Avrupa Bilim ve Teknoloji Dergisi
Sayı 51, S. 63-74, Ağustos 2023

Araştırma Makalesi

www.ejosat.com ISSN:2148-2683

European Journal of Science and Technology
No. 51, pp. 63-74, August 2023

Research Article

A Study of Predicting Arrival Patterns of Airport Passengers to the Counters on the Basis of International Terminal

Merve Gözde Sayın[1*], Kami Serdar Üçkardeş[2], Doruk Yarkın Aktaş[3], Mustafa Bolat[4], Murat Kerem Çelenli[5], Boran Dursun[6], Gökhan Koç[7]

[1*] Tav Technologies, İstanbul, Türkiye, (ORCID: 0000-0002-6213-2549), gozde.sayin@tav.aero
[2] Tav Technologies, İstanbul, Türkiye (ORCID: 0000-0002-0955-1170), kami.uckardes@gmail.com
[3] Tav Technologies, İstanbul, Türkiye (ORCID: 0000-0002-7647-1218), 15dorukyarkinaktas15@gmail.com
[4] Tav Technologies, İstanbul, Türkiye (ORCID: 0000-0001-8169-0629), mustafa.bolat@tav.aero
[5] Tav Technologies, İstanbul, Türkiye (ORCID: 0000-0002-6655-9144), murat.celenli@tav.aero
[6] Tav Technologies, İstanbul, Türkiye (ORCID: 0000-0003-0079-8588), borandursun59@gmail.com
[7] Tav Technologies, İstanbul, Türkiye (ORCID: 0000-0001-7433-2356), gokhan.koc@tav.aero

(İlk Geliş Tarihi 6 Aralık 2022 ve Kabul Tarihi 25 Mart 2023)

(**DOI:** 10.31590/ejosat.1214786)

ATIF/REFERENCE: Sayın, M.G., Üçkardeş, K.S., Aktaş, D.Y., Bolat, M., Çelenli, M.K., Dursun, B. & Koç, G. (2023). A Study Of Predicting Arrival Patterns Of Airport Passengers To The Counters On The Basis Of International Terminal. *Avrupa Bilim ve Teknoloji Dergisi*, (51), 63-74.

Abstract

AI-based passenger arrival predictions to the processing points are essential to ensure efficient management of many intertwined operational processes in the airport ecosystem. For example, to be able to analyze the number of ground service personnel that will be required in the following hours, days in different parts of the airport and for different types of operations, it is essential to predict how many passengers will come to the airport in the following time zones. Moreover, density-driven intelligent energy management and dynamic price offering options in different services could only be generated with accurate passenger arrival predictions. Passenger arrivals can be detected with various technologies such as computer vision, IoT, lidar, and radar. However, passenger boarding pass printing event messages from the CUPPS solution, which is implemented in İzmir Adnan Menderes Airport International Terminal, is used as the data source in this study. Also, Linear regression, FEDOT, LSTM, and hybrid methods are configured and compared to predict passenger arrival counts to the counters of the international terminal in the specified time slots

Keywords: Passenger Arrival, Pattern, LSTM, FEDOT, Regression, Prediction.

Havalimanı Yolcularının Uluslararası Terminal Bazında Kontuarlara Geliş Paternlerini Tahminleme Çalışması

Öz

Havalimanı ekosisteminde iç içe geçmiş birçok operasyonel sürecin verimli yönetimini sağlayabilmek için yolcu işlem noktalarındaki yolcu varış miktarlarını yapay zeka tabanlı sistemlerle tahmin edebilmek çok önemlidir. Örneğin, havalimanının farklı bölümlerinde ve farklı operasyon türleri için ilerleyen saatlerde, günlerde ihtiyaç duyulacak yer hizmet personeli sayısını analiz edebilmek için havalimanına kaç yolcunun geleceğini tahmin edebilmek gerekmektedir. Ayrıca, yoğunluğa dayalı akıllı enerji yönetimi ve farklı hizmetlerde dinamik fiyat teklifi seçenekleri ancak doğru yolcu varış tahminleri ile oluşturulabilir. Günümüzde, ilgi tahminlemenin temelini oluşturan veri havuzu bilgisayarlı görü, IoT, lidar, radar gibi çeşitli teknolojilerle beslenebilmektedir ama bu çalışmada İzmir Adnan Menderes Havalimanı Dış Hatlar Terminali'nde kullanılan CUPPS çözümü ile yazıcılara gönderilen yolcu biniş kartı basma

* Sorumlu Yazar: gozde.sayin@tav.aero

mesajları veri kaynağı olarak kullanılmıştır. Ayrıca Lineer regresyon, FEDOT, LSTM ve hibrit yöntemler konfigüre edilerek dış hat terminali bazında kaç adet yolcunun belirli zaman aralığında kontuarlara varacağını tahmin eden modeller geliştirilmiş ve birbirleri ile karşılaştırılmıştır.

Anahtar Kelimeler: Yolcu Varış, Patern, LSTM, FEDOT, Regresyon, Tahminleme.

1. Introduction

Airports are complex transportation centers where multiple stakeholders run many different operations either sequentially or simultaneously in a chain structure. Due to capacity constraints, passenger transaction congestion and flight delays can occur in this complex ecosystem, especially at large airports, and peak times, and managing passenger queues becomes difficult [1][2][3]. Congestion and related operational disruptions harm both passenger satisfaction and operating costs [4][5]. As a result of Takakuwa's Research, it is seen that international terminal passengers of Kansai Airport spends 25% of their time in waiting queues [22]. Also, according to Lin's research, even occupancy- forecast-based control could generate 18 to 30% energy saving if accurate passenger flow predictions achieved [15].

To be able to plan airport passenger processes in an efficient way, it is mandatory to know how many passengers will arrive at the airport at certain time intervals. Using this information, airports simulate passenger flows inside the terminal with different resource allocation combinations [6][7]. Yet, passenger arrival distribution is defined as stochastic in many types of research because each passenger's behavior changes depending on the type of passenger, time of day, seasonal factors, business or leisure type of passenger, full-fare, or special-fare, international or domestic, origin, destination, etc. [8][9]. Also, check-in rules of airports and countries are considered crucial factors that effects passenger arrival pattern in the terminal [11].

In M.N. Postorino's research, arrival behaviors are analyzed by clustering passengers by flight carrier types (Low-cost, Full), departure time of the day. According to the results of this research shows that in the early morning period, almost 70% of both carrier passengers arrive 60-90 min before their flights [10]. However, in the late afternoon period arrival distribution of low-cost carrier passengers spread over a greater period. Also, according to Alodhaibi's research, passenger arrival pattern in the international terminals is like most of the airports such as: 90% passengers arrive 60 minutes and peak hours are placed between 100-120 minutes (about 2 hours) before departure of the flights. Moreover, leisure passengers are coming earlier than business passengers. Last but not least, the peaks are shorter but busier in the mornings [12][13][14].

During the study, regression, LSTM, FEDOT and hybrit methods are developed and compared to achieve the best prediction accuary for passenger arrival patterns for İzmir Adnan Menderes Airport International Terminal.

2. State of the Art

The source, amount, and content of the data used in predicting passenger arrival behavior are significant for the success of the prediction models. For example, in some research, Bar Coded boarding pass printing events are consumed from check-in workstations and used to understand passenger patterns [10]. For this type of research, generating flight-specific predictions is achievable because each passenger data is collected with the attached flight information. Also, image recognition, lidar based solutions, RF, Wi-Fi, Bluetooth signal trackers, infrared, acoustic sensors are being used for people counting in much research [16][17][18]. However, with such solutions, it is not possible to separate specific flight passengers from other flights passengers.

Especially after 2010, the number of machine learning and deep learning-based passenger arrival pattern prediction and comparison studies has increased rapidly in the literature. For example, in the Monmousseau's article, three different models which are MSE loss trained LSTM200 architecture, 0.5- PMSE loss trained LSTM200 architecture and MSE loss trained Random Forest are tested and compared [19]. As a result, it is seen that LSTM models performed better in most of the cases, but Random Forest model generates better predictions for peak times. In another research, SVR, LSTM, hybrid version of SVR and LSTM, ARIMA and Fusion-KNN are compared for the prediction of passenger arrivals and hybrid version of SVR, and LSTM performed better for most of the cases [20]. Moreover, another model called SARIMA proposed and tested in Kunming International Airport for passenger flow prediction and achieved average error between 1% and 3% for short term predictions [21].

Most airports struggle to maintain a balance between demand and capacity given the constrained availability of infrastructural facilities and staff. Cost, operational effectiveness, and customer happiness are the three key aspects that need to be taken into account while balancing demand and capacity [23][24]. The crucial thing about predicting the number of arriving passengers to airport is the knowledge of arrival distribution, especially in smaller time intervals. Just as everything unplanned leads to chaos, the situation is similar for airports, so planned departure times also play a significant role [25][26]. According to Postorinoa's [26] study, several passengers show up at the security checkpoint at least two hours before their departure. Nevertheless, it depends on many features such as air company (Full carrier or low-cost carrier), access issues, airport facilities and shopping. Besides, Kulunu [24] said that baggage and security screening and check-in processes are the most critical procedures at the airports because they cause extra waiting times, long queues, and delays. They used a discrete event simulation (DES) to overcome these problems and to observe scenarios for passenger flow. D. Olaru and S. Emery [27] stated that Holland [28] and Davis [29] developed GA to mimic the passenger flow process in evolution and natural selection. Furthermore, they found that the seasonality and the day of the week plays a significant role in passenger flow. On the other hand, the space, number of check-in desks, and the number of security screening lines are the limitations. According to their model, more check-in counters and two screening lines might be the solution for the crowd.

3. Data Collection

The arrival times of the passengers at the counters were determined using the CUPPS system of İzmir Adnan Menderes Airport. Common Use Passenger Processing System (CUPPS) is a globally accepted standard introduced by the IATA, which describes the range of services, specifications, and standards to enable multiple airlines to share the same physical check-in or gate workstations simultaneously or consecutively. Each airline issues boarding passes containing IATA standardizing a barcode field to its passengers using CUPPS system workstations. Thus, within the scope of the study, the messages sent to the printers were parsed, and the arrival times of the passengers were determined as it is shown in Figure 1.

Figure 1. Parsed Boarding Pass Message Flow

According to the IATA standard; "Flight Date", "Origin Airport", "Destination Airport", "Carrier IATA" and "Flight Number" fields taken from the boarding pass barcode meta data are the fields used together with for the uniqueness of the flight. For the uniqueness of the passenger at tracking phase, it is sufficient to process the "Check-in Sequence Number" together with the flight uniqueness. All other data in the parsed boarding pass metadata were not used in this study because of the limits of GDPR.

4. Model Structures

During the study, linear regression, LSTM, FEDOT and hybrid version of LSTM and FEDOT methods are used, configured and compared to predict passenger arrival patterns to the checkin counters.

4.1. Linear Regression Model

To generate a successful prediction model for the defined problem, all flight data that can be reached before the day to be predicted has been analyzed. As a result, it is defined that only STD (Scheduled Time of Departure) and AC Type (Aircraft Type of Flights) could be known before the operation day. Therefore, entire regression model is built on these two parameters. In order to predict the number of passengers arriving at the terminal counters for half-hour intervals, various inputs have been created according to the beginning of the relevant time interval as its shown in Figure 2.

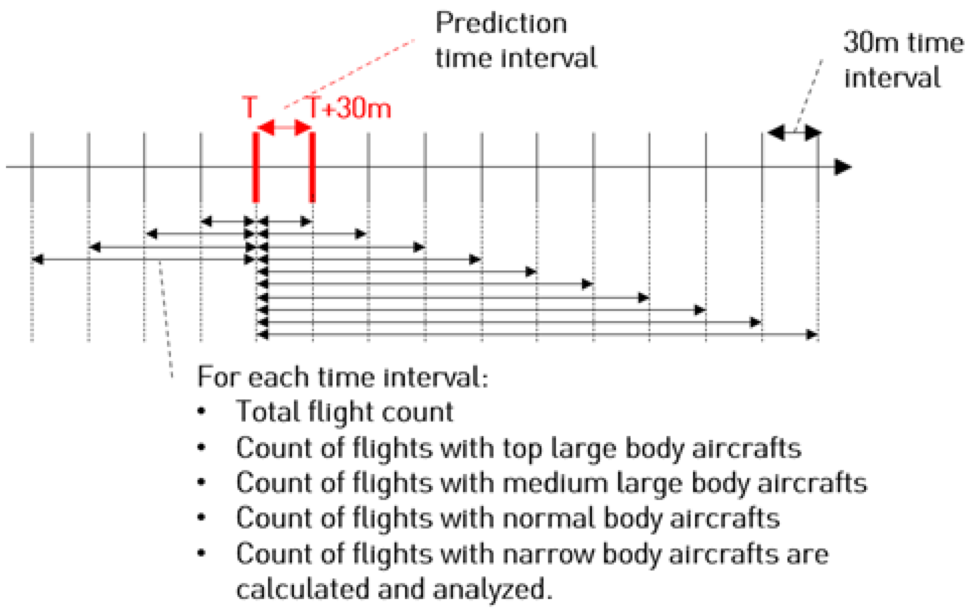

Figure 2. Analyzed regression inputs

As a result of possible input analysis, with 4 month sized operation data, only a few of them defined usefull for the regression model as it is shown in Figure 3.

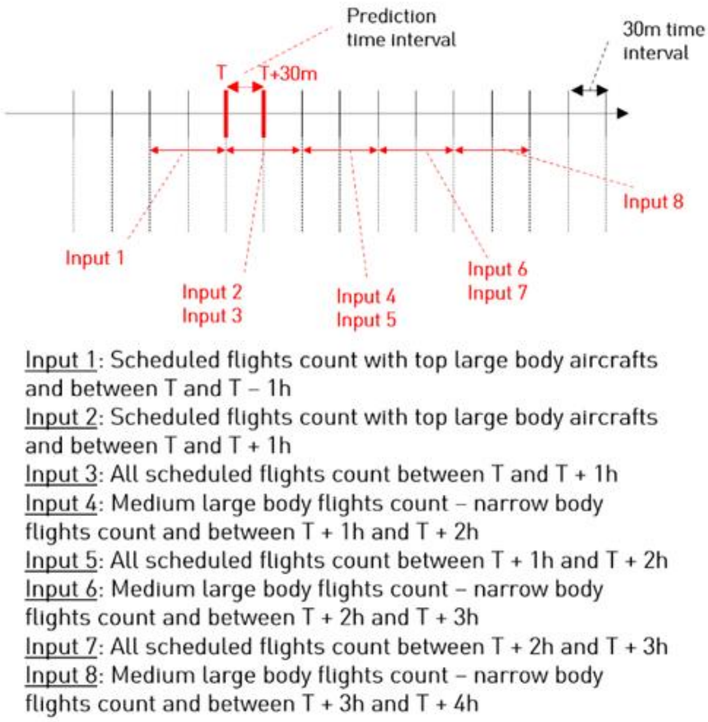

Figure 3. Succesfull regression model inputs

Regression model is developed and tested by 4 month sized operational data and Adjusted R Square is measured 0.843 for the real arrival counts at which is higher than 20 for defined 30 minute time-intervals. At the end of the study, performance of generated regression model is measured and compared with other models for new operation days. At Table 1. Generated regression model input parameters are shared in detail.

Table 1. Input parameters of generated regression model

	Coef.	Std. Err.	T Stat	p-Value

Input 1	-18.77	8.27	-2.27	0.02
Input 2	16.18	9.35	1.73	0.08
Input 3	-1.47	0.52	-2.80	0.01
Input 4	-8.37	5.51	-1.52	0.13
Input 5	19.47	0.47	41.61	0.00
Input 6	-6.50	5.97	-1.09	0.28
Input 7	21.76	0.39	55.71	0.00
Input 8	-9.02	6.87	-1.31	0.19

4.2. NN-Based Models

International passenger check-in records between 23.09.2021 and 19.07.2022 were extracted from DB. It was analyzed whether there was a problem in the format of the data and data flow, then the data were divided into 30-minute groups, according to the date column, and counted. The data was visualized to better observe the number of passengers in the grouped data and to check for outliers and missing data. Appropriate input sequence length determination plays a really important role when it comes to model performance. In this study, 48 (daily) and 48*7 (weekly) input sequence lengths were tried, and continued to work with 48*7 (weekly), which performed better. One of the purposes of scaling is to ensure that the effect of one of the features is not more than the other by bringing the inputs from different ranges into a single range. (Contribute equally to the output) There are studies showing that scaling improves model performance in the neural net as well, gradient descent converges much faster with feature scaling than without it is the reason for this improvement and that's why we scaled the data and saved the scaler as a pickle. The scaled array was later reshaped and made suitable for study afterward.

The first 75% of the 14,367 records are reserved for training and the remaining 25% for testing. Using the relevant module of the Keras library (TimeSeriesGenerator), sequence instances were created for both train and test data, afterward (336 (48*7) inputs and 1 output).

4.2.1. Simple RNN

A Recurrent Neural Network (RNN) is a type of artificial neural network, which can make predictions with sequential input values. An RNN has a memory that allows it to catch the pattern of the previous data and make predictions based on them. An RNN model with the following structure was created as it shown in Table 2 and parameters are shared in Table 3.

Table 2. RNN Structure

Layer (type)	**Output Shape**	**Param #**
simple_rnn_1 (SimpleRNN)	(none, 64)	4224
dense_6 (Dense)	(none, 32)	2080
dense_7 (Dense)	(none, 1)	33

Table 3. RNN parameters

Parameter Name	**Parameter Value**
Activation Function	ReLU
Loss Function	Mean Squared Error
Optimizer	Adam
Learning Rate	0.001
# of Epochs	51

The model was trained with the train generator and its performance on the test set was observed with the test generator. Also, a checkpoint was created to save only the best model.

4.2.2. Simple LSTM

Some disadvantages of the simple RNN are that; underperforms with longer sequences and is prone to vanishing or exploding gradients. Although vanishing and exploding gradients were not a problem at this stage because the models are not that complex and deep, a study was also performed with LSTM (Long-Short Term Memory) due to the long input sequence. An LSTM model with the following structure was created as it is shown in Table 4 and parameters are shared in Table 5.

Table 4. Simple LSTM Structure

Layer (type)	Output Shape	Param #
lstm (LSTM)	(none, 50)	10400
dense_4 (Dense)	(none, 32)	1632
dense_5 (Dense)	(none, 1)	33

Table 3. Simple LSTM parameters

Parameter Name	Parameter Value
Activation Function	ReLU
Loss Function	Mean Squared Error
Optimizer	Adam
Learning Rate	0.001
# of Epochs	8

The model was trained with the train generator and its performance on the test set was observed with the test generator. Also, a checkpoint was created to save only the best model.

4.2.3. Bidirectional LSTM

Unlike UniLSTM (Unidirectional LSTM), BiLSTM (Bidirectional LSTM) has a flow to both sides (both past to future and future to past) and this flow allows for a better understanding of the context. The bidirectional layer, which provides this both-sided flow, doubles the number of LSTM nodes, increasing the number of nodes from 50 in the base model (UniLSTM) to 100 in this model (BiLSTM). This structure is also frequently used in NLP (Natural Language Processing) studies. A BiLSTM model with the following structure was created as it is shown in Table 6 and parameters are shared in Table 7.

Table 6. Bidirectional LSTM Structure

Layer (type)	Output Shape	Param #
Bidirectional_lstm (Bidirectional LSTM)	(none, 100)	20800
dense_4 (Dense)	(none, 32)	3232
dense_5 (Dense)	(none, 1)	33

Table 7. Bidirectional LSTM parameters

Parameter Name	Parameter Value
Activation Function	ReLU
Loss Function	Mean Squared Error
Optimizer	Adam
Learning Rate	0.001
# of Epochs	12

The model was trained with the train generator and its performance on the test set was observed with the test generator. Also, a checkpoint was created to save only the best model.

4.2.4. Simple LSTM encoder-decoder Model

Although one step forecasting study was carried out, not multi-step time forecasting, a model was also established with LSTM with encoder-decoder, which is used quite frequently in sequence to sequence (seq2seq) studies. The encoder encodes the input sequence and the decoder decodes the encoded sequence and predicts output for each output sequence. LSTM is consisting of 50 layers were used in both the encoder and decoder parts of the model, and the repeat vector was used to repeat and shape the inputs. An LSTM (Encoder-Decoder) model with the following structure was created as it is shown in Table 8 and parameters are shared in Table 9.

Table 8. Simple LSTM encoder-decoder model structure

Layer (type)	**Output Shape**	**Param #**
Lstm_29 (LSTM)	(none, 50)	10400
Repeat_vector_7 (RepeatVector)	(none,1, 50)	0
lstm_30 (LSTM)	(none, 1, 50)	20200
Time_distributed_12	(none, 1, 1)	51

Table 9. Simple LSTM encoder-decoder model parameters

Parameter Name	**Parameter Value**
Activation Function	ReLU
Loss Function	Mean Squared Error
Optimizer	Adam
Learning Rate	0.001
# of Epochs	10

The model was trained with the train generator and its performance on the test set was observed with the test generator. Also, a checkpoint was created to save only the best model.

4.2.5. Comparison of 4 different NN-based models

The performance of the best-performing model among the 4 different models was tried to be improved by testing different parameter values, adding/removing different layers, and changing the number of nodes in the layers of the model. Performance metrics tracked mainly in the study are training time, r squared, and RMSE.

As a result of the test process, a 2-BiLSTM layered structure was created as it is shown in Table 10. 25 units and the ReLU activation function were used in the first LSTM layer, and 5 units and the tanh activation function were used in the second LSTM layer as it is shown in Table 11. Since the training time of the model was relatively long (Each epoch took appx. 45 mins.), 5 epochs were run. Although the number of epochs is low compared to other models, this model has the best performance and is currently used.

Table 10. Best performed model structure

Layer (type)	**Output Shape**	**Param #**
bidirectional_11	(none, 336, 50)	5400
bidirectional_12	(none, 10)	2240
Dense_16 (Dense)	(none, 32)	352
Dense_17 (Dense)	(none, 1)	33

Table 11. Best performed model parameters

Parameter Name	**Parameter Value**
Activation Function	ReLU/Tanh
Loss Function	Mean Squared Error
Optimizer	Adam
Learning Rate	0.001
# of Epochs	5

Comparison results of generated NN-based models are shared in Table 12.

Table 12. Comparison results of tested NN-based models

Model (Structure)	R Squared	RMSE
RNN	0.64	59.91
LSTM	0.65	58.84
LSTM (Encoder-Decoder)	0.66	57.91
BiLSTM	0.81	43.16
2-layered BiLSTM	0.82	42.81

4.3. FEDOT Models

The grouped records of the previous study (NN) were saved as a CSV and used in this study. Unlike the neural networks, forecast length is determined in Fedot, instead of input sequence length. Just as input sequence length determination is of critical importance in neural networks, forecast length determination has the same importance in Fedot. In this study, different forecast lengths were tried (48-daily, 336 weekly) and the study was completed with the length showing the best performance (48). The last 48 records were collected as test data and the remaining records as train data. 14319 data in training set and 48 data as test set defined. It is possible to accomplish regression, clustering, classification, and time series forecasting tasks with Fedot. After selecting the task in time series forecasting, defining the forecast length, and preparing the train and test sets, the initial configuration was completed.

After the basic configuration was completed, a simple pipeline consisting of two nodes was created to be run with default parameters and its performance was observed. The structure of the pipeline is shown in Figure 4.

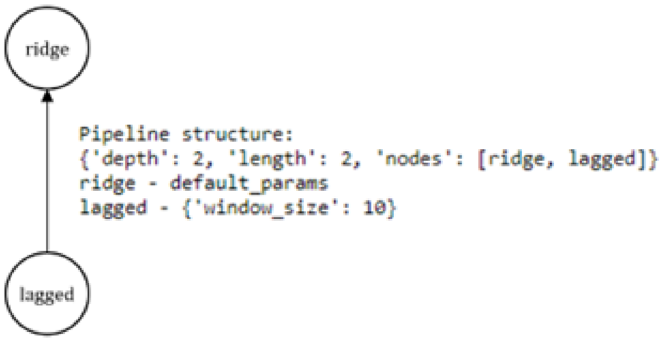

Figure 4. Initial FEDOT pipeline structure

Since the basic pipeline performance was not sufficient, first the window size parameter of the lagged layer was increased to 336, which is the neural network input sequence length as it is shown in Figure 5.

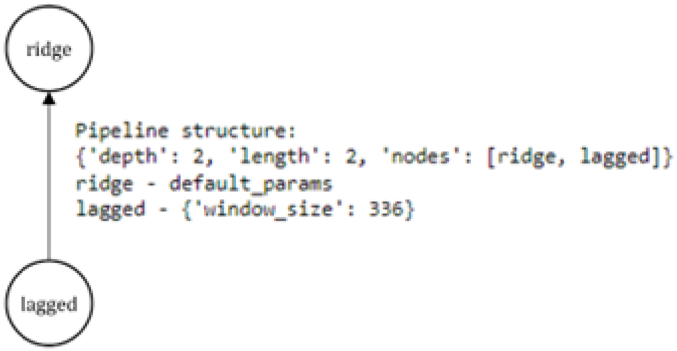

Figure 5. Second generated FEDOT pipeline structure

In another study, a relatively complex model consisting of two lagged layers was created. The window size of one of the lagged layers is set to 336 (48*7 / weekly) and the other to 1440 (48*30 /monthly). In the layer after the lagged layer different layers such as; knnreg, ridge, linear, lasso, etc. were tested and continued with the best performing structure. The thirdly created pipelines' structure is shown in Figure 6.

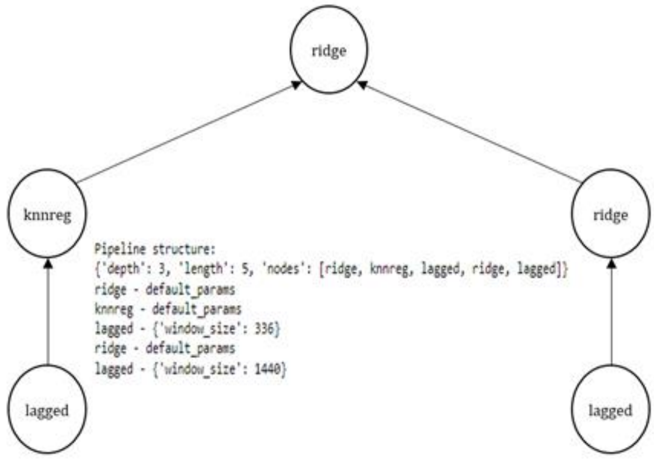

Figure 6. Third generated FEDOT pipeline structure

In the last study, a relatively more complex model consisting of 3 lagged layers was created. In this model, the window size of the lagged layers was determined as 48 (daily), 336 (48*7 / weekly), and 1440 (48*30 / monthly), respectively. In the layer after the lagged layer different layers such as; knnreg, ridge, linear, lasso, etc. were tested and continued with the best performing structure. The final manually created pipelines' structure is shown in Figure 7.

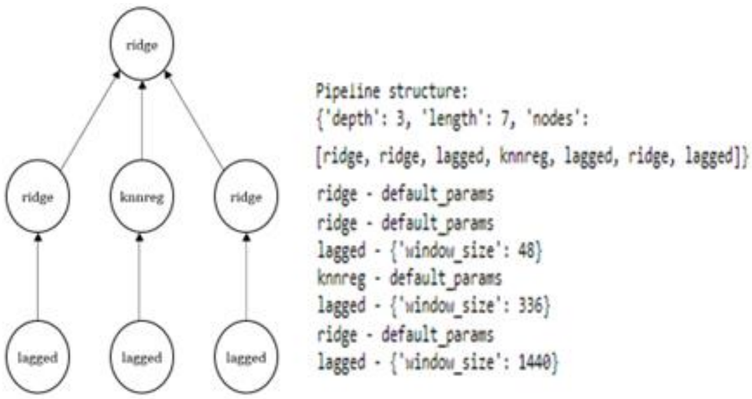

Figure 7. Final generated FEDOT pipeline structure

After creating and testing different simple and complex models, the parameter tuning phase was started with the model with the best performance. The basic pipeline consisting of 2 nodes (Lagged and ridge) showed the best performance in 4 different models, one of which was initial. Therefore, the parameter tuning phase was made for this model. Two different parameter tuning studies were carried out, one of which is a tuning study in which the two-node structure is preserved, and the layers are changed, and the other is a study in which both the two-node structure and the relevant layers are preserved and the values of these layers are tuned.

At the end of the study, when the structure and layers were preserved and only layer values tuned, there was a relative improvement in the model performance and the study was completed with this improved structure and saved for use in future studies. Final model's structure is shared in Figure 8.

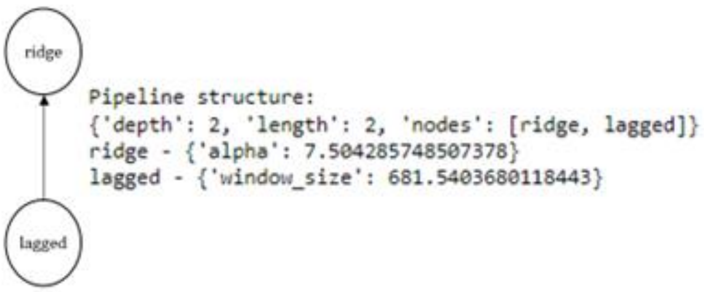

Figure 8. Best performed FEDOT pipeline structure

Comparison results of generated FEDOT models are shared in Table 13.

Table 13. Comparison results of tested NN-based models

Model (Structure)	R Squared	RMSE
Initial Pipeline	0.1	99.21
Basic Pipeline(Length=2)	0.87	37.85
Complex Pipeline (Length=5)	0.82	44.84
Complex Pipeline 2 (Length=7)	0.82	44.57
Tuned Pipeline	0.88	36.08

5. Prediction Models Performance Comparisons

At the end of the study, best configured linear regression, LSTM, FEDOT and hybrid version of LSTM and FEDOT models are tested and compared with a new operational data (246 different 30 minute sized time interval).

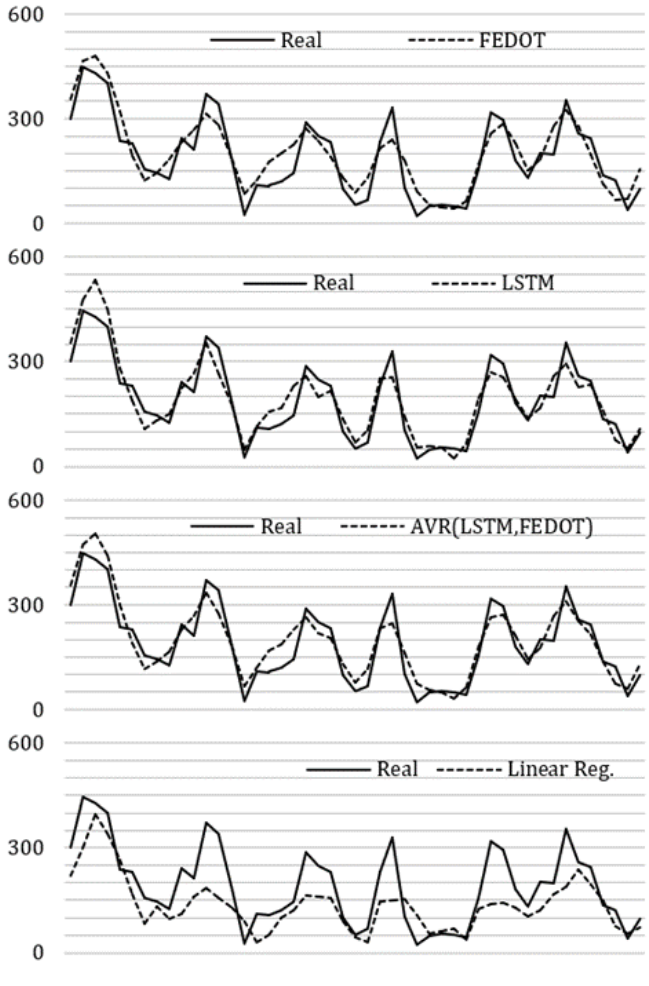

Figure 9. Predictions results with different models (1 operation day)

As it is seen in Figure 9 and Table 14, all models except regression generates accurate predictions for passenger check-in arrival counts of international terminal.

Table 14. Model performances and comparison

Model	R Squared
Best FEDOT	0.79
Best LSTM	0.86
AVR of best FEDOT and best LSTM	0.85
Linear Regression	0.40

6. Conclusion

As a result of the study it is seen that LSTM is the most succussful method on defined prediction scenarios. However, linear regression model performans results are way much below than the expectations. When the reason for the linear regression results to be so inconsistent was investigated, it was seen that the 4-month flight data used for the model coincided with the winter season, but the prediction tests were performed for the summer season flights. Also, in summer season, flights occupancies measured higher than winter season flights. The last reason for the difference was interpreted as the increase in the rate of charter flights in the summer season. Considering all these aspects together, using linear regression for passenger arrival pattern prediction at airports that do not have homogeneous flight processes may cause a significant decrease in prediction consistency at different times of the year.

6. Acknowledge

Gökhan Koç and Mustafa Bolat have researched, developed and tested LSTM and FEDOT based models together. Boran Dursun, Doruk Yarkın Aktaş, Murat Kerem Çelenli and Merve Gözde Sayın have worked on literature research about existing methods and outputs. Also they have generated several linear regression models together. Kami Serdar Üçkardeş has worked on data transmission architecture and also aggregated boarding pass data according to the needs of the study.

References

1. De Neufville, R. (2016). Airport systems planning and design. In Air Transport Management (pp. 89-106). Routledge.
2. Park, Y., & Ahn, S. B. (2003). Optimal assignment for check-in counters based on passenger arrival behaviour at an airport. Transportation Planning and Technology, 26(5), 397-416.
3. Fayez, M. S., Kaylani, A., Cope, D., Rychlik, N., & Mollaghasemi, M. (2008). Managing airport operations using simulation. Journal of Simulation, 2(1), 41-52.
4. Martín-Cejas, R. R. (2006). Tourism service quality begins at the airport. Tourism Management, 27(5), 874-877.
5. Kirschenbaum, A. A. (2013). The cost of airport security: The passenger dilemma. Journal of Air Transport Management, 30, 39-45.
6. Bevilacqua, M., & Ciarapica, F. E. (2010, December). Analysis of check-in procedure using simulation: a case study. In 2010 IEEE International Conference on Industrial Engineering and Engineering Management (pp. 1621-1625). IEEE.
7. Kalakou, S., Psaraki-Kalouptsidi, V., & Moura, F. (2015). Future airport terminals: New technologies promise capacity gains. Journal of Air Transport Management, 42, 203-212.
8. van Boekhold, J., Faghri, A., & Li, M. (2014). Evaluating security screening checkpoints for domestic flights using a general microscopic simulation model. Journal of Transportation Security, 7(1), 45-67.
9. Ashford, N. J., Stanton, H. M., Moore, C. A., AAE, P. C., & Beasley, J. R. (2013). Airport operations. McGraw-Hill Education.
10. Postorino, M. N., Mantecchini, L., Malandri, C., & Paganelli, F. (2019). Airport passenger arrival process: Estimation of earliness arrival functions. Transportation Research Procedia, 37, 338-345.
11. Manataki, I. E., & Zografos, K. G. (2009). A generic system dynamics based tool for airport terminal performance analysis. Transportation Research Part C: Emerging Technologies, 17(4), 428-443.
12. Ashford, N. J., Mumayiz, S., & Wright, P. H. (2011). Airport engineering: planning, design, and development of 21st century airports. John Wiley & Sons.
13. Cheng, L. (2014). Modelling airport passenger group dynamics using an agent-based method (Doctoral dissertation, Queensland University of Technology).
14. Alodhaibi, S., Burdett, R. L., & Yarlagadda, P. K. (2019). Impact of passenger-arrival patterns in outbound processes of airports. Procedia Manufacturing, 30, 323-330.
15. Jin, Y., Yan, D., Chong, A., Dong, B., & An, J. (2021). Building occupancy forecasting: A systematical and critical review. Energy and Buildings, 251, 111345.
16. Schreiber, D., & Rauter, M. (2012, June). A CPU-GPU hybrid people counting system for real-world airport scenarios using arbitrary oblique view cameras. In 2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (pp. 83-88). IEEE.
17. Mizutani, M., Uchiyama, A., Murakami, T., Abeysekera, H., & Higashino, T. (2020, March). Towards people counting using Wi-Fi CSI of mobile devices. In 2020 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops) (pp. 1-6). IEEE.
18. Kouyoumdjieva, S. T., Danielis, P., & Karlsson, G. (2019). Survey of non-image-based approaches for counting people. IEEE Communications Surveys & Tutorials, 22(2), 1305-1336.
19. Monmousseau, P., Jarry, G., Bertosio, F., Delahaye, D., & Houalla, M. (2020, February). Predicting passenger flow at Charles de Gaulle airport security checkpoints. In 2020 International Conference on Artificial Intelligence and Data Analytics for Air Transportation (AIDA-AT) (pp. 1-9). IEEE.
20. Guo, J., Xie, Z., Qin, Y., Jia, L., & Wang, Y. (2019). Short-term abnormal passenger flow prediction based on the fusion of SVR and LSTM. IEEE Access, 7, 42946-42955.
21. Li, Z., Bi, J., & Li, Z. (2017, December). Passenger flow forecasting research for airport terminal based on SARIMA time series model. In IOP conference series: earth and environmental science (Vol. 100, No. 1, p. 012146). IOP Publishing.
22. Takakuwa, S., Oyama, T., & Chick, S. (2003, December). Simulation analysis of international-departure passenger flows in an airport terminal. In Winter Simulation Conference (Vol. 2, pp. 1627-1634).

23. Guizzi, G., Murino, T., & Romano, E. (2009). A discrete event simulation to model passenger flow in the airport terminal. Proc. 11th WSEAS Int. Conf. Math. Methods Comput. Tech. Electr. Eng., pp. 427–434.
24. Munasingha, K., & Adikariwattage, V. (2020). Discrete Event Simulation Method to Model Passenger Processing at an International Airport. In 2020 Moratuwa Engineering Research Conference (MERCon) (pp. 401-406). doi: 10.1109/MERCon50084.2020.9185370.
25. Airport Cooperative Research Program (ACRP). (2010). Airport Passenger Terminal Planning and Design, Volume 1: Guidebook. Airport Cooperative Research Program. National Academies of Sciences, Engineering, and Medicine.
26. Postorino, M. N., Mantecchini, L., Malandri, C., & Paganelli, F. (2019). Airport Passenger Arrival Process: Estimation of Earliness Arrival Functions. Transportation Research Procedia, Volume 37.
27. Olaru, D. (2008). Simulation and GA-optimisation for modeling the operation of airport passenger terminals.
28. Holland, J. H. (1975). Adaptation in Natural and Artificial Systems. University of Michigan Press, Ann Arbor, USA.
29. Davis, L. D. (1991). Handbook of Genetic Algorithms. Van Nostrand, New York.