Marketing Automation in the Services Sector with Combined Artificial Intelligence

Lubomir Cvrk, Frantisek Mazuch, Tomas Sykora

***Abstract*—The article discusses the modeling of artificial intelligence to address the issue of predicting future demand in the service sector (hairdressing, massages, exercise, doctors, physiotherapists, etc.). Future demand is examined on a weekly granularity and may exhibit fluctuations, particularly declines. The intention is to proactively respond to these fluctuations through marketing automation and attract additional customers who will compensate for the decline in demand. The result is a combined artificial intelligence that predicts future revenues for entrepreneurs using time-series analysis (revenues are derived from demand). Subsequently, a binary classifier is applied to identified demand fluctuations to determine numbers of future customers. Thanks to artificial intelligence, marketing automation is made possible. The classifier minimizes the numbers of *false positives* and *false negatives* while maximizing *true positives*. The goal is to maximize the utilization of labor and capital for entrepreneurs, as well as optimize their marketing expenses. Discount offers will be sent to only a limited number of customers where there is a higher probability of response. Online advertising will only be operational for the necessary duration, allowing entrepreneurs to achieve cost savings.**

***Keywords*—marketing automation, artificial intelligence, time-series analysis, classification, long short-term memory, gradient boosted trees.**

I. Introduction

The fluctuation in demand has a negative impact on businesses in the service sector. Temporary declines in demand, in particular, lead to inefficiencies in the allocation of labor and capital, resulting in losses as fixed costs (employees, rent, energy) must still be covered. Larger business entities or entire reservation platforms therefore seek intelligent tools for demand management. This enables them to address anticipated declines in demand in an informed and automated manner.

Marketing campaigns serve as tools to support demand. These campaigns include direct methods to engage specific customers, as well as online advertising. During periods of decreased demand, campaigns need to be targeted at customers who are more likely to respond to the marketing offers.

In marketing, there is generally a conversion ratio, indicating a certain percentage of engaged customers who will actually purchase the offered services.

AI-driven marketing brings a new dimension of efficiency to the service sector by launching demand-increasing campaigns precisely when necessary. Simultaneously, investment funds are saved during periods when demand stimulation is not needed (campaigns automatically stop). The above-mentioned automation through artificial intelligence increases both the cost and time efficiency of businesses in the service sector overall.

The aim of the research, therefore, is to explore possibilities for systematically capturing the issue of fluctuations in future demand for services using artificial intelligence. The objective is to predict future demand fluctuations for services within a horizon of at least 2 to 4 months. This timeframe allows for the implementation of marketing campaigns that attract customers and thus mitigate the predicted decline in demand. At the same time, it is essential to ensure a sufficient level of forecast reliability for commercial deployment.

II. The State-of-the-Art in Demand Prediction

The prediction of future demand is not a new area in science and research, but the previous attention of the scientific community has been focused on areas other than the provision of services to end customers. Published works that are relatively close to the subject of our research include the following.

A. Service-oriented Production

Article [1] utilizes a structural equation model to describe the relationships between customer satisfaction and influencing factors. Demand prediction is designed using least-squares support vector machine, an algorithm whose training time grows quadratically with the amount of training data.

B. Demand Management in Taxi Services

Authors of [2] proposed a complex deep neural network for short-term (tens of minutes into the future) demand prediction in the personal transportation sector. They model demand and supply of personal transportation services in five-minute intervals, considering factors such as location, traffic infrastructure utilization, weather, completed orders, and delays. The model is quite complicated with a high number of parameters, leading to high hardware requirements during training.

Works [3] and [4] share the same topic as [2]. These papers study short-term demand prediction for personal transportation in urban agglomerations, aiming to optimize the spatial distribution of taxi drivers based on predicted demand in minutes to tens of minutes into the future. The system recommends to taxi drivers which urban areas to head to, where demand is expected to be higher, with the goal of satisfying demand as quickly as possible (who arrives first gets the customer).

C. Demand for Energy (Smart Grids)

A comparison of multiple machine learning models (especially linear regression – SVR, i.e., regression SVM, and a three-layer feed-forward neural network) for predicting electricity demand for the next 7 hours is proposed in [5]. The authors achieved the best results with the SVR algorithm.

D. Demand in Supply Chains

Article [6] addresses the problem of predicting future demand for specific products in terms of units sold each month using time series analysis based on a classic recurrent neural network. The observed error rate beyond the 10th month into the future exceeded 16%.

E. Rationale for Research

To the best of our knowledge, there is no existing paper on demand modelling focused on the Services business sector. From the literature survey, and also from the authors' experience, it is not possible to take the design of a machine learning model from other works and conduct a training using the available data. Each scientific paper builds AI models on different principles and with different application goals. Also, different scientific works handle completely different input data, which are always unique to the research subject. Therefore, we conducted research and development in the given topic.

III. Demand Prediction with Classification

A. Data Sources

The prediction of future demand can be tackled based on reservation data acquired and stored in reservation systems. After initially exploring possible methods to address this issue and considering the content, structure, and volume of available data, we proceeded with researching a binary classifier. This classifier is designed to categorize customers into one of two classes, where a customer either makes a reservation (positive class) or does not make a reservation (negative class) for services from a specific entrepreneur in a given week of the year.

B. Theoretical Foundations

The goal of machine learning is to find generalized information and patterns in data using computer algorithms. Essentially, the algorithms used can be divided into several groups:

- Supervised learning,
- Unsupervised learning,
- Combinations of the above,
- Reinforcement learning.

The type of problem being addressed is a *classification task* – a *supervised learning* problem.

Classification methods are based on the assumption that individual data objects (examples) belonging to the same concept have similar characteristics. If each example is described by n attributes, it can be represented as a point in an n-dimensional attribute space. Objects belonging to the same concept then form *clusters* in this space, and the goal of the classification task is to find a *function* that determines the membership of examples in these clusters. To formalize these considerations, let's draw inspiration from the description of the classification task in the book [7].

The analyzed data is stored in a matrix D, consisting of n rows and m columns.

$$D = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1m} \\ x_{21} & x_{22} & \cdots & x_{2m} \\ \vdots & \vdots & & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nm} \end{bmatrix}$$

The rows of the matrix represent individual examples, with the i-th object being the row X_i.

$$X_i = \begin{bmatrix} x_{i1} & x_{i2} & \cdots & x_{im} \end{bmatrix}$$

The columns of the matrix represent attributes. Let the j-th attribute is denoted as A_j.

$$A_j = \begin{bmatrix} x_{1j} \\ x_{2j} \\ \vdots \\ x_{nj} \end{bmatrix}$$

In order to classify the data, it is necessary for there to exist an attribute that contains information about the assignment of the object to a class. Let's call this attribute the *target attribute* and denote it by the symbol C. The other, non-target attributes A_j are referred to as *input attributes*.

$$C = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

By adding the target attribute to the matrix, we obtain data suitable for machine learning. These data are referred to as *training data*, and we denote the corresponding matrix as D_{TR}.

$$D_{TR} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1m} & y_1 \\ x_{21} & x_{22} & \cdots & x_{2m} & y_2 \\ \vdots & \vdots & & \vdots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nm} & y_n \end{bmatrix}$$

By classification task, we mean the task of finding knowledge represented by a function f that would allow assigning a suitable value to the target attribute for the values of input attributes of a given object. During classification, a value for the target attribute is deduced for the values of the input attributes X of a particular object. Let's denote this deduced value as y'.

$$y' = f(X)$$

The derived value y' can, of course, differ from the actual value y for a specific object, and therefore it is necessary to *calculate the classification error*, which indicates to what extent the function f covers correct values.

To find the classification function f, there are various methods available, and a variety of them are used in practice. These methods differ in expressive power as well as in the representation of learned knowledge.

Within the research, we experimented with methods such as *Support Vector Machines*, *Gradient Boosted Trees* from *XGBoost* and *Sklearn* libraries, and *recurrent neural networks*.

C. Experimental Datasets

The experimental dataset included 3,070,710 reservations made by 947,670 customers from 537 providers in the *Beauty* sector. From this dataset, 37,484 reservations were selected for experiments, involving 10,571 clients specifically from the *Barbershop* sector, created between May 1, 2021, and May 1, 2022. This subset is referred to as the *small dataset*.

The *large dataset* was extracted from the same source, covering reservations made from May 1, 2021, to May 1, 2023, and included reservations from 77,605 customers.

In these data, it is necessary to identify and extract attributes that will form the input vector X_i for the machine learning algorithm used. Binary and integer attributes can be used in their existing form for the input vector, while categorical attributes need to be vectorized. This is achieved using the *One-Hot Encoding* method, which involves creating as many new attributes as the original attribute can take on values. These new attributes are binary and set to 1 if the original attribute matched a given value, and 0 otherwise.

The *created and paid orders* form the basis for *positive examples* in the training data (the target attribute takes on the value 1). Therefore, negative examples must be *computed* from existing data.

Negative examples need to be inserted between every two reservations of one client at one provider. Predictions are planned in weekly intervals (sufficiently accurate). Therefore, negative examples must be generated for each week between two reservations, with one negative example per customer, provider, and week. The reference time for each week is chosen as Monday at 00:00 UTC.

D. Imbalanced Dataset

For example, men visit barbershops regularly, about once a month. Therefore, the dataset for each such customer contains (on average) 12 positive examples and 40 negatives for a men's haircut service in a year.

Due to the nature of the problem, the target attributes of datasets for practically all types of businesses in the service sector are *imbalanced*, meaning they contain significantly more negative examples (with a target attribute of 0). To train a quality machine learning model, it is necessary to work with a much larger training set than would be typical for a balanced dataset. This is primarily to ensure an adequate number of positive examples in the dataset. Fortunately, our data sets contain enough positive examples.

E. Input Vector

Attributes used for input vector computation are in the Tab. 1. The last attribute is the target attribute of 0 or 1 (negative or positive class respectively) from which the algorithm learns the class of the given training data example.

Tab. 1: Source Attributes for Input Vector

Attribute name	Attribute description	Range
`client_id`	Customer identifier.	int
`business_id`	Business (entrepreneur) identifier.	int
`service_tag`	Machine readable service tag.	categorical
`business_category_tag`	Main business category of the entrepreneur.	categorical
`location_tag`	The most detailed known location tag of the business.	categorical
`time_since_last_visit`	The number of seconds since the visit. Related to Monday 00:00.	int
`weeks_since_last_visit`	The number of weeks since last visit.	int
`months_since_last_visit`	The number of months since the last visit.	int
`quarters_since_last_visit`	The number of quarters since the last visit.	int
`booking_week`	The number of the week in the year.	int
`target`	**Target value {0, 1}**	**int**

F. Results Evaluation

Theoretical foundations

For measuring the quality of results, basic evaluation metrics for classification were used, including *Precision*, *Recall*, *F-measure*, and *Accuracy* [8]. To calculate these metrics, each example in the dataset needs to be classified into one of four categories:

- *TP (true positives)* – examples correctly classified as positive.
- *FP (false positives)* – examples incorrectly classified as positive.
- *TN (true negatives)* – examples correctly classified as negative.
- *FN (false negatives)* – examples incorrectly classified as negative.

The metrics can then be derived from the sizes of these individual classes as follows [8]:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1measure = \frac{2 \cdot TP}{2 \cdot TP + FP + FN}$$

Given that the dataset for this classifier is imbalanced, relying solely on the *Accuracy* metric is not sufficient, as it tends to favor the negative class, which is prevalent in the dataset (due to a high number of *TN*). Instead, the trade-off between *Precision* and *Recall* metrics is more informative.

Miss Rate Metric and Threshold Determination

As a supplement in the search for the optimal threshold separating positive and negative classes, and for comparing the accuracy of individual models in this research, we proposed the *Miss Rate (MR)* metric. This metric serves the purpose of threshold tuning and as a tool for evaluating the efficiency of different classifier variants. The metric is focused on cases where the goal is to *minimize both FP and FN* while *maximizing TP*.

MR is derived from error impact to true positives ($ErrorImpact_{TP}$). Let's denote *C* as the number of items classified in the positive class:

$$ErrorImpact_{TP} = \frac{FP + FN}{TP}; \; TP = C \cdot Precision;$$

$$FP = C(1 - Precision); \quad Recall = \frac{TP}{TP + FN}$$

$$FN = \frac{TP}{Recall} - TP = \frac{C \cdot Precision}{Recall} - C \cdot Precision =$$

$$= \frac{Precision}{Recall} \cdot C(1 - Recall)$$

Hence, *FN* is a certain multiple of the complement to *Recall*. For simplicity and assuming dimensionlessness of the resulting value, we can neglect this multiple represented by the fraction of *Precision / Recall* and consider *FN* as simply the complement to *Recall* times *C*. The complement to *Recall* is also sometimes referred to in the literature as the *false negative rate*. When applied to $ErrorImpact_{TP}$ we receive the *Miss Rate*:

$$MissRate \approx \frac{C(1 - Precision) + C(1 - Recall)}{C \cdot Precision} =$$

$$= \frac{C[(1 - Precision) + (1 - Recall)]}{C \cdot Precision} = \frac{2 - Precision - Recall}{Precision}$$

Specific values of *Precision* and *Recall* are derived from the tested threshold. Because we reduce the calculation of *FN* by the above-mentioned method, we decrease the impact of *FN* in the *MR*, simultaneously increasing the impact of *FP*. The influence of *FP* needs to be emphasized in order to minimize it during classification. In business practice in the service sector, *FP* means that the classifier has included customers in the positive set who will not make a reservation. This would have a negative impact on business management because space would be reserved for these customers even when it is not necessary. The business capacity would be considered exhausted (or more occupied) even though it would still be possible to continue acquiring customers via marketing campaigns.

However, optimizing the metric for simple *FP* minimization is not feasible. *FP* itself is zero at very high thresholds where a low number of *TP* and a high number of *FN* is found. The *MR* ultimately minimizes the counts of *FP* and *FN* while maximizing *TP*. The optimum of all the conflicting dimensions mentioned above is located at its minimum which refers to the optimal threshold.

Thus, if the classifier is correctly assembled and trained, correctly classified elements should outweigh the misclassified ones above a certain threshold and *MR* values will appear in the interval of *<0,1>*. The value of *Miss Rate* will then be possible to display in a single graph with *Precision* and *Recall*.

G. Best Classifier Results

The algorithm that achieves the best performance in terms of accuracy on both small and large datasets is the *GBT* (*Gradient Boosted Trees*) with a tree depth of 3. This classifier achieves the highest *Recall* values of all. Training this algorithm on a large dataset (1.8 million training examples) for 5,000 estimators takes over 12 hours of machine time, making it by far the most resource-intensive. *GBT* is the only algorithm on the datasets that achieved the *Accuracy* value of 0.9 on the validation set (Fig. 1).

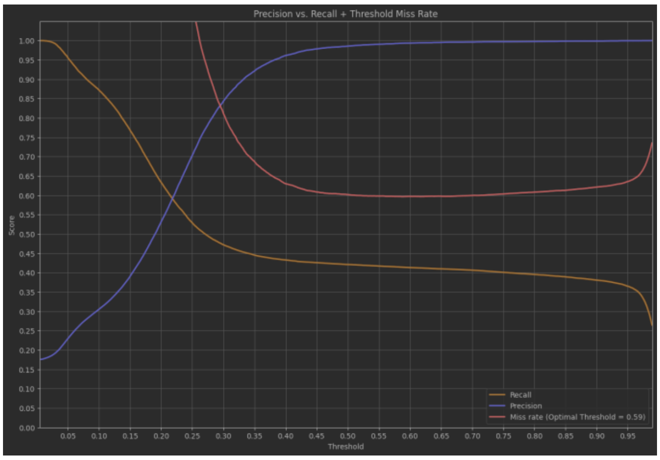

Fig. 1: Precision/Recall/MR Results of the Most Accurate Classifier

```
Test accuracy: 0.90
              precision    recall  f1-score   support

           0       0.89      1.00      0.94    153664
           1       0.99      0.42      0.59     32759

    accuracy                           0.90    186423
   macro avg       0.94      0.71      0.77    186423
weighted avg       0.91      0.90      0.88    186423

Train accuracy: 0.90
              precision    recall  f1-score   support

           0       0.89      1.00      0.94   1385305
           1       0.99      0.42      0.59    292502

    accuracy                           0.90   1677807
   macro avg       0.94      0.71      0.77   1677807
weighted avg       0.91      0.90      0.88   1677807
```

Min Miss Rate = 0,59

Second best result reached the *XGBoostClassifier* with the *Accuracy* value of 0.89 and *Recall* value of 0.40 at the *Threshold* of 0.5. Training this algorithm takes only 1 201 seconds of machine time, which makes it 40x better in speed than *GBT*.

H. Costs of Classification

The results show that the operating costs of running a binary classifier with the *XGBoost* algorithm are much lower than with *GBT*. However, *XGBoost* still has relatively high costs. It can be generalized that deploying the *XGBoost* classifier on a large scale to make demand predictions for each customer would increase the infrastructure costs by 24% which is too much. Therefore, we had to continue the research and find another way of predicting the customer's demand on the large scale (10M+ of customers).

IV. Demand Prediction with Time-Series Analysis

The prediction of future demand can only be approached based on reservation data acquired and stored in the reservation systems. After conducting research on a binary classifier, which exhibited certain challenges from a cost perspective, we proceeded with further investigations by reformatting the problem from predicting the future demand of specific customers to the time-series analysis problem. This is done based on the *accumulated revenues of businesses in individual weeks*.

Hypothesis: It should be possible to use the revenue predictor for all businesses in the system, generating an estimate of future revenues for several weeks ahead. This will provide an overview of future fluctuations in demand, as *revenues inherently reflect the demand* that leads to their creation.

The output of the revenue prediction can be presented in a graphical representation of future revenue values, clearly displaying the predicted fluctuations. Subsequently, businesses will be able to respond to the forecasted demand fluctuations using the automated marketing tools.

A. Capturing the Time-Series Problem

Reservation systems store historical data on bookings. The date of booking is recorded for each reservation. Each reservation has an associated monetary value. The sum of reservation prices and other sales made through a mobile terminal in stores or via online apps for a specific week provides the total revenue for that week. This yields 52 input values for the year.

The output must be predictions of future entrepreneur revenues for a horizon of 16 weeks in the form of specific predicted revenue amounts.

The task at hand is a regression task – predicting numerical values.

Statistical analysis of business revenue time series values indicates that these are unstructured data, despite containing some visible patterns, such as a decline in revenue for almost all service entrepreneurs for example during the Christmas holidays.

The forecast must cover 16 steps into the future, maintaining the dynamic nature of this parameter to easily predict different number of future time spans, without adjusting the model's structure.

The output model must be dynamic. It will be retrained weekly with new revenue data from the past weeks. This ensures constant accuracy.

Data research revealed the following phenomena:

- Data are entirely continuous; entrepreneurs operate permanently. This is the predominant phenomenon (94% of entrepreneurs).
- Seasonal business type, where revenues are generated, for example, from April to October, and this pattern repeats for a specific entrepreneur (5% of entrepreneurs). Data can be considered continuous; it's necessary to train this data outside the season.
- An entrepreneur interrupts and later resumes operations. This is a random event that cannot be predicted. Data inconsistency cannot be addressed without external input. After resuming operations, prediction accuracy would be negatively affected for at least the length of the time window. In this case, we will consider the entrepreneur as entirely new.

B. Investigation of Classic Approaches to Time-Series Prediction

In the forecasting of time series, linear methods such as *ARIMA* [15] traditionally prevail because they are well-understood and effective for many problems. However, these classical methods also suffer from some limitations, such as:

- *Focus on complete data*: Missing or damaged data are generally not supported.
- *Focus on linear dependencies*: The assumption of a linear relationship excludes more complex joint distributions.
- *Focus on fixed temporal dependence*: The relationship between observations at different times and subsequently, the number of historical observations provided as input data, must be diagnosed and specified.
- *Focus on one-dimensional data*: Many real-world problems have multiple input variables.
- *Focus on single-step predictions*: Many real-world problems require predictions with a long-time horizon.

Machine learning methods can be effective for more complex time series forecasting problems with multiple input variables, intricate nonlinear relationships, and missing data. To perform well, classical methods often require manually crafted features prepared either by *domain experts* or *professionals with expertise* in the field. Existing techniques have often relied on manually prepared features, the creation of which was costly and required expert knowledge in the given domain [14].

C. Transformation of Time Series into Supervised Learning

Supervised learning is machine learning where the model is created based on known training examples. All literature identifies the *Sliding Window* method as the only possible method for preparing training data. In this method, the time series is progressively traversed, and individual windows form training examples. This applies whether using classical feed-forward neural networks [9], [10], [11], recurrent networks with attention mechanisms [12], or convolutional networks deployed for time series problems [13].

Example: For time series of 1 to 10 (with steps by one) the train examples can be prepared the following way:

```
x1, x2, x3, y
 1,  2,  3,  4
 2,  3,  4,  5
 3,  4,  5,  6
...
```

In this case, the size of the time window is 3, and by incrementally shifting the window through the dataset, we obtain a training set where each column is a *feature*, and each row is a *sample*. The *target value* is represented by the column *y*.

D. Preparation of Training Vectors for Recurrent Neural Networks

For predicting time series using recurrent neural networks, it is necessary to ensure special normalization. The following transformations of revenue values on input and output were performed:

- Application of the natural logarithm to convert multiplicative factors into additive ones.
- Utilization of min-max normalization to bring values into the interval *<-1, 1>*.

The output consists of individual samples of normalized data for a given window size, which is dynamic. These samples then serve as training examples for the neural networks.

E. Evaluating Accuracy of Predictions

The *RMSE* (root mean squared error) metric will be used for evaluating prediction accuracy of future revenues:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}$$

RMSE is expressed in the same units as the time series and can be presented as a percentage of the average revenue values. For example, if the average revenue is 100,000 CZK and the *RMSE* is 12,000 CZK, the prediction error is 12% of the revenue. This means that the actual values are likely to fall within ±12% of each prediction (confidence interval).

F. Comparison of traditional methods and LSTM networks

To compare the performance of the traditional statistical methods (especially *ARIMA*) and neural networks, we conducted a series of experiments. The results indicated that *LSTM* neural networks on the provided data would achieve better outcomes than traditional methods (Fig. 2). The average *RMSE* of *LSTM* was in average lower by 19,77% than the average *RMSE* of *ARIMA*.

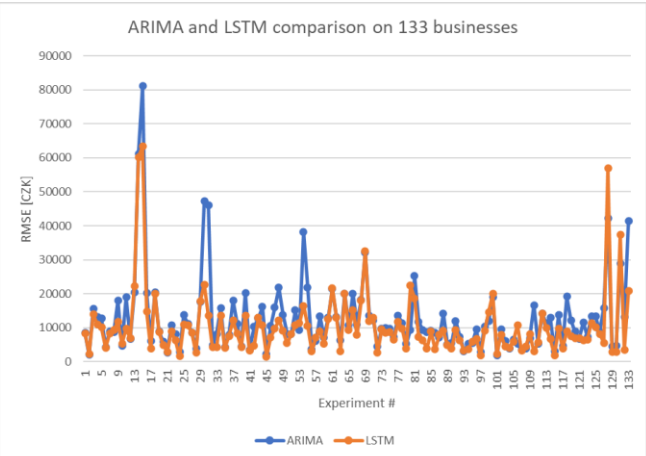

Fig. 2: *ARIMA* and *LSTM* Comparison Graph

G. Selecting the Optimal Neural Network and Parameters

Experiments were conducted on a small dataset to evaluate which of the 14 selected neural networks (see Tab. 2) would achieve the lowest *RMSE* on the provided data. The dataset consisted of bookings from 102 entrepreneurs, with average weekly revenues amounting to 76,029 CZK. Due to the stochastic nature of training neural networks, each experiment was trained 10 times, and the resulting values were averaged. The difference in results between individual trials varied by a maximum of 7%. Best results were achieved in training 100 epochs (with batch size of 256, and window size of 16), as seen on the Tab. 2.

Tab. 2: Best Results of 14 Neural Networks

Algorithm	ARMSE*	ARTAR**
vanillaLSTM	11 663,60	15,34%
reservioAttention50x100LSTM	11 824,00	15,55%
reservioMidLSTM	11 987,20	15,77%
reservioTinyLSTM	12 310,20	16,19%
reservioAttention100x50LSTM	12 341,60	16,23%
stackedLSTM	12 374,40	16,28%
reservioAttention50x50LSTM	12 378,00	16,28%
reservioAttention200x100LSTM	12 498,20	16,44%
reservioAttention128x64LSTM	12 600,40	16,57%
reservioAttention128x128LSTM	12 791,80	16,83%
reservioAttention128x256LSTM	12 796,40	16,83%
bidirectionalLSTM	13 496,20	17,75%
reservioDeepLSTM	14 134,60	18,59%
reservioRNN	15 166,40	19,95%

**ARMSE* = Average *RMSE*
** *ARTAR* = Average *RMSE* to *Average Revenue*

The winner was the *vanillaLSTM* neural network defined in [16] with the *RMSE* of 11,663.60 CZK. That is 15.34% of the average revenues in the dataset for the 16 weeks into the future. This result can be considered very good and commercially viable. In shorter-term predictions, the error is expected to decrease.

The Impact of Sliding Window Size to RMSE

An investigation of the dependence of the size of the sliding window parameter on the accuracy achieved by the selected neural networks was performed. The experiments were trained with the above settings for 50 and 100 epochs, window sizes from 2 to 32. Training more epochs than 100 led to worse results. Tab. 3 displays the best results and Fig. 3 shows all results graphically.

Tab. 3: Window Sizes and Related Average RMSE

Algorithm / Window	20	22	24	26	28	30	32
reservioMidLSTM	15,9%	16,1%	16,6%	16,6%	16,7%	16,9%	17,1%
vanillaLSTM	15,3%	16,1%	16,3%	15,6%	16,1%	15,6%	16,0%
reservioAttention50x100	16,1%	15,8%	16,2%	16,4%	17,9%	16,5%	16,2%

At the same time, it should be emphasized that the size of the window limits the usability of the algorithm. It is necessary to have just as many values as the window length to perform the prediction. For this reason, we decided to use a window size of **16 weeks**, where the value of the measured RMSE is still very close to the lowest measured value.

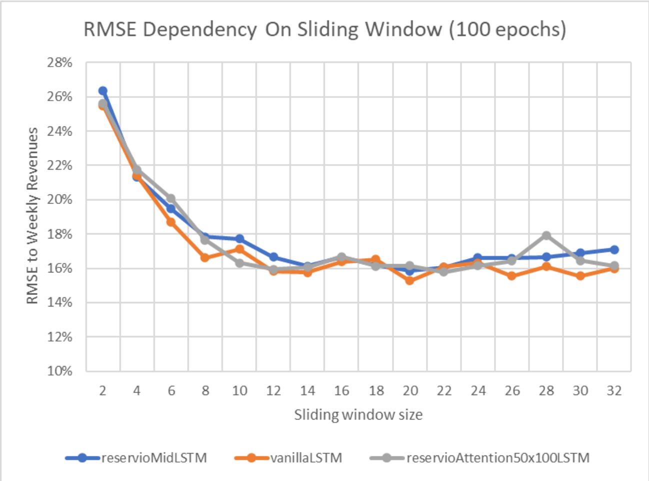

Fig. 3: RMSE dependency On Sliding Window

The Impact of Dataset Size to RMSE with 16 Weeks Window

Regarding the *RMSE-to-revenues* and its relation to the size of the dataset with 16 weeks window, Fig. 4 shows the results.

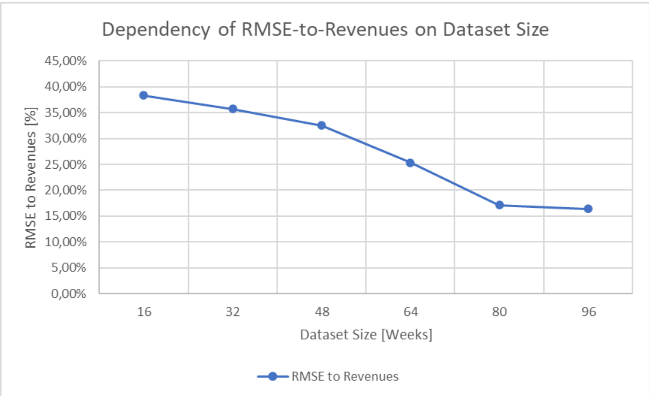

Fig. 4: *RMSE-to-Revenues* Dependency on the Dataset Size

Experiments showed that beyond a dataset size of 80 weeks, the *RMSE* did not significantly decrease. However, it is still possible to expect further improvement, as the experiment did not have enough data to reach the plateau of this curve.

H. *Costs of Revenues Predicion*

The operating costs of a neural network predicting demand through future revenues are far cheaper compared to classifiers. The advantage lies primarily in the fact that predictions do not need to be computed for a large number of customers but for the number of businesses serving these customers, which is a fraction of the customers value. For 500k customers the costs are expected as low as 4,975 CZK yearly, unlike the *XGBoost* classifier, which is estimated to cost 198,365 CZK yearly (Amazon f1.2xlarge instance with the price of $1,73 per hour of machine time).

V. Automated Marketing Synthesis

The resulting artificial intelligence combines the *LSTM* neural network with the *binary classifier* in the following scheme:

1. Detection of fluctuations in the future revenues of individual entrepreneurs using *LSTM* neural network.
2. Evaluate future revenues and look for drop below 85% of the average sales over the last 16 weeks.
3. If the drop below 85% is detected, activate the classifier and compute the demand as follows:

$$D = TP + FN$$

Demand is represented by the number of customers in the positive class + *false negatives* for the given week, which can also be computed as:

$$D = C \cdot Precision + \frac{C \cdot Precision}{Recall} - C \cdot Precision$$

$$D = C \cdot \frac{Precision}{Recall}$$

The production capacity of a business (denoted as *K*) refers to the maximum number of customers that can be served within a given time period (week). Then, the anticipated utilization *U* of the business in a given period computes as:

$$U = \frac{D}{K} \cdot 100\ [\%]$$

Once *U* is known, missing demand *MD* is computed as follows:

$$MD = K \cdot \left(1 - \frac{D}{K}\right)$$

MD represents the number of customers which should be acquired by the marketing tools. Hence, marketing automation directly addresses a specific number of existing customers and sends them a discount offer for the given week. The number of targeted customers depends on the conversion ratio (the response) to this type of campaign for the respective business, typically ranging up to 10%. Except of direct marketing, the automation can switch the previously configured online advertising campaigns to running state. *MD* can be evaluated every day and once $MD \rightarrow 0$ for all fluctuations detected, online ad campaigns should be automatically stopped.

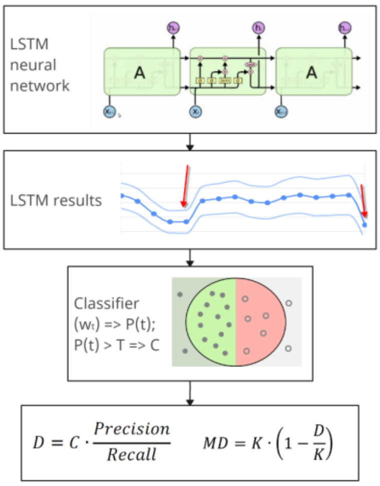

Fig. 4: Combined AI for Automated Marketing

The customers who will receive the sale offer will be taken from the negative class of the classifier result sorted by the reservation probability in the descending order. It is not desirable to invest in motivation in any member of the positive class because they will almost certainly make the purchase in the given week.

VI. Conclusion

In conclusion, the integration of time-series analysis and a binary classifier presents a robust and highly effective solution for predicting and managing future demand in the service sector. The proposed approach not only enhances the efficiency of marketing automation but also contributes significantly to the optimization of labor, capital utilization, and reduction of marketing expenses for entrepreneurs in the discussed industries.

The findings of the study indicate that the binary classifier employed in the model achieves the accuracy of 90%. This high level of accuracy ensures that the identified demand fluctuations are reliably classified, minimizing both *false positives* and *false negatives* while maximizing *true positives* by using the proposed *Miss Rate* metric.

The time-series analysis component of the model demonstrates good predictive capabilities, with a *Root Mean Square Error* (*RMSE*) as low as 15% of the weekly revenues of the business over a 16-week future period. This level of accuracy in predicting future revenues provides entrepreneurs with a valuable tool for proactive decision-making in response to demand fluctuations. The low *RMSE* underscores the reliability of the model in capturing and forecasting the nuances of weekly revenue patterns, enabling businesses to plan and allocate resources with a high degree of confidence.

In essence, the combination of a precise time-series analysis and highly accurate binary classifier positions the proposed artificial intelligence model as a powerful tool for businesses aiming to navigate the dynamic landscape of service sector demand. The demonstrated accuracy and reliability make this approach a practical and impactful solution for optimization of business operations and driving automated cost-effective marketing strategies.

Acknowledgment

The article was based on the results of the project titled *Research and Development of Artificial Intelligence Increasing Economic Efficiency in Service Business,* reg. no. *CZ.01.1.02/0.0/0.0/20_321/ 0024529*, under the *Applications program*, with the supervising authority being the Ministry of Industry and Trade of the Czech Republic. The project was co-financed by the European Union.

References

[1] Cao, J., Jiang. Z., Wang, K. Customer demand prediction of service-oriented manufacturing incorporating customer satisfaction. In International Journal of Production Research, vol. 54, issue 5. ISSN 0020-7543. 2016. IF 9,2.

[2] Wang, D., Cao, W., Li, J., Ye, J. DeepSD: Supply-Demand Prediction for Online Car-Hailing Services Using Deep Neural Networks. In proceedings of 2017 IEEE 33rd International Conference on Data Engineering (ICDE), San Diego, USA, 2017, pp 243-254.

[3] W. Li et al., A General Framework for Unmet Demand Prediction in On-Demand Transport Services. In IEEE Transactions on Intelligent Transportation Systems, vol. 20, no. 8, pp. 2820-2830, Aug. 2019.

[4] Li, W., Cao J., Guan, J., Zhou, S., Liang, G., So, W., Szczecinski, M. A General Framework for Unmet Demand Prediction in On-Demand Transport Services. In IEEE Transactions on Intelligent Transportation Systems. ISSN 0305-215X, 2017.

[5] P. Mirowski, S. Chen, T. K. Ho and C. Yu, Demand forecasting in smart grids. In Bell Labs Technical Journal, vol. 18, no. 4, pp. 135-158, March 2014.

[6] Kochak, A., Sharma, M. Demand Forcasting using Neural Network for Supply Chain Management. In International Journal of Merchanical Engineering and Robotic Research.

[7] Berka, P. Dobývání znalostí z databází. Book, 368 p. Academia, ISBN 8020010629, 2005.

[8] Olson, D. L., Delen, D.: Advanced Data Mining Techniques, Springer, 1st edition (February 1, 2008), page 138, ISBN 3-540-76916-1.

[9] Chung, I.H., Williams, D.W., Do, M.R. For Better or Worse? Revenue Forecasting with Machine Learning Approaches. In Public Performance & Management Review (journal, IF 3.1), vol. 45 issue 5. ISSN 1530-9576, 2022.

[10] Pereira L.N., Cerquiera, V. Forecasting hotel demand for revenue management using machine learning regression methods. In Current Issues in Tourism (journal, IF 8.0), ISSN 1368-3500, 2022.

[11] Su, Y., Backlund, P., Engröm H. Data-driven method for mobile game publishing revenue forecast. In Service Oriented Computing and Applications (IF 1.3). ISSN 1863-2394. 2022.

[12] Qin, Y., Song, D., Chen, H., Cheng, W., Jiang, G., Cottrell, G.W. A Dual-Stage Attention-Based Recurrent Neural Network for Time Series Prediction. Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI-17). 2017.

[13] Li, Y., Yu, R., Shahabi, C., Liu, Y. Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting. 6th International Conference on Learning Representations, 2018.

[14] Gamboa, J. Deep Learning for Time-Series Analysis. University of Kaiserslautern, Germany, 2017.

[15] Harvey, A. C. ARIMA models. In Time Series and Statistics, London:Palgrave Macmillan UK, pp. 22-24, 1990, ISBN 978–1-349-20865-4.

[16] Brownlee, J. Deep Learning for Time Series Forecasting. Machine Learning Mastery Publishing. 2018.