DOI: 10.58895/ksp/1000162754-13 erschienen in:
**Proceedings. 33. Workshop Computational Intelligence - Berlin, 23. – 24. November 2023**
DOI: 10.5445/KSP/1000162754 | https://www.ksp.kit.edu/site/books/e/10.5445/KSP/1000162754/

# Machine Learning Forecasting of Daily Delivery Positions: A Modern Take on Industrial Workforce Planning

Lukas Hans[1], Patrick Eichenseer[2], and Matthias Groß[1]

[1]Institute for Data Science, Engineering, and Analytics, TH Köln
Steinmüllerallee 1, 51643 Gummersbach, Germany
E-Mails: lukas.hans@th-koeln.de, matthias.gross2@th-koeln.de

[2]Head of Operational Excellence, DEHN SE
Umelsdorfer Str. 8, 92280 Utzenhofen, Germany
E-Mail: patrick.eichenseer@dehn.de

## 1 Introduction

The logistics industry plays a pivotal role in global trade, and efficient warehouse operations are essential for the seamless movement of goods. In recent years, prevailing job market conditions have presented significant difficulties in recruiting skilled workers in warehouse operations [1]. This shortage of skilled logistics workers has challenged companies to meet dynamic market demands and hindered effective workforce planning. The labor shortage results in increased operating costs and decreased overall efficiency due to suboptimal resource utilization. The key challenge for warehouse managers is the fluctuating and unpredictable nature of customer demand. Short-term customer orders, seasonal fluctuations, and rapidly changing market demands make it difficult for companies to forecast and plan their logistics workforce accurately. These dynamic demands often result in overstaffing during off-peak periods and understaffing during peak periods. Understaffing leads to unmet customer demand and, in some cases, customer churn. It also runs counter to a common warehouse goal of maximizing service levels (i.e., the promise of fast and accurate delivery) as a measure of differentiation from competitors. Overstaffing leads to underutilization and inefficiencies. This results in financial losses, as customer order fulfillment through picking at the point of delivery is the most costly activity in the warehouse [2, 3, 4].

To tackle the pressing issues in workforce scheduling, it is essential to develop accurate forecasting frameworks that can efficiently predict delivery positions. Accurate forecasting enables companies to optimize their employee staffing in logistics, reducing the risks of both overstaffing and understaffing within the limited workforce. Currently, the forecast process is heavily reliant on the personal expertise and judgment of individual team members. These individuals draw on their years of experience and intuition to estimate future demand, utilizing the number of pre-orders already recorded in their software system as a key input. This approach, although common in small and medium-sized warehouses, is inherently subjective and susceptible to human error and bias.

In this context, integrating Machine Learning (ML) methods presents a promising solution to enhance workforce scheduling efficiency [5, 6]. ML algorithms offer the capability to process vast amounts of data, identify complex patterns, and make data-driven predictions. By reframing the workforce optimization problem as a forecasting challenge, ML models can be leveraged to provide accurate and reliable one-day-ahead delivery position predictions. This approach not only improves transparency and explainability in the estimation process but also enhances the overall scheduling efficiency.

## 2 Problem Statement and Data Description

In the complex domain of workforce planning within logistics, we aim to develop an accurate mathematical representation of the challenge. Instead of directly tackling workforce scheduling, we've transformed it into a prediction problem, using various ML models to forecast the next day's delivery positions. The problem is defined mathematically as minimizing the root-mean-square error between the actual and predicted number of delivery positions using the optimal model and its parameters.

$$i^* = \text{argmin}_i \quad \sqrt{\frac{1}{n}\sum_{t=1}^{n}(D_t^{act} - D_t^{pred,i})^2} \tag{1}$$

With

- $D_t^{act}$: The actual number of delivery positions for day $t$.
- $D_t^{pred,i}$: The predicted number of delivery positions for day $t$, using the $i$-th ML model.

Simultaneously, we aim for the predicted delivery positions, $D_t^{pred,i^*}$, from the best model to closely match the needed number of workers, $W_t$. The relationship between these is determined by past company data and workforce planning rules. This relationship is given by

$$W_t = h(D_t^{pred,i^*}) \tag{2}$$

where $h(\cdot)$ represents the historical link between delivery positions and required workforce. Nonetheless, our focus is on pinpointing the most effective ML technique, rather than deducing the exact form of $h(\cdot)$. The study relies on delivery position data from a collaboration between the Technical University of Cologne and a major electrical engineering firm, with adjustments made using a constant factor X to safeguard financial confidentiality. Table 1 provides a comprehensive statistical overview of the initial time series data.

The actual data entries amount to 1337 data points, given the non-operational days like weekends and winter breaks. Following an 80/20 split for partitioning, the training set contains 1069 entries, and the test set has 268. The initial data inspection highlighted significant downward fluctuations, leading to the removal of entries identified as outliers and those with fewer than 200 delivery positions. This refined dataset, now with 1321 entries, presents a more consistent distribution conducive to analysis. Cleaning paved the way for feature engineering, introducing variables like time details, lag intervals, and multiple rolling means to serve as inputs for the ML models.

Table 1: Feature summary

| | Start Date | End Date | Time Span | |
|---|---|---|---|---|
| Lagerbewegung Zeitpunkt | 2018-01-02 | 2023-05-08 | 1952 days | |
| | Mean | Std. | Min | Max |
| Lagerbewegung Pick ERP-Auftrag | 2860.26 | 697.91 | 1 | 4525 |

# 3 Methods & Metrics

For forecasting, five primary models were utilized: Random Forest, XGBoost, LightGBM, Support Vector Regression (SVR), and Convolutional Neural Networks (CNN). Random Forest is an ensemble method known for its robustness and ability to manage non-linear relationships [7]. XGBoost and LightGBM are both gradient boosting frameworks, with the former being praised for its speed and flexibility [8], and the latter for efficiency and leaf-wise tree growth [9]. SVR excels in high-dimensional spaces and offers kernel function flexibility [10]. CNNs, while dominant in image classification, have shown prowess in time-series forecasting, capturing both local and global temporal dependencies [11].

Beyond the primary models, two ensemble strategies, Stacking and Averaging, were evaluated. Averaging involves computing the mean of individual model predictions, noted for its robustness [12]. Stacking harnesses multiple base models, in our case leveraging ridge regression as the meta-model [13].

Hyperparameter tuning for all models was automated using the Optuna package, streamlining optimal value discovery [14].

Model performance was gauged using Root Mean Square Error (RMSE) and Mean Absolute Error (MAE). While RMSE emphasizes large errors, offering a penalty, MAE offers a balanced view on error magnitude [15, 16]. Both metrics help in a holistic evaluation of model accuracy.

Table 2: Performance metric of the different modeling approaches

| Algorithm | MAE | RMSE |
|---|---|---|
| LightGBM | 335.56 | 440.63 |
| CNN | 382.83 | 518.00 |
| SVR | 356.51 | 470.66 |
| XGBoost | 331.86 | 437.15 |
| Random Forest | 323.78 | 429.55 |
| Stacking Ensemble with CNN | 683.47 | 796.16 |
| Average Ensemble with CNN | 318.91 | 416.48 |
| Stacking Ensemble without CNN | 365.08 | 476.35 |
| Average Ensemble without CNN | 313.98 | 413.96 |

# 4 Results

In evaluating the base models, tree-based algorithms - LightGBM, XGBoost, and Random Forest - showed close performance, with RMSEs between 429 to 441 and MAEs ranging from 323 to 336. The SVR demonstrated a conservative forecast, with MAE and RMSE values at 356.51 and 470.66, respectively. Conversely, the CNN trailed in performance, especially post-Christmas, resulting in an MAE of 382.83 and an RMSE of 518.

The ensemble strategies exhibited contrasting results. The stacking ensemble, despite capturing the time series' structure, tended to overestimate with MAEs and RMSEs at 683.47 and 796.16. Contrarily, the average ensemble outperformed all models, achieving an MAE of 318.91 and an RMSE of 416.48.

Considering the subpar performance of the CNN, ensembles were recomputed without CNN inputs. This led to an overall improvement. The average ensemble showed an MAE of 313.98 and an RMSE of 413.96, while the stacking ensemble registered 365.08 and 476.35 for MAE and RMSE, respectively. Even though the average ensemble's metrics were superior, the stacked method better captured individual peaks.

The performance metrics for all the discussed modeling approaches are summarized below:

## 5 Discussion & Outlook

Besides confirming a fundamental predictability in the used dataset, this work also showed that tree-based models like Random Forests, XGBoost and LightGBM are suitable algorithms for this task. Beyond that, stacking and averaging ensemble methods using these models proved to further increase the forecasting effectiveness.

While the SVM approach performed slightly worse but in the same approximate range as the tree-based models, CNNs clearly showed inferior results, especially when exposed to irregularities like the post-christmas dip in the used dataset. The small size of the dataset can be assumed as a possible reason for this lack in performance. The neural network might not have been exposed to a sufficient amount of data to pick up the more complex patterns. As the dataset grows over time, we could expect an improvement in CNNs' performance.

Going forward with these approaches, a possible next step is the incorporation of external variables that might influence the number of delivery positions. These variables can range from calendrical data to pre-order information and even weather data.

Furthermore, different model architectures should be examined, as this work mainly focused on tree-based models. Diversifying the model architectures could boost the forecasting effectiveness even further.

In summary, this research offers a promising start toward optimizing workforce scheduling in small and medium-sized warehouses in the logistics sector by leveraging ML methods.

## Acknowledgments

The work was partly funded by the German Federal Ministry for Economic Affairs and Climate Action (BMWK) as part of Project "IMProvT II" (FKZ: 03EN2086 A-E).

## References


[1] “Skilled labour shortage: Freight, logistics sector needs 18,000 workers for jobs,” June 2023. [Online; accessed 11. Sep. 2023].

[2] T. van Gils, K. Ramaekers, A. Caris, and R. B. M. de Koster, “Designing efficient order picking systems by combining planning problems: State-of-the-art classification and review,” *Eur. J. Oper. Res.*, vol. 267, pp. 1–15, May 2018.

[3] Y.-C. Ho and J.-W. Lin, “Improving order-picking performance by converting a sequential zone-picking line into a zone-picking network,” *Comput. Ind. Eng.*, vol. 113, pp. 241–255, Nov. 2017.

[4] S. Vanheusden, T. van Gils, A. Caris, K. Ramaekers, and K. Braekers, “Operational workload balancing in manual order picking,” *Comput. Ind. Eng.*, vol. 141, p. 106269, Mar. 2020.

[5] M. Witthaut and C. Culotta, “Machine Learning in der Logistik – Eine empirische Studie über die Anwendung in deutschen Unternehmen,” *Logistics Journal : nicht referierte Veröffentlichungen*, vol. 2021, June 2021.

[6] M. Akbari and T. N. A. Do, “A systematic review of machine learning in logistics and supply chain management: current trends and future directions,” *Benchmarking: An International Journal*, vol. 28, pp. 2977–3005, Mar. 2021.

[7] L. Breiman, “Random forests,” *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[8] T. Chen and C. Guestrin, “Xgboost: A scalable tree boosting system,” in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pp. 785–794, 2016.

[9] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, “Lightgbm: A highly efficient gradient boosting decision tree,” *Advances in neural information processing systems*, vol. 30, pp. 3146–3154, 2017.

[10] H. Drucker, C. J. C. Burges, L. Kaufman, A. Smola, and V. Vapnik, "Support vector regression machines," in *Advances in Neural Information Processing Systems* (M. Mozer, M. Jordan, and T. Petsche, eds.), vol. 9, MIT Press, 1996.

[11] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[12] T. G. Dietterich, "Ensemble methods in machine learning," in *International workshop on multiple classifier systems*, pp. 1–15, Springer, 2000.

[13] D. H. Wolpert, "Stacked generalization," *Neural Networks*, vol. 5, no. 2, pp. 241–259, 1992.

[14] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 2623–2631, 2019.

[15] R. J. Hyndman and G. Athanasopoulos, *Forecasting: principles and practice*. OTexts, 2018.

[16] T. Chai and R. R. Draxler, "Root mean square error (rmse) or mean absolute error (mae)?–arguments against avoiding rmse in the literature," *Geoscientific Model Development*, vol. 7, no. 3, pp. 1247–1250, 2014.