Electronics and Information Technologies, 2025, 30, 99–112
http://publications.lnu.edu.ua/collections/index.php/electronics/index

UDC: 004.9

MODELING RETAIL SALES USING AUTOREGRESSIVE INTEGRATED MOVING AVERAGE AND LONG SHORT-TERM MEMORY FORECASTING METHODS

Oleksii Kachmar *, ***Roman Shuvar***, ***Igor Kolych***
Department of System Design,
Ivan Franko National University of Lviv
50 Drahomanova Str.,79005 Lviv, Ukraine

Kachmar, O., Shuvar, R., & Kolych, I. (2025). Modeling Retail Sales Using Autoregressive Integrated Moving Average and Long Short-Term Memory Forecasting Methods. *Electronics and Information Technologies*, 30, 99–112. https://doi.org/10.30970/eli.30.8

ABSTRACT

Background. Forecasting retail sales is crucial for modern supply chain and inventory management. Traditional statistical models alone can be insufficient due to the large amounts of data generated by extensive retail chains. Combining time series analysis with machine learning can improve forecast accuracy.

Materials and Methods. This research used the M5-forecasting accuracy dataset, containing over 30,000 time series of store-item daily sales. The study involved data preprocessing to handle any missing values and splitting the series into training and hold-out test sets. Three forecasting methods were applied. The first method accounted for autoregressive and moving average components. The second approach explicitly included trend and seasonality by decomposing the series into those components, fitting a model to the trend-adjusted series, and then reintroducing the seasonal part. Third, a long short-term memory deep learning regressor was trained to capture longer-range dependencies. The evaluation on the test set was performed using the Mean Absolute Error (MAE). Residual analysis examined autocorrelation and the distribution of errors.

Results and Discussion. A focus on one item showed a strong weekly cycle. The first autoregressive approach without explicit seasonality partially captured the data but left some significant autocorrelation in the residuals. The second autoregressive variant that considered trend and weekly seasonal decomposition achieved the best short-term predictive accuracy, reflected by lower MAE. The deep learning regressor, implemented in a recursive multi-step setup, did not outperform the autoregressive one, partly due to error accumulation and possibly incorrect choice of its architecture.

Conclusion. The study indicates that for retail data with clear weekly fluctuations, auto-regressive moving-average models enhanced by trend and seasonal decomposition can provide robust forecasts. Neural network methods can model non-linearities but require more specialized sequence-to-sequence configurations to avoid cumulative forecast errors. Future work can involve combining methods for multi-horizon and hierarchical retail time series.

Keywords: Time series analysis, machine learning, retail forecasting, ARIMA, LSTM, seasonality

INTRODUCTION

Retail sales forecasting is pivotal for demand-driven supply chain management, strategic pricing, and minimizing waste from unsold inventory. Historically, simple models such as Moving Averages or Exponential Smoothing were used to predict consumer demand. With the exponential growth of data volumes, modern forecasting efforts leverage more sophisticated approaches, including autoregressive integrated moving average (ARIMA) models and neural networks.

However, successful forecasting in the retail sector encounters several challenges. Seasonality with weekly, monthly, or yearly patterns. Trend components reflect broad, gradual changes in consumer behavior. Promotional spikes and special events.

As shown in [1], machine learning models that leverage generalization across products and stores can improve forecasting accuracy even with limited historical data. While our study evaluates ARIMA and long short-term memory (LSTM) separately, the findings in [1] support the idea that different modeling approaches capture distinct sales patterns. This reinforces the value of comparing classical methods like ARIMA with deep learning models such as LSTM to assess their strengths across scenarios.

Accurate retail demand forecasting is essential for optimizing inventories, streamlining operations, and enhancing customer satisfaction. Recent studies demonstrate the potential of machine learning (ML) and deep learning (DL) methods to handle various complexities in sales data, such as seasonality, special events, and regional differences [3]. The M5 Competition [2] further highlights the importance of hierarchical retail datasets, driving advancements in both established and novel forecasting algorithms.

Several publications show the advantages of ML-based approaches. For example, [6] treats sales forecasting largely as a regression problem, citing improvements over time-series models like ARIMA. In [6], using additional (exogenous) data enhances XGBoost performance, while [4] and [5] underscore the effectiveness of LSTM and convolutional neural networks (CNN), as well as hybrid models (XGBoost-LSTM). Time Series forecasting with LSTM is broadly described in the article [4]. The author describes so-called “Multistep” forecasting or “Walk-Forward” forecasting that will be leveraged in this study.

Training of deep neural networks may be expensive in terms of time and resources. Alternatively, to training or model fine-tuning, in-context learning was introduced in large language models. The idea is that giving examples to model works similarly to fine-tuning it for foundational models. Authors of Lag-Llama, the open-source base model for univariate probabilistic forecasting [8], decided to replicate this idea for time-series forecasting to make the model adjust to specific data more seamlessly. The novelty of this study lies partially in the way models produce forecasts. As the output model predicts mean, standard deviation, and degrees of freedom estimates of the predicted distribution, from which model forecasts are then sampled.

Despite these strides, challenges remain in forecasting the complex seasonality of retail sales. Therefore, the purpose of this study is to develop and experimentally validate a demand forecasting approach that leverages ML and DL techniques under real-world retail constraints. By applying these approaches to the M5-forecasting accuracy dataset, we aim to examine how trend, seasonality, and advanced recurrent neural network architectures influence forecast performance.

MATERIALS AND METHODS

Data Description

This study used the M5-forecasting accuracy dataset [7], comprising over 30,000 unique time series from Walmart stores across different US states. Each product-store combination details daily unit sales from 2011 to 2016. It includes sales data for specific products at individual store locations, incorporating contextual factors such as pricing, promotions, and product/store characteristics. The dataset is substantial, spanning several dozen gigabytes, making it highly valuable for time series research. The dataset exhibits variability across different time periods, including daily, weekly, and monthly fluctuations. Due to the dynamic nature of sales and the numerous influencing factors, developing accurate forecasting models is challenging. Forecasting requires consideration of future time horizons, in addition to historical sales data. Robust evaluation metrics are necessary to assess the accuracy of predictions. This dataset is widely used for testing and benchmarking forecasting algorithms.

Forecasting Approaches

Three main forecasting approaches were explored.

First, we used ARIMA models. It incorporates autoregressive (AR) terms. This element of the model indicates that the current value of the series can be represented as a linear combination of the previous values and model parameters. [2]

$$Y_t c + \phi_1 * Y_{t-1} + \ldots + \phi_p Y_{t-p} + \varepsilon_t \tag{1}$$

where ϕ is slope model parameter,
c – intercept model parameter,
and p – a hyperparameter of the model.

In order to define hyperparameter *p* in AR we shall use the autocorrelation function.

$$r_k = corr(y_t, y_k) = \frac{cov(y_t, y_{t-k})}{\sigma_t \sigma_{t-k}} \tag{2}$$

Differentiating for stationarity (I). Includes differentiation of the series to achieve stationarity. For example:

$$Y_t{}' = Y_t - Y_{t-1} \tag{3}$$

Additionally, moving average (MA) terms. It represents a series as a weighted combination of error values. The MA process must be stationary [2]

$$Y_t = \mu + \varepsilon_t + \theta_1 * \varepsilon_{t-1} + \ldots + \theta_q \varepsilon_{t-q} \tag{4}$$

where θ_n – slope model parameter,
μ – average of the series, also can be an intercept,
and ε_n – past errors obtained from AR.

To define hyperparameter *q* in MA we shall use the partial autocorrelation function.

$$r_k = corr(y_t, y_k) = \frac{cov(y_t, y_{t-k})}{\sigma_t \sigma_{t-k}} \; and \; \forall m \; (0 < m < k) \colon y_{t-m} = const \tag{5}$$

Note: partial correlation function on the contrary to correlation function helps to determine direct effect of y_{t-k} on y_t.

Second, an ARIMA variant explicitly accounted for trend and seasonal effects by decomposing the time series (Fig. 1) into trend, seasonal, and residual components, then applying ARIMA on the deseasonalized series and re-adding the seasonal parts [2].

Why is time series decomposition needed?

- Modeling: to predict the trend and add a seasonal component.
- Using the seasonal component as a feature for the model.
- Using knowledge of the trend and seasonality for more useful actions. For example, if we know some periodicity, we can decompose it into a Fourier series, generating the corresponding periodicity $\sin(kx)$ and $\cos(kx)$ and feed these functions to the input of a linear model, thus we can get a good basic solution.

Third, we implemented a Long Short-Term Memory (LSTM) regressor from the class of recurrent neural networks (Fig. 2), training it to capture longer-range patterns [3].

For LSTM regression in one step h_t is projected onto the linear layer of the network.

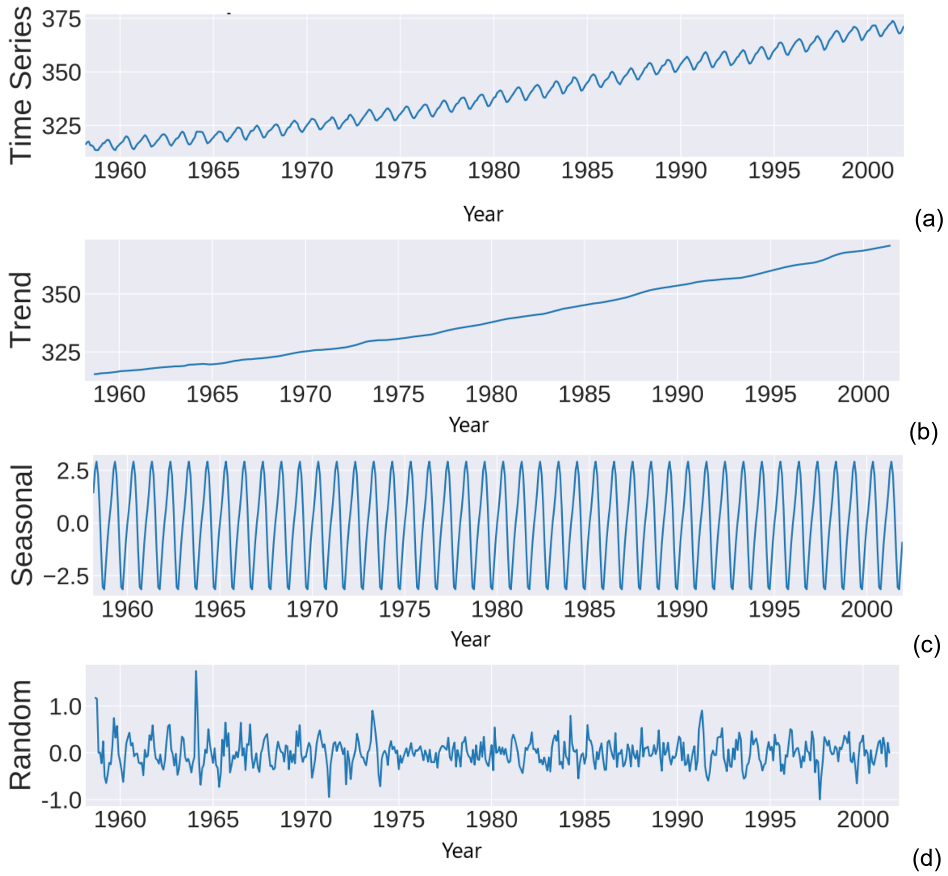

Fig. 1. Additive Decomposition of Time Series: Original (a), Trend (b), Seasonality (c), and Residuals (d) [10].

In a chosen **recursive** multi-step forecasting strategy, the model's previous predictions serve as inputs to forecast further ahead. The idea of this approach can be analytically described as follows [3]:

$$prediction(t+1) = model\big(observ(t), observ(t-1), \dots, obs(t-n)\big), \tag{6}$$

$$prediction(t+2) = model\big(prediction(t+1), obs(t), \dots, obs(t-n)\big). \tag{7}$$

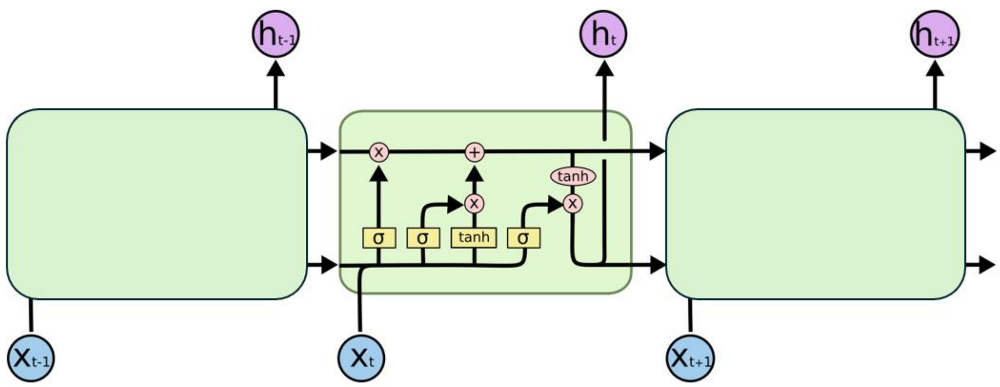

Fig. 2. Schematic representation of LSTM cells [11]: $\mathrm{h_t}$ – hidden state output; x_t – time series input value.

Evaluation

We evaluated performance using MAE, SMAPE, MASE to quantify forecast accuracy, while the analysis of residuals and autocorrelation confirmed how well each model explained the underlying patterns.

Used Software

The experiments in this study were conducted using Python 3.11. For time series modeling, the statsmodels library was employed to implement ARIMA models and perform STL decomposition. LSTM networks for deep learning-based forecasting were developed using PyTorch.

RESULTS AND DISCUSSION

Case Study on a Single Product with ARIMA

For our experiment we picked an example of an M5 dataset. It is the daily sales of a product item called "FOODS_3_090_CA_3," where "FOODS_3" indicates a department-subcategory (food items), "090" identifies the product, and "CA_3" denotes a particular store in California. Initial exploration revealed a clear weekly seasonal pattern and some irregular peaks likely driven by promotions.

The experiment aims to evaluate the forecasting performance of the ARIMA (7,0,7) model on a given time series dataset, assessing its ability to capture trends and fluctuations in both training and test sets.

The ARIMA (7,0,7) model was trained on historical data, with actual values represented in blue for the training set and red for the test set (Fig. 3 and Fig. 4). Model predictions were compared against actual values, with forecasted values shown in purple for the training set and orange for the test set.

The results indicate that the model successfully captures the general trends and periodic fluctuations in the training set. However, discrepancies were observed, particularly in areas with sharp peaks and drops, suggesting potential limitations of the approach.

The next experiment investigates an alternative forecasting method that separately models trend and seasonality components using an additive decomposition approach. The goal is to improve forecasting accuracy by isolating these components before making predictions.

The time series was decomposed into trend, seasonality, and residuals. The trend component was forecasted using ARIMA (7,0,7), removing seasonal fluctuations for a clearer long-term prediction. After predicting the trend, the seasonal pattern was

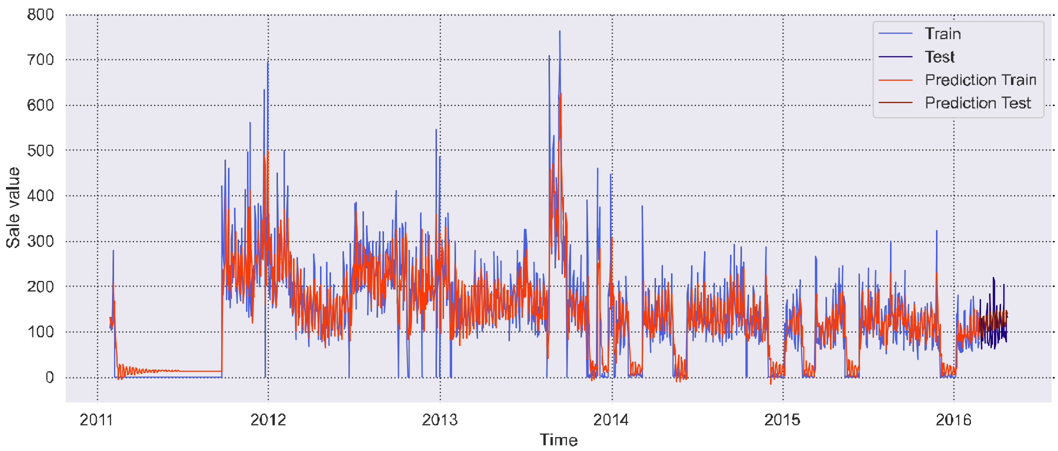

Fig. 3. Sales forecasting with the ARIMA (7, 0, 7) on train and test sets.

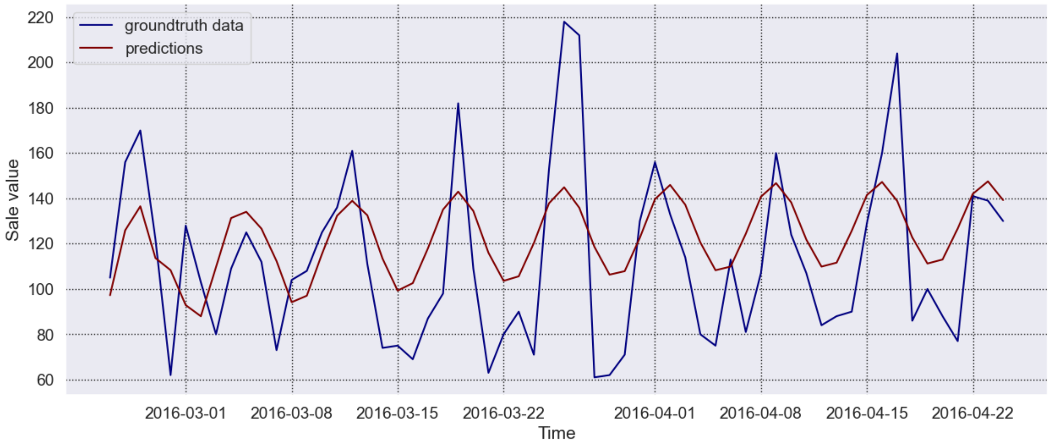

Fig. 4. Sales forecasting with the ARIMA (7, 0, 7) model, test set of sales for the last month.

reintroduced to reconstruct the complete forecast (Fig. 5 and Fig. 6). The approach provides improved metrics compared to a standard ARIMA model, though deviations still exist at extreme peaks and dips.

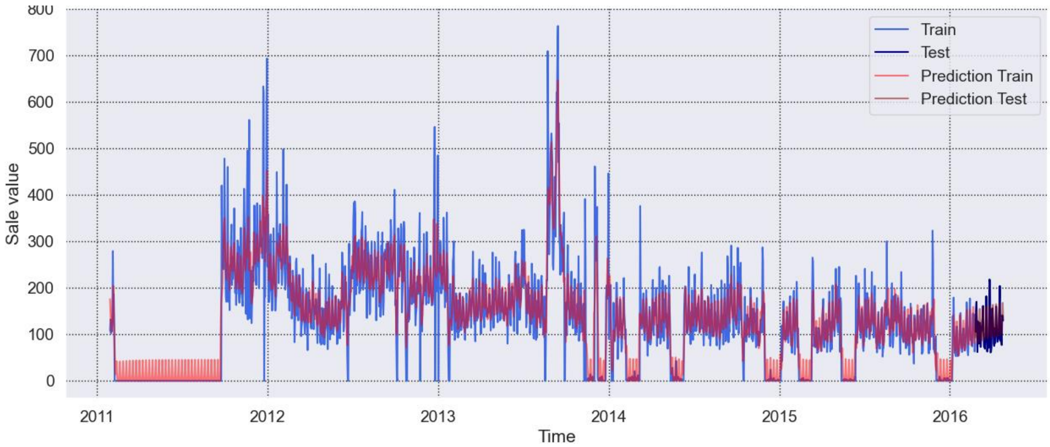

Fig. 5. Forecasting a time series using ARIMA(Trend) + Seasonality.

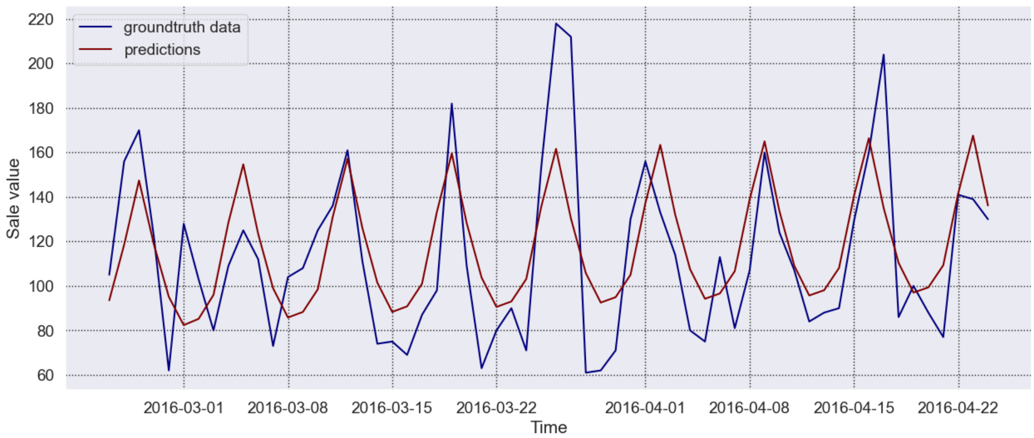

Fig. 6. Forecasting a test set of time series using ARIMA(Trend) + Seasonality.

The time series was decomposed into trend, seasonality, and residuals. The trend component was forecasted using ARIMA (7,0,7), removing seasonal fluctuations for a clearer long-term prediction. After predicting the trend, the seasonal pattern was reintroduced to reconstruct the complete forecast (Fig. 5 and Fig. 6). The approach provides improved metrics compared to a standard ARIMA model, though deviations still exist at extreme peaks and dips.

LSTM Performance

LSTM regressor was trained to predict the next-day sale.

LSTM Configuration.

The network consists of two stacked LSTM layers, each with a hidden state size of 256 units. The input sequence length (window size) is set to 30-time steps. No dropout regularization is applied between layers. The model is trained for a maximum of 269 epochs using the Adam optimizer with a learning rate of 1×10^{-4}.

Early Stopping.

To prevent overfitting and reduce training time, we apply early stopping based on validation loss. The model is saved whenever the validation loss improves beyond a set threshold, and training halts if no improvement occurs for 100 consecutive epochs. The best model (from epoch 169) is retained for evaluation. Learning curves are shown in Fig. 7 and Fig. 8.

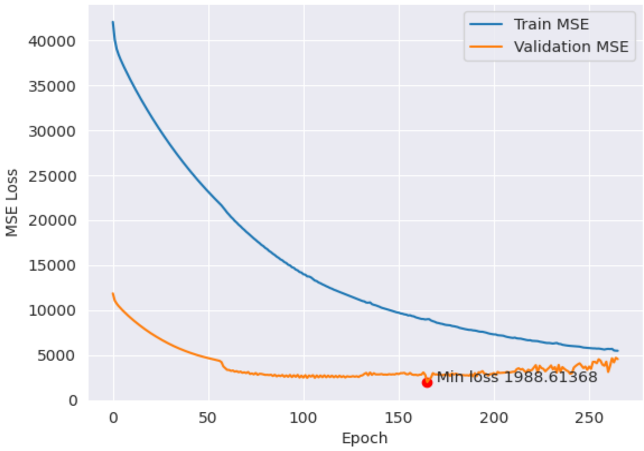

Fig.7. LSTM Training and validation mean squared error (MSE) loss curves for next-day sales prediction.

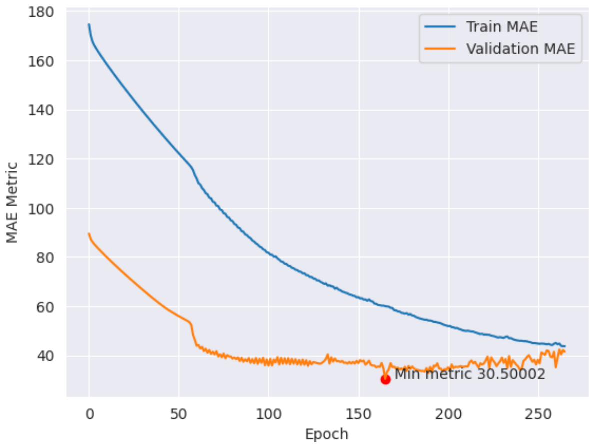

Fig. 8. LSTM training and validation mean absolute error MAE metric curves for next-day sales prediction.

LSTM was trained to perform next-step prediction on historical sales data and employed recursive inference to produce multi-step forecasts. Evaluation shows (Fig. 9 and Fig. 10) that while the model achieves good performance on the test set (MAE = 36.0, symmetric mean absolute percentage error (SMAPE) = 30.4), it performs worse on the training set (MAE = 77.0, SMAPE = 61.4) due to recursive error accumulation and the presence of higher variance in earlier historical data. This discrepancy highlights the limitations of recursive forecasting on long historical sequences.

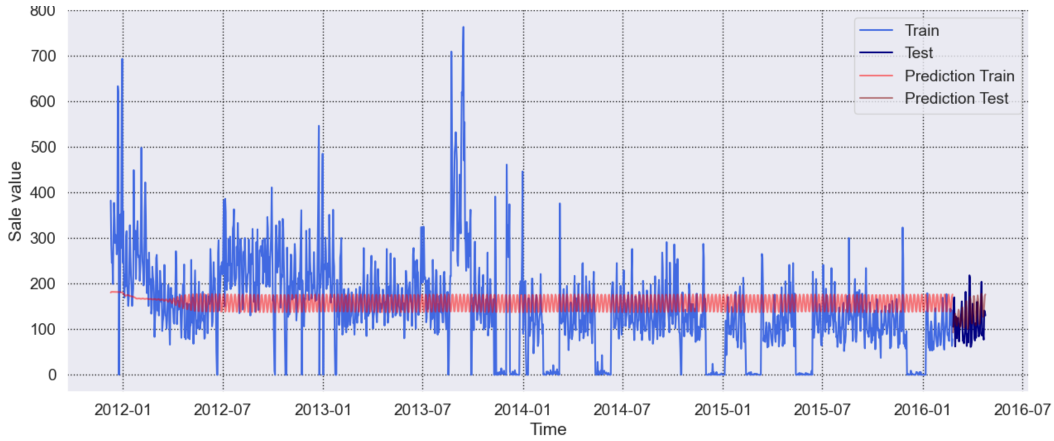

Fig. 9. Forecasting a time series using pretrained LSTM Regressor with recursive multistep approach.

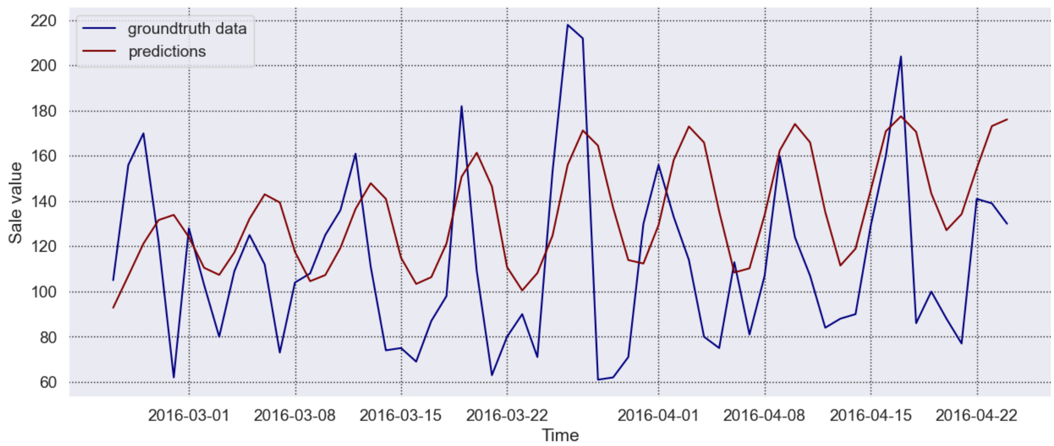

Fig. 10. Forecasting a test set of time series using pretrained LSTM Regressor with recursive multistep approach.

This approach seems to be working satisfactorily for shorter test sequences but does not work efficiently for longer-term train series forecasting. Its quality also seems to be dependent on the initialized sequence to regress from. Thus, we think that adopting a more specialized sequence-to-sequence design could improve results. Sequence-to-sequence approach for long-term time series forecasting is nicely described in the work [9].

Evaluation metrics

Table 1 presents train and test MAE metrics for LSTM regressor, Prophet, ARIMA (7,0,7) and for ARIMA trend forecasting with added seasonality.

Table 1. **Evaluation metric of models on train and test sets**

Models	Evaluation metrics					
	Train MAE	Test MAE	Train SMAPE	Test SMAPE	Train MASE	Test MASE
LSTM Regressor	77.05	36.01	61.38	30.42	11.05	4.25
Prophet (out of the box)	56.24	62.97	71.51	83.18	5.415	5.37
ARIMA (7,0,7)	37.55	26.86	65.19	24.25	2.28	2.99
ARIMA trend + seasonality	30.35	21.83	45.55	19.95	2.16	2.30

LSTM Regressor shows the significant gap between training and test MAE (from 77.05 to 36.01), likely due to recursive multistep inference on training data leading to error accumulation. Lower test errors imply that when evaluated on shorter, realistic forecast horizons, the LSTM performs better. LSTM has the highest mean absolute scaled error (MASE) on train and test sets, which means that the prediction is the worst against naïve forecast.

Prophet out of the box with weekly seasonality performs poorly for this time series. It may need tuning. MAE and MASE are similar across train and test. Prophet is not overfitting but is underperforming overall. High SMAPE and MASE indicate poor predictive quality and high deviation from naive forecasts.

ARIMA (7,0,7) performs better than previous models. Its average error is only 26 product units on the test set and 37 product units train set. It indicates that ARIMA (7,0,7) managed to handle linear dependency in data well and forecast sales better in this case than LSTM Regressor with a recursive approach to forecasting. Its test MASE (2.99) is lower than LSTM's, indicating fewer cumulative errors across the forecast horizon and higher proximity to the naive forecast errors.

ARIMA with trend and added seasonality performs the best on train and test sets. This model achieves the best overall performance, with the lowest errors across all metrics for both train and test sets. It indicates that there are trends and seasonal weekly oscillations that ARIMA manages to capture carefully.

Residual Analysis

To compare the results of ARIMA (7,0,7) and ARIMA with trend and added seasonality, we conducted comprehensive residual analysis, including examination of residuals versus ground truth sales, distributional fit, and autocorrelation.

The residuals from an ARIMA (7,0,7) model are illustrated on Fig. 11a. The scatter plot reveals a structured pattern. The residuals reflect a noticeable downward trend in residuals for higher sales values, indicating potential model bias and underperformance in high-sales as well as low-sales regions.

The empirical distribution has noticeably heavier tails, thus it is better approximated by the Laplace distribution (fitted mean = 14.7, STD = 23.8) when compared with the normal distribution (fitted mean = 10, STD = 30.3). This reflects the presence of outliers or unmodeled structure. While most lag values fall within the 95% confidence bounds, several statistically significant autocorrelations remain – especially at lower lags (e.g., lags 1, 2, 7, and 14).

These values indicate the presence of remaining temporal structure in the residuals, violate the assumption of white noise – indicating that certain data features remain badly modeled.

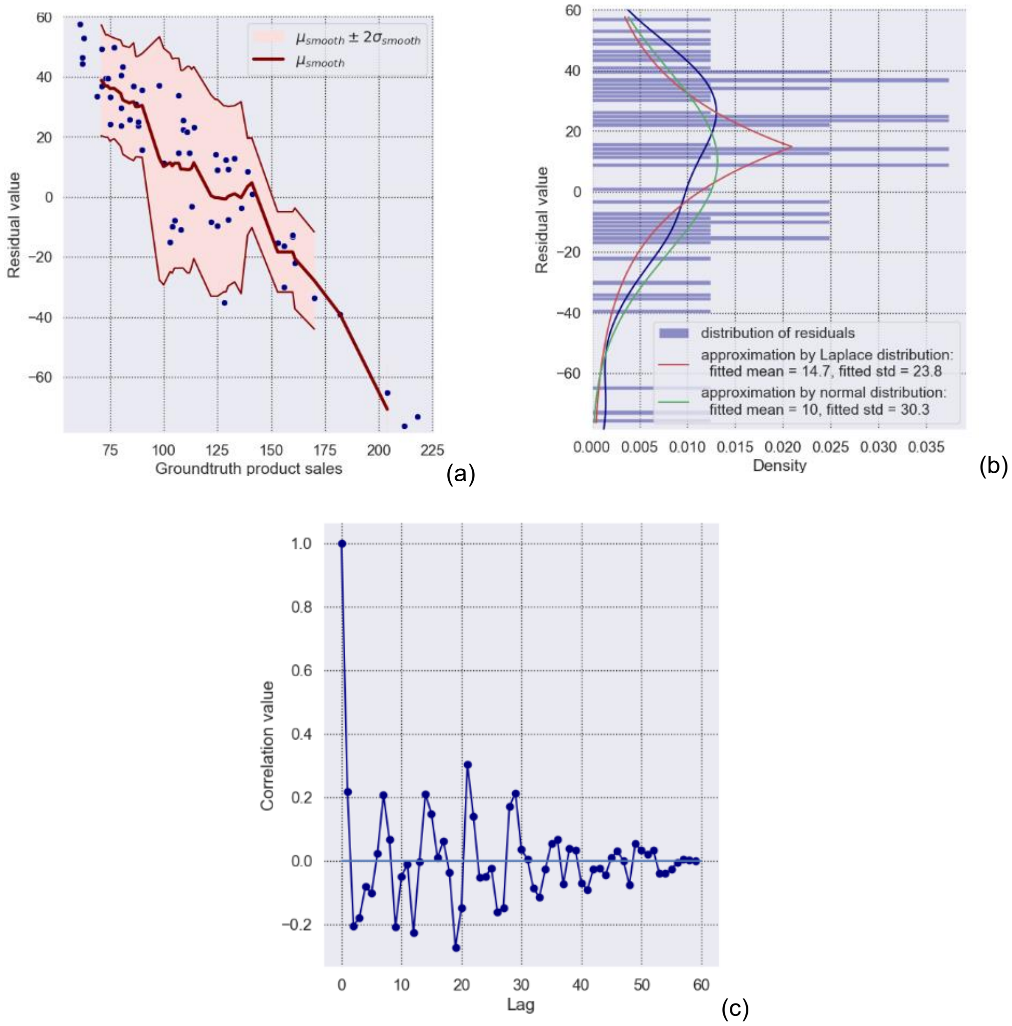

Fig. 11. Scatterplot (a), distribution of residuals (b) and autocorrelation function of residuals (c) for modeling with ARIMA (7,0,7).

Residual plots from the ARIMA with trend + seasonality approach (Fig. 12) reflect the fact that the smoothed residual mean is flatter, and the variance is more stable across sales values. Residual values related to small sales are lower though outliers remain for high sales.

This plot shows improved bias handling due to trend modelling and proper seasonal treatment.

The Laplace approximation fitting remains better than the normal fit, but the mean (closer to 0) and reduced standard deviation indicate more centred, less dispersed residuals. This reflects improved modelling of systematic patterns, though some non-normality remains.

The updated ACF shows weaker and fewer significant lags, indicating a closer approximation to white noise. This confirms that accounting for seasonality and trend improved the temporal independence of residuals.

This result suggests the model more adequately captured the main structure – both trend and seasonal cycles.

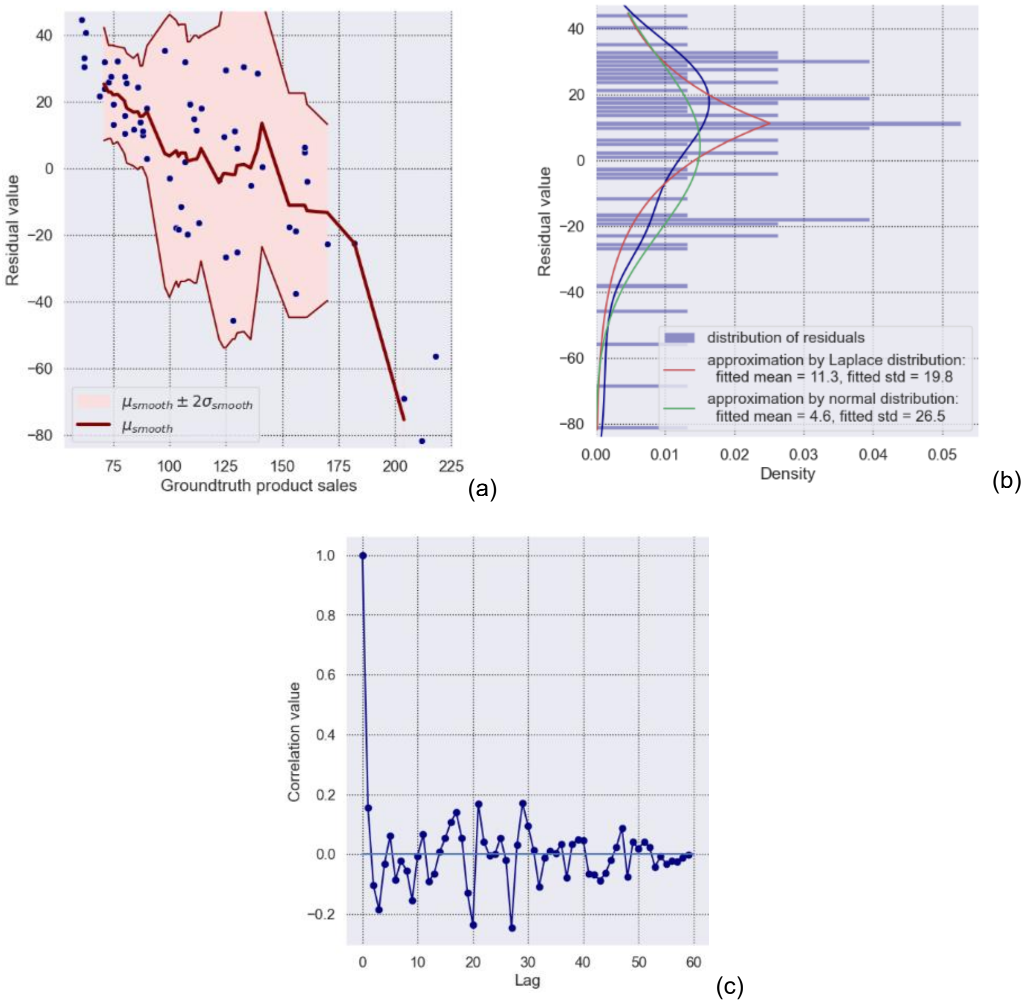

Fig. 12. Scatterplot (a), distribution of residuals (b) and autocorrelation function of residuals (c) for ARIMA(Trend) + Seasonality modeling.

Diebold–Mariano test

We applied the Diebold-Mariano (DM) test to statistically compare the predictive accuracy of several forecasting models on the test set. The test evaluates whether the difference in forecast errors between two models is statistically significant. A positive test statistic indicates the second model has lower forecast error (is more accurate), while a negative value favors the first model. The p-value assesses the significance of this difference. The results are attached to Table 2.

From the results on a test set, we can see that ARIMA significantly outperforms non-tuned Prophet. ARIMA with seasonality significantly outperforms Prophet even more. ARIMA with trend and seasonality significantly improves ARIMA. No significant difference between LSTM and Prophet. ARIMA significantly outperforms LSTM. ARIMA with seasonality significantly outperforms LSTM even more.

Table 2. **Diebold–Mariano test results**

Models' comparison	Test Statistic	P-value
ARIMA vs Prophet	–6.73	7.78×10^{-9}
ARIMA trend + seasonality vs Prophet	–7.86	9.27×10^{-11}
ARIMA trend + seasonality vs ARIMA	–3.63	5.9×10^{-4}
LSTM vs Prophet	–1.05	0.296
LSTM vs ARIMA	6.22	5.4×10^{-8}
LSTM vs ARIMA trend + seasonality	6.25	4.99×10^{-8}

CONCLUSION

In this paper, we focused on predicting retail sales trends using time series analysis and machine learning algorithms. Forecasting methods, including autoregression (AR), moving average (MA), integrated autoregression with moving average (ARIMA), seasonal ARIMA (SARIMA), and long short-term memory (LSTM) played an important role in the study. We also compared the results to Prophet model with default hyperparameters. Based on experiments conducted with sales data we've folded up our conclusions.

Seasonality Matters: The presence of distinct weekly patterns proved vital for accurate forecasting; explicitly modeling seasonality with ARIMA significantly reduced errors.

Trend Decomposition: Decomposing the data into separate trend and seasonal components allowed a more precise fit, which is especially helpful in retail domains with strong cyclical effects.

Neural Networks Caveats: While LSTM can capture complex patterns, performance depends on architecture design, training strategy, and the amount of data. A single-step recurrent approach can lead to escalating errors in multi-step forecasts. The quality of forecasts is dependent on the initialized sequence to regress from.

Practical Implications: For retail businesses, relatively straightforward ARIMA extensions remain a strong baseline. More complex models are promising but require careful tuning.

These findings underscore the importance of robust time series decomposition and targeted model selection. The results of this work can be used to improve solutions in the field of inventory management and sales planning in retail. Future work will explore hybrid methodologies, combining the interpretability of statistical models with the flexibility of deep learning architectures for hierarchical and multi-horizon sales forecasts.

ACKNOWLEDGMENTS AND FUNDING SOURCES

The authors sincerely gratitude EPAM University for course materials that were used in this article for theoretical description of ARIMA and certain plotting logic.

The authors received no financial support for the research, authorship, and/or publication of this article.

COMPLIANCE WITH ETHICAL STANDARDS

Conflict of Interest: The authors declare that they have no competing interests.

AUTHOR CONTRIBUTIONS

Conceptualization, [O.K., R.S.]; methodology [O.K.]; validation, [O.K., R.S., I.K.]; formal analysis, [O.K.].; investigation, [O.K.] writing – original draft preparation, [O.K.]; writing – review and editing, [R.S., I.K.]; visualization [O.K.], supervision, [R.S., I.K.].

All authors have read and agreed to the published version of the manuscript.

REFERENCES

[1] Pavlyshenko, B.M. (2019). Machine-learning models for sales time series forecasting. *Data Stream Mining and Processing*, *4*(1), 15. https://doi.org/10.3390/data4010015

[2] Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2021). The M5 competition: Background, organization, and implementation. *International Journal of Forecasting*. https://doi.org/10.1016/j.ijforecast.2021.07.007

[3] Pongdatu, G. A. N., & Putra, Y. H. (2018). Seasonal time series forecasting using SARIMA and Holt Winter's exponential smoothing. *IOP Conference Series: Materials Science and Engineering, 407*, 012153. https://doi.org/10.1088/1757-899x/407/1/012153

[4] Wang, Y., Zhu, S., & Li, C. (2019). Research on multistep time series prediction based on LSTM. *2019 3rd International Conference on Electronic Information Technology and Computer Engineering (EITCE)*, Xiamen, China, 1155-1159. https://doi.org/10.1109/EITCE47263.2019.9095044

[5] Wei, H., & Zeng, Q. (2021). Research on sales forecast based on XGBoost-LSTM algorithm model. *Journal of Physics: Conference Series, 1754*(1), 012191. https://doi.org/10.1088/1742-6596/1754/1/012191

[6] Massaro, A., et al. (2021). Augmented data and XGBoost improvement for sales forecasting in the large-scale retail sector. *Applied Sciences, 11*(17), 7793. https://doi.org/10.3390/app11177793

[7] Kaggle. (2020). M5 Forecasting dataset. Publicly available at: https://www.kaggle.com/competitions/m5-forecasting-accuracy/data

[8] Rasul, K., Ashok, A., Williams, A. R., Ghonia, H., Bhagwatkar, R., Khorasani, A., Bayazi, M. J. D., Adamopoulos, G., Riachi, R., Hassen, N., Biloš, M., Garg, S., Schneider, A., Chapados, N., Drouin, A., Zantedeschi, V., Nevmyvaka, Y., & Rish, I. (2024). Lag-Llama: Towards foundation models for probabilistic time series forecasting. *arXiv*. https://doi.org/10.48550/arXiv.2310.08278

[9] Polykhov, M. Time series forecasting using LSTM (Прогнозуваннння часових рядів методом LSTM). *Ekmair*. https://ekmair.ukma.edu.ua/items/7f55fa50-bf7c-4ba3-a7a0-cab60010af06

[10] Xu, Chengjin, Nayyeri, Mojtaba, Alkhoury, Fouad, Shariat Yazdi, Hamed, Lehmann, Jens. (2020). Temporal Knowledge Graph Embedding Model based on Additive Time Series Decomposition. https://www.researchgate.net/publication/344450419_Temporal_Knowledge_Graph_Embedding_Model_based_on_Additive_Time_Series_Decomposition

[11] Olah, C. (2015). Understanding LSTM Networks. colah's blog. https://colah.github.io/posts/2015-08-Understanding-LSTMs/

МОДЕЛЮВАННЯ РОЗДРІБНИХ ПРОДАЖІВ З ВИКОРИСТАННЯМ МЕТОДІВ АВТОРЕГРЕСІЙНОГО ІНТЕГРОВАНОГО КОВЗНОГО СЕРЕДНЬОГО ТА ДОВГОСТРОКОВОЇ ПАМ'ЯТІ

Олексій Качмар *, ***Роман Шувар***, ***Ігор Колич***
Кафедра системного проєктування,
Львівський національний університет імені Івана Франка
вул. Драгоманова 50, 79005 Львів, Україна

АНОТАЦІЯ

Вступ. Прогнозування роздрібних продажів має критичне значення для ефективного управління запасами та ланцюгом поставок. Через складність і обсяг сучасних роздрібних даних традиційних статистичних моделей часто недостатньо. Інтеграція методів машинного навчання з аналізом часових рядів дозволяє суттєво покращити точність прогнозів.

Матеріали та методи. У дослідженні використано набір даних M5, що включає понад 30 000 часових рядів щоденних продажів товарів у магазинах. Проведено очищення та попередню обробку даних, включно з обробкою пропусків, а також розділенням рядів на навчальні та тестові вибірки. Було застосовано три підходи до прогнозування. Перший - класична авторегресивна модель з ковзними середніми (ARMA), що не враховує явним чином сезонність. Другий метод із розкладанням часового ряду на трендову й сезонну компоненти, побудовою моделі для скоригованого ряду, а потім відновленням повного прогнозу з урахуванням сезонності. Третій - модель глибокого навчання на основі мережі типу «довга короткочасна пам'ять» (LSTM), здатна виявляти довготривалі залежності. Для оцінювання використовувалась метрика середньої абсолютної помилки (MAE), а також аналіз автокореляції залишків.

Результати. Аналіз продажів окремого товару виявив чітко виражену тижневу сезонність. Авторегресивний підхід без декомпозиції частково відображав структуру, але залишав суттєву автокореляцію в залишках. Другий метод, який враховував тренд та сезонну компоненту, показав найкращі результати за MAE, досягнувши найточнішого короткострокового прогнозу. Нейронна мережа LSTM, реалізована у багатокроковому рекурсивному режимі, не перевершила ARMA-модель через накопичення помилок і, ймовірно, неідеальну конфігурацію архітектури.

Висновки. Для роздрібних часових рядів із регулярною тижневою сезонністю найкращу точність забезпечують авторегресивні моделі з урахуванням тренду й сезонної декомпозиції. Нейронні мережі мають потенціал у моделюванні складних залежностей, однак вимагають ретельного налаштування, щоб уникнути накопичення помилок. У майбутньому варто дослідити комбіновані підходи для багаторівневих і багатогоризонтних прогнозів.

Ключові слова: аналіз часових рядів, машинне навчання, роздрібне прогнозування, ARIMA, LSTM, сезонність

Received / Одержано	Revised / Доопрацьовано	Accepted / Прийнято	Published / Опубліковано
28 April, 2025	09 May, 2025	13 May, 2025	26 June, 2025