[Makara Journal of Technology](https://scholarhub.ui.ac.id/mjt)

Volume 25 | Number 2 Article 7

8-2-2021

Predictive Delivery Man Assignment Problem using Deep Learning

Rahmadini Payla Juarsa
Agro-industrial Engineering Study Program, Institut Pertanian Bogor, Bogor 16680, Indonesia, rahmadini_pj@apps.ipb.ac.id

Taufik Djatna
Agro-industrial Engineering Study Program, Institut Pertanian Bogor, Bogor 16680, Indonesia

Recommended Citation

Juarsa, Rahmadini Payla and Djatna, Taufik (2021) "Predictive Delivery Man Assignment Problem using Deep Learning," *Makara Journal of Technology*: Vol. 25 : No. 2 , Article 7.
DOI: 10.7454/mst.v25i2.3700
Available at: https://scholarhub.ui.ac.id/mjt/vol25/iss2/7

Makara J. Technol. 25/2 (2021), 97‒104
doi: 10.7454/mst.v25i2.3700

Predictive Delivery Man Assignment Problem using Deep Learning

Rahmadini Payla Juarsa* and Taufik Djatna

Agro-industrial Engineering Study Program, Institut Pertanian Bogor, Bogor 16680, Indonesia

**E-mail: rahmadini_pj@apps.ipb.ac.id*

Abstract

Dispatching is a critical part in current online shopping. It relates to how the delivery man assignment should minimize cost along with the service from a source to an end customer with an appropriate scheduled time. The problem arises as neither enough products to deliver nor delivery men are available for dispatch, resulting in suboptimal service and a waste of money. The study aimed to formulate the cost of restaurant dispatching for inducing a deep learning-based solution with the gated recurrent unit recurrent neural network to receive hourly order data and to engage the result for near feature delivery man schedule with minimum cost. The result showed that cost formulation minimized the number of delivery men times the wage per hour with the constraints of each delivery man carrying a maximum of five orders in one way and 11 work hours/day. The deep learning input model used 1078 historical data which were filtered using the Savitzky-Golay method. The root mean square errors of training and testing were 2.35 and 2.41, respectively. Moreover, the number of delivery men every hour was found in a range from one to four people. Furthermore, the deep learning approach saved costs of up to 43.8%.

Abstrak

Prediksi Masalah Penugasan Kurir dengan Pembelajaran Mendalam. Pengiriman merupakan salah satu hal yang penting dalam model belanja online. Hal itu berhubungan dengan bagaimana penugasan kurir untuk mengantar pesanan hingga ke konsumen dengan cepat dan biaya yang ditimbulkan sesedikit mungkin. Permasalahan yang terjadi adalah jika ada ketidakseimbangan antara jumlah kurir dan jumlah pesanan yang harus diantarkan sehingga menyebabkan pelayanan tidak optimal dan berakhir pemborosan. Tujuan dari penelitian ini adalah memformulasi model biaya pengiriman yang terdapat di restoran, menyusun solusi *deep learning gated recurrent Unit* (GRU) r*ecurrent neural network* (RNN) untuk mendapatkan data pesanan setiap jam, dan menggunakan hasil yang didapat untuk menyusun jadwal penugasan kurir yang menghasilkan biaya minimum. Hasil yang didapat adalah formulasi dari biaya penugasan adalah meminimumkan jumlah kurir dikali dengan upah tiap jamnya, dengan batasan masing-masing kurir maksimum membawa 5 pesanan dalam sekali jalan dan 11 jam kerja/hari. Input dalam model deep learning adalah 1078 data historis pesanan online yang sudah disaring menggunakan metode Savitzky-Golay. RMSE dari data pelatihan dan data percobaan masing-masing 2.35 dan 2.41. Jumlah kurir yang didapat dari metode ini adalah 1-4 orang dari yang sebelumnya 3-4 orang. Metode ini mampu menghemat biaya hingga 43.8%.

Keywords: assignment problem, delivery order, gated recurrent unit, predictive deep learning

1. Introduction

The development of various restaurant industries now requires them to innovate to become more enduring and enticing to costumers than before [1]. One of the most desirable innovations is providing delivery order service to easily reach many customers. Delivery order service is offered online. An order is sent to a customer's address by a delivery man.

The ordering time until the product reaches the customer is a critical part that guarantees customer satisfaction. If the delivery takes too long, the customer may not order again [2]. Thus, companies must have a sufficient number of delivery men, so that orders can be delivered as quickly as possible.

Job assignment problem (JAP) refers to the assignment of *n* delivery man to *n* other task items to obtain the maximum profit [3]. The main task is the delivery of several orders every hour with the constraints of each delivery man only able to carry a maximum of five orders. The most common way to solve JAP is to use matrix cost with the Hungarian method to find a suitable combination of workers and their jobs. However, this method can only keep a fixed number of workers and jobs.

Moreover, the problem that arises is that the number of online orders is not always constant within the operating hours of companies. By contrast, the number of employed delivery men is always constant, which can undoubtedly cause problems in dispatching and make delivery men idle and overwhelmed at certain hours. As a result, losses in terms of costs, time, and orders that cannot be delivered to customers occur. Companies make every effort to minimize the costs incurred due to hiring delivery men. The number of workers or delivery men changes unlike in normal JAP.

The most commonly used forecasting method is autoregressive integrated moving average (ARIMA). This method can only use numerical and categorical data and cannot handle fluctuating and complex data patterns. ARIMA still has many errors in predicting actual data patterns [4]. Other researchers use artificial neural network (ANN) to forecast time series [5][6]. Although the performance of ANN is better than that of ARIMA, ANN still exhibits low performance in certain cases. The reason is that ANN only has forward and backward approaches, and it is not always suitable with fluctuating time-series data [7]. Other methods can be used to overcome the shortcomings of this method.

Given that delivery men are dynamic, and data patterns are numerical, categorical, and fluctuating, this study applies the deep learning method in the form of the gated recurrent unit (GRU) recurrent neural network (RNN) to predict delivery man assignments. Doing so makes a balance between the products delivered and the number of employed delivery men to achieve the minimum cost. This method is selected because it can learn how to use online order historical data that determine the order pattern and prediction. Another reason is because the learning method is beneficial for handling large and varied data [8]. In addition, this method can learn, memorize, and create relationships among data [9].

The first objective of the study is to formulate the cost of restaurant dispatching. The second is to induce an in-depth learning approach with the GRU RNN to receive hourly order predictive data. The last objective is to engage the results for near feature delivery man with minimum cost and balance the numbers of orders and employed delivery men. The data are collected in a restaurant in Bogor, Indonesia named Ayam Geprek Pedjoang.

The paper is organized in the following manner. Section 2 provides a review of related studies on predictive time series using RNN. Section 3 describes the methodology for this study. Section 4 discusses the results, strengths, and weaknesses of this work. Finally, Section 5 provides the conclusion and suggestions for further research.

2. Related Works

Zhang and Peng [10] investigated JAP by using several fixed jobs and workers. Although a fuzzy approach was applied, the problem of changing jobs and workers could not be solved. The same thing occurred in the research of [10] who applied the fuzzy assignment problem with the ranking method.

Several research uses deep learning to assess predictions. Certain researchers apply RNN to predict voltage instability [11], early-stage malware with an accuracy of up to 94% [12], air quality [13], and network traffic prediction [14]. Kaneko and Yada [15] applied deep learning to predict sales from retail stores with an accuracy of up to 86%. The number of product inputs in his work was more than 10,000 items, indicating the deep learning capabilities that must be applied for diverse inputs.

In general, the RNN mechanisms that are often used are GRU and long short-term memory (LSTM). Cho *et al*. [16] proposed GRU, whereas Hochreiter and Schmidhuber proposed LSTM. The advantages of GRU are it can analyze the time scale by maintaining gradients during the training process [8]. In addition, GRU is more profitable because its computation is simpler than ANN, suggesting that it adequately processes data fast [13]. GRU can also effectively analyze data in certain cases, such as in image classification [9] and traffic flow prediction [17].

Other studies also apply the Savitzky-Golay method to filter input data. This method can filter noisy data due to fluctuations without removing important information from such data. The principle is based on fitting data with least-squares procedures [18]. Some are on infrared gas detection [19], fruit quality evaluation [20], and seismic random noise reduction [21].

3. Methods

This work was done in three stages. First, the cost of solving the JAP of delivery order was formulated with an input which was conditioned from the delivery order mechanism. Second, the GRU RNN approach was compiled by inputting online order historical data every hour. The expected result of this step was the prediction pattern of the number of orders per hour used to determine delivery man schedules. Last, delivery man schedules were arranged. The cost reduction that balanced delivery man availability and the total order were also calculated. This stage is illustrated in Figure 1.

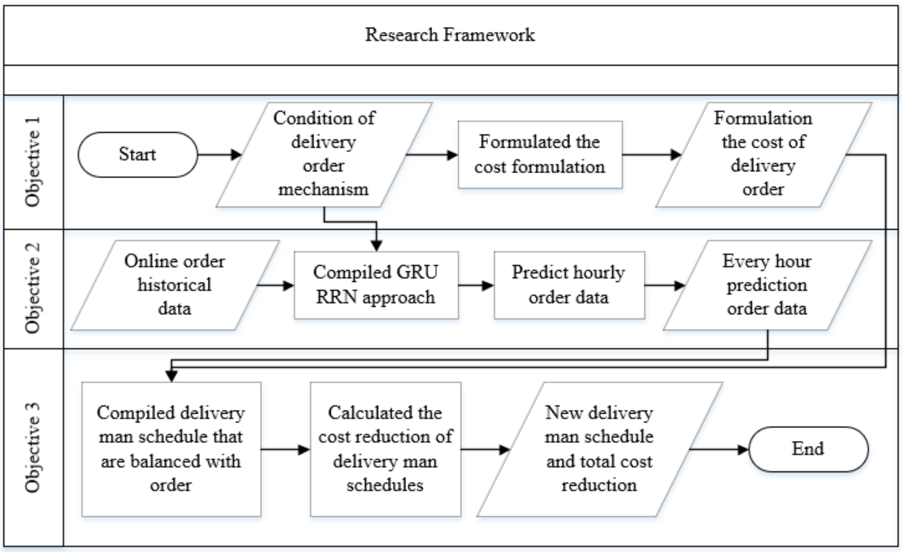

Figure 1. Research Framework

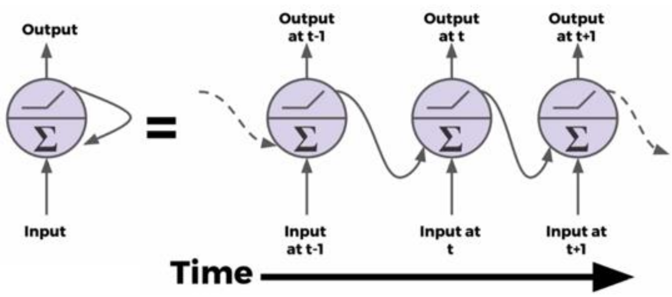

Figure 2. RNN Architecture

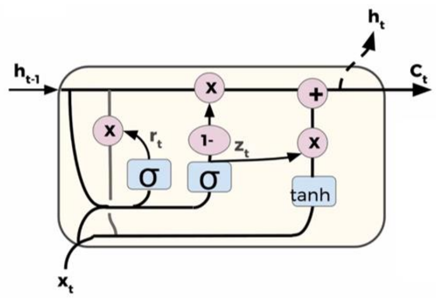

Figure 3. GRU Architecture

Cost formulation. This stage required the input data conditions of the delivery order mechanism. These conditions were used to formulate cost. This stage was essential to find existing constraints in the system.

GRU RNN approach compilation. The results of the formulation stage were used to compile the in-depth learning approach. This stage required the additional input of online order historical data in each hour. The data were filtered using the Savaitzky-Golay method to reduce errors arising from data fluctuations [20]-[23]. The approach was then normalized because doing so accelerates the training process [22].

The general RNN architecture is illustrated in Figure 2. In this study, RNN simultaneously described time sequence and dependence on various scales. The input sequence in an RNN was processed one element at a time instead of processing it in a hidden unit that contains information on all the previous elements of the sequence [23]. RNN had more than one cycle. Thus, following the pattern from the end unit to itself was possible. RNN contained hidden unit **h**, which ensured a deep learning mechanism and output **y**. This unit operated on a variable-length sequence $x = (x_1, x_2, \ldots, x_t)$.

At each time step t, the hidden state $\mathbf{h}_{<t>}$ of RNN was updated using Eq. (1).

$$\mathbf{h}_{<t>} = f\ (\mathbf{h}_{<t-1>}, x_t) \tag{1}$$

where f was a non-linear activation function. F f had to be simple as sigmoid or as complex as GRU.

The hidden layers contained a GRU mechanism for memorizing and updating data. A typical RNN cell is shown in Figure 3. In the GRU, the cell contained two gates: reset gate **r** and update gate **z** [20]. In the GRU cell, the process was computed using Eqs. (2)–(5):

$$\mathbf{r}_t = \sigma(\mathbf{W}_r \,.\, [\mathbf{h}_{t-1}, x_t] + \mathbf{b}_r) \tag{2}$$

$$\mathbf{z}_t = \sigma(\mathbf{W}_z \,.\, [\mathbf{h}_{t-1}, x_t] + \mathbf{b}_z) \tag{3}$$

$$\mathbf{g}_t = f(\mathbf{W}_g \,.\, [\mathbf{h}_{t-1} * \mathbf{r}_t, x_t] + \mathbf{b}_g) \tag{4}$$

$$\mathbf{h}_t = (1 - \mathbf{z}_t) * \mathbf{h}_{t-1} + \mathbf{z}_t * \mathbf{g}_t \tag{5}$$

with,
W = weight matrices, **g** = candidate state,
b = bias, **C** = cell state, **h** = hidden state,
r = reset gate, **z** = update gate

The model was evaluated using the mean square error (MSE) for history loss during training and validating and root mean square error (RMSE) for evaluating its performance. A small RMSE was preferred [24]. RMSE evaluated the variance of error between the predicted value and the actual value. MSE and RMSE were calculated using Eqs. (6) and (7), respectively.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - y_{di})^2 \qquad (6)$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - y_{di})^2} \qquad (7)$$

where n is the number of points, y_i is the predicted value, and y_{di} is the actual value.

Cost reduction and delivery man schedule calculation. The formulation produced in the first stage and the prediction results from the second stage were used to prepare delivery man schedules. Subsequently, the newly generated costs were calculated to make new delivery man schedules. The results were compared with the costs generated before using the deep learning method.

4. Result and Discussion

Cost formulation. Cost formulation was performed using the data on delivery order condition.. The case studied in this work was a delivery order at Ayam Geprek Pedjoang restaurant with main products derived from chicken. Data were obtained by conducting interviews with the restaurant owners. The results obtained are as follows:

- Three delivery men work during weekdays, whereas four people work during weekends. The reason for the slight difference is because more orders are delivered on weekends than on weekdays.
- Orders are delivered from 10:00 a.m. to 9:00 p.m., except holidays, such as Islamic holy days.
- Each delivery man can deliver a maximum of five products derived from chicken (category 1). Table 1 presents the grouping categories.

The assumption in this case was one delivery man receives a fee of IDR 10.000/hour. Predictions were also used for category 1 because products in this category are the most ordered online. The constructed formulations followed Eq. (8), and the JAP matrix cost is shown in Table 2.

$$\min \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_{ij} , \qquad (8)$$

where

$x_{ij} = 1$ when ith worker assigned to jth job

$x_{ij} = 0$ when ith worker is not assigned to jth job

$(i = 1, 2, .., n)$

$(j = 1, 2, .., 11)$

maximum number of goods delivered by each delivery man ≤ 5

weekend orders > weekday orders

c = cost of assignment i to j

x_{ij} = worker i assigned to job j

i = number of workers

j = number of jobs (maximum of 11 jobs because of operation hours)

Table 1. Product categories

Category		
1	2	3
1. Original geprek chicken 2. Rica-rica geprek chicken 3. Matah chili geprek chicken 4. Lombok chili geprek chicken 5. Salt chili geprek chicken 6. Roasted geprek chicken 7. Padang sauce chicken 8. Black pepper chicken 9. Roasted chicken 10. Rica roasted chicken 11. Rice 12. Savings package 1 13. Savings package 2 14. Super saver package 1 15. Super saver package 2	1. Iced tea 2. Bargain tea 3. Lemon tea 4. Milk tea 5. Thai tea 6. Mineral water 7. Cold mineral water 8. Orange juice	1. Kangkung vegetable 2. Tofu 3. Tempe 4. Fried cabbage 5. Crispy mushrooms

Table 2. JAP Matrix Cost

Worker/Job	Job				
	Mon 10.00–11.00	Mon 11.00–12.00	.	.	Sun 20.00–21.00
Delivery man I	10	10	.	.	10
Delivery man II	10	10	.	.	10
Delivery man III	10	10	.	.	10
Delivery man IV	0	0	.	.	10

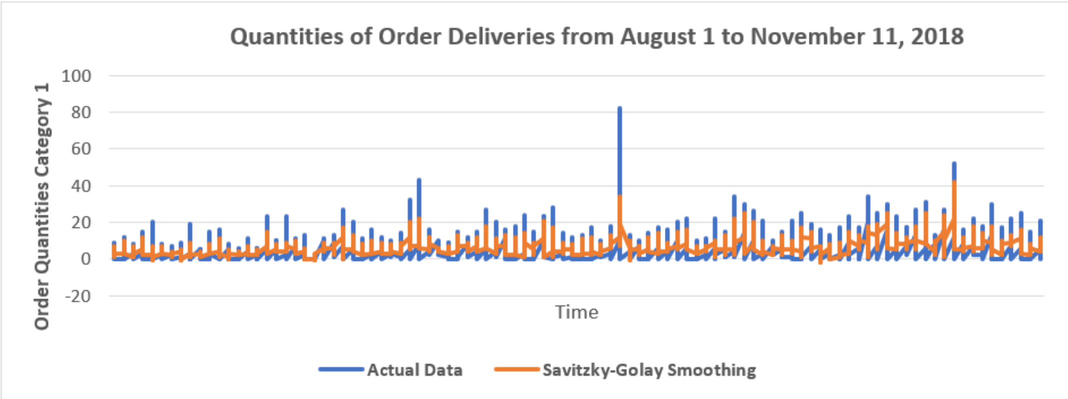

Figure 4. Comparison Between Actual and Smoothing Data

GRU RRN approach. This step predicted the model of the number of orders within a week. The model was then used to determine delivery man schedules. The limit used was (7). Online order historical data were required. Such data were filtered using the Savitzky-Golay method with 15 window lengths and 7 degrees of a polynomial in python. A comparison of actual and filtered data is illustrated in Figure 4, which shows 35% of data smoothing that represents actual data.

RNN algorithm involves certain hyperparameters, which are set by researchers themselves for constructing models [25]. Table 3 shows the model hyperparameter summary. The model was made using a Jupyter notebook with the python programming language. To help build the model, API Keras and Tensorflow were used in this research.

RNN architecture was then built. The model had 11 nodes in the input layer from look_back, which indicates the time window for how much previous data should be used to predict the next value. These 11 nodes represent 11 data points in one day. Batch_size indicates how much data are processed for each batch. Delay is the target data that must be predicted, in this case, one hour ahead. Adam optimizer was also applied. Loss_function is the value to ensure that the training mechanism works well. Epoch refers to how many times the training is run. The model had two GRU hidden layers. Each layer had 300 nodes. The output layer had one node. The model architecture is illustrated in Figure 5. Rectified linear unit (ReLu) activation function was applied on the hidden layer and sigmoid on the output layer. The data range that resulted from the normalization was [0, 1], suggesting that ReLu and sigmoid were relatively suitable [26].

Table 3. Model Hyperparameter Summary

No	Hyperparameter	Value
1	Look_back	11
2	Batch_size	11
3	Delay	1
4	Optimizer	Adam
5	Loss_function	MAE
6	Epoch	10
7	Hidden layer activation function	ReLu
8	Output layer activation function	sigmoid

The online order historical data comprised 1,078 points (14 weeks of observation), which were then divided into three parts. The data consisted of 78.6% (847 points) for training, 14.3% (154 points) for validating, and 7.1% (77 points) for testing. Table 4 presents the results after 10 times of running the epoch or iteration for the training model. Figure 6 shows the history loss after 10 epochs. It depicts that the line trends were declining. In addition, the gap between the two lines was not too far. This model was neither overfitting nor under-fitting [27].

The model trained to make predictions was used and ran to determine the RMSE value to perceive its performance. The RMSE results for each training dataset (train + validation) and testing dataset indicated 2.35 and 2.41 average errors, respectively. Figure 7 displays the performance plot of the model. Table 5 and Figure 8 show the prediction results for the testing dataset every hour.

Cost reduction. The formulation results from the first step and the prediction results from the second step were used to determine the cost of the delivery order system. The data were combined to determine delivery man schedules. The results of the new JAP matrix and cost reduction are explained in Table 6 and 7, respectively.

Model assumptions and constraints were considered in all calculations. The results revealed that the old system employed three to four delivery men, and such hiring cost IDR 2,530,000/week, whereas the new system employed one to four delivery men and hiring cost IDR 1,108,140/week. In this schema, the cost reduction reached 43.8%.

Strengths and Weaknesses. The advantage of this method is it learns or obtains insights from the customer order data pattern. The model also includes numerical and categorical data. It is relatively more accurate than using the previous system that does not use data. In addition, this method quickly determines delivery man schedules and can adapt to possible future changes in customer order patterns. The disadvantage of this method is large data are required for its improvement.

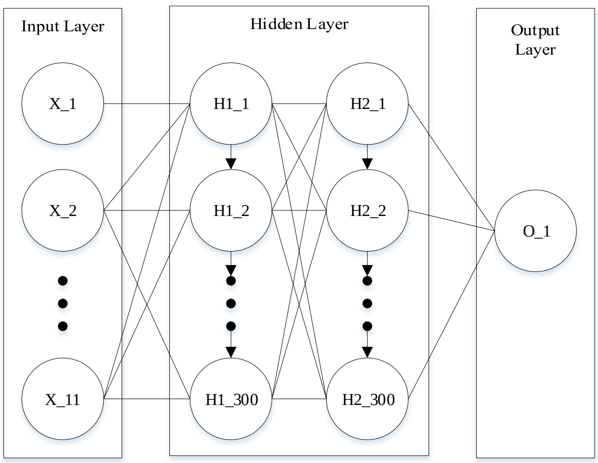

Figure 5. Model Architecture

Table 4. History Loss from 10 Epochs

Parameter	Category	
	Train	Validation
MSE	0.0031	0.0094
Accuracy	0.0024	0.0076

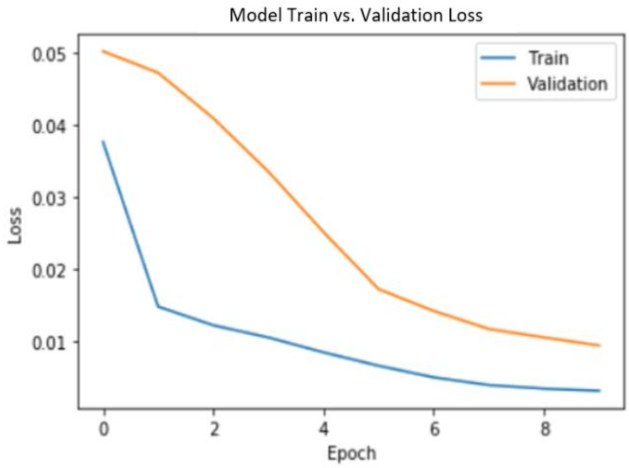

Figure 6. Error History Loss of the Training Set

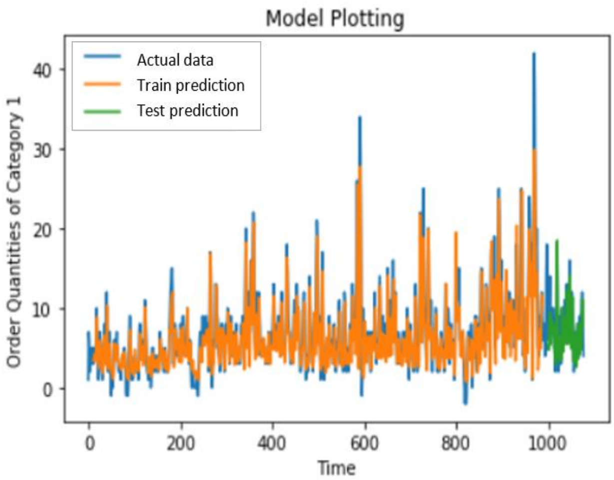

Figure 7. Performance Plot of the Model

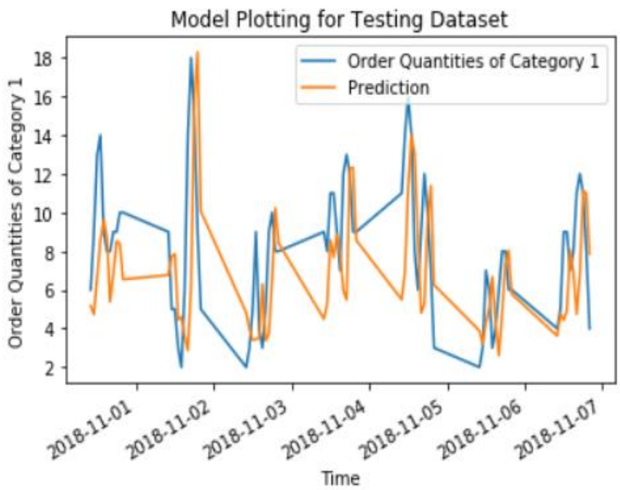

Figure 8. Prediction Plot for the Testing Dataset

Table 5. Model Prediction on the Testing Dataset

No	Time	Smoothing Data	Prediction
1	2018-11-05 10:00:00	6	5
2	2018-11-05 11:00:00	9	5
3	2018-11-05 12:00:00	13	7
.	.	.	.
.	.	.	.
75	2018-11-11 18:00:00	11	11
76	2018-11-11 19:00:00	8	11
77	2018-11-11 20:00:00	4	8

Table 6. New JAP Matrix

Worker/Job	Job				
	Mon 10.00–11.00	Mon 11.00–12.00	.	.	Sun 20.00–21.00
Delivery man I	10	10	.	.	10
Delivery man II	10	0	.	.	10
Delivery man III	0	0	.	.	0
Delivery man IV	0	0			0
Delivery man V	0	0	.	.	0

Table 7. Cost Reduction Summary

No.	Day	Work Hour	Order Prediction	Delivery Man Availability	New Delivery Man
1	Mon	10.00–11.00	5	3	2
2	Mon	11.00–12.00	5	3	1
3	Mon	12.00–13.00	7	3	2
4	Mon	13.00–14.00	9	3	2
5	Mon	14.00–15.00	10	3	2
6	Mon	15.00–16.00	9	3	2
7	Mon	16.00–17.00	5	3	2
8	Mon	17.00–18.00	7	3	2
9	Mon	18.00–19.00	8	3	2
10	Mon	19.00–20.00	8	3	2
11	Mon	20.00–21.00	7	3	2
12	Tue	10.00–11.00	7	3	2
13	Tue	11.00–12.00	8	3	2
14	Tue	12.00–13.00	8	3	2
15	Tue	13.00–14.00	4	3	1
16	Tue	14.00–15.00	5	3	1
17	Tue	15.00–16.00	4	3	1
18	Tue	16.00–17.00	3	3	1
19	Tue	17.00–18.00	6	3	2
20	Tue	18.00–19.00	16	3	4
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.
70	Sun	13.00–14.00	5	4	1
71	Sun	14.00–15.00	8	4	2
72	Sun	15.00–16.00	7	4	2
73	Sun	16.00–17.00	5	4	1
74	Sun	17.00–18.00	7	4	2
75	Sun	18.00–19.00	11	4	3
76	Sun	19.00–20.00	11	4	3
77	Sun	20.00–21.00	8	4	2

5. Conclusion and Recommendation

A deep learning approach was successfully applied to balance delivery man assignments in a restaurant delivery order system that aims to minimize dispatching cost. Formulating a delivery order cost model minimizes cost, which was a function of the number of delivery man times the hourly fee, with constraints on working hours and maximum orders delivered by each delivery man. This formulation was performed to compile the GRU RNN deep learning model that produces a model that can determine order prediction each hour with the RMSE training dataset value of 2.35 and testing dataset value of 2.41. The formulation and prediction results were then used to calculate the cost reduction that can be achieved by applying the new delivery man schedules. The number of delivery men every hour was found in a range from one to four people in the new system, whereas in the old system were from three to four-person. The results showed that with a deep learning approach, the delivery order system minimized costs up to 43.8% of the previous system. The recommendation for future research can add factors and other relevant data to improve the model.

Acknowledgments

The authors want to thank Ayam Geprek Pedjoeang for their cooperation in this research.

References

[1] R. Aryanto, A. Fontana, A.Z. Afiff, Strat. Hum. Res. Man. Procedia Soc. Behav. Sci. 211 (2015) 874.
[2] A. Pizam, V. Shapoval, T. Ellis, Int. J. Contemporary Hosp. Man. 28 (2016) 2.
[3] R. Burkard, M. Dell'Amico, S. Martello, Introduction in Assignment Problem, Siam, Philadelphia, 2009, p.1.
[4] G. Werner, S. Yang, K. McConky, Proceedings of the 12th Annual Conference on Cyber and Information Security Research, Oak Ridge, 2017, p.4.
[5] A.A. Oluyinka, A.C. Korede, J. Appl. Math. (2014) 1.
[6] C.W.M. Noor, R. Mamat, G. Najafi, W.B.W. Nik, M. Fadhil, IOP Conf. Ser. Mater. Sci. Eng. 100 (2015) 1.
[7] T. Kim, T. Yoon, Int. J. Mach. Learn. Compu. 5/6 (2015) 471.
[8] L. Mou, P. Ghamisi, X. Zhu, IEEE Tran. Geosci. Rem. Sensing. 5 (2017) 3639.
[9] J. Liu, C. Wu, J. Wang, Inf. Sci. 423 (2018) 50.
[10] B. Zhang, J. Peng, Appl. Math. Model. 37 (2013) 6458.
[11] A.N. Gani, V.N. Mohamed. Intern. J. Fuzzy Math. Arch. 2 (2013) 8.
[12] A.M. Ibrahim, N.H. El-Amary. J. Elect. Syst. Info. Technol. 5 (2018) 216.
[13] M. Rhode, P. Burnap, K. Jones. Comp. Sc. 77 (2018) 578.
[14] V. Athira, P. Geetha, R. Vinayakumar, K.P. Soman. Proc. Comp. Sci. 132 (2018) 1394.
[15] R. Vinayakumar, K.P. Soman, P. Poornachandran. International Conferences on Advances in Computing, Communications and Informatics, Karnataka, 2017,.p.2353.
[16] Y. Kaneko, K. Yada. International Conference on Data Mining Workshops, Barcelona, 2016, p.531.
[17] K. Cho, Bv. Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, Learning phrase representations using RNN encoder-decoder for statistical machine translation, arXiv:1406.1078v3 [cs.CL], 2014.
[18] S. Hochreiter, J. Schmidhuber. Neur. Comput. 9 (1997) 1735.
[19] R. Fu, Z. Zhang, L. Li. 31st Youth Academic Annual Conference of Chinese Association of Automation, Wuhan, 2016, p.324.
[20] A. Savitzky, M.J. Golay. Anal. Chem. 36 (1964) 1627.
[21] J. Li, H. Deng, P. Li, B. Yu. Appl. Phys. B. 120 (2015) 207.
[22] H. Wang, J. Peng, C. Xie, Y. Bao, Y. He. Sens. 15/5 (2015) 11889.
[23] Y. Liu, B. Dang, Y. Li, H. Lin, H. Ma. 2016. Acta Geophys. 64 (2016) 101.
[24] T. Salimans, D.P. Kingma. 30th Conference on Neural Information Processing Systems, Barcelona, 2016, p.1.
[25] Y. LeChun, Y. Bengio, G. Hinton. Nat. 521 (2015) 436.
[26] N. Kamairuddin, S.S.A. Gani, H.R.F. Masoumi, M. Basri, P. Hashim, N.M. Mokhtar, M.E. Lane. RSC Adv. 5 (2015) 68632.
[27] J. Zbontar, Y. LeChun, Computing the stereo matching with a convolutional neural network. Computer Vision Foundation, arXiv:1409.4326v2 [cs.CV], 2015.
[28] A. Krizhevsky, I. Sutskever, G.E. Hinton. 2012. Proceeding Advance Neural Information Processing Systems, Nevada, 2012, p.1097.
[29] Y. Gu, B.K. Wyle, S.P. Boyte, J. Picotte, D.M. Howard, K. Smith, K.J. Nelson, Remote Sensing. 8 (2016) 1.