AI-DRIVEN RPA FOR BACK-OFFICE MANAGEMENT OF WORK CAPACITY

RPA BAZAT PE IA PENTRU GESTIONAREA BACK-OFFICE A CAPACITATII DE LUCRU

Radu Florin NEGOITA[1], THEODOR BORANGIU[2]

Abstract: *The study covered in this paper focuses on automating back-office processes for managing capacity in hospitality industry. This procedure, which is a component of the hospitality industry's back-office service management, is carried out using the robotic process automation (RPA) technology and integrated with back-office workflows in the operations management software (OMS) system. The end-to-end (E2E) automation of the organization, a novel approach that combines AI methods into one E2E platform with RPA capabilities at its core, was the key inspiration behind this service model, where prediction of hotel occupancy is performed using AI methods based on artificial neural networks, for the best staffing level.*

Keywords: Service capacity management, service demand, prediction, artificial neural network, staffing optimization, work shift scheduling, RPA.

Rezumat: *Studiul prezentat în acest articol se concentrează pe automatizarea proceselor back-office pentru gestionarea capacității în industria ospitalității. Această procedură, o componentă a managementului serviciilor back-office, este realizată utilizând tehnologia de automatizare a proceselor robotice (RPA) și integrată cu fluxul de lucru back-office în sistemul software de gestionare a operațiunilor (OMS). Automatizarea end-to-end (E2E) a organizației, o abordare nouă de combinare a tehnologiei de inteligență artificială într-o platformă E2E cu capabilități RPA, a fost principala sursă de inspirație pentru acest model de servicii, în care predicția gradului de ocupare hoteliera este realizata utilizând metode de IA bazate pe rețele neuronale artificiale pentru a calcula nivelul optim de angajati.*

Cuvinte cheie: Gestionarea capacității de servicii, cererea de servicii, predictie, rețele neuronale artificiale, optimizarea personalului, planificarea turelor, RPA.

---

[1] Ph.D. student, Doctoral School of Automatic Control and Computers, University POLITEHNICA of Bucharest, Romania, e-mail: radu-florin.negoita@stud.acs.upb.ro
[2] Professor, Dept. of Automation and Industrial Informatics, University POLITEHNICA of Bucharest, Romania, e-mail: theodor.borangiu@upb.ro

1. Introduction

The main concern that the management of service providing companies must deal with is matching daily service capacity with customer demand in a dynamic business and market environment. Two basic strategies address the problem of variable service demand: operations-oriented and marketing-oriented. *Operations-oriented* strategies adjust the level of service capacity; they increase capacity utilization by better matching the available service capacity to the demand. This first approach is centred around the concept of *productive capacity* and on the management's ability to exploit the forms of productive capacity - work, equipment, facilities, and infrastructure - efficiently (produce maximum output with fixed or limited resources in short run perspective, yield-oriented) and effectively (ensure excellent service delivery on long run perspective, result-oriented), as profitably as possible [1]. *Marketing-oriented* strategies consist in managing the level of service demand by altering or smoothing customer demand, which requires knowledge of demand patterns and drivers as well as the ability to segment market into customer classes [2]. Most service firms combine these two approaches in *yield* strategies [3].

Service capacity, defined in terms of an achievable level of output per unit time (e.g., workload of busy hotel staff) respectively of a constrained supporting facility (e.g., number of hotel rooms), is a perishable commodity because producing and consuming a service occur simultaneously (e.g., accommodation); this means that service capacity cannot be saved for sale at a later time in many service sectors among which hospitality.

Managing productive service capacity requires permanent service costing, i.e., identifying the costs related to designing, delivering and supporting the service for which facilities, equipment, infrastructure and staff work represent direct key cost components [4].

In the present analysis, it is also considered that capacity in this high-contact service sector is not limited by factors such as work skills classification or availability of equipment and consumables, because proper actions can be taken whenever necessary: professional staff selection and training, well-organized procurement, efficient supply and logistics [5].

Chasing demand is a first operations-oriented strategy to stretch or shrink productive capacity to adapt to the expected level of demand [6].

Adjusting capacity to match demand is a second operations-oriented strategy derived from the chasing demand approach which, unlike stretching capacity, directly adjusts the productive capacity in the range [optimum-maximum available] level as needed to balance demand fluctuations (in principal increases) [7, 8].

Smoothing demand is a marketing-oriented strategy that consists in swaying or even modifying the type of demand by trying to move customer requests to time periods with lower capacity utilization, respectively by depressing demand when it exceeds the optimum level of productive capacity [9].

An important first step in balancing demand and capacity is to establish the nature of service demand patterns by market segments: predictable cycle of demand levels or random variations of the demand. For *predictable cyclical demand*, it is necessary to establish the reasons for cyclical variations and the cycle length which are usually influenced by the facility's location, market segment and societal context. In the hospitality industry, the duration of the demand cycle varies from one week (for urban hotel location and business/weekend segments) to one year (for tourist resorts, varies by month/season and reflects seasonal climate characteristics, school vacation, employment calendars, public and religious holidays) [10].

When demand frequently exceeds limited capacity, the above actions may not succeed balancing supply and demand; then it is necessary to *inventory demand* at the time it occurs, which can be performed in two ways: establishing a reservation system (e.g., preselling the accommodation service in the hospitality industry), and using a formal queuing system that includes a mechanism to soothe waiting customers. Inventory demand is further refined in demand forecast and made operational by inventory planning, demand forecasting and analytics [11].

For a service organization, coordinating service capacity with dynamic varying, market-driven customer demand implies the correlation and synchronization of front-office and back-office processes - the first category addressing customer relationship management (CRM) and the second inventory planning, forecasting demand, analytics, and strategic decision making. *Operations management soft-ware* (OMS) systems assists plan, coordinate and monitor back-office service operations, digitalize daily front-office workflows of front-line personnel, identify, and mitigate operational process bottlenecks [12]. Service operations management refer to task automation, business process management and streamlining operations to improve the efficiency of employees in service delivery.

Intelligent Process Automation (IPA) uses IT capabilities to automate OMS system and network operational processes while interacting with elements like user platforms, databases and hardware infrastructure. While service automation focusses on automation of end-to-end service as a design-oriented approach that considers the whole service value chain, *Robotic Process Automation* (RPA) is more business-focused on the automation of back-office processes of the organisation. RPA represents a promising solution based on software robots (*bots*) to automate OMS tasks [13]. To implement service OMS, RPA must be extended beyond its rigid rule-based methods by combining bots with artificial intelligence (AI) techniques to forecast demand, analyse (big) market data, optimize capacity utilization, operations planning and staffing levels, and extract insights on customer perception of service quality [14, 15].

The paper presents an RPA solution that automates the service capacity management processes for prediction-based optimization of the staffing level needed in hotels. Chapter 2 presents the global service OMS system controlling demand and adjusting capacity in yield strategy. Chapter 3 presents an aggregate RPA system developed to automate capacity management processes. Experimental results are given in Chapter 4 that presents the advantages of extending RPA bots with AI techniques for end-to-end process automation. Conclusions and perspectives of future work are outlined in Chapter 5.

2. Service OMS system with front-office and back-office process integration

Back-office processes in the OMS system for services are automated using RPA software robots that employ AI techniques, thus achieving intelligent process automation (IPA). RPA bots for strategic processes handle a large volume of data and transfer the results of these calculations as configuration parameters and operational data to front-office processes that interact with customers, ensuring the integration of operation flows according to the principle of end-to-end automation (E2E), Fig. 1.

The main idea behind the proposed OMS system for services is the complete automation (or hyper-automation) of the organization, representing a new direction of unifying automation technology into a comprehensive end-to-end (E2E) platform with RPA capabilities at its core.

This platform encompasses the construction, management, and operation of software robots equipped with process analysis tools to address a wider range of automated management and operational functions for knowledge-intensive services, with quick business impact reporting. For the considered OMS system, E2E has been extended with IPA technologies to provide attributes of hyper-automation to the overall service system [16]:

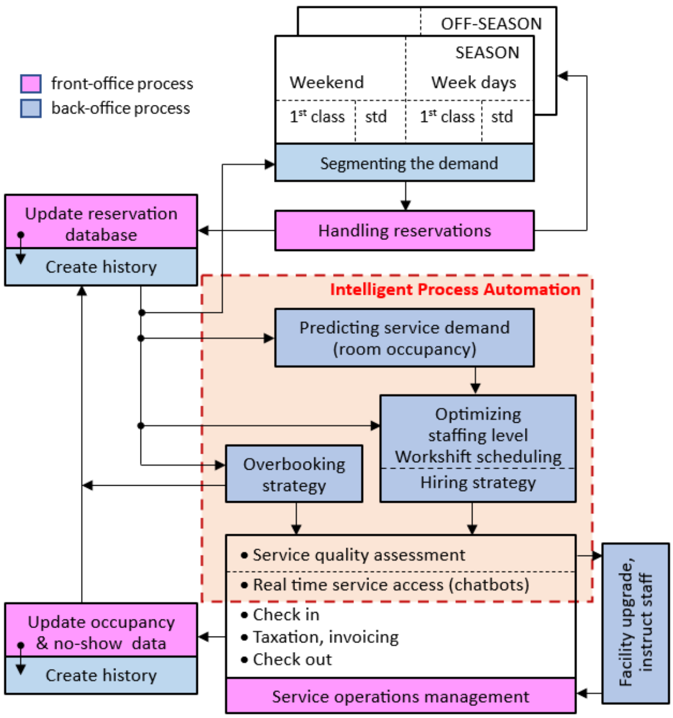

Figure 1. OMS service model for fully automated services. Front- and back-office operations connected by IPA (intelligent process automation).

Applied artificial intelligence techniques such as machine learning (ML), prediction, natural language processing (NLP), optical character recognition (OCR), and process mining analysis have been utilized to develop intelligent process automation (IPA) methods and tools for services. These techniques involve segmentation and prediction of service demand, sentiment analysis for assessing customer expectations and perception, quality evaluation of services, simulation of human conversations to understand customer inquiries and provide automated responses, as well as training software robots to read, see, learn, compare, analyze, and support decision-making [17].

The model in Fig. 1 represents the significant sets in terms of automating core front-office and back-office processes driven by software robots and guided by AI techniques that optimize the yield strategy by continuously controlling service demand and appropriately adjusting the utilization of hotel facilities and critical resources for service capacity.

Inventorying demand is represented by front-office processes that manage service requests (initiating and canceling reservations), confirmed requests (check-in, no-shows), and fulfilled requests (completed and interrupted, check-out), and update associated databases (reservation database, occupancy of accommodation facilities, no-shows). The content of these databases is accessed by software robots that automate front-office service operations such as check-in, check-out, taxation, and invoicing.

Elements of the yield strategy are selected and configured by a set of back-office processes that utilize market data, information about competitor firms, and regulations in the field [18].

Automating these action flows requires aggregating multiple data sources from the OMS system: the reservation platform, hotel registration desk, back-office data of organizational management, service delivery lists, human resources register, and integrating them into a continuously updated database to enable intelligent automation of back-office processes: a) demand inventorying (reservation system), and b) service delivery (registration system), see the central zone in Fig. 1.

In the back-office area, demand prediction - an extremely proactive process - will be used to determine the number of employees required on a weekly basis, the volume of items needed weekly for accommodation facilities, inventory levels, and marketing budgets. Demand prediction for intelligent automation of processes associated with the registration system forms the basis of the business plan to meet demand and improve profitability in the hotel business [19].

3. RPA solution

To achieve the AI prediction function and its integration into the RPA bot structure for calculating the staffing needs in services with labor cost optimization, a dataset containing daily customer records (confirmed bookings) over a period of 3 years was used. An artificial neural network-based prediction algorithm developed in the Python programming language was applied to this dataset.

To perform this prediction, a database containing all the relevant information for such a procedure is required. This includes customer data, information about the requested service type (duration, start date of stay, etc.), and the temporal evolution of the demand for this service (date of request initiation, subsequent modifications in the request) and its status (waiting, check-in, in-service, check-out).

To train the RPA bot performing the prediction, experimentation and analysis of results were conducted using the dataset available at https://www.kaggle.com/jessemostipak/hotel-booking-demand. To perform the prediction of hotel facility occupancy, the following ANN structure was defined:

- Input layer: It consists of 26 neurons, each represented by the columns of the dataset, denoted as I1, I2, I3, ..., I26 in Figure 5.1.
- Two hidden layers: The first hidden layer includes neurons H11, H12, ..., H1100, and the second hidden layer includes neurons H21, H22, ..., H2100. Each hidden layer consists of 100 neurons.
- Output layer: It consists of 2 neurons representing the accuracy of the prediction algorithm and its error (O1 and O2).

The operation of ANNs can be divided into two stages: Forward Propagation and Backpropagation. These methods are performed sequentially to obtain the prediction and calculate the error. Forward propagation involves multiplying the feature values by weights, adding bias values, and then applying an activation function to each neuron in the neural network. Multiplying the feature values by weights and adding bias values to each neuron is essentially the application of linear regression.

Fig. 2 presents the logic behind the forward propagation method of the ANN's operation, which consists of two functions: the sum function and the activation function. As shown in Fig. 2, the input to the first hidden layer is defined by equation (1), which represents step 1 where the sum of the products of the input values and their associated weights is calculated.

$$Y_{in} = \sum_{i,j=1}^{n} X_i * W_{ij} \tag{1}$$

Next, an activation function is applied to the result obtained in step 1, according to equation (2).

$$Y_{out} = F(Y_{in}) \tag{2}$$

The activation function is a crucial component of the ANN algorithm. It transforms the input data into output data by applying a mathematical operation of weighted sum of the inputs. Common activation functions include sigmoid, tanh, and rectified linear unit (ReLU). The choice of activation function depends on the nature of the problem and the analyzed data. The purpose of the activation function is to introduce nonlinearity to the data, which helps in identifying basic patterns. Additionally, the activation function is used to scale the value within a certain range. Multiple types of functions can be used to achieve the necessary scaling:1) Sigmoid function $f(x) = \frac{1}{1+e^{-x}}$ which scales the values between 0 and 1 and is commonly used for binary classification of results., 2) Tanh function $f(x) = \tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$ which scales the values between -1 and 1, 3) ReLU function (Rectified Linear Unit), $f(x) = \begin{cases} 0, x < 0 \\ x, x \geq 0 \end{cases}$ which results in the same value if the number is greater than 0 and 0 otherwise, 4) Leaky ReLU function, $f(x) = \begin{cases} 0.01x, x < 0 \\ x, x \geq 0 \end{cases}$, similar to the ReLU function, except that when the value is not greater than 0, it returns one-hundredth of the initial value.

The calculation function of the parameter Y_{out} is performed for each hidden layer, and the final result of the last layer is used as the input value for the output node. Equation (3) illustrates the method of calculating the input value Z_{in}.

$$Z_{in} = \sum_{j,k=1}^{n} Y_{out} * W_{jk} \tag{3}$$

After the completion of the calculation of Z_{in} the activation function will be applied to this variable, and the resulting output of the robot will be the prediction of the occupancy level for the next month, according to equation (4).

$$Output = F(Z_{in}) \tag{4}$$

Backpropagation is performed to find the optimal value of weights for the model by iteratively updating the parameters obtained through partial differentiation of the gradients of the loss function with respect to the parameters. To find the optimal value of the parameters, an optimization function is applied for reverse propagation. The available optimization functions are: Gradient Descent, Adam, Gradient Descent with momentum, and RMS Prop (Root Mean Square Prop). However, the optimization method used is Adam (Adaptive Moment Estimation).

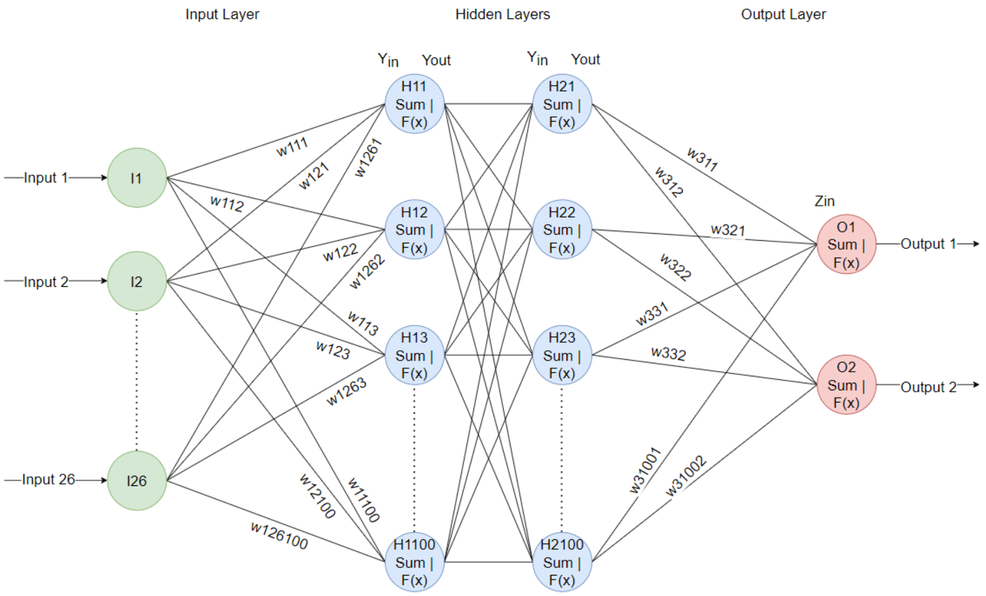

Figure 2. Representation of the ANN layers, inputs, outputs and associated weights

Adam is an optimization algorithm used in deep learning to update the model weights [21]; it is an extension of the SGD (Stochastic Gradient Descent) optimization algorithm that combines the benefits of both SGD and the RMSProp optimization algorithm (Root Mean Squared Propagation) [20]. Fig. 3 presents the RPA "Code" object for the Python implementation code of the defined ANN architecture for the prediction function.

```
import keras
from keras.layers import Dense
from keras.models import Sequential

model  = Sequential()
model.add(Dense(100, activation = 'relu', input_shape = (26, )))
model.add(Dense(100, activation = 'relu'))
model.add(Dense(2, activation = 'sigmoid'))
model.compile(optimizer = 'adam', loss = 'binary_crossentropy', metrics = ['accuracy'])
model_history = model.fit(X_train, y_train, validation_data = (X_test, y_test), epochs = 100)
```

Figure 3. Python code used to implement theANN architecture

In this RPA object, two modules from the Keras library, Dense and Sequential, are initialized [22,23]. The first module is responsible for initializing the neural networks, while the second module initializes the ANN layers. Next, the input and hidden layers are added, and then the model is compiled using the classifier.compile function. The first parameter is "adam," the optimization algorithm used, and the second parameter is the mean squared error, which is used to calculate the error function. For binary output in an ANN, the binary_crossentropy function is used, while the categorical_crossentropy function is used for multiple output variables. The third parameter, "metrics," represents the evaluation criterion of the model to improve its performance. The model is trained using the "model.fit" function, using the dataset of confirmed hotel booking records from the last 12 months to make predictions for the next month. The experimental database includes records (confirmed bookings) for the prediction month, allowing for comparison between the predicted and actual number of confirmed bookings (which can be translated into the occupancy percentage of the hotel facility for the target month) to determine the accuracy and error of the prediction model. After the model is compiled, the RPA robot will use it to extract the calculated prediction of the occupancy percentage of the hotel facility for the current month.

For robot training, 100 epochs are used as the training period. An epoch represents a single iteration over the entire training dataset of the model. During an epoch, the model's weights are updated based on the loss gradients with respect to the weights, using the Adam optimization algorithm. The number of epochs is a hyperparameter that determines how many times the model will see the entire training dataset. The chosen number of epochs is important because too few epochs can lead to insufficient adaptation, while too many epochs can lead to overfitting.

The Python script that performs the prediction of the monthly occupancy rate of the hotel facility (derived from the predicted number of confirmed bookings) is called in the flow of actions in the intelligent RPA automation diagram, Fig. 4.

The development of this IPA-type RPA process is based on a Blue Prism process flow diagram, where the first step involves extracting historical data regarding the recorded confirmed hotel bookings for the past 12 months and the records for the next month. The records from the database for the month being predicted are only used in the validation experiments of the prediction model and are compared with the prediction result to determine the accuracy of the method. After validation, the intelligent RPA bot will only use the records of confirmed bookings from the past 12 months for the monthly prediction.

To extract data, a VBO(Visual Business Object) called ”**Data – SQLServer:: Set Connection**” was used to establish the connection with the existing database. On the left side of the diagram, in the SQL sequence block, the necessary information for establishing the connection is provided: the server name, the database name, and the credentials used for authentication in the database.

After establishing the connection, the next step is to extract the required information from the database through an SQL query (”**SELECT * FROM BookingTable WHERE BookingTable.Year = 2022 OR (BookingTable.Year = 2022 AND BookingTable.Month = ‚January’)**”). The information resulting from this query will be saved in a local collection variable called "Results".

To use the extracted information from the database, an Excel instance was created (”**MS Excel VBO::Create Instance**”) and associated with a workbook (”**MS Excel VBO::Create Workbook**”), and then the resulting collection from the query was saved in the created Excel file. The next step in the RPA action chain diagram involves opening the "cmd" application using the Blue Prism object, ”**Utility - Environment::Start Process**”, with the following execution code: *If Arguments<>"" Then System.Diagnostics.Process.Start (Application, Arguments) Else System.Diagnostics.Process.Start(Application) End If,* where *Arguments* represent the application to be opened, and the name is passed to the robot through the local variable "TerminalName" (in this case, "cmd").

The RPA bot uses an "Attach" activity (described in Chapter 2) to attach to the opened application, and then it launches the Python script. To run the script, the "Navigate" activity is used, through which the robot navigates to the location where the script is located (”**CD C" & "<{SHIFT}:>{SHIFT}" & FilePath**”), and then it passes the script's file path as a parameter (which is stored in the local variable "PredictionFilePath") to execute it using the VBO code "py PredictionScript.py". After the execution is complete, the last step the robot needs to perform in the prediction process is to save the obtained result inside an Excel file. This result serves as an input to the automated process performed by the RPA bot, which optimizes the frontline staffing requirements.

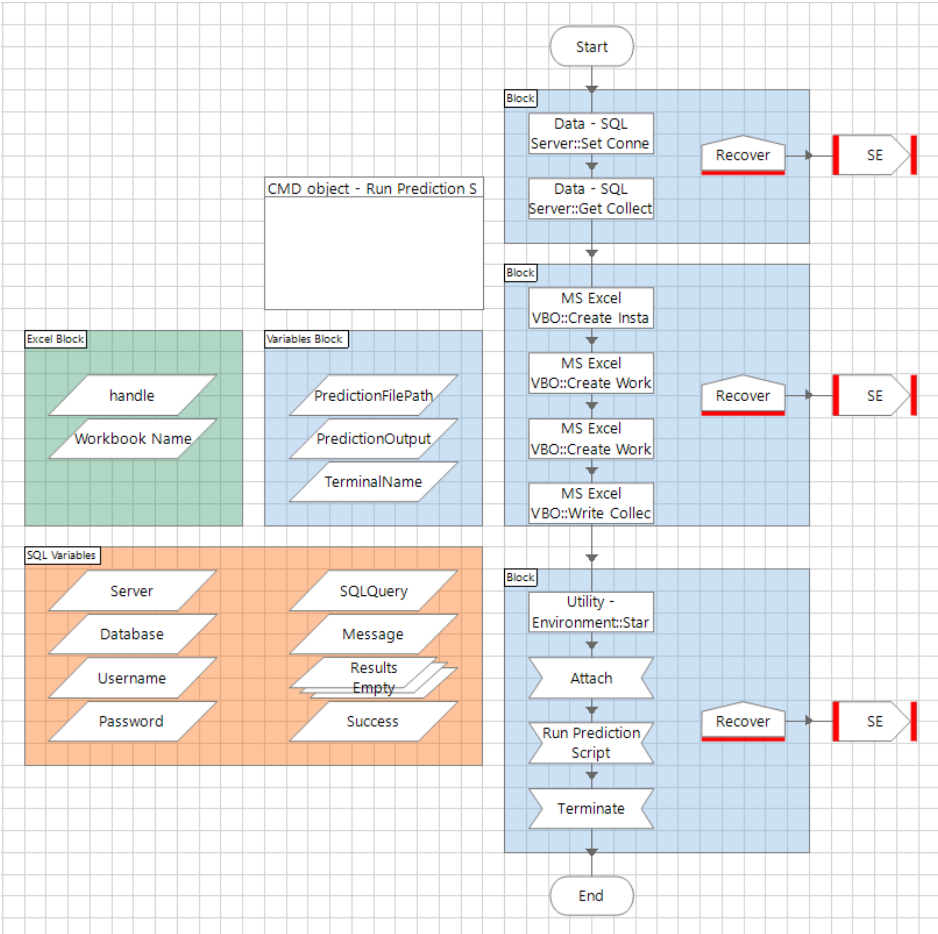

Figure 4 Graphical representation of the intelligent RPA subprocess for calculating the hotel occupancy prediction

4. Experimental results

Running the RPA robot that performs prediction using artificial neural networks provides three results: the predicted occupancy rate for the next month (integer value), the prediction accuracy (performance of the executed prediction model), and the prediction error.

For prediction, the monthly occupancy coefficients in a year (integer values) from the stored history of confirmed records in these 12 months were used as input data. With this data, using the ANN algorithm, the robot calculates the numerical occupancy coefficients for the next month.

Table 1. Results of the RPA bot's run for ANN prediction of the occupancy for 12 months

Year	Period *t* (month)	Hotel Occupancy	Prediction	Prediction Error
[y-1]	1 (jan)	98		
	2 (Feb)	85		
	3 (Mar)	92		
	4 (Apr)	95		
	5 (May)	136		
	6 (Jun)	140		
	7 (Jul)	148		
	8 (Aug)	154		
	9 (Sep)	141		
	10 (Oct)	126		
	11 (Nov)	110		
	12 (Dec)	120		
[y]	13 (Jan)	105	102	3
	14 (Feb)	89	87	2
	15 (Mar)	96	94	2
	16 (Apr)	100	98	2
	17(May)	139	136	3
	18 (Jun)	146	143	3
	19 (Jul)	156	152	4
	20 (Aug)	160	156	4
	21 (Sep)	149	146	3
	22 (Oct)	136	133	3
	23 (Nov)	118	115	3
	24 (Dec)	128	125	3

First, y_pred is used to obtain a matrix of predicted probabilities for each class. To convert these probabilities into binary results, the argmax method was used, where y_pred_classes = y_pred.argmax(axis=-1). This way, a one-dimensional matrix is obtained that contains the predicted results for the dataset used. In this binary value vector, each value of 1 represents a predicted confirmed reservation (occupancy), and the sum of all these values represents the total number of predicted confirmed reservations made by the robot for the target month. The RPA robot was iteratively run to calculate the prediction for a longer period (months t=13-24 of the year [y]) based on the actual occupancy data in months t=1-12 of the year [y-1], and the results of the runs are included in Table 1 (the rows marked in green indicate the peak tourist season).

The last column of this table represents the prediction error. To validate the method, the prediction result was compared to the actual occupancy (known from experiments in the larger history of more than 12 months stored in the database). In the current operation of the RPA robot, the actual occupancy value will only be known after the expiration of the month for which the robot made the prediction. This way, the accuracy (precision) and prediction error are evaluated.

Fig. 5 provides a graphical representation of the prediction accuracy calculated by the RPA robot based on ANN in the experiments. It can be observed that for the training dataset, the accuracy starts from 0 and eventually reaches a value of 0.99, while for the testing dataset, the accuracy reaches a value of 0.98. In this graphical study, the Y-axis represents the model's accuracy, and the X-axis represents the training epochs completed by the algorithm to finalize the prediction calculation.

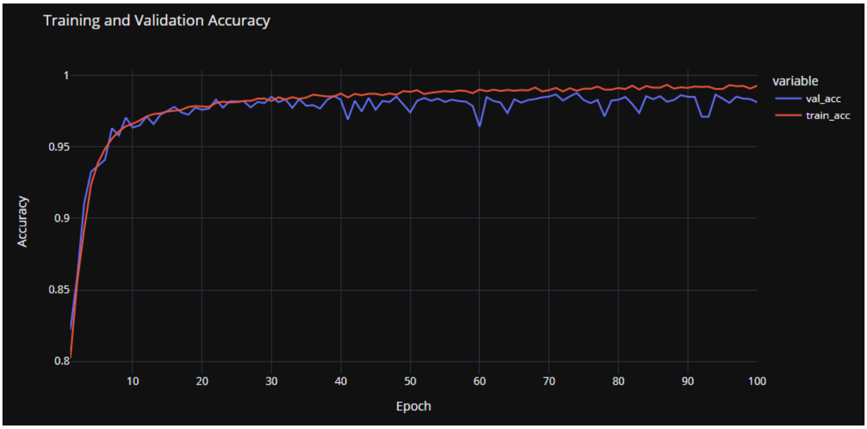

Figure 5 Comparative graphical representation of the accuaracy of the prediction model for the learning dataset and the test dataset

5. Conclusions

In the study, a mixed yield approach is used to explain a method for automating the back-office process of balancing service capacity and demand. In an integrated operations management software system with intelligent process automation capabilities, where the machine learning capability is leveraged by incorporating artificial neural networks into the prediction process, this process, which is a part of the back-office service management, is coordinated with the aid of RPA technology with customer interaction workflows.

The development of a service OMS model with complete automation of back-office operations using RPA technology is the key contribution of the study that was just described. The software robots of the suggested RPA system comprise AI-based algorithms capable of forecasting demand using the artificial neural networks in order to predict the demand for the upcoming season.

The experimental findings from the Blue Prism RPA software serve as validation for the intelligent bots created to automate back-office service management. These bots make use of past client requests (reservations, registrations) to predict and manage future demand segment evolutions. Operating autonomy in dynamic environments, high-speed, error-free computing that is 15 times quicker than a human agent than that of human specialists, and embedded intelligent process automation are their primary selling points. The goal of future research will be to create solutions for the end-to-end automation of service management systems, such as:

• Process and task mining tools to find potential for automation.

• Workload automation solutions, low-code development tools, integration platform as a service, and RPA bots.

• Business logic tools, such as intelligent business process management, decision management, and business rules management, to make it easier to modify and reuse automations.

REFERENCES

[1]. *J.A. Fitzsimmons*, *M.J. Fitzsimmons*, Service Management: Operations, Strategy and Information Technology, 6th edition, ISBN: 978-0073403359, McGraw-Hill, 2008

[2]. *J. Wirtz*, Balancing Capacity and Demand in Service Operations, Winning in Service Markets Series, Vol. 7, WS Professional, World Scientific Publishing, ISBN: 9781944659295, 2018

[3]. *R.L. Rajani, G.S. Heggde, and R. Kumar*, Services Redesign Strategies for Demand and Capacity Management and Impact on Company Performance, Vision: The Journal of Business Perspective, MDI, doi: 10.1177/09722629221107238, 2022

[4]. *Stanford UIT*, Service Management. Improving the end-to-end service experience for the Stanford community, Service Management Toolkit, https://uit.stanford.edu/service-management, accessed 24 November 2022

[5]. *L. Li, D.E. Gray, A.J. Lockwood, and D. Buhalis*, Learning About Managing the Business in the Hospitality Industry, Human Resource Development Quarterly, 24 (4), pp. 525-559, doi: 10.1002/hrdq.21171, Wiley, 2013

[6]. *R.P. Fisk*, Coping With Fluctuating Demand For Services, Interactive Services Marketing, 14, https://college.cengage.com/business/fisk/interactive/3e/assets/students/outlines/chapter14.html, Cengage, 2018, accessed December 2022

[7]. *P. Chetty*, Strategies to match demand and capacity. [online] Project Guru, 2010. Available at: https://www.projectguru.in/strategies-to-match-demand-capacity/, accessed 5 January 2023

[8]. *C. Morariu, O. Morariu, and T. Borangiu*, Offer request management in service oriented holonic manufacturing, UPB Scientific Bulletin, Series D: Mechanical Engineering 76(1):257-276, January 2014
[9]. *K. Granger*, The fundamentals of demand planning and forecasting, eazystock, Jan. 4, 2022, https://www.eazystock.com/blog/the-fundamentals-of-demand-planning-and-forecasting/
[10]. *S. Ross*, Operating Income vs. Gross Profit, Investopedia, August 2022, accessed January 2023, https://academy.investopedia.com/products/financial-modeling
[11]. *N. Thomopoulos*, Demand Forecasting for Inventory Control. In: Demand Forecasting for Inventory Control. Springer, Cham, https://doi.org/10.1007/978-3-319-11976-2_1, 2015
[12]. *P. Ilyusenko*, Operations Management Software 360-Degree Overview, ScienceSoft 2023, https://www.scnsoft.com/operations-management#service-oms
[13]. *H.P. Fung*, Criteria, Use Cases and Effects of Information Technology Process Automation (ITPA). Advances in Robotics & Automation 3: 124, doi: 10.4172/2168-9695.1000124, 2014
[14]. *S. Kumar*, S, Robotic Process Automation, Int. Research J. of Modernization in Engineering Technol. and Science, Vol. 2, Issue 7, pp. 1248-1251, 2020
[15]. *C. Sutherland*, Framing a Constitution for Robotistan – Racing with the Machine for Robotic Automation. Racing with the Machine of Robotic Automation, HfS Research, 2013
[16]. Blue Prism (2021). Intelligent Automation Platform, Enterprise Datasheet, https://www.blueprism.com/products/intelligent-rpa-automation/
[17]. Schmid, A., Sobottka, T., Luther, S., Sihn, W. (2022). Demand Planning Falcon. Zielgenaue, stochastische Bedarfsvorhersagen mit einer neu entwickelten digitalen Planungsmethode, Industrie 4.0 Management, 6(38), pp. 47-50, https://doi.org/10.30844/IM_22-6_47-50
[18]. Li, L., Gray, D.E., Lockwood, A.J., Buhalis, D. (2013). Learning About Managing the Business in the Hospitality Industry, Human Resource Development Quarterly, 24(4), pp. 525-559, doi: 10.1002/hrdq.21171, Wiley
[19]. Kliewer, N., Ehmke, J.F., Borndörfer, R. (2017) (eds). Operations Research Proceedings 2017, Ann. Int. Conf. GOR Society, doi: 10.1007/978-3-319-89920-6
[20]. Wilson, A,C., Roelofs, R., Stern, M., Srebro, N., Recht, B. (2017). The Marginal Value of Adaptive Gradient Methods in Machine Learning, arXiv:1705.08292v2, doi: 10.48550/arXiv.1705.08292, https://arxiv.org/abs/1705.08292
[21]. Zhang, J. (2020). Optimisation Algorithm - Adaptive Moment Estimation (Adam), published in Towards Data Science, Jan 21, 2020, https://towardsdatascience.com/optimisation-algorithm-adaptive-moment-estimation-adam-92144d75e232
[22]. Keras (2022). Keras - Dense Layer, TutorialsPoint, https://www.tutorialspoint.com/keras/keras_dense_layer.htm
[23]. Chollet, F. (2020). The Sequential model, Developer Guides Keras, https://keras.io/guides/sequential_model/

Acknowledgements

The results presented in this article were been funded by the Ministry of Investments and European Projects through the Human Capital Sectoral Operational Program 2014-2020, Contract no. 62461/03.06.2022, SMIS code 153735.