBM follows closely. The LSTM struggles on meshes because short sequences and limited spatial context make it hard to generalise beyond momentum, and Sendai’s small sample triggers LightGBM split warnings, so we prefer Random Forest for Sendai deployment. Ward forecasts can lean on Linear Regression for sub-millisecond updates when latency matters, while meshes warrant Random Forest because its accuracy gains outweigh a few extra milliseconds once training coverage exceeds 50%. Figure 8 shows city-split SHAP beeswarms for the tree models (ward LightGBM, mesh Random Forest) plus the available LSTM variants. Mesh LSTM SHAP is provided for Tokyo (validation) and the full test set; a Sendai-specific mesh LSTM plot was not generated.

Across model families (Linear Regression, Random Forest, LightGBM, and LSTM) the SHAP beeswarms in Fig. 8 give a consistent feature hierarchy. The four-quarter moving average of prices dominates with the widest SHAP spread, followed by one- and four-quarter lags; QoQ/YoY growth ranks next, especially on meshes where local series are noisy. Structural signals (age, size, transaction counts), spatial identifiers, and hedonic indices cluster around zero, implying limited marginal impact once momentum is present. This pattern shows the models behaving like compact autoregressive forecasters whose predictive power comes primarily from recent price trajectories, with smooth trends and short-run momentum carrying far more weight than static structure or location labels.

Drilling deeper, ward LightGBM shows mild

negative SHAP mass for average age and near-zero impact for counts, confirming structural nuance adds little once momentum is known. Mesh Random Forest has a small bump for `WardHedonicIndex` where mesh lags are sparse; its missingness flag adds a bounded positive push, signalling deference to ward context during cold starts. Tokyo meshes exhibit tighter SHAP clouds than Sendai; Sendai shows wider QoQ spread and mild negative skew in mesh RF and ward LightGBM/LSTM, consistent with thin-sample regularisation when local momentum is weak. LSTM SHAP mirrors the tree rankings, largely relearning momentum.

## 6 Conclusions and Discussion

Hedonic Forecast demonstrates that a data-clean hedonic pipeline plus lightweight modelling can close the three-month information gap that plagues Japan’s official indices. By combining ward baselines, mesh ensembles, SHAP narratives, and an interactive dashboard, we deliver forecasts that are both accurate (ward MAE 18.6k JPY/m<sup>2</sup>, mesh MAE 22.6k JPY/m<sup>2</sup>) and explainable to practitioners who are new to machine learning. The visual layer keeps non-technical stakeholders in the loop by highlighting which features move prices in each geography, and the code path from raw MLIT inputs to plots is fully scripted for reproducibility.

Limitations remain. MLIT releases omit micro amenities, so nonlinear models still default to lag structure; we could buy REINS listings, prototype SUUMO scraping with ToS care, and merge OpenStreetMap POIs. LightGBM was unstable on Sendai’s small sample, so default to Random Forest for Sendai meshes/wards. Automating quarterly reruns, adding conformal prediction intervals, and ablating hedonic fallbacks or residual spatial autocorrelation (e.g., Moran’s I) would tighten robustness in future work.

All team members contributed equally to all aspects of the project and report writing.

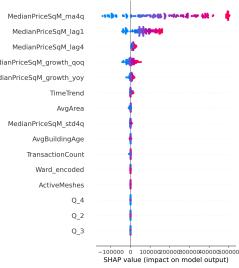


Figure 1: \*  
Tokyo Ward  
LightGBM

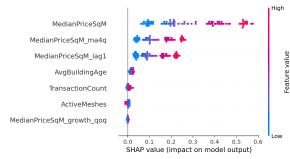


Figure 5: \*  
Tokyo Ward LSTM



Figure 2: \*  
Sendai Ward  
LightGBM

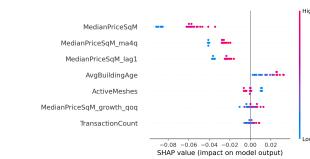


Figure 3: \*  
Tokyo Mesh RF

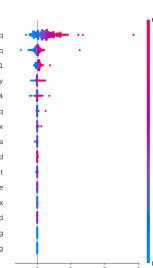


Figure 4: \*  
Sendai Mesh RF

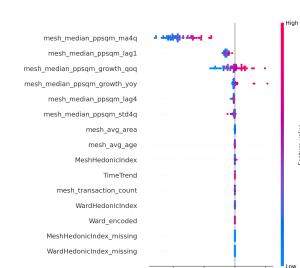


Figure 8: City-split SHAP beeswarms for ward LightGBM, mesh Random Forest, and LSTM variants. Momentum dominates; Sendai shows wider spreads and mild negative skew where data are sparse. Mesh LSTM SHAP uses Tokyo validation and overall test (no Sendai-specific mesh LSTM SHAP available).

## Appendix

## Efficiency Tables

Tables 8 and 9 report inference-time scaling for the classical models (Linear Regression, Random Forest, LightGBM) at 25%, 50%, 75%, and 100% of training data. LSTM timing is excluded because sequence training and SHAP export are orders slower and hardware-dependent.

## References

- [1] Ministry of Land, Infrastructure, Transport and Tourism (MLIT). (2005–present). *Real Estate Information Library*. Retrieved from <https://www.reinfolib.mlit.go.jp/>
  - [2] Ministry of Land, Infrastructure, Transport and Tourism, Real Estate and Construction Economy Bureau. (2020). *Methodology*

of JRPI: Japan Residential Property Price Index. Retrieved from <https://www.mlit.go.jp/common/001360414.pdf>

- [3] Ministry of Land, Infrastructure, Transport and Tourism. (2015). *Japan Residential Property Price Index and Residential Transaction Volume (August 2015)*. See p. 6: “Time lag: About 3 months” and coverage by geography. Retrieved from <https://www.mlit.go.jp/common/001110934.pdf>
  - [4] Yoshida, T., & Seya, H. (2024). Spatial prediction of apartment rent using regression-based and machine learning-based approaches with a large dataset. *The Journal of Real Estate Finance and Economics*, 69(1), 1–28. <https://doi.org/10.1007/s11146-022-09929-6>
  - [5] Chen, X., Wei, L., & Xu, J. (2017). House price prediction using LSTM.

Table 8: Inference-time efficiency scaling for ward-level models across training fractions.

Model	Fraction	Train Rows	Val Rows	Test Rows	Infer Time (s)	Test MAE	Test RMSE	$R^2$
LinearRegression	0.25	377	224	364	0.0011	104,324	109,369	0.937
RandomForest	0.25	377	224	364	0.0450	88,365	170,694	0.846
LightGBM	0.25	377	224	364	0.0045	106,527	192,861	0.804
LinearRegression	0.50	754	224	364	0.0015	24,371	36,728	0.993
RandomForest	0.50	754	224	364	0.0436	87,944	171,270	0.845
LightGBM	0.50	754	224	364	0.0049	111,882	206,757	0.774
LinearRegression	0.75	1,131	224	364	0.0016	19,298	30,074	0.995
RandomForest	0.75	1,131	224	364	0.0490	77,972	167,894	0.851
LightGBM	0.75	1,131	224	364	0.0047	76,600	169,087	0.849
LinearRegression	1.00	1,507	224	364	0.0015	18,595	29,651	0.995
RandomForest	1.00	1,507	224	364	0.0808	44,763	113,298	0.932
LightGBM	1.00	1,507	224	364	0.0045	48,039	116,653	0.928

Table 9: Inference-time efficiency scaling for mesh-level models across training fractions.

Model	Fraction	Train Rows	Val Rows	Test Rows	Infer Time (s)	Test MAE	Test RMSE	$R^2$
LinearRegression	0.25	6,705	4,188	6,770	0.0020	97,903	210,560	0.858
RandomForest	0.25	6,705	4,188	6,770	0.099	45,518	168,424	0.909
LightGBM	0.25	6,705	4,188	6,770	0.058	45,942	167,110	0.911
LinearRegression	0.50	13,409	4,188	6,770	0.0022	85,203	203,366	0.868
RandomForest	0.50	13,409	4,188	6,770	0.082	33,971	152,441	0.926
LightGBM	0.50	13,409	4,188	6,770	0.049	36,607	134,185	0.942
LinearRegression	0.75	20,114	4,188	6,770	0.0017	87,921	199,769	0.872
RandomForest	0.75	20,114	4,188	6,770	0.176	25,335	102,214	0.967
LightGBM	0.75	20,114	4,188	6,770	0.049	27,902	124,372	0.951
LinearRegression	1.00	26,818	4,188	6,770	0.0028	83,896	197,137	0.876
RandomForest	1.00	26,818	4,188	6,770	0.142	24,303	126,879	0.949
LightGBM	1.00	26,818	4,188	6,770	0.047	27,702	126,969	0.948

- arXiv preprint arXiv:1709.08432.  
<https://arxiv.org/abs/1709.08432>
- [6] Hyndman, R. J., Koehler, A. B., Snyder, R. D., & Grose, S. (2002). A state space framework for automatic forecasting using exponential smoothing methods. *International Journal of Forecasting*, 18(3), 439–454. <https://www.sciencedirect.com/science/article/pii/S0169207001001108>
- [7] Haque, D. (2024). Transforming Japan real estate. arXiv preprint arXiv:2405.20715. <https://arxiv.org/abs/2405.20715>
- [8] Wen, Q., Zhou, T., Zhang, C., Chen, W., Ma, Z., Yan, J., & Sun, L. (2023). Transformers in time series: A survey. In *Proceedings of IJCAI 2023 (Survey Track)*. <https://www.ijcai.org/proceedings/2023/0759.pdf>
- [9] Muto, S., Sugasawa, S., & Suzuki, M. (2023). Hedonic real estate price estimation with the spatiotemporal geostatistical model. *Journal of Spatial Econometrics*. <https://link.springer.com/article/10.1007/s43071-023-00039-w>
- [10] Mizuho Research & Technologies, Ltd. (2022). *Prediction of land prices using machine learning with mesh-based neighbor features*. JSAI Special Interest Group on Financial Informatics (SIG-FIN-029-61). Retrieved from [https://www.jstage.jst.go.jp/article/jaisigtwo/2022/FIN-029/2022\\_61/\\_pdf](https://www.jstage.jst.go.jp/article/jaisigtwo/2022/FIN-029/2022_61/_pdf)
- [11] Statistics Bureau of Japan. (n.d.). *The Standard Grid Square and the Grid Square Code used for the Statistics*. Retrieved from <https://www.stat.go.jp/english/data/mesh/02.html>

- [12] National Research Institute for Earth Science and Disaster Resilience (NIED). (n.d.). *What is the 250m-mesh code?* J-SHIS FAQ. Retrieved from <https://www.j-shis.bosai.go.jp/en/faq-250mmesh>
- [13] Otsuki, K. (2023). *A Study on Data-Cleansing Methods and Model-Update Algorithms for Real Estate Price Forecasting Models* [Doctoral dissertation, Hitotsubashi University]. Hitotsubashi University Repository. <https://hit-u.repo.nii.ac.jp/records/2048234>
- [14] Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems 30* (pp. 4765–4774). <https://arxiv.org/abs/1705.07874>
- [15] Lundberg, S. M., Erion, G., Chen, H., DeGrave, A., Prutkin, J. M., Nair, B., Katz, R., Himmelfarb, J., Bansal, N., & Lee, S.-I. (2020). From local explanations to global understanding with explainable AI for trees. *Nature Machine Intelligence*, 2(1), 56–67. <https://doi.org/10.1038/s42256-019-0138-9>
- [16] Sen, D., Deora, B. S., & Vaishnav, A. (2025). Explainable deep learning for time series analysis: Integrating SHAP and LIME in LSTM-based models. *Journal of Information Systems Engineering and Management*, 10(16s). <https://jisem-journal.com/index.php/journal/article/view/2627>
- [17] Kuber, V., Yadav, D., & Yadav, A. K. (2022). Univariate and multivariate LSTM model for short-term stock market prediction. *arXiv preprint arXiv:2205.06673*. <https://arxiv.org/abs/2205.06673>
- [18] Hyndman, R. J., & Koehler, A. B. (2006). Another look at measures of forecast accuracy. *International Journal of Forecasting*, 22(4), 679–688. <https://robjhyndman.com/papers/mase.pdf>
- [19] Tashman, L. J. (2000). Out-of-sample tests of forecasting accuracy: An analysis and review. *International Journal of Forecasting*, 16(4), 437–450. <https://www.sciencedirect.com/science/article/pii/S0169207000000650>
- [20] Peng, Z., & Inoue, R. (2022). Identifying multiple scales of spatial heterogeneity in housing prices based on eigenvector spatial filtering approaches. *ISPRS International Journal of Geo-Information*, 11(5), 283. <https://doi.org/10.3390/ijgi11050283>