MJI
Media Jurnal Informatika
Vol. 18, No. 1, June 2026
p-issn : 2088-2114, e-issn : 2477-2542

Stock Demand Forecasting for MSME E-Commerce Using LSTM and Facebook Prophet: A Comparative Study

Fatimatuzzahra [a,1,*], Akmal Amilunnizar [a,2], Khen Dedes [a,3] , Helyatin Nisyak [b,4] , Nadzirotul Fitriyah [b,5]

[a] Politeknik Negeri Jember
[b] Universitas Ibrahimy
[1] fatimatuzzahra@polije.ac.id*; [2] uzumakisujai@gmail.com; [3] khen_dedes@polije.ac.id; [4] helyatinnisa@gmail.com; [5] nadzirotulfitriyah@gmail.com;
* corresponding author

ARTICLE INFO

Article history
Received 2026-06-04
Revised 2026-06-23
Accepted 2026-06-27

Keywords
E-Commerce;
Stock Forecasting;
Long Short-Term Memory;
Facebook prophet;
MSMEs Competitiveness.

ABSTRACT

Manual sales processes and stock management in micro, small, and medium enterprise (MSME) settings often lead to limited market reach and inefficient inventory management. The purpose of this research is to solve these operational problems by designing and developing a web-based e-commerce system equipped with an integrated monthly stock demand forecasting module to enhance the competitiveness of MSMEs in the roster industry. The system was developed following the waterfall methodology and it has a decoupled architecture to separate the artificial intelligence computational workloads from the core application. Two time series forecasting models Long Short-Term Memory (LSTM) and Facebook Prophet were applied and compared for forecasting stock requirements from intermittent, zero-inflated demand patterns of historical sales data. System functionality was validated using User Acceptance Testing and the forecasting accuracy was measured using Root Mean Square Error (RMSE) and Weighted Mean Absolute Percentage Error (WMAPE). The performance evaluation showed that the unscaled LSTM model outperformed the linear additive regression method of Facebook Prophet in terms of lower physical volume deviation and consistent operational error in the course of the evaluation period. The developed platform provides a reliable data-driven decision support for inventory management. The incorporation of forecasting using neural networks in the e-commerce system has reduced the risk of stockouts, expanded the market, and proved the increase in the business competitiveness of artisan MSMEs.

1. Introduction

Micro, Small, and Medium Enterprises (MSMEs) form the backbone of the Indonesian economy, contributing significantly to national Gross Domestic Product (GDP) and employment. However, certain sectors such as the roster manufacturing industry in Malang, East Java still rely heavily on traditional operational methods. Preliminary observations and interviews reveal that local roster craftsmen predominantly use manual processes for stock recording, order management, and product marketing. This dependence on conventional techniques, combined with limited digital literacy, leads to restricted market reach, frequent inventory inaccuracies, and delayed order fulfillments. As a result, there is an urgent need to digitize these business processes to boost efficiency and competitiveness.

A key gap in current research lies in the practical integration of modern time-series forecasting models directly into functional e-commerce platforms tailored for local MSMEs. While prior studies have demonstrated the effectiveness of advanced predictive algorithms in supply chain and inventory management, most focus on theoretical model evaluations rather than real-world system implementation. For instance, deep learning models especially Long Short-Term Memory (LSTM) have shown exceptional accuracy in e-commerce sales forecasting, achieving error rates below 1% in specific datasets [1]. Comparative studies across diverse domains for example, daily bakery sales [2], multi-category retail [3], aerosol supply chains [5], and even public health and stock index forecasting [11]–[12], have consistently shown that LSTM excels at capturing complex, long-term non-linear patterns, while Facebook Prophet demonstrates strong performance in handling seasonal variations. Further advancements include hybrid models that combine LSTM and Prophet, successfully applied to dynamic tasks like electricity demand management [4]. Autoregressive models with external inputs (ARIMAX) have also proven effective in sales forecasting applications [13].

Despite these proven capabilities, the state-of-the-art research remains largely confined to model performance benchmarks without integrating these tools into practical, user-friendly e-commerce systems for MSMEs. To bridge this gap, this study proposes a responsive, web-based electronic commerce platform with an embedded monthly stock forecasting module designed specifically for roster craftsmen. The innovation lies not only in the integration of AI-driven forecasting but also in the system's architecture and algorithmic tuning for real-world use.

The proposed solution features a decoupled software architecture that separates the web-based sales and inventory interface from the artificial intelligence computational engine. This design ensures optimal server performance, preventing latency during forecasting tasks. Furthermore, this study presents a novel comparative analysis using zero-inflated, real-world sales data a common characteristic of intermittent retail demand in craft-based MSMEs. Specifically, we evaluate an unscaled LSTM (LSTM No-Scale) model trained on raw, native-scale data against the Facebook Prophet algorithm. This approach preserves physical volume sensitivity, ensuring predictions reflect actual stock levels without distortion from normalization processes.

Ultimately, this research delivers a fully functional, integrated platform that goes beyond digitizing sales it actively mitigates stockout risks and enhances business competitiveness through proactive, data-driven decision-making. By combining intelligent forecasting with accessible technology, it empowers artisan MSMEs to thrive in the digital economy.

2. Method

This methodology section outlines the systematic steps taken to develop and test the integration of an inventory forecasting model into the e-commerce information system of MAS Roster Malang. In general, the methodological structure of this research encompasses six main components: (1) the comparative research approach used, (2) the object and scope of the sales system being studied, (3) data collection techniques for historical transactions, (4) the software and frameworks (such as Laravel and Flask) implemented, (5) the system development procedures using the Waterfall method, and (6) data analysis techniques utilizing evaluation metrics (such as MAE, RMSE, and WMAPE) to determine the best forecasting model between Long Short-Term Memory (LSTM) and Facebook Prophet.

2.1 Type and Approach of Research

This study employs a comparative quantitative approach combined with a software development method. The comparative approach is applied to test and compare two time-series forecasting models, namely the Facebook Prophet additive regression algorithm and the Long Short-Term Memory (LSTM) artificial neural network, to find the champion model with the best generalization capabilities. For system development, this research implements the Waterfall method, which is linear and sequential, to ensure comprehensive documentation from the beginning to the end of the development cycle [15]-[16].

2.2 Object and Scope of Research

The main object of this research is a web-based e-commerce system integrated with a monthly stock forecasting feature for the MAS Roster Malang MSME. The scope of the forecasting computation is exclusively focused on the retail order pathway (*continuous demand*) to reduce probability distortion often caused by bulk order anomalies (*lumpy demand*). This sales management system is limited to a web platform (excluding mobile applications) and does not include automated payment system integration or payment gateways.

2.3 Data Collection Techniques

Field data collection was conducted through direct observation and informal interviews with the owner of MAS Roster Malang to analyze the business workflow. Primary data for machine learning computations were gathered by extracting historical sales transaction documents (secondary data) of the MSME from 2023 to 2025. This transaction data contains temporal dimensions (transaction dates), product types, and sales volume quantities, which were subsequently transformed into a time-series matrix dataset format.

2.4 Tools and Materials Used

The hardware environment utilized a Intel Core i7-1165G7 processor, 16 GB DDR4 RAM, and Intel Iris Xe integrated graphics laptop for development and local server hosting, alongside an Android smartphone running Android 14 with a Qualcomm Snapdragon 8 Plus Gen 1 processor and 8 GB RAM for responsive user interface testing. The software ecosystem was built on a decoupled architecture framework. The transactional frontend and backend were constructed using the Laravel (PHP) framework and MySQL for relational database management. The predictive analytics microservice was engineered using Python within a Jupyter Notebook environment, deployed via the Flask framework. Essential libraries included Pandas for data manipulation, Scikit-Learn and TensorFlow/Keras for building the LSTM architecture, and the Facebook Prophet library for statistical regression [8].

2.5 Research Procedures or Stages

The study was executed through six sequential phases in accordance with the Waterfall methodology: (1) *Requirements Analysis*: Gathering functional specifications through MSME observation and dataset extraction. (2) *System Design*: Creating Entity Relationship Diagrams (ERD) to structure the database and Use Case diagrams to map the decoupled architecture flow. (3) *Implementation*: Translating the conceptual designs into the Laravel e-commerce codebase and the Flask AI microservice. (4) *Testing*: Conducting Alpha testing (Black-Box fungsionality) and Beta testing (User Acceptance Testing) involving end-users and IT experts [17]. (5) *Deployment*: Transitioning the system to a live web server environment. (6) *Maintenance*: Monitoring model performance and executing periodic data retraining.. Specifically for the artificial intelligence module employs a specialized data processing pipeline that begins with rigorous cleaning, retail segmentation, and monthly temporal aggregation. This specific aggregation strategy is implemented to mitigate the prevalence of zero-inflated values characteristic of intermittent demand. Furthermore, comprehensive preprocessing encompassing the resolution of missing values and structural normalization is applied, as the integrity of these initial stages exerts a profound influence on the ultimate accuracy of time-series forecasting models [18]. The data is then partitioned using the Sequential Holdout technique chronologically with a ratio of 80% training data and 20% testing data to prevent future data leakage.

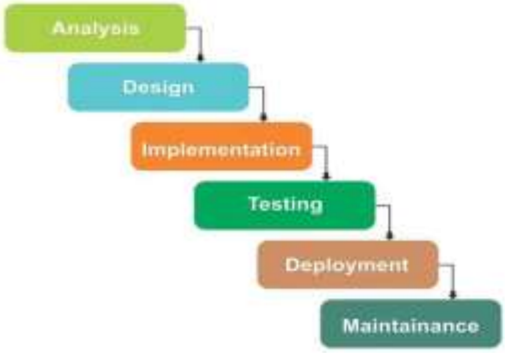

Figure 1 Waterfall Method Stages (Widyantoro dkk., 2025)

This research will implement the Waterfall software development method. The Waterfall method is a linear and sequential software development approach, where each development phase must be completed sequentially before entering the next phase. This approach was chosen because the project has relatively clear and well-defined requirements at the initial stage, and it emphasizes the completeness of documentation at each stage. Simulations of the Waterfall method indicate that appropriate resource allocation management can significantly accelerate project completion [7].

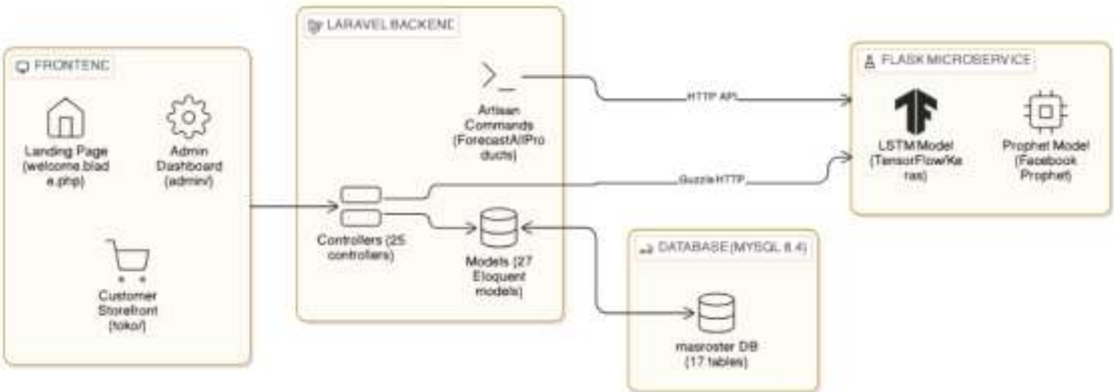

Figure 2 System Architecture

The developed system architecture consists of four main components: the frontend, backend, database, and forecasting module. Users interact with the system via the web-based frontend to manage products, record transactions, and monitor inventory after securely logging in. The backend handles the core business logic and processes these activities, while the database stores all operational and historical sales data.

Integrated into this ecosystem is a Python-based forecasting module that utilizes LSTM and Facebook Prophet to predict future stock requirements. Historical sales data from the database undergoes pre-processing and model training to generate accurate predictions, which are seamlessly displayed on the system dashboard. This integration allows MSMEs to automatically obtain data-driven stock insights, ultimately improving sales efficiency and optimizing inventory procurement in a planned manner.

.

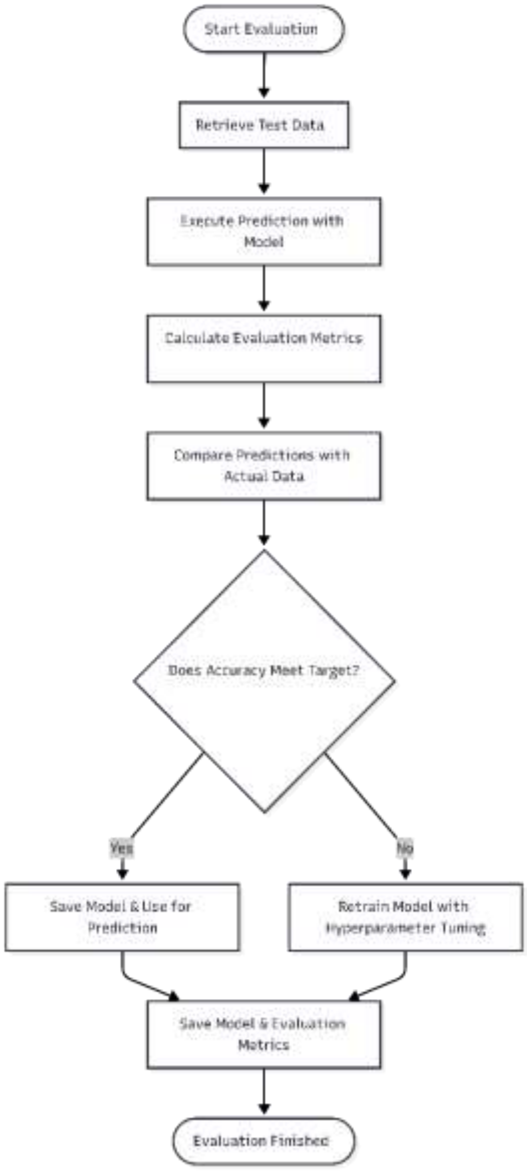

Figure 3 Evaluation Flows

The evaluation of the LSTM and Prophet models was conducted utilizing three assessment metrics, namely Weighted Mean Absolute Percentage Error (WMAPE), Mean Absolute Error (MAE), and Root Mean Square Error (RMSE). The selection of these three metrics aims to provide a comprehensive view of prediction errors in the field of time-series forecasting [6] . The objective of this evaluation is to assess the extent of the models' accuracy and precision in predicting stock requirements based on existing data.

2.6 Data Analysis Techniques

The prediction reliability of the forecasting algorithms is quantitatively evaluated using three main error metrics: Weighted Mean Absolute Percentage Error (WMAPE), Mean Absolute Error (MAE), and Root Mean Square Error (RMSE) [19]. WMAPE is specifically used to address division-by-zero issues in sporadic data. On the other hand, the e-commerce system's functionality is evaluated through a User Acceptance Testing (UAT) procedure using the Black-Box method. The testing is divided into Alpha Testing (functional validation by IT experts) and Beta Testing (usability evaluation by MSME administrators) by collecting feedback through a Likert Scale-based questionnaire.

3. Results and Discussion

This section presents and evaluates the outcomes of implementing the web-based e-commerce system integrated with the stock forecasting feature for the MAS Roster Malang MSME. The discussion covers the performance evaluation of the Long Short-Term Memory (LSTM) and Facebook Prophet models in predicting inventory needs, as well as the system's operational viability based on user acceptance testing.

3.1 Presentation of Research Results

This subsection presents the empirical outcomes of the forecasting model evaluation and the implementation of the e-commerce system. The performance of the forecasting algorithms was measured using RMSE, MAE, and WMAPE metrics on the testing dataset. The evaluation specifically compared two variations of the Long Short-Term Memory (LSTM) artificial neural network: the LSTM Advanced architecture (which utilized MinMaxScaler normalization) and the LSTM No-Scale architecture (which operated without data scaling intervention). The comparative results revealed that the LSTM No-Scale model achieved superior stability and consistently outperformed the Advanced model across the evaluated products. For instance, on the baseline product (Roster Beton 20 x 20 Lubang 4), eliminating the scaling mechanism successfully reduced the RMSE from 58.90 to 56.84, the MAE from 48.53 to 43.36, and the WMAPE from 42% (0.42) to 35% (0.35).On the other hand, the Facebook Prophet algorithm achieved an RMSE of 60.28, an MAE of 34.52, and a WMAPE of 49% on specific baseline products, but struggled with extreme volatility.

Table 1. Statistical Result LSTMs Results

Product	Adv RMSE	Adv MAE	Adv WMAPE	NoSc RMSE	NoSc MAE	NoSc WMAPE
Roster Beton 20 x 20 Lubang 4	58.90	48.53	0.42	56.84	43.36	0.35
Roster Beton 20 x 20 Cawang Tonjok	86.98	75.49	1.03	83.54	68.16	0.84
Bovenlis Beton 30 x 60 Polos	40.93	33.08	0.96	37.56	30.54	0.84
Bovenlis Beton 40 x 70 Polos	46.78	39.78	12.90	42.62	37.01	11.62
Roster Beton 20 x 20 Berlian	87.33	84.07	0.66	84.36	80.04	0.67

Table *1* presents a comprehensive performance comparison between two variations of the Long Short-Term Memory (LSTM) architecture: LSTM Advanced (utilizing *MinMaxScaler* normalization) and LSTM No-Scale (operating without scaling intervention) across five top product entities. The evaluation demonstrates that the LSTM No-Scale model consistently outperforms the Advanced architecture. For instance, on the baseline product (*Roster Beton 20 x 20 Lubang 4*), eliminating the data scaling mechanism successfully reduced the RMSE from 58.90 to 56.84, the MAE from 48.53 to 43.36, and the WMAPE from 42% (0.42) to 35% (0.35). This empirical superiority reveals a fundamental computational insight regarding zero-inflated intermittent demand: conventional decimal compression through *MinMaxScaler* distorts the temporal signal by forcing zero-demand periods and sporadic extreme surges into an overly narrow decimal range. By retaining the data's raw native scale, the LSTM No-Scale model maintains its sensitivity to map the true hierarchy of order volatility without losing the physical context of the items, thereby establishing it as the definitive champion model for this segment.

Table 2. Statistical Result Prophet Results

Product	Data Row	Month	Train	Test	RMSE	MAE	WMAPE
Roster Beton 20 x 20 Lubang 4	54	32	25	7	60.28	34.52	0.49
Roster Beton 20 x 20 Cawang Tonjok	25	30	24	6	76.69	41.34	0.79
Bovenlis Beton 30 x 60 Polos	66	33	26	7	48.69	44.18	1.05
Bovenlis Beton 40 x 70 Polos	81	33	26	7	46.99	37.81	8.87
Roster Beton 20 x 20 Berlian	21	22	17	5	70.12	38.60	0.77

Table 2. *Statistical Result Prophet Results* outlines the dataset profiling and the comprehensive evaluation metrics resulting from the implementation of the Facebook Prophet algorithm on the top five product entities. The testing scheme was conducted by applying a hold-out split partition, where historical observations were divided into training and testing ranges based on the chronological proportion of observation months for each product.

Based on the analytical review of the test results, the Prophet algorithm demonstrated its optimal operational performance on the *Roster Beton 20 x 20 Lubang 4* entity, successfully suppressing the percentage error (WMAPE) to 49% (0.49) with an absolute volume deviation (MAE) of 34.52 units. However, the stability of these metrics proved highly fragile when the Prophet architecture was confronted with products exhibiting extreme void volatility. This structural failure is transparently recorded in the *Bovenlis Beton 40 x 70 Polos* entity. Although the model recorded seemingly moderate physical error values (an RMSE of 46.99 and an MAE of 37.81 units), the WMAPE metric for this product skyrocketed to 8.87, which is equivalent to an 887% error rate.

This massive escalation in percentage error is not a result of a system computational error, but rather empirical proof of the inherent limitation of the additive regression architecture. The Prophet algorithm is mathematically designed to map smooth and continuous trend waves. When this model is forced to process intermittent demand characteristics dominated by zero-inflated data points, Prophet experiences curve adjustment disorientation. When the actual testing data reveals a value of zero, even the slightest inaccurate guess from Prophet will be accumulated as an almost infinite percentage error. This condition mathematically justifies the conclusion that the baseline Prophet algorithm is too rigid and poses a high risk if utilized as the primary forecasting model for the Masroster retail inventory system, which has a highly sporadic order history.

Regarding system validation, User Acceptance Testing (UAT) was conducted through Alpha and Beta testing. The Beta testing, which involved end-users evaluating the system using a Likert scale, yielded highly positive results [20]. The e-commerce module scored an 85% acceptance rate for its memorability and 83.3% for efficiency. Similarly, the AI forecasting dashboard achieved a 90% acceptance rate for memorability, proving that the complex predictive data was successfully translated into an intuitive and user-friendly interface.

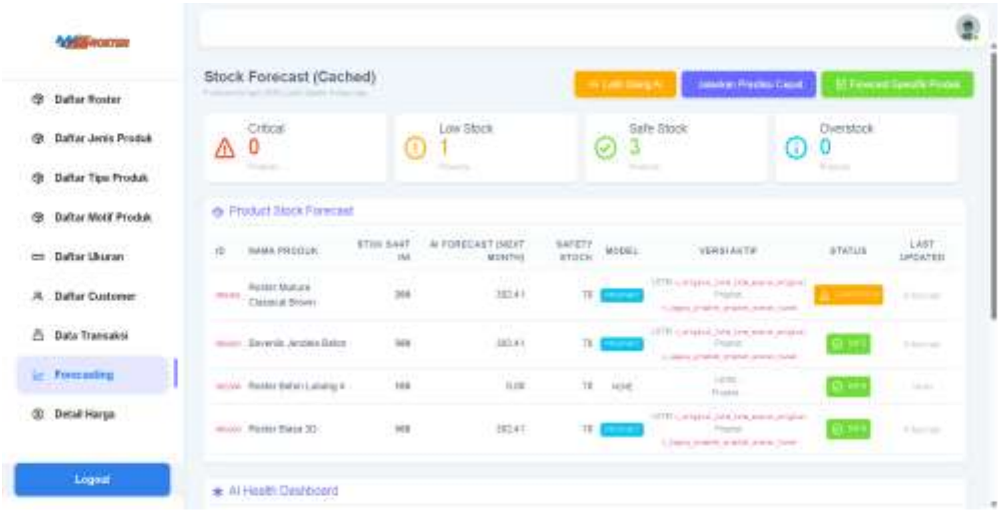

Figure 4 Forecasting Dashboard

Figure 4 illustrates the implementation of the Forecasting Dashboard within the MasRoster E-Commerce and Inventory Management System. This interface serves as a decision-support module that presents AI-based demand forecasting results for future inventory planning. The dashboard includes a data freshness notification that alerts administrators when forecasting data stored in the database is outdated and requires regeneration through the forecasting process. Additionally, summary indicators provide an overview of inventory conditions by categorizing products into stock status groups such as *Low Stock*, *Safe Stock*, and *Overstock*.

The main component of this interface is a forecasting table that compares current inventory levels with predicted demand. The table displays product information, current stock quantities, forecasted demand values, and the forecasting model used for each prediction, which in this implementation is Facebook Prophet. To support inventory decision-making, the system applies a rule-based classification mechanism using a predefined safety stock threshold. Based on this evaluation, products are automatically categorized according to their inventory risk level, enabling administrators to identify replenishment priorities and optimize stock management activities.

The forecasting dashboard is fully integrated with the web-based e-commerce system and the machine learning module. Historical sales data stored in the MySQL database are processed by forecasting models developed in Python, including LSTM and Facebook Prophet. Prediction results are then stored back into the database and displayed through the dashboard interface. This integration enables real-time access to forecasting information and supports data-driven inventory planning, helping the organization reduce stock shortages, avoid overstock conditions, and improve overall supply chain efficiency.

Table 3. Forecasting Accepting Index

Aspect	Acceptance Indexs	Percentage
Learnability	(8.75 / 10) × 100%	87.5%
Efficiency	(7.67 / 10) × 100%	76.70%
Memorability	(9.0 / 10) × 100%	90%
Errors	(8.0 / 10) × 100%	80%
Satisfaction	(7.75 / 10) × 100%	77.5%

Table 3 presents the user acceptance indices for the AI forecasting dashboard, highlighting a highly positive reception from the factory administrators. The system truly excels in Memorability, earning an impressive 90%, which indicates that despite the complex predictive logic running behind the scenes, the visual interface is incredibly intuitive and easy for users to remember without needing to relearn. This is closely followed by Learnability at 87.5%, proving that administrators can quickly grasp how to read the AI stock projections. While overall Satisfaction (77.5%) and Error prevention (80%) reflect a stable and reliable user experience, the Efficiency aspect scored the lowest at 76.70%. This reveals a valuable human insight: although the forecasting tool successfully supports daily logistical decisions, future system updates could focus on streamlining how fast the data is presented, helping administrators extract those crucial insights even quicker.

Table 4. E-commerce Accepting Index

Aspect	Acceptance Indexs	Percentage
Learnability	(8 / 10) × 100%	80%
Efficiency	(8.33 / 10) × 100%	83.30%
Memorability	(8.5 / 10) × 100%	85%
Errors	(7.5 / 10) × 100%	75%
Satisfaction	(7.75 / 10) × 100%	77.5%

Table 3 presents the user acceptance indices for the Masroster e-commerce ecosystem, reflecting a solid and positive reception from its everyday users. The platform truly shines in Memorability, achieving the highest score of 85%, which confirms that the shopping flow—from browsing the catalog to the final checkout process—feels remarkably natural and intuitive for both customers and staff to remember. The system also proves highly practical for daily business operations, earning an 83.30% in Efficiency and 80% in Learnability. While overall user Satisfaction is comfortably strong at 77.5%, the Errors aspect scored the lowest at 75%. This reveals a valuable human insight: although the digital shopping experience is generally smooth and well-received, future system enhancements should focus on building smarter error-prevention features to gently guide users and prevent accidental input mistakes when filling out transaction forms.

3.2 Analysis of Findings

The evaluation indicates that the LSTM No-Scale architecture performs consistently better on intermittent demand data compared to the standard LSTM equipped with a MinMaxScaler. Scaling the data compressed the zero-inflated transactions and sporadic demand surges into an overly narrow decimal range, which distorted the temporal signal. By retaining the raw native scale, the LSTM No-Scale model accurately mapped the hierarchy of order volatility without losing the context of physical quantities.

Conversely, while Prophet performed well on continuous baseline products, it suffered from severe structural failure when faced with extreme zero-inflated intermittent demand. For example, on certain products, Prophet's WMAPE surged to 887%. This occurs because Prophet's additive regression architecture is designed for smooth, continuous trends and overfits when attempting to minimize squared errors on sparse data. This finding aligns with previous studies indicating that traditional statistical models struggle with non-linear, zero-inflated demand compared to robust deep learning architectures [5], although deep learning models inherently require sufficient data density to function optimally [3].

3.3 Implications of the Results

The integration of the LSTM forecasting model into the Laravel-based e-commerce system provides significant practical implications for the MAS Roster MSME. By utilizing a decoupled architecture where the e-commerce backend runs on PHP and the AI computation is isolated in a Python Flask microservice, the system delivers asynchronous batch forecasting without causing server latency or bottlenecks. Practically, this empowers the MSME management to proactively determine adaptive safety stocks. The predictive dashboard prevents capital from freezing due to dead stock during zero-demand periods, while simultaneously mitigating the risk of revenue loss (lost sales) from sudden stockouts, thereby increasing the company's competitiveness.

3.4 Limitations of the Study

Despite the successful implementation, this study encountered several limitations. First, the forecasting model was only highly accurate for the retail segment (continuous demand). The bulk order segment (lumpy demand) remains challenging to predict accurately due to extreme sparsity and random volume surges. Second, the dataset utilized was limited to a three-year historical sales period (2023–2025). This restricted volume of data (data starvation) limited the deep learning model's ability to recognize long-term seasonal patterns effectively. Future research should explore hybrid models, such as Prophet-LSTM, or specific sparse data algorithms like Croston’s Method, and incorporate external variables like promotional data to improve accuracy on bulk orders.

4. Conclusion

This research introduces a web-based e-commerce platform featuring a monthly stock demand forecasting module, specifically designed for an artisan micro, small, and medium enterprise in the garment craft industry. Utilizing the waterfall technique and a decoupled design, it incorporates a PHP backend and a Python Flask microservice, improving efficiency by reducing server latency in batch forecasting. The study evaluates two time-series forecasting models, demonstrating that the LSTM No-Scale architecture surpassed the LSTM Advanced (MinMaxScaler), attaining reduced RMSE (from 58.90 to 56.84), MAE (from 48.53 to 43.36), and WMAPE (from 42% to 35%) for baseline products. This benefit is ascribed to maintaining the data's original scale, which safeguarded the model's sensitivity to zero-inflated sporadic demand. Conversely, Facebook Prophet encountered difficulties with sparse, zero-inflated items, resulting in performance concerns that culminated in a WMAPE of 887% for a particular product.

User acceptability testing confirmed the system's efficacy, with the e-commerce platform achieving an 85% memorability score and an 83.3% efficiency score, and the AI forecasting dashboard attained a memorability score of 90% and a learnability score of 87.5%. Nonetheless, constraints are there concerning the generalizability of the results. The models proficiently managed retail (continuous) demand but encountered difficulties in forecasting bulk (lumpy) demand. Furthermore, the three-year history dataset (2023–2025) constrained the deep learning models' capacity to identify long-term seasonal patterns. Future research should explore hybrid architectures, such as Prophet-LSTM, sparse-data algorithms like Croston's Method, and the incorporation of external variables to improve forecasting accuracy across diverse demand sectors.

Acknowledgment

The author thanks Fatimatuzzahra, S.Kom., M.Kom. for her continuous guidance, supervision, and valuable insights throughout the development of this research. Appreciation is also extended to Mr. Muhammad Fahmi Amin and the entire management of MAS Roster Malang for providing the essential sales dataset and granting permission to conduct this study.

Declarations

Author contribution. A. Amilunnizar designed the system architecture, performed the data analysis, implemented the machine learning models (LSTM and Facebook Prophet), and wrote the manuscript. F. Fatimatuzzahra provided supervision, validated the research methodology, and reviewed the final manuscript. Funding statement. This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors. Conflict of interest. The authors declare no conflict of interest. Additional information. No additional information is available for this paper.

Data and Software Availability Statements

The historical sales datasets analyzed during the current study are proprietary to MAS Roster Malang and are not publicly available due to business confidentiality. However, aggregated data and the web-based e-commerce system's source code developed during the study are available from the corresponding author upon reasonable request.

References

[1] S. Zhou, "E-commerce sales forecast based on neural network LSTM," in *Proceedings of the 6th EAI International Conference on Computer Science and Artificial Intelligence*, 2023, doi: 10.4108/eai.26-5-2023.2334251.

[2] I. G. T. Suryawan, I. K. N. Putra, P. M. Meliana, and I. G. I. Sudipa, "Performance comparison of ARIMA, LSTM, and Prophet methods in sales forecasting," *Sinkron: Jurnal dan Penelitian Teknik Informatika*, vol. 8, no. 4, pp. 2410–2421, 2024, doi: 10.33395/sinkron.v8i4.14057.

[3] D. Brykin, "Sales forecasting models: Comparison between ARIMA, LSTM and Prophet," *Journal of Computer Science*, vol. 20, no. 10, pp. 1222–1230, 2024, doi: 10.3844/jcssp.2024.1222.1230.

[4] S. Albahli, "LSTM vs. Prophet: Achieving superior accuracy in dynamic electricity demand forecasting," *Energies*, vol. 18, p. 278, 2025, doi: 10.3390/en18020278.

[5] N. Sunendar, H. P. Putro, and R. Hesananda, "Prediksi penjualan aerosol menggunakan algoritma ARIMA, LSTM dan GRU," *Insologi: Jurnal Sains dan Teknologi*, vol. 4, no. 1, pp. 113–126, 2025, doi: 10.55123/insologi.v4i1.4868.

[6] A. A. Fauziyyah, J. P. A. Brahmana, P. L. Simatupang, E. B. Soewono, and H. Hayati, "Development of a sales forecasting application using the autoregressive integrated moving average method with external input (ARIMAX)," *Media Jurnal Informatika*, vol. 12, no. 2, pp. 413-424, Dec. 2025, doi: 10.35194/mji.v17i2.5693.

[7] A. Saravanos and M. X. Curinga, "Simulating the software development lifecycle: The Waterfall model," *Applied Systems Innovation*, vol. 6, p. 108, 2023, doi: 10.3390/asi6060108.

[8] S. Saini and A. Sharma, "Impact of data preprocessing on Prophet-LSTM hybrid models for time series forecasting," *Jurnal Kuey*, vol. 6, no. 1, pp. 32–41, 2024. [Online]. Available: https://kuey.net/index.php/kuey/article/download/7207/5358/14154

[9] S. Gordon, J. Crager, C. Howry, A. I. Barsdorf, J. Cohen, M. Crescioni, B. Dahya, P. Delong, C. Knaus, D. S. Reasner, S. Vallow, K. Zarzar, and S. Eremenco, "Best practice recommendations: User acceptance testing for systems designed to collect clinical outcome assessment data electronically," *Therapeutic Innovation & Regulatory Science*, vol. 56, pp. 442–453, 2022, doi: 10.1007/s43441-021-00363-z.

[10] B. Al-Asaad, R. J. Mahmod, and M. J. Hashim, "The comparative analysis of SARIMA, Facebook Prophet, and LSTM for public-health time-series forecasting," *Frontiers in Public Health*, 2022, doi: 10.3389/fpubh.2022.994098.

[11] H. Patel, B. K. Bolla, S. E., and D. Reddy, "Comparative study of predicting stock index using deep learning models," arXiv preprint arXiv:2306.13931, 2023. [Online]. Available: https://arxiv.org/abs/2306.13931

[12] P. Piotrowski, I. Rutyna, D. Baczyński, and M. Kopyt, "Evaluation metrics for wind power forecasts: A comprehensive review and statistical analysis of errors," *Energies*, vol. 15, p. 9657, 2023, doi: 10.3390/en15249657.

[13] V. Bhaskarreddypogu and D. Prasadu, "Demand forecasting in e-commerce fashion retail: A comparative study of generative AI, LSTM and ARIMA models," *Journal of Information Systems Engineering and Management*, vol. 18, no. S, pp. 1–12, 2024. [Online]. Available: https://www.jisem-journal.com/

[14] A. Kumar and P. Nagpal, "Deep learning based simulation for new product demand estimation," *Issues in Information Systems*, vol. 25, no. 4, pp. 236–251, 2024, doi: 10.48009/4_iis_2024_119.

[15] X. Qi, K. Hou, T. Liu, Z. Yu, S. Hu, and W. Ou, "From known to unknown: Knowledge-guided transformer for time-series sales forecasting in Alibaba," arXiv preprint arXiv:2109.08381, 2021. [Online]. Available: http://arxiv.org/abs/2109.08381

[16] M. Sukel, S. Rudinac, and M. Worring, "Multimodal temporal fusion transformers are good product demand forecasters," arXiv preprint arXiv:2307.02578, 2023. [Online]. Available: http://arxiv.org/abs/2307.02578

[17] D. Arnas, H. Ariessanti, D. Aryani, and I. Sutanto, "Sistem penjualan aplikasi perancangan pada sistem penjualan aplikasi e-commerce berbasis web menggunakan metode extreme programming," *ICIT Journal*, vol. 11, no. 1, pp. 77–92, 2025, doi: 10.33050/icit.v11i1.3454.

[18] M. Khusnah, R. Gernowo, and B. Surarso, "Implementasi e-commerce dengan sistem informasi rekomendasi menggunakan metode collaborative filtering untuk pengembangan penjualan pada UMKM," *Jurnal Sistem Informasi Bisnis*, vol. 15, no. 1, pp. 134–141, 2025, doi: 10.21456/vol15iss1pp134-141.

[19] A. Widyantoro, F. F. A. Bina, T. Prayoga, R. Safei, and M. A. Arrasid, "Systematic literature review: Membandingkan pendekatan metode Agile dan Waterfall dalam pengembangan perangkat lunak," *Journal of Comprehensive Science*, vol. 4, no. 1, 2025

[20] T. Rahman, A. Andrean, and M. Putra, "Design and testing of e-commerce website shoe sales with Waterfall method and User Acceptance Testing (UAT)," *Voteteknika*, vol. 13, no. 1, 2025, doi: 10.24036/voteteknika.v13i1.133273.