IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 19 March 2023, accepted 4 May 2023, date of publication 9 May 2023, date of current version 17 May 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3274669

APPLIED RESEARCH

A Holistic Approach on Airfare Price Prediction Using Machine Learning Techniques

THEOFANIS KALAMPOKAS, KONSTANTINOS TZIRIDIS, NIKOLAOS KALAMPOKAS, ALEXANDROS NIKOLAOU, ELENI VROCHIDOU, AND GEORGE A. PAPAKOSTAS

MLV Research Group, Department of Computer Science, International Hellenic University, 65404 Kavala, Greece

Corresponding author: George A. Papakostas (gpapak@cs.ihu.gr)

This work was supported in part by Hellenic Academic Libraries Link (HEAL-Link) Greece.

ABSTRACT Globalization of markets involves new strategies and price policies from professionals that contribute to global competitiveness. Airline companies are changing tickets' prices very often considering a variety of factors based on their proprietary rules and algorithms that are searching for the most suitable price policy. Recently, Artificial Intelligence (AI) models are exploited for the latter task, due to their compactness, fast adaptability, and many potentials in data generalization. This paper represents an analysis of airfare price prediction towards finding similarities in the pricing policies of different Airline companies by using AI Techniques. More specifically, a set of effective features is extracted from 136.917 data flights of Aegean, Turkish, Austrian and Lufthansa Airlines for six popular international destinations. The extracted set of features is then used to conduct a holistic analysis from the perspective of the end user who seeks the most affordable ticket cost, considering a destination-based evaluation including all airlines, and an airline-based evaluation including all destinations. For the latter cause, AI models from three different domains and a total of 16 model architectures are considered to resolve the airfare price prediction problem: Machine Learning (ML) with eight state-of-the-art models, Deep Learning (DL) with six CNN models and Quantum Machine Learning (QML) with two models. Experimental results reveal that at least three models from each domain, ML, DL, and QML, are able to achieve accuracies between 89% and 99% in this regression problem, for different international destinations and airline companies.

INDEX TERMS Airfare price, artificial intelligence, deep learning, machine learning, prediction model, pricing models, regression, quantum machine learning.

I. INTRODUCTION

Approximately 50 years ago airline flights were considered a luxury. Airline companies were launching more domestic flights than international while pricing policies for flight tickets were static. To increase profitability, airline companies adopted management and economical software systems to perform route optimizations, reservation adaptation, and dynamic pricing. An evolution in airline companies was the adoption of yield management [1], which was a variable pricing strategy based on understanding, anticipating, and influencing consumer behaviour so as to reach the highest revenues. As a consequence, airline companies started to pay more attention to customers' preferences and experiences during flights, simultaneously increasing the destinations at an international level. Thus, airline flights became accessible to all potential consumers since dynamic pricing and extra flight services increased the competition between airline companies. Moreover, in recent years, the ability to shop online revolutionized many different fields and became a trend among modern people, seeking the most favorable offers and prices. Currently, there are several websites that support secure flight booking, listing the same flight routes from all airline companies towards getting the most competitive flight deals. Moreover, sharing flight experiences through rating systems provides a great amount of useful information produced daily by airline customers, that are exploited by pricing policy systems to adapt the airfare price, even minutes before a flight. To this end, it is clear that market globalization and technology evolution have affected airline

The associate editor coordinating the review of this manuscript and approving it for publication was Junhua Li.

companies at a level where the mainstream price optimization systems may not track the changes and reach the adaptation speed that is required. The latter increased the demand for more sophisticated algorithms and software for dynamic price policy optimization. For this reason, Artificial Intelligence (AI) algorithms are currently considered for airfare price estimation, towards achieving efficient and more realistic results with higher speed.

Artificial Intelligence attracts high interest from the research community in many research fields. Machine Learning (ML) was the first introduced domain of AI by Walter Pitts and Warren McCulloch [2] in 1943 where a mathematical model of a biological neuron was proposed with no learning capabilities. Seven years later, in 1950, Frank Rosenblatt proposed the perceptron ......(()) [3] as the first neural network (NN) with learning abilities. Perceptron was an inspiration for researchers to design and implement subsequently many well-known ML models like SVM [4], kNN [5], and Boosting methods [6]. ML models couldn't robustly generalize without a supporting feature extraction mechanism. The latter requirement was handled by the Deep Learning (DL) domain, increasing the computational demands and reducing the execution time. The flagship for the rise of the DL domain was the introduction of convolutional neural networks (CNN) [7] by Fukushima in 1980 who used a NN for visual pattern recognition. A distinct boost towards this effort came from Yann LeCun in 1990 [8], who used CNN models with backpropagation learning in order to recognize handwritten digits from images. DL models have automated the feature extraction process giving the capability to fabricate more complex algorithms and applications [9], [10] that impact human daily lives. However, even today, due to the huge data growth rate and despite the evolution of computational hardware (GPUs), there is still a need for faster and more compact ML and DL algorithms.

An attempt to overcome the confronted limitations of ML and DL algorithms, is the combination of quantum mechanics under quantum computing techniques with ML and DL methods. The quantum computing domain was formed in the 90s where quantum algorithms have been proposed for dealing with challenging problems like number factorization Shor's algorithm in 1994 [11] or Grover's algorithm in 1997 [12]. These algorithms became the reason for the fabrication of quantum computers with IBM leading the field. During the same decade, quantum machine learning (QML) started to grow with the introduction of quantum neural networks (QNN) [13] in 1999, where quantum circuits and Grover's algorithm were used to mimic neural network procedures. That work inspired many researchers to experiment with the QML domain. Therefore, during the years 1990-2010, a lot of QML algorithms were introduced, including quantum multilayer perceptron (QMLP) [14], quantum support vector machine (QSVM) [15] and more. Until today, quantum machine learning is expanding even in the industry with applications and algorithms that are executed in real quantum hardware, despite the fact that the available quantum hardware is limited and the computational demands of QML models for a classical computer are very high. Moreover, many QML methods are highly related to classical methods, such as the QNN training for classical data, where the optimizer and loss are computed based on their classical form. The above facts enhanced the evolution rate of QML in the market and research.

This work comes as a follow-up of a previous work [16] on Airfare price prediction. A set of features that characterize a typical flight are extracted and used under the scheme of airfare price prediction for different airline companies and destinations in order to highlight their level of competition and provide a holistic approach to the problem. Moreover, the range of ML, DL, and QML models' applicability and performance is examined holistically in airfare price prediction. Two experiments are conducted; in the first experiment, the problem is studied from the destination perspective (destination-based approach) for each airline company, and the AI models from the above three domains are applied to the same set of destinations for different airline companies in order to highlight similarities among the performance of models; in the second experiment, the ML, DL, and QML models are applied in datasets for each Airline company (airline-based approach), independently from the destination. It should be highlighted here, that this work is the first reported attempt towards a holistic approach to the problem of airfare price prediction, where the problem is examined as a whole, covering both approaches, from the side of destinations and from the side of airline companies. Moreover, it should be noted that QML has never been applied before to the airfare price prediction problem, as far as our knowledge.

Based on the above, the main contributions of the proposed work can be summarized as follows:

1) Investigation of the relation of pricing policies among different airline companies.
2) Investigation of features' influence to the airfare prices prediction problem.
3) Application of QML models in airfare price prediction for the first time in the literature.
4) Comparative performance analysis of ML, DL and QML models for airfare price prediction.

The rest of this paper is organized as follows: Section II summarizes the related work on airfare price prediction. In Section III materials and methods are introduced, refereeing to data and algorithms that have been used for the implementation of this work. Section IV describes the experimental setup, while in Section V the experiment results are presented and discussed. In Section VI, quantum machine learning results are presented and compared to classical models. Finally, Section VII concludes the paper and presents further potential research directions.

II. RELATED WORK

Market globalization along with the evolution of airfare price policies resulted in a great amount of relevant information and, subsequently, high research interest in airfare price prediction. In terms of AI and data analysis, this information is translated to data with many attributes and in amounts that could be characterized as big data, especially when the change rate of air ticket prices and services is such high. The airfare price prediction problem can be exploited under various scopes, like customer segmentation, ticket purchase timing, air tickets demand prediction, and more, as presented in a review by Abdella et al. [17] regarding the target application problem and the solutions. In general, the subject of airfare price prediction is in the spotlight for three decades; a search on Scopus on the term ''airfare price prediction'' returned 24 documents, from 2003 to date, with most of the work being implemented in the last three years. Vu et al. [18] implemented an airfare price prediction application with two ML models, exploiting features around time to describe Vietnamese national airline company flights. Compared to the proposed approach, fewer models have been presented and only one airline company has been considered, while the main focus was on consumers' target applications. In [19], a different approach was presented. A custom recurrent neural network (RNN) was constructed and compared to classical ML models in airfare price prediction under events like a basketball match. Features that described basketball matches and airline flights were combined in one dataset, achieving high prediction accuracies. The same approach was followed in [20]. The authors proposed a framework that could gather information for air tickets from various sources, such as consumers' interests, air tickets availability, distance, and more, to predict airfare prices by using ML models. In [21], airfare price prediction was implemented in the domestic markets of USA and India. The authors exploited ML models and reported an 88% score in price prediction. In [22], Joshi et al. adopted a similar approach with fewer ML models, by investigating new features, like flight duration, and achieved up to 90% prediction score. In [23] feature selection algorithms were applied along with hyperparameter methods to find the optimal model parameters and set of features for flight description in order to predict airfare price prediction. In [24] explainability for the problem under study has been introduced towards a deeper insight into the models that could provide an efficient solution, in order to give robust and explainable predictions.

In general, all related works are based on similar approaches. The proposed work differs in the following distinct points: (1) in the selected feature sets, (2) the data collection sources and (3) the target of the application. Compared to all previous research on the same field, the present work: (4) exploits more technologies and (5) attempts to extract useful information for airline companies' competition and consumers' behavior, through the (6) comparative performance of several algorithms that are introduced to the problem for the first time, (7) providing two evaluation perspective approaches towards a holistic investigation of the problem under study. It is obvious that the diversity of the proposed approach makes a direct comparison with other methods meaningless, since there is no point of common reference, and this would not lead to any conclusion. Comparative performance analysis is therefore provided with common points of reference (dataset, features, target of application) within this work, between the selected ML, DL and QML models for the problem of airfare price prediction.

III. MATERIALS AND METHODS

In this section, the proposed holistic approach is described, focusing on the used data and the selected methods. Datasets, features description and visualization material are presented to highlight the level of competition and globalization affection in airfare tickets between destinations from different airline companies. Moreover, in this section, the ML, DL, and QML models that are employed are presented and a short description for each one is given to underline the differences in performance and capabilities between them.

Fig. 1 graphically illustrates the steps of the proposed methodology. In the first step of the methodology, four airlines and six destinations are considered. The extracted features are applied to eight ML and six DL models towards evaluating the best performing model. Evaluation is performed in two different perspectives. In the first experiment, destination-based evaluation takes place, where the same set of destinations are applied to the models regardless the airline. In the second experiment, airline-based evaluation is examined, where the data from each airline company is applied to the models, for all destinations. The best two performing ML models of the first step of the methodology are used in the second step and are extended to the quantum domain. More specifically, in the second step of the methodology, the two best performing airlines and their best three performing destinations from step 1 are examined, towards comparing ML models and the corresponding QML models. Comparative evaluation is considered from the same two perspectives as in step 1.

A. DATA PRESENTATION AND DESCRIPTION

The focus of this work is on the prediction of airfare prices for six different destinations for four airline companies. The airline companies are: Aegean Airlines, Austrian Airlines, Lufthansa Airlines and Turkish Airlines. The destinations of interest are the following:

1) Thessaloniki (SKG) – Amsterdam (AMS), (1907 Km)
2) Thessaloniki (SKG) – Stockholm (ARN), (2157 Km)
3) Thessaloniki (SKG) – Brussels (BRU), (1812 Km)
4) Thessaloniki (SKG) – Paris (CDG), (1863 Km)
5) Thessaloniki (SKG) – Lisbon (LIS), (2747 Km)
6) Thessaloniki (SKG) – Vienna (VIE), (985 Km)

The flight data are collected for the period of one year.[1] It should be clarified here that flight data are not for exactly one year, due to the fact that some airline companies did not provide the same flights for all destinations all over the year, mainly due to demand variations. Table 1 summarizes

[1]The dataset is provided via GitHub: (https://github.com/MachineLearningVisionRG/AirD)

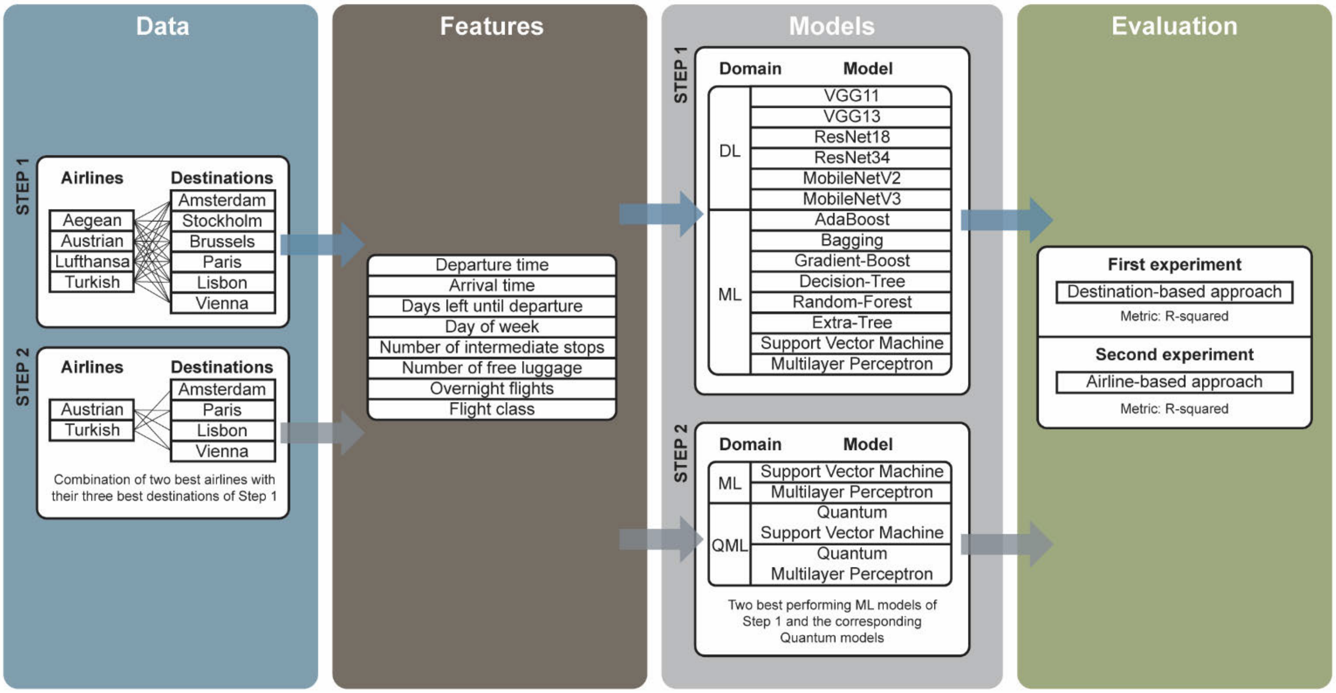


FIGURE 1. **The proposed holistic approach to airfare price prediction.**

TABLE 1. **Number of data flights for each destination and airline company.**

Airline	Destination						
	AMS	ARN	BRU	CDG	LIS	VIE	All
Aegean	17754	17756	15427	18374	16598	9727	95435
Lufthansa	4219	5328	4787	3851	5628	6092	29900
Turkish	1515	1129	1124	1765	1288	1583	8391
Austrian	850	373	285	607	607	1010	3191

the amounts of data flights for each destination and for each airline company.

As it can be observed from Table 1, Aegean has the biggest number of flights; this is attributed to the fact that Aegean is a Greek company and therefore it is on its national base since SKG airport is in Thessaloniki, Greece. Moreover, it can be observed that the number of flights exposes similarities between airline companies. More specifically Aegean, Turkish and Austrian share similar amounts of flights for some destinations (e.g., for Turkish airlines, from SKG to ARN and BRU, etc.). Amsterdam (AMS) and Paris (CDG) are among the most popular destinations for Aegean, Turkish and Austrian airlines. As destinations' distance increases, the number of flights is increased too, according to Table 1. A reason for this might be the capacity of variations in airfare ticket characteristics, like services, which increases the possibility for higher profits for the airline company.

In this work, the most descriptive features that affect the airfare price and were publicly available, were selected. For each flight data, a set of eight features (0:7) was used. Due to the difficulty of collecting flight data manually, Data Mining techniques were applied to acquire as many data as possible. Finally, for each flight the following eight features were considered:

1) Feature 0: Departure time
2) Feature 1: Arrival time
3) Feature 2: Days left until departure (0 - 350+)
4) Feature 3: Day of week (1-7)
5) Feature 4: Number of intermediate stops (0 - 2)
6) Feature 5: Number of free luggage (0 - 2)
7) Feature 6: Overnight flight (yes - 1 or no - 0)
8) Feature 7: Flight class (three-digit number, each digit 0 - 5)

Regarding feature 7, note that flight class is a three-digit integer number. Each digit independently represents a flight class, considering up to three correspondences per voyage. For instance, if the third digit of fight class is not zero, it means that the flight had two intermediate stops, thus, the voyage involved three corresponding flights in total, and each of the three digits informs about the involved ticket class. If the third digit is zero, it means that there was no third flight (only two flights) and so on. Every digit's value is ranged from 0 to 5, depending on the flight class of each of the corresponding flights, as follows:

1) Economy class– 1
2) Economy Standard class – 2
3) Economy Premium class – 3
4) Business class – 4
5) First class – 5
6) No flight – 0

In what follows, a feature correlation analysis is presented based on the Pearson correlation [25] coefficient, in order to justify the eight features' selection and to highlight similarities among airline companies pricing policies through data analysis. Pearson correlation coefficient takes values in the range -1 and 1 for each combination of features. Its mathematical formulation is:

$$\mathrm{r} = \frac{\sum (\mathrm{x_i} - \bar{x})(\mathrm{y_i} - \bar{\mathrm{y}})}{\sqrt{\sum (\mathrm{x_i} - \bar{\mathrm{x}})^2 \sum (\mathrm{y_i} - \bar{\mathrm{y}})^2}} \tag{1}$$

In Equation (1) $\mathrm{x_i}$ and $\mathrm{y_i}$ are the values of the data sample, $\bar{\mathrm{x}}$ and $\bar{\mathrm{y}}$ represent the mean of feature values and r is the correlation coefficient. The results of Pearson correlation for the airfare price dataset for all destinations and airline companies, are presented in Fig. 2 in the form of heatmaps.

Austrian and Turkish airlines, as it can be observed from Fig. 2(b) and Fig. 2(d) have very few flights in the selected destinations and, thus, the number of stops (feature 4) has a low diversity and the correlation coefficients of this value equal to zero. A first notice is that Aegean displays more light colors in its heatmap, translated to less correlations between features in destinations of greater distance (SKG_ARN, SKG_LIS) compared to other destinations which seem to have darker color values, translated to stronger correlations. The same observation can be made for Austrian, Turkish and Lufthansa, but only in the destination SKG_ARN. It is also important to mention that for every airline company and destination, it seems that flight class (feature 7) and price have a strong correlation despite the differences in the number of flights of each company. Based on this fact, it is easy to conclude that flight class has a strong impact on the competition between airline companies. In Fig. 3, the heatmaps of Pearson correlation coefficient for each airline company are presented with the destination as an extra feature (feature 8).

As observed in Fig. 3, flight class and destination display a strong correlation with price for every airline company. Moreover, Austrian and Lufthansa have similar correlations for departure and arrival time, while Aegean and Turkish have opposite correlations for the same two features. A more general notice is that Aegean and Turkish airlines have more diversity in correlations between features, while Austrian and Lufthansa tend to be smoother. Finally, considering Fig. 2 and Fig. 3, similarities between airline companies and their price policies can be highlighted despite the huge differences between the amounts of flights from each company. It can be concluded that by conducting a small data analysis, market globalization and competition level can be spotted between airline companies and their flight destinations. In what follows, the AI models that have been used in this work are presented along with their characteristics.

B. MODELS PRESENTATION AND DESCRIPTION

In this section the total of the 16 selected models from the three domains (ML, DL, QML) are presented and analyzed along with their characteristics.

TABLE 2. Selected machine learning (ML) models.

Model	Algorithm type
AdaBoost Regressor	Boosting family
Bagging Regressor	Boosting family
Gradient Boost Regressor	Boosting family
Decision Tree Regressor	Tree based
Random Forest Regressor	Tree based
Extra Tree Regressor	Tree based
Support Vector Regressor (SVM)	Kernel function
Multi-Layer Perceptron (MLP)	Neural Network

Starting from the ML domain, eight state-of-the-art models were selected and presented in Table 2.

AdaBoost regressor [26] comes from the 'Boosting' family of algorithms, forming a strong learner from a composition of weak learners. Very often these learners are Decision Trees [27] where iteratively AdaBoost adapts their errors and combines them sequentially to create a strong ensemble model that will decrease bias and variance in the training data. A disadvantage of this algorithm is its sensitivity to noise and overfitting with the increase of dataset features and size. *Bagging regressor* [28] adopts a variation of the same approach as AdaBoost. Weak learners in Bagging are created in parallel and, thus, independently of each other, while in AdaBoost they are created sequentially. In addition, Bagging decreases the variance more than the bias and it is proposed to resolve overfitting issues. A reported disadvantage is its sensitivity to noise data and the construction of ideal global solutions in a large number of features and data. Finally from Boosting family, *Gradient Boost algorithm* [29] is also selected. Gradient Boost can produce new models (often Decision Trees) to be maximally correlated with the negative gradients of a loss function (often Mean Squared Error) to minimize it with minimum iterations. Through this learning process, Gradient Boost models have achieved remarkable performances in pattern recognition applications. A disadvantage of these models is in large-scale datasets, where they might stack in local minima, leading to underfitting problems. The *Decision Tree Regressor* is the oldest Tree-based model. It can separate data iteratively based on given parameters, where leaves make predictions and nodes partition input data. Decision Tree algorithms have human understandable interpretations and have achieved high performances in many applications. Their disadvantage is that they may be unstable in a high number of features and data, leading to large sizes of trees. Thus, they can perform more robustly at a local than a global level which is preferable in more complex problems. *Random Forest* [30] algorithm provided the solution to Decision Tree disadvantages, using majority voting from boosting algorithms or averages on random decision trees to control the increase and structure of the trees during the learning process. Random Forest share the same disadvantages as Decision trees in more complex data, however they report better performances. *Extra Tree* [31] or Extremely Randomized Tree is an algorithm similar to Random Forest with the difference

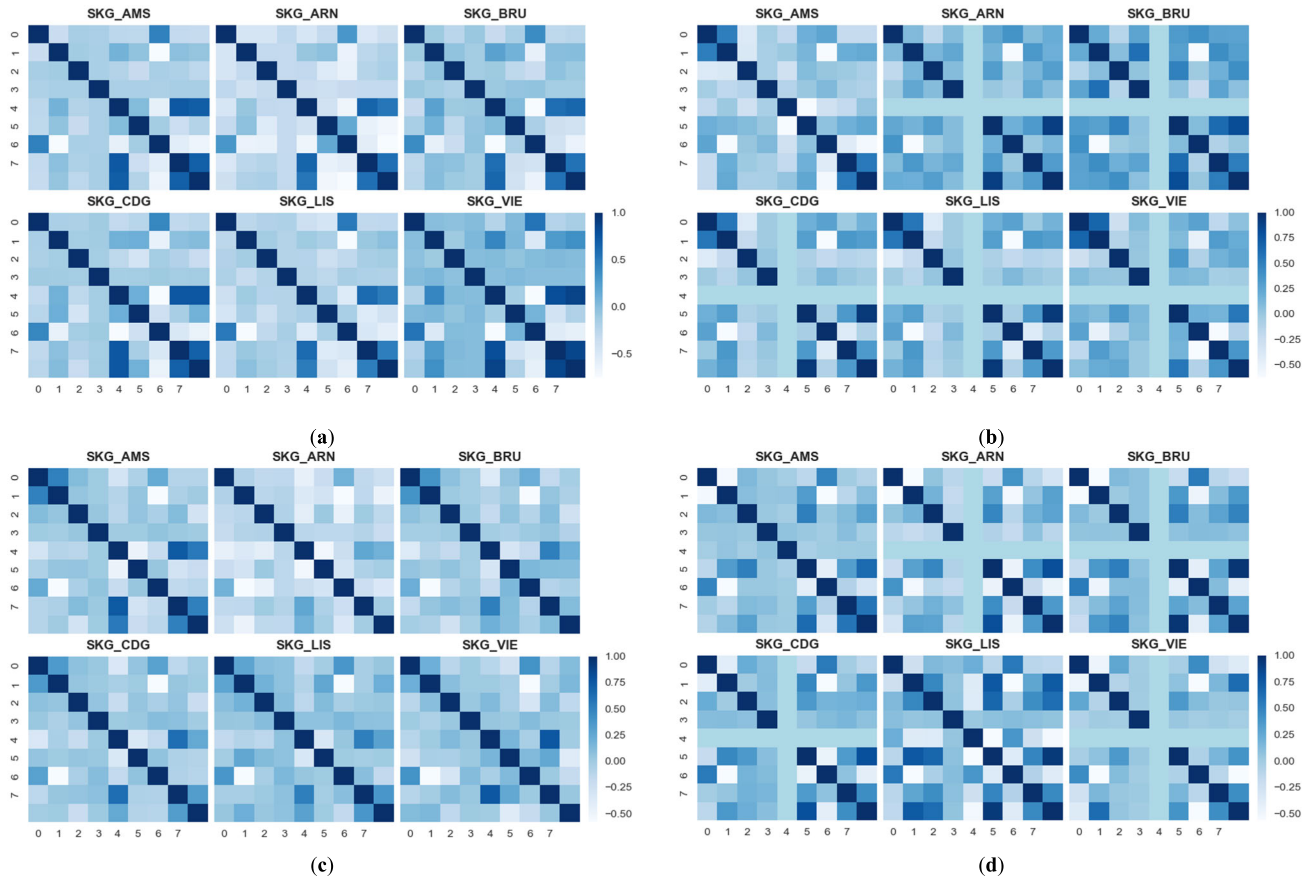


FIGURE 2. Pearson correlation coefficients heatmaps for each destination: (a) Aegean airlines correlation coefficients; (b) Austrian airlines correlation coefficients; (c) Lufthansa airlines correlation coefficients; (d) Turkish airlines correlation coefficients.

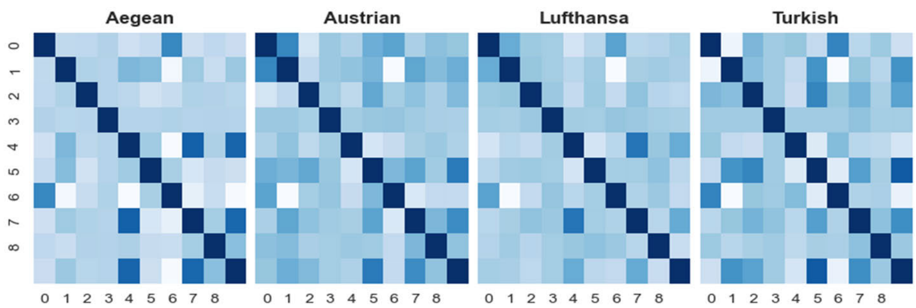


FIGURE 3. Pearson correlation coefficient for each airline company.

that, the construction of decision rule and selection of split values is random. *Support Vector Machines* (SVM) [4] apply kernel functions in a generalization learning process which is a different approach compared to the previous ML models. More specifically, SVM attempt to construct a hyper-plane in a high dimensional feature space that will separate training data based on the target. The high dimensional feature space is acquired through kernel functions such as radial base, linear, and polynomial. Kernel functions give the capability to separate data linearly even if it is uncertain, through data transformation in a higher dimensional feature space. SVM performance is proven robust in handling sparse or large datasets but suffers from increased training time with sensitivity in feature scale. Finally, *Multi-Layer Perceptron* (MLP) [3] was also selected. MLP is the first proposed neural network with an architecture of one input layer, one or more hidden layers and an output layer. Each neuron is connected to each one in the next layer with a weight value. Using backpropagation as a learning method, weight values are adjusted to minimize the loss function and approximate the target values. The advantage of MLP is the large feature handling and the automated feature extraction process.

TABLE 3. Selected deep learning (DL) models.

Model	Training parameters (Millions)
VGG11	133
VGG13	133
Resnet18	11.4
Resnet34	21.5
MobileNetV2	3.4
MobileNetV3	2.4

TABLE 4. Selected quantum machine learning (QML) models.

Model	Algorithm type
Quantum Support Vector Regressor (QSVM)	Quantum kernel
Quantum Multiplayer Perceptron (QMLP)	Quantum neural network with 48 parameters

However, MLP has been proven sensitive in feature scaling under complex problems, as well as computationally costly especially under hyperparameters tuning.

To this end, the above-mentioned ML models have been selected in this work for airfare price prediction, based on their efficient mechanisms, and considering their state-of-the-art performances during the previous years. It should be mentioned that MLP is the basic architecture that inspired the DL domain to produce more sophisticated and complex neural network architectures to overcome performance disadvantages and achieve more robust features in the feature extraction phase.

The selected six models in this work from the DL domain are included in Table 3.

From 1989 until today the DL domain is specialized in the formation of large and complex neural network architectures with most known the CNNs [7], in order to solve challenging problems with complex data like images. Under some level of complexity, handcrafted features cannot produce robust descriptions of data and so ML models cannot perform well. Based on this fact, the DL models have automated the feature extraction procedure and under large architectures (especially in network depth) rich features can be produced even for complex data. The most common model scheme of the DL domain is CNN, which consists of five fundamental units. First, the most important layer is the Convolutional Layer, which consists of convolutional filters (or kernels). Each filter is convolved with the input 2D data to produce feature maps. Kernels are randomly initialized, and they slide in the input data where the dot product is calculated in each slide. Kernel values, namely weights, adjust during training. Second, the pooling layers are applied to sub-sample the feature maps to produce smaller maps, maintaining most of the dominant features. The pooling process is applied with various methods like average, min-max, or custom methods. Then, activation functions take place to map input data with target values through the weighted summation of convolutional layers neurons weights. Thus, it is determined if neurons are contributing to the corresponding target value of a given input data or not. Activation functions give the ability to CNNs to form non-linear correlations between input and target data. Finally, a Fully Connected Layer is usually used to make predictions for input data. Despite the model's architecture, CNN requires a learning process like backpropagation [8]. In this process, an optimization algorithm is applied to minimize the loss function which accepts a target and predicted values in order to calculate the error between them. Based on the above technology, *VGG* [32] was proposed in 2014 by the Visual Geometry Group, under a variety of architectures from 11 to 19 layers with max-pooling adoption. The VGG model has been characterized as state-of-the-art, having small filter sizes, and a large network architecture consisting of 61 to 140 million parameters in VGG19. In the next year, *ResNet* was proposed to overcome the gradient vanish problem. However, while the depth of the CNN network increased the dimension of the features also increased and in contradiction, the loss was optimized to local minima. In that case, a part of the network usually at the start had a low contribution to the prediction. This phenomenon was noticed in VGG and attempted to be resolved by ResNet [33] where multiple residual blocks were used to shorten the connections between layers and, thus, the network could take more layers with stable performance and simpler architecture. It was also proposed under various architectures with 85 million parameters in Resnet50. Finally, *MobileNetV2* [34] was also selected in this work, as a CNN architecture that focuses on the balance between performance and speed. It consists of 3 convolutional layers with a filter size of 1×1 in order to reduce computation time. In addition, the latest version MobileNetV3 [35] was proposed for mobile processing units having less than 2 million parameters.

A disadvantage of the DL technology is that operates non-optimally and is based on statistical methods, considering that CNN treats neuron weights as a whole, even though some weights might not have a high contribution to the predictions of an input datum. This fact justifies the long training times that are required. Based on the above, it is clear that the CNN models' design needs improvement, and thus an effort was given by the research community during the last years to produce sophisticated mechanisms that will make CNN architectures more robust and exclusive to the problem through attention mechanisms, custom losses, and layers or even model design under new domains on which these models will be structured in a more compact way and with more generalization capabilities.

Towards this direction, the QML domain was formed with some well-known models mainly from the ML domain to be implemented under quantum mechanics and quantum computing principles. The selected QML models in this work are presented in Table 4.

Unfortunately, to date, there are many limitations in the QML domain, such as the availability of quantum hardware and the high computational demands by quantum simulations on classical machines. Thus, hybrid algorithms are proposed that operate between classical and quantum data rather than fully quantum. Until today, it is not clear how to express everything under quantum mechanics principles. Therefore, depending on the target application problem, QML models may shift between full and hybrid forms. When QML models are used to solve quantum problems every part of the process could be expressed under quantum mechanics and computing rules since input data and target values are quantum states. In contrast, when QML models are applied in classical data, encoding is required in quantum states and decoding of the output quantum state is required to obtain the target value. Under this scheme, the optimization algorithms that are being used for learning remain classical. This fact is observed mainly in regression problems, where the output is a numerical value, rather than in classification, where the output labels are mapped with qubit states that are described from probability distributions. QML models are benefited from quantum mechanics principles due to the fact that qubits are the basic information unit, since 2^N classical states can be found for N number of qubits. The latter gives a tremendous capacity in information encoding compared to classical bits. Additionally, qubits based on quantum mechanics can be observed also as particles or waves. This gives the capability to store information in its particle form and process it as a signal. Another advantage is the entanglement under which all the possible states of a qubit can exist simultaneously which provides a great parallelization process ability. A disadvantage is that during qubit measurement the entanglement state is lost and, thus, a new circuit has to be formed and executed. Moreover, entanglement produces noise, which affects neighbor qubits, resulting in unstable quantum states.

To this end, *QSVM* [15] is proposed in this work, with kernel function described through entangled qubits in the same number as the features set. Entanglement is applied through rotation gates and the optimal weight value is approximated to form the hyperplane that will optimally separate data. Considering that a qubit can fit more states than a classical bit, the feature space dimension of the quantum kernel can be much higher than the classical and, thus, the data separation hyperplane can be approached optimally and quicker under quantum hardware. Especially in classification problems, QSVM has proven its superiority over the classical SVM in many well-known benchmark datasets for applications related to cancer, fraud detection, etc. Unfortunately, there are very limited real-world applications using QSVM since as the number of features increases the number of qubits increases too, and thus, the computational demands are extremely high for simulation on classical machines, considering that the available quantum hardware has a limit in the number of qubits and availability in general.

Neural networks have succeeded in remarkable results in many applications with many capabilities, leading to classical neural networks under quantum implementations. More specifically *QMLP* [14] architecture consists of parameterized quantum circuits with tunable phase parameters, which represent neuron weights. Input data are encoded as two angles of a qubit and, thus, the number of qubits is double the number of features. This increase of qubits amount is applied to increase the dimension of the feature space that creates redundancy, resulting in possible feature enhancement. With this first step, classical data are transformed into quantum states of untangled qubits which, in other words, can be expressed as a high dimension feature map in multidimensional Hilbert space. Next, a variational circuit is applied to entangle the encoded qubits and tune the phase parameters of rotation gates to extract the enhanced feature map. Finally, a measurement is applied for all qubits in one. The measured value represents the network weight of a given input datum, and a classical linear model is applied to estimate the predicted value as an output layer. In general, there is no prior rule regarding the quantum circuit design and the quantum gates selection for each problem. This verifies the fact that this technology is very new, and it might provide solutions to present and future challenges of ML and DL models. Under the learning process, a classical optimization algorithm and classical loss function are required to adjust the phase parameters of the quantum circuit based on the minimization of the loss between the predicted and real target value. The latter is the most applicable structure of the quantum neural network that is similar to the classical one with some additions. Quantum circuits or quantum layers have more compact structures since qubits can encode a huge amount of information and thus complex features can be extracted even form small architectures. Exploiting entanglement from quantum mechanics, QMLP can process all possible combinations for a given datum in parallel resulting in a tremendous speed-up. Unfortunately, this speed cannot be noticed at the moment since information encoding from classical to quantum state puts a huge time overhead in the process. Another problem is that in general quantum computing works under linear principles, while non-linear principles do not exist; thus, QMLP architecture applies classical activation functions, often sigmoid, to map input and quantum weight values to the output.

Based on the above, the scope of the presented holistic approach is to apply all the above-mentioned 16 models from ML, DL, and QML domains, for airfare price prediction, and comparatively analyze the results. In the following section, the experimental setup followed in this work is presented in detail.

IV. EXPERIMENTAL SETUP

In this work, two experiments are conducted in order to cover the proposed holistic approach for the target application problem. In the first experiment, namely the destination-based approach, the selected models from ML, DL, and QML domains are used to find the best choice for each destination per Airline Company. With this experiment, it can be concluded the optimal set of models that describe the same destinations for separate airline companies, having similar airfare price prediction accuracies. To accomplish that, the entire dataset was split for each destination for each airline company. More specifically, 24 datasets were created for four airline companies and six destinations. In the second experiment, namely the airline-based approach, the same strategy as the first experiment was followed, with the intention this time to locate the best models for each airline company that could describe all six destinations at the same time. For this reason, the dataset was split into four parts, based on the four selected airline companies. After that, a new feature was added to the four datasets, which describe the destination, ranging from 0 to 6, to make the dataset of the second experiment more distinct among the ML, DL, and QML models. For DL models the datasets features' values are normalized and converted to images in order to be used as inputs in the CNN models.

QML models were excluded at this phase since 28 different experiments would take a very long time to be processed, considering the amounts of flights in Table 1. This exception is also justified by the units of time during the learning process of the models from each domain, where for ML and DL the training process took hours and, therefore, for QML models would require days for only one destination. To computationally verify this, consider that in order to simulate on a classical computer a six-dimensional qubit state, a 64-dimensional vector is required since $2^6 = 64$. This computation for a classical computer is hard since bits can be only in one state at a time. Based on the proposed dataset of features, 8 qubits are used for the first experiment and 9 for the second. Thus, the dimensions are 256 and 512, respectively, for each datum. In addition, the computational branches in QMLP are doubled for the weight values of feature and, thus, 65.536 and 262.144 dimensions, respectively, are finally required, translated in millions of flops for a classical machine. Based on the above, only the comparison of QML with ML models for only two airlines, Austrian and Turkish, have been conducted, only for the three best destinations from each of the two selected airline companies.

In general, for each experiment including the QML domain, the prediction accuracy (%) was measured and recorded, using Cross Validation method after a fit-and-predict phase completed with a dataset split in 80% percent for training and 20% for validation. The prediction accuracy is measured by R-squared (R^2) score metric:

$$R^2 = 1 - \frac{\sum (y_i - \hat{y}_i)^2}{\sum (y_i - y)^2} \quad (2)$$

where, y_i is the target value, $\hat{y}_i$ is the predicted value and y is the mean of all target values. In Equation (2), the numerator is the sum squared of regression, which is the difference between predicted and real values. The denominator is the sum of the total squared and expresses the distance of the real data from the mean of the total. R^2 score measures the variation between the predicted and input data and takes values from 0 to 1. For the error rate, Mean Squared Error is selected:

TABLE 5. Frameworks for the implementation of the experiments.

Framework	Domain	Models
PyTorch	DL	[36]
Scikit-Learn	ML	[37]
PennyLane	QML	[38]
Qiskit	QML	[39]

$$MSE = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \quad (3)$$

where, y_i is the real value and $\hat{y}_i$ is the predicted value and the sum of squared differences between these two values is divided by the total number of data, expressing the distance of each input datum from the regression line that is formed by each model. Thus, the less the error, the closer to the regression line the input datum is.

Table 5 includes information regarding the software frameworks that were exploited to conduct the experiments at each domain.

From PyTorch [36] all the presented CNN [7] models were applied along with the learning process on a GPU unit. From Scikit-Learn [37] all the ML models were used and fitted on a CPU unit. From PennyLane [38] QMLP network was formed and executed on a simulator that benefited CPU unit. Under the same principles, QSVM was applied from the Qiskit framework [39]. The hardware specifications where all the above experiments have been conducted are presented below:

- CPU: AMD Ryzen™Threadripper™2920X, 12 cores (24 threads), 3.5GHz base clock.
- RAM: 32 GB DDR 4.
- GPU: NVIDIA GeForce RTX 2060 SUPER 8 GB VRAM.
- STORAGE: Viper M.2 vpn100 3450 MB/s-read, 3000 MB/s-write.

In what follows, the experimental results for the ML and DL models are comparatively presented and analyzed, including ML and DL models, for four airline companies and six destinations, by conducting two experimental approaches. Then, the best ML models are compared with the QML models' performance results for two airline companies and three destinations, for the same two experimental approaches.

V. EXPERIMENTAL RESULTS OF STEP 1: ML VS DL

In this section the results for both experimental approaches, for each ML and DL model are presented, by using the data of four airlines and six destinations.

A. FIRST EXPERIMENT OF STEP 1: THE DESTINATION-BASED APPROACH

Tables 6 to 9 include the experimental results for each airline company and destination for the first experiment. The best scores for each destination are marked in bold in the Tables. An observation that can be derived from the following tables is regarding the model with the best score for each

TABLE 6. First experiment results (R^2) for the aegean airline. Best results for each destination are marked in bold.

Domain	Model	AMS	ARN	BRU	CDG	LIS	VIE	Mean
ML	AdaBoost	0.92	0.85	0.81	0.89	0.80	0.93	0.86
	Bagging	0.92	0.86	0.86	0.91	0.81	**0.95**	0.88
	Gradient-Boost	0.89	0.80	0.81	0.84	0.73	0.94	0.83
	Decision-Tree	0.90	0.79	0.77	0.89	0.73	0.93	0.83
	Random-Forest	0.92	0.85	0.81	0.91	0.81	0.94	0.87
	Extra-Tree	0.93	0.86	0.81	0.92	0.79	0.92	0.87
	Support Vector Machine	0.86	0.80	0.72	0.75	0.76	0.92	0.80
	Multilayer Perceptron	0.94	0.90	0.86	**0.93**	**0.90**	**0.95**	**0.91**
DL	VGG11	0.94	**0.91**	0.85	0.91	0.89	**0.95**	0.90
	VGG13	**0.95**	**0.91**	0.86	0.91	0.89	**0.95**	**0.91**
	ResNet18	0.94	0.88	0.85	0.90	0.87	0.94	0.89
	ResNet34	0.93	0.89	0.85	0.91	0.89	0.94	0.90
	MobileNetV2	0.93	0.87	**0.87**	0.90	0.83	0.91	0.88
	MobileNetV3	0.93	0.82	0.85	0.84	0.82	0.93	0.86

destination, as for all destinations by considering the Mean performance (last column of each Table). Therefore, information about airfare price policies and competition levels between airline companies can be extracted.

From Table 6 it is obvious that the best models for each destination are the neural networks from the DL domain. Bagging, Multilayer Perceptron, Random Forest, and Extra-Tree from the ML domain are following in performance. According to Table 6, it can be concluded that for the Aegean airline, AMS and VIE are the most important destinations compared to the rest of the destinations, since at these destinations almost all models achieve their highest scores, greater than 86%. Based on Table 1, AMS is the destination with the highest number of flights for Aegean and based on Fig. 2(a) it seems that for AMS there are many available flights despite the variety of ticket classes, so the distribution of prices is normal. The same fact involves the VIE destination. Additionally, in Fig. 2(a), VIE has darker color compared to the rest destinations, and since it is the closest destination to SKG it can be concluded that there are many flights to VIE with similar prices. Based on the above it can be assumed that Aegean ticket price strategy aims to attract a variety of consumer groups for AMS destination, rather than VIE, where ticket classes and a variety of services are limited.

In Table 7 the best model for the Austrian airline is Extra-Tree-Regressor with 99% in VIE destination. It can be observed that ML and DL achieve the highest scores with less difference between them, compared to the previous airline company performances. This is justified considering the number of flights from Table 1, as Austrian airlines have at least 50% fewer flights than Aegean. Even with fewer flights for each destination compared to Aegean in CDG, LIS and VIE, all models achieve high performance scores. In addition, according to Fig. 2(b), Austrian airline has stronger correlations for these destinations. Thus, it seems that Austrian airline attempts a competitive policy with many flights that have similar ticket classes and number of stops along with the amount of luggage. On the contrary, for the destinations AMS and BRU, price and ticket classes have high variation with small number of flights, which justifies the results of the DL models. It seems that AMS and BRU are not among the destination that Austrian tries to compete with.

For the case of Lufthansa, as it can be seen in Table 8, the results were very poor compared to the other airlines in general. The best model is the MLP in CDG destination from the ML domain. Based on Table 8, the highest and similar results of the models are in AMS, ARN and CDG, which can be justified by the number of flights in Table 1. It seems that Lufthansa tries to be more competitive with Aegean Airline rather than Austrian since VIE is not in the scope of concern for its price policy. Finally, Lufthansa seems to differ from all airline companies in its general price strategy, since for four out of six destinations ML and DL models have the highest difference in performance compared to the rest of the airlines.

Finally, for the Turkish airline, it can be observed in Table 9 that the best scoring destinations include LIS, AMS, CDG and VIE. More specifically, for destination AMS, the best models are AdaBoost and Random Forest with a score of 93%. For LIS the best models are from both ML and DL domains with scores of up to 97%. For Turkish airline, almost in all destinations, the models bring similar results, except for ARN destination, revealing that it is not so preferable due to its price strategy. In general, based on the results of Table 9, Turkish airline attempts to be competitive through similar ticket classes and prices, considering its number of flights. A final notice is that Turkish and Austrian have more similar price strategies since ML and DL models for four out of six destinations share common performances.

In Table 10 the three best scoring destinations for each airline company are presented. From Table 10 it can be concluded that destination AMS is the best for Aegean, VIE for Austrian, CDG for Lufthansa and LIS for Turkish airlines. Another fact is that destination CDG is among the best performing for all airline companies. In general, it seems that all airline companies are being competitive with Aegean airline, which has the most flights. The latter can be observed especially in VIE destination, which is the nearest to SKG and, thus, the ticket prices for each airline company are similar but with different number of flights and services. Another notice that justifies this fact is, that even Lufthansa is the second

TABLE 7. First experiment results (R^2) for the Austrian airline. Best results for each destination are marked in bold.

Domain	Model	AMS	ARN	BRU	CDG	LIS	VIE	Mean
ML	AdaBoost	0.24	0.90	0.53	0.96	0.97	0.98	0.76
	Bagging	0.30	0.73	0.61	**0.98**	**0.98**	0.95	0.75
	Gradient-Boost	0.31	**0.97**	0.54	**0.98**	**0.98**	0.98	0.79
	Decision-Tree	0.45	0.93	0.59	0.97	0.96	0.95	0.80
	Random-Forest	0.43	0.93	0.61	0.97	**0.98**	0.97	0.81
	Extra-Tree	0.47	0.69	0.57	0.97	0.97	**0.99**	0.77
	Support Vector Machine	**0.79**	0.88	0.78	0.85	0.82	0.94	0.84
	Multilayer Perceptron	0.64	0.93	0.61	0.97	**0.98**	0.97	0.85
DL	VGG11	0.65	0.94	0.64	0.96	0.97	0.97	0.85
	VGG13	0.59	0.93	0.65	0.96	0.97	0.97	0.84
	ResNet18	0.61	0.91	0.65	0.97	**0.98**	0.97	0.84
	ResNet34	0.58	0.95	0.64	**0.98**	**0.98**	0.97	0.85
	MobileNetV2	0.69	0.89	**0.81**	0.94	**0.98**	0.94	**0.87**
	MobileNetV3	0.67	0.93	0.80	**0.98**	**0.98**	0.90	**0.87**

TABLE 8. First experiment results (R^2) for the Lufthansa Airline. Best results for each destination are marked in bold.

Domain	Model	AMS	ARN	BRU	CDG	LIS	VIE	Mean
ML	AdaBoost	0.87	0.88	0.61	0.82	0.48	0.42	0.68
	Bagging	0.85	0.87	0.62	0.8	0.50	0.44	0.68
	Gradient-Boost	0.92	0.77	0.65	0.95	0.47	0.37	0.68
	Decision-Tree	0.84	0.87	0.58	0.95	0.45	0.45	0.69
	Random-Forest	0.89	0.75	0.66	0.95	0.48	0.38	0.68
	Extra-Tree	0.90	0.76	0.76	0.95	0.50	0.46	0.72
	Support Vector Machine	0.77	0.87	0.62	0.86	0.46	0.38	0.66
	Multilayer Perceptron	**0.95**	**0.92**	0.78	**0.98**	0.60	0.60	**0.80**
DL	VGG11	0.94	0.89	**0.79**	0.96	0.60	0.59	0.79
	VGG13	**0.95**	0.89	**0.79**	0.96	0.59	0.59	0.79
	ResNet18	**0.95**	0.87	0.77	0.96	0.59	0.59	0.78
	ResNet34	0.94	0.90	0.78	0.97	**0.61**	0.61	**0.80**
	MobileNetV2	0.87	0.83	**0.79**	0.95	0.57	**0.64**	0.77
	MobileNetV3	0.85	0.77	0.72	0.94	**0.61**	0.63	0.75

TABLE 9. First experiment results (R^2) for the Turkish Airline. Best results for each destination are marked in bold.

Domain	Model	AMS	ARN	BRU	CDG	LIS	VIE	Mean
ML	AdaBoost	**0.93**	**0.93**	0.93	0.93	0.93	0.93	**0.93**
	Bagging	0.92	0.92	0.92	0.93	0.92	0.92	0.92
	Gradient-Boost	0.77	0.52	0.54	0.93	0.46	0.93	0.69
	Decision-Tree	0.89	0.89	0.90	0.89	0.91	0.90	0.89
	Random-Forest	**0.93**	**0.93**	0.93	0.93	0.93	0.93	0.93
	Extra-Tree	0.78	0.53	0.60	0.95	0.62	0.93	0.73
	Support Vector Machine	0.88	0.83	0.89	0.89	0.84	0.94	0.87
	Multilayer Perceptron	0.89	0.90	0.94	**0.96**	**0.97**	0.94	**0.93**
DL	VGG11	0.89	0.89	0.93	0.95	0.96	0.95	0.92
	VGG13	0.89	0.90	0.94	0.95	**0.97**	0.94	**0.93**
	ResNet18	0.88	0.88	0.94	0.94	**0.97**	0.94	0.92
	ResNet34	0.90	0.89	0.95	**0.96**	0.96	0.95	**0.93**
	MobileNetV2	0.88	0.85	0.93	0.90	0.91	0.93	0.90
	MobileNetV3	0.89	0.89	**0.95**	0.94	0.91	**0.96**	0.92

TABLE 10. Summary of the experimental results.

Airline company	Best destination	R^2
Aegean	AMS, VIE, CDG	0.95, 0.95, 0.93
Austrian	VIE, LIS, CDG	0.99, 0.98, 0.98
Lufthansa	CDG, AMS, ARN	0.98, 0.95, 0.92
Turkish	LIS, CDG, VIE	0.97, 0.96, 0.96

biggest airline company in SKG, it does not concern much for VIE destination.

Finally, one model is able to better approach the airfare price prediction problem for all destinations and for each airline company. Based on the previous conclusions, a second experiment is conducted to prove the extracted assumptions. From a technological point of view, it seems that the complexity of the problem is not very high and both ML and DL models can achieve similar scores especially in Austrian airline, however with DL models ranking first in most of the cases. Even for destinations where price tickets are not normal distributed (imbalanced data), DL models performed better due to their inherent mechanisms that give the ability to produce more rich features like max pooling in VGG or non-linear activation functions. Moreover, MobileNet performed better

TABLE 11. Results (R^2) for the second experiment. Best scores for each airline are marked in bold.

Domain	Model	Aegean	Austrian	Lufthansa	Turkish	Mean
ML	AdaBoost	0.86	0.76	0.83	0.65	0.77
	Bagging	0.87	0.76	0.82	0.70	0.78
	Gradient-Boost	0.87	0.76	0.83	0.68	0.78
	Decision-Tree	0.85	0.73	0.78	0.65	0.75
	Random-Forest	0.90	0.77	0.82	0.68	0.79
	Extra-Tree	0.87	0.78	0.83	0.71	0.79
	Support Vector Machine	0.89	0.83	0.84	0.81	0.84
	Multilayer Perceptron	0.92	0.86	0.85	0.92	0.88
DL	VGG11	**0.93**	0.87	**0.87**	0.95	**0.90**
	VGG13	**0.93**	0.87	**0.87**	0.94	**0.90**
	ResNet18	0.90	0.88	**0.87**	0.95	**0.90**
	ResNet34	0.91	**0.89**	0.86	0.96	**0.90**
	MobileNetV2	0.92	0.79	0.85	0.96	0.88
	MobileNetV3	0.86	0.84	0.86	**0.97**	0.88

TABLE 12. Quantum models first experiment results (R^2) for the Austrian Airline. Best performances for each destination are marked in bold.

Domain	Model	CDG	LIS	VIE	Mean
ML	Support Vector Machine (SVM)	0.86	0.82	0.94	0.87
	Multilayer Perceptron (MLP)	0.93	0.90	0.95	0.92
QML	Quantum Support Vector Machine (QSVM)	0.88	0.85	0.95	0.89
	Quantum Multilayer Perceptron (QMLP)	**0.95**	**0.94**	**0.96**	**0.95**

in most of the cases. The latter verifies the relatively low complexity of the problem under study. Considering the model structure and the target application, MobileNets could be the most suited models for traveler's applications considering that they would adapt and predict tickets prices with high scores even in a mobile device.

B. SECOND EXPERIMENT OF STEP 1: THE AIRLINE-BASED APPROACH

Table 11 summarizes the experimental results from the second experiment, in which airlines were studied independently, for all destinations.

In Table 10, all models were comparatively evaluated for all destinations, giving similar results per Airline, with DL models ranking first in all cases. Turkish Airlines report a higher performance, reaching 97% with MobilNetV3. ML models did not perform so well in the experiment with Turkish airline, compared to the first experiment. In general, the superiority of DL models in a higher amount of data is clear compared to the ML domain.

A reason for this poor performance could be that Turkish airline has 8391 data flights for six different destinations, having a distribution of many low-price tickets and a small number of more expensive tickets, which mainly affects the ensemble models. In contrast, DL models prove that they can adjust weights along with the features and targets in a more flexible way, and by using supplementary methods like pooling, they can achieve a higher score under more complex data.

VI. EXPERIMENTAL RESULTS OF STEP 2: ML VS QML

For the comparison of QML with ML models, the two best performing models of ML domain of the first step of the experiments were extended to the quantum domain, for the two best performing airlines and their three best performing destinations. Therefore, in this step of the methodology, only Austrian and Turkish Airlines have been considered (based on Table 2) for only three destinations: CDG, LIS, VIE for Austrian (based on Table 7) and AMS, CDG, VIE for Turkish (based in Table 9). In what follows, experimental results for the selected destinations and airlines are presented for both experiments (destination-based and airline-based approaches).

A. FIRST EXPERIMENT OF STEP 2: THE DESTINATION-BASED APPROACH

Table 12 summarizes the results of the first experiment. As it can be observed, QMLP clearly holds the first place compared to classical MLP and SVM, achieving enhanced performances by 3% and 8%, respectively, based on the Mean performance for all three destinations. Since QMLP has a more compact structure based on Table 4 with a similar feature enhancement capacity to a CNN, its generalization capability is very high. Regarding the pair of models SVM and QSVM, quantum kernels have proven better, compared to the classical since they can construct larger feature dimensions that might lead to linear data separation even under complex data structures or sparse patterns. In addition, it should be noted that QML models are examined not at their full potential, since not all the capabilities of their learning process are feasible to be explored due to huge time and resource requirements. In Fig. 4, a comparative illustration is presented, including all domain models for the Austrian airline and the three selected destinations.

Considering that Austrian airline has a small number of flights for the three selected destinations which are close to

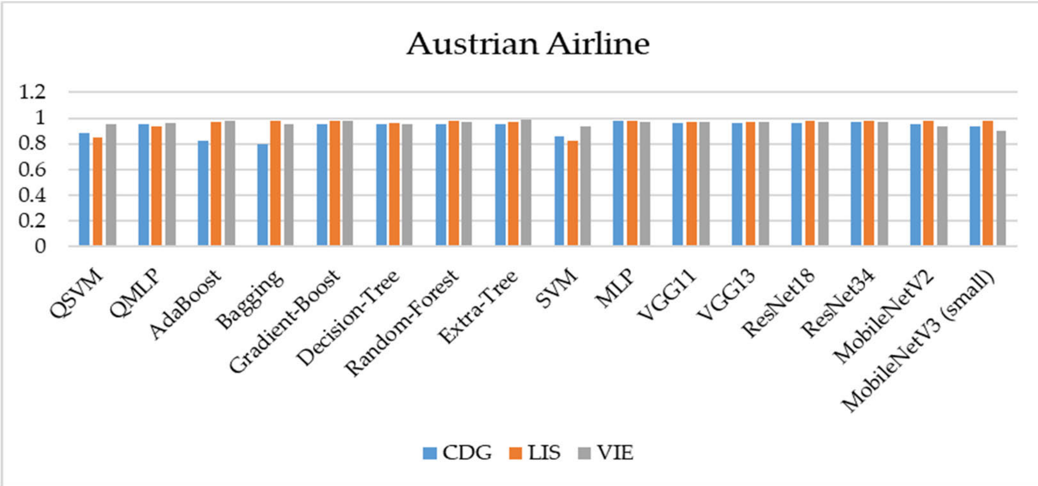


FIGURE 4. Bar plot for all domain models in three destinations of Austrian airline.

TABLE 13. Quantum models first experiment results (R^2) for the Turkish Airline. Best performances for each destination are marked in bold.

Domain	Model	AMS	CDG	VIE	Mean
ML	Support Vector Machine (SVM)	0.88	0.89	0.94	0.90
	Multilayer Perceptron (MLP)	0.89	**0.96**	0.94	0.93
QML	Quantum Support Vector Machine (QSVM)	**0.92**	0.91	0.95	0.92
	Quantum Multilayer Perceptron (QMLP)	0.93	0.95	**0.96**	**0.94**

each other, the performance scores for all domain models result in small differences between them. Even so, QML models perform better and closer to DL domain models, compared to ML, especially for QMLP which ranks first among all models in all cases. This performance similarity between QML and DL models is justified from the dimensions of features space under quantum principles that are closer to CNN rather to ML models, but with a simpler structure considering that 16 qubits represent 8 flight features and their corresponding neurons. The dimension of a qubit is 2^N in classical machines, where N is the number of qubits, therefore, for the proposed problem QMLP constructs a 65.356-dimensional feature map. Moreover, the classical gradient-based optimization algorithm requires the construction and evaluation of several quantum circuits in its gradient iteration which is computation costly. All above justify higher hardware resource demands of QML in a classical machine, compared to the other two domains. Results for Turkish airlines are included in Table 13.

It is clear that QML models overall perform better than classical ML models based on the results of Table 13. In two out of the three destinations, QML models performed better, justifying their superiority. QSVM can construct a hyperplane through a quantum kernel with much higher dimensional feature space than classical kernels and, thus, the separation of Turkish flights data is much bigger than with SVM despite the fact that there are only a few flights in the dataset with imbalanced price groups. QMLP also shares the same advantage compared to MLP, since the feature map is in higher dimensional feature space, plus the variational circuits which exploit entanglement to tune and feature enhance at the same time, leading to higher redundancy than classical models, and therefore, leading to better results. QMLP has a much more compact structure with much bigger generalization capabilities.

Even so, the difference in the results included in Table 13 is not as high as it should be, for the reason that QML models have huge time and resource demands as already mentioned and, thus, the optimal time of the training process was not achieved. However, under quantum hardware machine specifications, this process is expected to be very fast, and the structure of these models can be characterized as very small in parallel. Fig. 5 illustrates the bar plots for all domain models for the three selected destinations of the Turkish airline.

It is clear that QSVM and QMLP come first compared to ML models in all three destinations, not only as a matter of score but also as a matter of sustainability, despite the diversity of data involved in the above destinations. It can be concluded that QML performance is comparable to the performance of DL domain models. A general notice is that QML models seem to have the generalization capabilities of DL models, but with a simpler structure according to quantum hardware capacities. It should be mentioned that QML models however, come last with respect to speed and flexibility of resources management. In the following subsection, the comparison between ML and QML continues by conducting the second experiment, referring to all destination flight datasets for the two airline companies.

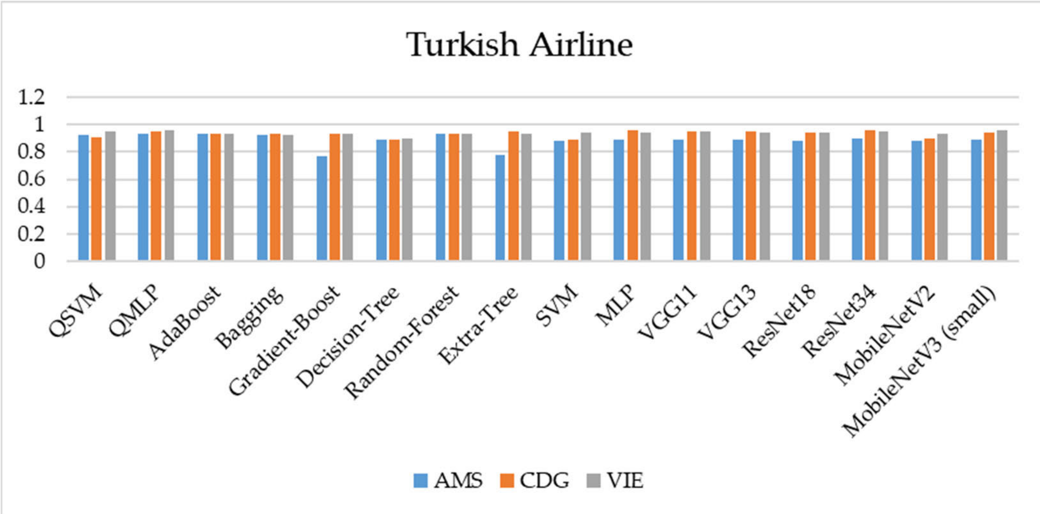


FIGURE 5. Bar plot for all domain models in three destinations of Turkish airline.

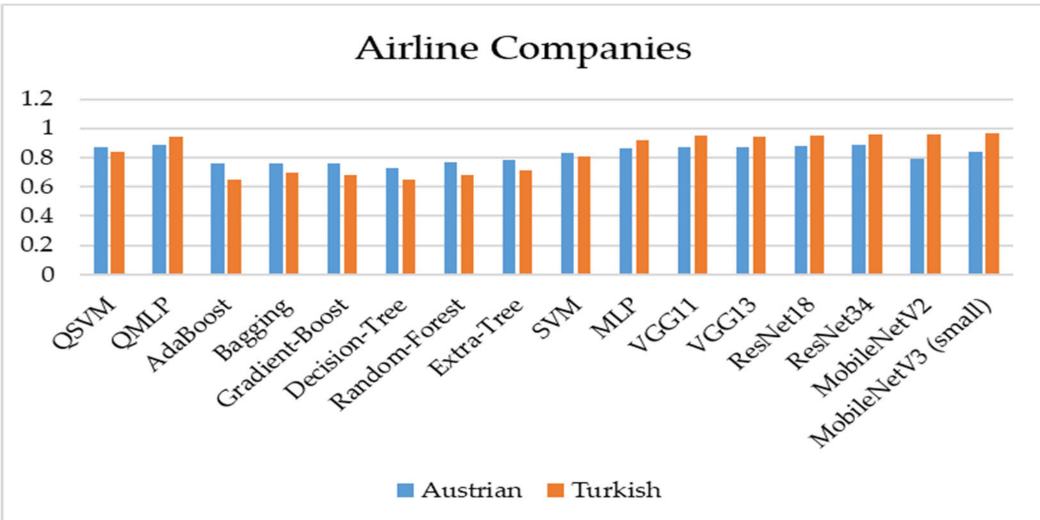


FIGURE 6. Bar plot for all domain models in three destinations of Turkish and Austrian airlines.

B. SECOND EXPERIMENT OF STEP 2: THE AIRLINE-BASED APPROACH

In the second experiment, all destinations are considered for the two airline companies. Results are summarized in Table 14. Based on the experimental results, QML models reveal the optimal performance for both airlines. Despite Austrian airline's imbalanced distribution of airfare price groups and a few number of flights, still results are ranking high for all models. However, for QSVM, the above fact seems to have a smaller impact compared to SVM, reporting a score difference of 4%. For QMLP and MLP the same notice can be made. The Turkish airline shares common strategies with Austrian, but with a more normal distribution in airfare prices and services groups. Same with the previous airline company, QML models come first compared to ML models for Turkish airline flights. Another similar conclusion to the previous experiment's, is that QML models achieve performance scores closer to DL models rather than to ML models, as it can be observed from the bar plots illustrated in Fig. 6.

Despite their similar performance to DL models, QML models are competitive and among the best models with almost similar performances for both airline companies. However, QML models' performance cannot be deployed in real-world applications towards tracking and adjusting airfare price predictions based on a variety of sources for feature values. The latter is the main disadvantage of QML models, which will take time to be covered. ML models have shown the poorest average performance. Considering the evolution and growth of data amounts and the involved complexity of

TABLE 14. Quantum models second experiment results (R^2) for Turkish and Austrian airlines. Best performances for each airline are marked in bold.

Domain	Model	Austrian	Turkish	Mean
ML	Support Vector Machine	0.83	0.81	0.82
	Multilayer Perceptron	0.86	0.92	0.89
QML	Quantum Support Vector Machine	0.87	0.84	0.85
	Quantum Multilayer Perceptron	**0.89**	**0.94**	**0.91**

the problem under study, ML models are not recommended for airfare price prediction. Finally, considering the factor of computational efficiency, DL models are the most suited for the problem, since they can be executed in a variety of devices, which shows flexibility in the manageability of computational resources with higher potential in the approximation of optimal solutions for airfare price prediction problem.

VII. DISCUSSION AND CONCLUSION

In this work, the focus is on the airfare price prediction holistic approach, considering different datasets and technologies that could be applied. To this end, four airlines and six destinations were considered. To resolve the problem under study, eight ML models, six DL models, and two QML models have been employed and comparatively evaluated. Experimental results reveal that at least three models from each domain ML, DL, and QML are able to achieve accuracies between 89% and 99% in this regression problem, for different international destinations and airline companies. Results reveal that by using AI models and flight features that are available to customers before purchase, the airline company ticket price policy can be efficiently analyzed. More features are publicly available and by using the above technologies, robust simulations for flight tickets' price optimization and customer demand could be approximated, towards providing rich information to airline companies to build their optimal price strategy. However, even under a small set of features, all model domains are able to extract patterns from the given flight data and can find similarities between them. In this work, two different approaches have been investigated and analyzed: one based on the destinations (for all airlines) and one based on the airline companies (for all destinations). Future work from the perspective of the airline-based target application, could include the same airline companies and destinations studied from different airports to examine if the information could be efficiently extracted. Moreover, the same problem could be studied as a classification problem through customer segmentation, based on the flight features set.

From a technological point of view, QML models have been studied under a regression application, which is limited in the literature, since the advantage of QML models in classical data is controversial, considering the limitations and the available quantum resources along with the computational demands in classical machines. Despite limitations like the number of qubits and noise levels in quantum machines, the availability of quantum hardware must be increased and become friendlier in order to pave the way for QML solutions to be applied to more real-world applications.

In this work, QML models for airfare price prediction achieved higher results in most cases compared to ML and DL models, despite the reported disadvantages and confronted difficulties. It could be therefore concluded that future approaches to airfare price prediction based on the QML domain could provide efficient solutions, especially with the expectation that the amount, complexity, and diversity of data will grow. Future work around QML methods in airfare price prediction, includes the investigation of various different methods for data encoding in quantum states, and more quantum models like quantum Boltzmann machines, which will be able to generate flight data based on given air tickets feature sets and distributions. The resulted QML-based application could be used as an airfare price policy generator.

REFERENCES

[1] S. Netessine and R. Shumsky, "Introduction to the theory and practice of yield management," *INFORMS Trans. Educ.*, vol. 3, no. 1, pp. 34–44, Sep. 2002.

[2] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *Bull. Math. Biophys.*, vol. 5, no. 4, pp. 115–133, Dec. 1943.

[3] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain," *Psychol. Rev.*, vol. 65, no. 6, pp. 386–408, 1958.

[4] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *Proc. 5th Annu. workshop Comput. Learn. theory*, Jul. 1992, pp. 144–152.

[5] E. Fix and J. L. Hodges, "Discriminatory analysis. Nonparametric discrimination: Consistency properties," *Int. Stat. Rev./Revue Internationale de Statistique*, vol. 57, no. 3, p. 238, Dec. 1989.

[6] R. E. Schapire, "The strength of weak learnability," *Mach. Learn.*, vol. 5, no. 2, pp. 197–227, Jun. 1990.

[7] K. Fukushima, "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," *Biol. Cybern.*, vol. 36, no. 4, pp. 193–202, Apr. 1980.

[8] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[9] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis, "Mastering the game of go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, Jan. 2016.

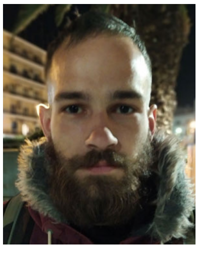
[10] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Commun. ACM*, vol. 63, no. 11, pp. 139–144, Oct. 2020.

[11] P. W. Shor, "Algorithms for quantum computation: Discrete logarithms and factoring," in *Proc. 35th Annu. Symp. Found. Comput. Sci.*, 1994, pp. 124–134.

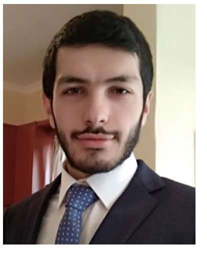
[12] L. K. Grover, "A framework for fast quantum mechanical algorithms," in *Proc. 30th Annu. ACM Symp. Theory Comput. (STOC)*, 1998, pp. 53–62.

[13] M. Andrecut and M. K. Ali, "Quantum associative memory," *Int. J. Mod. Phys. B*, vol. 17, no. 12, pp. 2447–2472, May 2003.

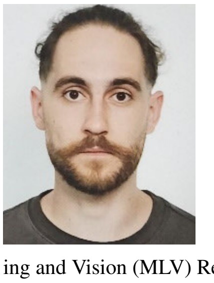
[14] M. V. Altaisky, N. E. Kaputkina, and V. A. Krylov, "Quantum neural networks: Current status and prospects for development," *Phys. Particles Nuclei*, vol. 45, no. 6, pp. 1013–1032, Nov. 2014.
[15] V. Havlíček, A. D. Córcoles, K. Temme, A. W. Harrow, A. Kandala, J. M. Chow, and J. M. Gambetta, "Supervised learning with quantum-enhanced feature spaces," *Nature*, vol. 567, no. 7747, pp. 209–212, Mar. 2019.
[16] K. Tziridis, T. Kalampokas, G. A. Papakostas, and K. I. Diamantaras, "Airfare prices prediction using machine learning techniques," in *Proc. 25th Eur. Signal Process. Conf. (EUSIPCO)*, Aug. 2017, pp. 1036–1039.
[17] J. A. Abdella, N. Zaki, and K. Shuaib, "Automatic detection of airline ticket price and demand: A review," in *Proc. Int. Conf. Innov. Inf. Technol. (IIT)*, Nov. 2018, pp. 169–174.
[18] V. H. Vu, Q. T. Minh, and P. H. Phung, "An airfare prediction model for developing markets," in *Proc. Int. Conf. Inf. Netw. (ICOIN)*, Jan. 2018, pp. 765–770.
[19] F. Huang and H. Huang, "Event ticket price prediction with deep neural network on spatial–temporal sparse data," in *Proc. 35th Annu. ACM Symp. Appl. Comput.*, Mar. 2020, pp. 1013–1020.
[20] T. Wang, S. Pouyanfar, H. Tian, Y. Tao, M. Alonso, S. Luis, and S. Chen, "A framework for airfare price prediction: A machine learning approach," in *Proc. IEEE 20th Int. Conf. Inf. Reuse Integr. Data Sci. (IRI)*, Jul. 2019, pp. 200–207.
[21] E. A. Kuptsova and S. K. Ramazanov, "Analysis of artificial neural networks training models for airfare price prediction," *Artif. Intell.*, vol. 25, no. 3, pp. 45–50, Oct. 2020.
[22] N. Joshi, G. Singh, S. Kumar, R. Jain, and P. Nagrath, "Airline prices analysis and prediction using decision tree regressor," in *Proc. Int. Conf. Recent Develop. Sci., Eng. Technol.*, in Communications in Computer and Information Science, 2020, pp. 170–186.
[23] R. R. Subramanian, M. S. Murali, B. Deepak, P. Deepak, H. N. Reddy, and R. R. Sudharsan, "Airline fare prediction using machine learning algorithms," in *Proc. 4th Int. Conf. Smart Syst. Inventive Technol. (ICSSIT)*, Jan. 2022, pp. 877–884.
[24] S. Sutthithatip, S. Perinpanayagam, and S. Aslam, "(Explainable) artificial intelligence in aerospace safety-critical systems," in *Proc. IEEE Aerosp. Conf. (AERO)*, Mar. 2022, pp. 1–12.
[25] K. Pearson, "VII. Note on regression and inheritance in the case of two parents," *Proc. Roy. Soc. London*, vol. 58, nos. 347–352, pp. 240–242, Dec. 1895.
[26] Y. Freund and R. E. Schapire, "Experiments with a new boosting algorithm," in *Proc. 13th Int. Conf. Mach. Learn.*, 1996, pp. 148–156.
[27] A. D. Gordon, L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, "Classification and regression trees," *Biometrics*, vol. 40, no. 3, p. 874, Sep. 1984.
[28] L. Breiman, "Bagging predictors," *Mach. Learn.*, vol. 24, no. 2, pp. 123–140, Aug. 1996.
[29] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *Ann. Statist.*, vol. 29, no. 5, pp. 1189–1232, Oct. 2001.
[30] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, pp. 5–32, Oct. 2001. [Online]. Available: https://link.springer.com/article/10.1023/a:1010933404324
[31] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," *Mach. Learn.*, vol. 63, no. 1, pp. 3–42, Apr. 2006.
[32] K. Simonyan, and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," Apr. 2015, *arXiv:1409.1556*.
[33] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[34] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4510–4520.
[35] A. Howard, M. Sandler, B. Chen, W. Wang, L. Chen, M. Tan, G. Chu, V. Vasudevan, Y. Zhu, R. Pang, H. Adam, and Q. Le, "Searching for MobileNetV3," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1314–1324.
[36] A. Paszke et al., "PyTorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems*, vol. 32, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds. Curran, 2019. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2019/file/bdbca288fee7f92f2bfa9f7012727740-Paper.pdf
[37] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and É. Duchesnay, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12 no. 10, pp. 2825–2830, Jan. 2012.
[38] V. Bergholm et al., "PennyLane: Automatic differentiation of hybrid quantum-classical computations," 2018, *arXiv:1811.04968*.
[39] M. Treinish et al., "Qiskit/qiskit-metapackage: Qiskit 0.43.0 (0.43.0)," Zenodo, May 2023. Accessed: May 12, 2023, doi: 10.5281/zenodo.7897504.



THEOFANIS KALAMPOKAS was born in September 1994. He received the degree in software engineering and the M.Phil. degree in advanced technologies in informatics and computers from the Department of Computer Science, International Hellenic University, Kavala, in 2018 and 2021, respectively, where he is currently pursuing the Ph.D. degree with the Department of Computer Science. Since 2021, he has been a member of the Machine Learning and Vision (MLV) Research Group, where he participates in research projects. His main research interests include image processing, artificial intelligence, machine-learning and deep-learning techniques, and quantum computing.

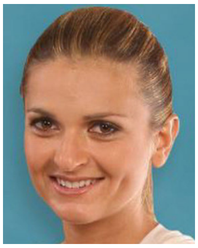


KONSTANTINOS TZIRIDIS was born in October 1995. He received the degree in software engineering and the M.Phil. degree in advanced technologies in informatics and computers from the Department of Computer Science, International Hellenic University, Kavala, in 2018 and 2021, respectively, where he is currently pursuing the Ph.D. degree with the Department of Computer Science. Since 2021, he has been a member of the Machine Learning and Vision (MLV) Research Group, where he participates in research projects. His main research interests include image processing, artificial intelligence, and machine-learning and deep-learning techniques.



NIKOLAOS KALAMPOKAS was born in August 1998. He received the degree from the Department of Industrial Design and Production Engineering, University of West Attica, in 2021. He is currently pursuing the master's degree with the Department of Image Analysis and Machine Learning, Uppsala University. He was with start-up company Gaspardesk, as a Software Developer, specializing in APIs integrations and implementations. As a researcher, he is a member of the Machine Learning and Vision (MLV) Research Group.

ALEXANDROS NIKOLAOU received the degree in software engineering and the M.Phil. degree in advanced technologies in informatics and computers from the Department of Computer Science, International Hellenic University, Kavala, in 2018 and 2023, respectively. As a researcher, he is a member of the Machine Learning and Vision (MLV) Research Group, where he participates in research projects.



ELENI VROCHIDOU received the Diploma degree in embedded systems, the M.Sc. degree in automatic control systems, and the Ph.D. degree in signal processing from the Department of Electrical and Computer Engineering, Democritus University of Thrace (DUTH), Greece, in 2004, 2007, and 2016, respectively.

She is currently a part-time Lecturer with the Department of Computer Science, International Hellenic University (IHU), where she taught bachelor's courses (2005–2022) and master's courses (2018–2022). Her research interests include intelligent systems, signal processing, pattern recognition, and embedded systems. In these areas, she has several publications in international scientific conferences, international scientific journals, and book chapters. As a researcher, she is a member of the Machine Learning and Vision (MLV) Research Group, where she participates in research projects.

GEORGE A. PAPAKOSTAS received the Diploma, M.Sc., and Ph.D. degrees in electrical and computer engineering from the Democritus University of Thrace (DUTH), Greece, in 1999, 2002, and 2007, respectively. Currently, he is a tenured Full Professor with the Department of Computer Science, International Hellenic University, Greece. He has 15 years of experience in large-scale systems design, as a senior software engineer and a technical manager. He is also the Head of the Machine Learning and Vision (MLV) Research Group. He has (co)authored more than 210 publications in indexed journals, international conferences, and book chapters, one book (in Greek), two edited books, and six journal special issues. His publications have more than 3300 citations with H-index is 33 (Google Scholar). His research interests include machine learning, computer/machine vision, pattern recognition, and computational intelligence. He is a member of the IAENG, MIR Labs, EUCogIII, and the Technical Chamber of Greece (TEE). He served as a reviewer for numerous journals and conferences.