Comparison Study of Machine Learning Techniques to Predict Flight Energy Consumption for Advanced Air Mobility

Robert Selje II* and Reagan Rubio†
New Mexico State University, Las Cruces, New Mexico, 88003

George Gorospe‡
NASA Ames Research Center, Moffett Field, CA, 94035, USA

Liang Sun§
New Mexico State University, Las Cruces, New Mexico, 88003, USA

This paper addresses the need to predict the flight energy consumption of aerial vehicles in the presence of wind using machine learning techniques. The presented work is critical to achieving sustainable and efficient operations for Advanced Air Mobility (AAM) and to evaluating the readiness of the ground-supporting energy infrastructure, e.g., electric grid and AAM portals. The flight energy consumption is described using the "energy per meter" (EPM) metric. We present a comparison study of influential machine learning techniques in predicting EPM using real-world flight test data. We presented new results of using the Decision Tree, Random Forest, and linear regression techniques, along with our previous results using the Recurrent Neural Network and Feed Forward Neural Network techniques. The comparison results show that the Linear Regression method outperforms other methods on the basis of the Mean Squared Error and error variance.

I. Introduction

The pressing need for clean-energy aerial vehicles to achieve net-zero greenhouse gas emissions from the U.S. aviation sector by 2050 [1] has led industrial, government, and academic stakeholders to reimage air transportation for people and goods. This brings the concept of Advanced Air Mobility (AAM) [2]. Active research is underway to enable AAM and resolve the current transportation and aircraft operations challenges. The advanced development and usage of various Uncrewed Aerial Vehicles (UAVs) and Vertical Take-Off and Landing (VTOL) aircraft in recent years have driven many researchers to reimage how goods are shipped and how people will travel in the near future. NASA is conducting investigations in Advanced Air Mobility (AAM) to resolve the current challenges in transportation and guide research activities supporting aircraft development for emerging aviation markets [3]. AAM will enable consumers to access on-demand air mobility, package delivery, and emergency services for travel across local, regional, intraregional, and urban areas [4, 5].

The carbon dioxide (CO_2) emissions from the growing number of fossil fuel-based vehicles continue to increase levels of greenhouse gases and have raised concern from governments and researchers around the world for irreversible climate damage. One particular study shows that 26% of the worldwide CO_2 emissions are caused by traffic [6]. Many governments worldwide have issued laws banning the production of fossil fuel-based vehicles and moving towards clean energy vehicles. One clean energy vehicle that has gained the attention of researchers and private industry is UAVs, which have been used in shipping goods in recent years. In addition, the industrial, government, and academic stakeholders are investigating large electric VTOLs (eVTOLs) for the future transportation of humans and cargo for AAM. There is a pressing need to charge electric drones to minimize downtime and keep vehicles moving quickly.

As we progress toward clean energy vehicles, there is an increasing need to predict the flight energy consumption needed by AAM operations. The same motivation for clean energy vehicles also drives the need to harvest energy from renewable resources and limit the usage of coal-based power plants. A significant increase in energy produced

*Ph.D. student, Klipsch School of Electrical and Computer Engineering, P.O. Box 30001, MSC 3-O, Las Cruces, NM 88003.
†Undergraduate student, Department of Mechanical and Aerospace Engineering, P.O. Box 30001, MSC 3450, Las Cruces, NM 88003.
‡Senior Research Engineer, NASA Ames Research Center, Moffett Field, CA 94035-1000
§Associate Professor, Department of Mechanical and Aerospace Engineering, New Mexico State University, Jett Hall, 1040 S. Horseshoe Street, Las Cruces, NM 88003, USA. AIAA Senior Member.

from renewable sources (such as wind and solar) is needed to assure the sustainability and dependability of electricity delivery in the expanding metropolis. However, renewable energy resources are more unreliable than traditional power plants due to varying weather conditions and the time it takes for these devices to replenish their energy banks from renewable sources like solar, water, and wind. Therefore, there is a pressing need to provide the UAV with the necessary energy required to complete the task and reduce the overall burden on renewable energy resources.

Unlike traditionally gas-powered vehicles, recharging an eVTOL battery can be a time-consuming process. Energy gained from renewable sources can be unreliable, cloud cover can reduce the amount of energy available, and on a calm day, no energy would be generated from a wind turbine. Therefore there is a pressing need to charge the eVTOL battery only to the level that is required to complete the task, reducing the overall burden on renewable energy resources and service delay. To optimize energy usage in this way, we must first be able to predict how much energy an eVTOL would use over a set trajectory. Predicting an aircraft's energy usage is not a simple task as there is no standardized method for predicting eVTOL energy consumption. Physicis-based models have been proposed to estimate eVTOL flight energy consumption, each model producing different results with varying levels of accuracy [7]. For example, Dorling et al. [8] presented a linear model of drone energy consumption that is based on the weight of payloads that the drone is carrying, whereas Shabarani et al. [9] calculate the energy consumed as a unit per distance.

The main challenges in validating and adopting the physics-based energy-consumption models come from the inaccessibility of key parameters in the existing physics-based models, such as the drag coefficient and downwash coefficient. Also, there have been few physics-based models that reflect the impact of the wind on vehicle energy consumption. Machine learning (ML) techniques have been leveraged in describing complex relationships of variables in many applications. However, very few research works for data-driven flight energy consumption modeling have been found in the literature.

In this work, we focus on the energy consumption of a level flight, which is evaluated using the "energy per meter" (EPM) metric [7]. For a level flight, we assume that EPM is a function of the wind speed, wind angle, UAV heading (yaw) angle, and UAV ground velocity at a given time instant. Such information is readily available given any UAV. Knowing EPM allows the UAV to plan and take action to minimize energy consumption in real-time. A UAV can also determine the energy needed along a particular path using EPM. We expect a UAV to have the ability to calculate the EPM at an exact moment in time with the current knowledge without having to maintain historical information.

The contribution of this paper includes a comparison study of various state-of-the-art machine learning techniques to predict EPM. We used real-world flight test data [10] to train and validate the ML algorithms. We applied the Decision Tree Regressor (DTR), Random Forest Regressor (RFR), and Linear Regression Regressor (LRR) techniques for EPM prediction. We compared the results with our previous results of using the Recurrent Neural Network (RNN) and Feed Forward Neural Network (FFNN) techniques. The comparison results show that the Linear Regression method outperforms other methods on the basis of the Mean Squared Error and its variance.

The remainder of the paper is structured as follows. Section II introduces the data set used to train and validate the different machine learning techniques. The fundamentals for each machine learning technique discussed in the paper are presented in Section III, along with the methodology of applying them for energy expenditure estimation. Section IV discusses the results of the research. Section V contains the conclusion and future work.

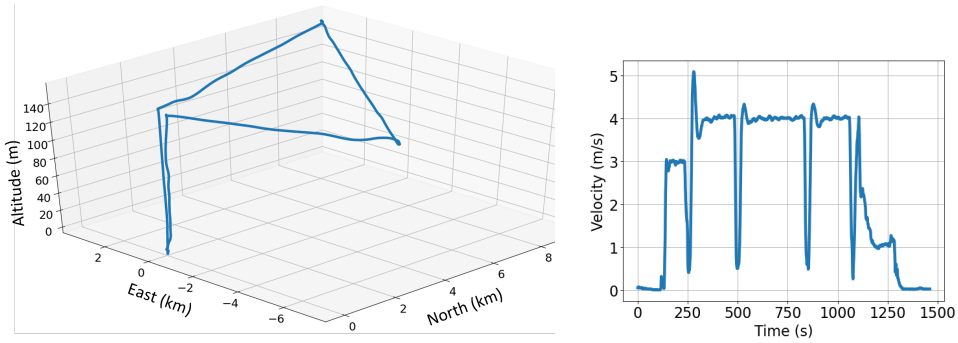
II. Real-World Flight Test Data

This work utilized the data set presented in [10], which contains nearly 11 hours of flight time and covers approximately 65 kilometers of flight distance using the DJI Matrice 100 quadcopter. The drone flies the same triangular pattern for all 201 flights and utilizes onboard sensors, including GPS, IMU, voltage and current sensors connected to the battery, and an ultrasonic anemometer, to collect data on vehicle states, battery states, and wind velocities. Various flight operational parameters were generated using different combinations of payload weights (0 g, 250 g, and 500 g), velocities (4 m/s, 6 m/s, 8 m/s, 10 m/s, and 12 m/s), and cruise altitude (25 m, 50 m, 75 m, and 100 m).

Figure 1 shows an example flight where Fig. 1(a) is the flight path, and the ground velocity during the flight is shown in Fig. 1(b). We display the flight path with respect to the takeoff position. The UAV takes off at the origin, flies a triangular pattern, and lands near the takeoff location. If we compare the flight path with the velocities for the flight, we can see that the ground velocity is stable along each triangle leg. Such behavior can be seen between 800 seconds and 1020 seconds in Fig. 1(b). The large oscillations in the velocity figure are due to the drone stabilizing or turning.

In this paper, the true EPM values are given by

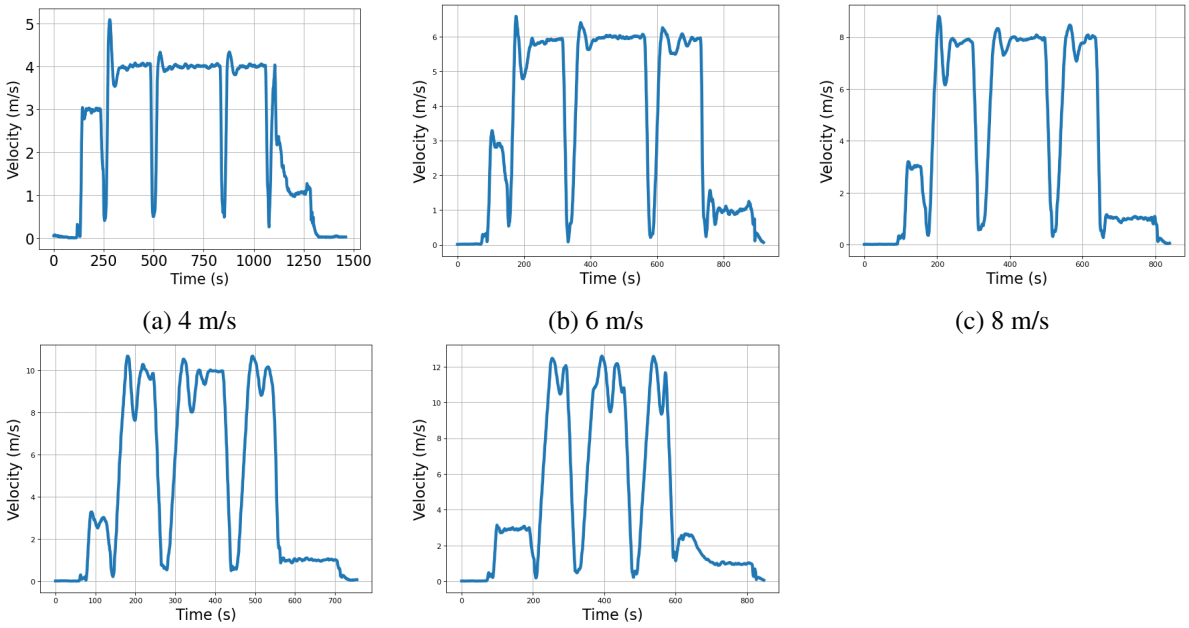
$$EPM = \frac{energy}{distance} = \frac{power}{ground\ speed} = \frac{voltage \cdot current}{ground\ speed}. \tag{1}$$



(a) UAV path (b) Ground velocity

Fig. 1 The flight path and ground velocity of Flight 258 in the data set [10].

As we focus on EPM for a level flight, the data used to train and validate the proposed ML techniques are selected from portions of each flight in the dataset where the velocity does not deviate more than ±0.3 m/s for a significant period, which are assumed to represent level flights.



(a) 4 m/s (b) 6 m/s (c) 8 m/s

(d) 10 m/s (e) 12 m/s

Fig. 2 Representative different ground velocities in the data set [10].

It is also vital to consider how the different velocities perform during a flight. Figure 2 shows the entire flight for each of the five velocities evaluated in the dataset. It can be seen that the period with stable velocity becomes shorter and shorter as the velocity increases, limiting the amount of data available to train an ML algorithm. In this work, we only used the data for the flights where the UAV was flown at 4 m/s and 6 m/s.

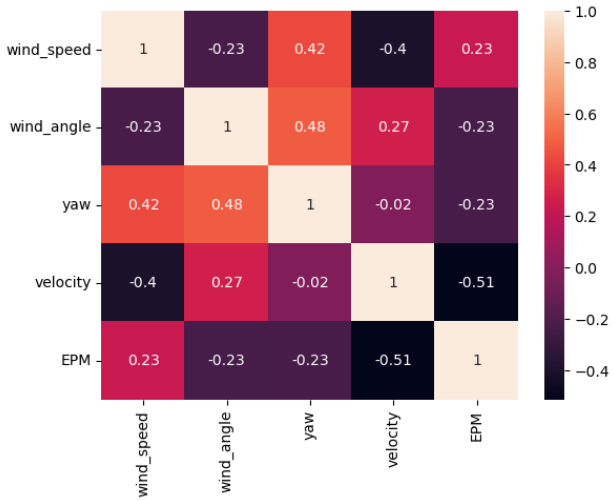


Fig. 3 Linear dependence between pairs of features

III. Methodology

A. Correlation Matrix

Before applying an ML technique to predict the EPM from the given features, we need to understand better how the features and target variables relate. One technique to show the relationship between two variables is the correlation matrix. The matrix uses the Pearson product-moment correlation coefficient to measure the strength and direction between the variables [11] [Add a citation here]. The correlation coefficients range between -1 (perfect negative correlation) and 1 (perfect positive correlation). The closer the absolute value of the correlation coefficient is to one, the more likely the relationship between the variables can be modeled. On the other hand, a correlation coefficient of zero shows no correlation, and it is harder to model the relationship. Figure 3 shows the correlation matrix for EPM and four input features. It can be seen that ground velocity has a more significant correlation with EPM than the other three inputs (or features).

B. Decision Tree Regressor

The decision tree regressor is a nonlinear, supervised learning technique that predicts a continuous target variable [11]. The algorithm utilizes a flowchart-like structure that breaks the samples into smaller subsets. Atop the decision tree is the root node that contains all the samples. The root node is split into multiple children nodes containing a subset of the samples. The splitting process continues until the leaf node can represent the sample. Each node decides the best feature to split the samples up that maximizes the information gain, which is the difference between the impurity of the parent node and the sum of the child node impurities. The information gain is given by

$$IG(D_p, f) = I(D_p) - \sum_{j=1}^{m} \frac{N_j}{N_p} I(D_j), \tag{2}$$

where D_p is the parent node's dataset, f is the feature that will be split, N_j is the number of samples of the j^{th}node, N_P is the number of samples of the parent node, D_j is the dataset as the j^{th}node, and $I(\cdot)$ is the impurity measure of the node calculated by the mean squared error. Decision trees are computationally inexpensive to construct and are fast as the algorithm splits the data based on a feature for each node. However, the trained model tends to overfit and show bias toward the training data and has poor generalization performance.

C. Random Forest Regressor

Random forest regressors [11] are often used to reduce the overfitting problem from the decision tree regressor and capture the continuity and differentiability of the desired prediction. A random forest is a collection of decision

trees where the predicted target variable is the average prediction of all the individual trees. Instead of utilizing all the features when splitting a decision tree's node, features are randomly selected to determine the best information gain from Eqn. (2). The random selection of features allows each decision tree to be constructed differently, improving the model's generalization performance while being less sensitive to outliers in the dataset.

D. Linear Regression

A multivariate linear regression model fits a linear plane to the dataset by learning a set of weights $(w_0, w_1, w_2, ..., w_d)$ that minimized the overall error. The equation for a linear regression model is given by

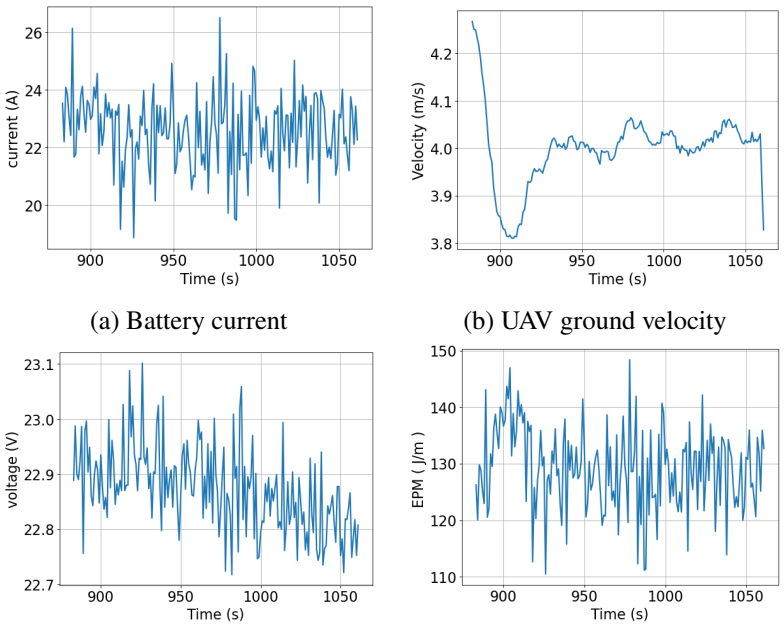
$$\hat{y} = w_0 + w_1 \boldsymbol{x}_1 + w_2 \boldsymbol{x}_2 + ... + w_d \boldsymbol{x}_d, \tag{3}$$

where d is the number of input features [11]. We can see from the equation that changing any weight value will significantly affect the shape of the line used to fit the data. The cost function used to find the line that best fits the data is given by

$$J(\boldsymbol{w}) = \frac{1}{2n} \sum_{i=1}^{n} (\hat{y}(x_i) - y_i)^2 \tag{4}$$

where n is the number of samples, $x_i$is the i^{th}feature, $\hat{y}()$ is the predicted value, and y_i is the truth value of the i^{th}feature [11]. Linear regressor models can produce reasonable estimates for different degrees of polynomials. However, these models are susceptible to outliers.

IV. Results



(a) Battery current

(b) UAV ground velocity

(c) Battery voltage

(d) Computed EPM groud truth

Fig. 4 The computed EPM ground truth, the battery current and voltage, and the UAV ground velocity from flight 258 in the data set [10].

The features used to train and validate the network include the data where the velocity ranged from 3.7 m/s to 4.3 m/s, which consists of 23,162 samples. The data was then split into training and testing datasets using the 60-40 split, with 13,834 data points used for training and 9,328 data points used for testing. In this section, we will describe the results for the decision tree regressor, random forest regressor, and linear regression regressor and discuss the values of the tuned hyperparamers. All models were trained using the scikit-learn library [12], an open-source Python library

for predictive data analysis. The trained models will then be compared to our previous work, where we used a recurrent neural network [13] and a feed-forward neural network to determine the EPM. The two neural networks from the previous work were trained using the Keras library [14].

Before we discuss the results, it is crucial first to examine the actual EPM signal. The results were all validated on flight 258 from the dataset [10]. We chose the flight portion between timesteps 833 and 1062 in flight 258 as it had near-constant velocity. Figures 4a-c shows the components that make up the EPM signal, while the calculated EPM signal from Eqn. (1) is shown in Fig. 4d. We can see from the figures that there is much variation in the current, voltage, and velocity signals, which is credited to the sensors' noise and the UAV's instantaneous velocity. Combining the variations in all three signals when calculating the EPM magnifies the variation observed in the actual signal of Fig. 4d.

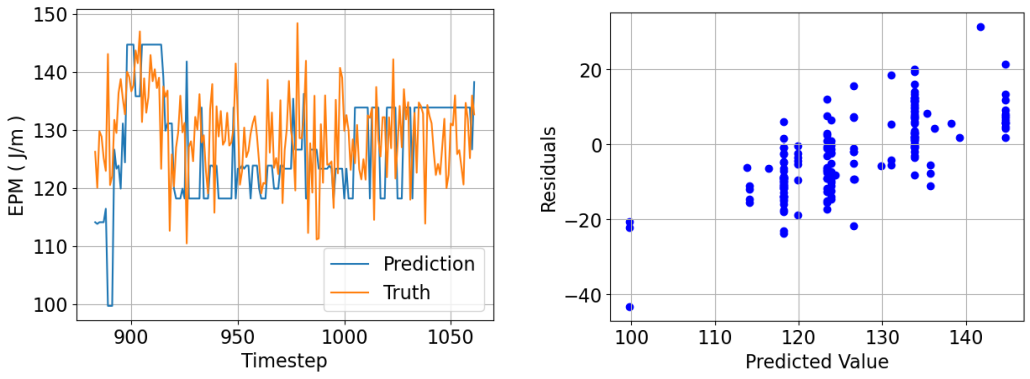
The trained models discussed in this section look to predict the EPM signal shown in Figur 4d. We aim for the trained models to accurately predict the instantaneous EPM along the flight path and closely track this actual EPM signal. Therefore, the models will be evaluated based on the mean squared error between the predicted EPM signal and the actual EPM signal. It is important to note that the sum of the energy for the flight interval will be the total energy necessary for the drone to perform the flight over the interval. The necessary energy for the flight interval shown in Fig. 4d is 23,127.62 J.

A. Results using Decision Tree Regressor

Table 1 Decision Tree Hyperparameters

Hyperparameter	Depth	Minimum Sample Split	Minimum Sample Leaf	Maximum Features
Value	95	28	1	4

The hyperparameters used to train the decision tree regressor are described in Table 1. The depth of the decision tree significantly affects the trained model to overfit the data. As the depth of the tree increases, the model fails to generalize the data as it captures the actual pattern of the training data. Therefore, it is essential to limit the depth of the decision tree to generalize the data accurately. Two additional hyperparameters that can assist with limiting the decision tree's depth are the minimum sample split and minimum sample leaf. Both hyperparameters restrict the growth of the decision tree by preventing the algorithm from splitting internal tree nodes. The minimum sample split specifies the minimum number of samples each tree node must have, while the minimum sample leaf defines the minimum number of samples each leaf node must have. Restricting the minimum samples at each node throughout the tree prevents the model from overfitting and helps generalize the data better. The final hyperparameter we used to train the decision tree regressor was the maximum features hyperparameter. Limiting the features the model can use to split the data at each node helps generalize the data by increasing randomness and limiting the sensitivity to outliers.



(a) Predicted and true EPM evolutions

(b) Residuals vs predicted EPM values

Fig. 5 Validation results using the trained Decision Tree model with the hyperparameters in Table 1.

The predicted EPM from the trained decision tree regressor follows the overall trend of the actual EPM signal as shown in Fig. 5a. The maximum depth hyperparameter prevents the trained model from overfitting. Instead, the trained

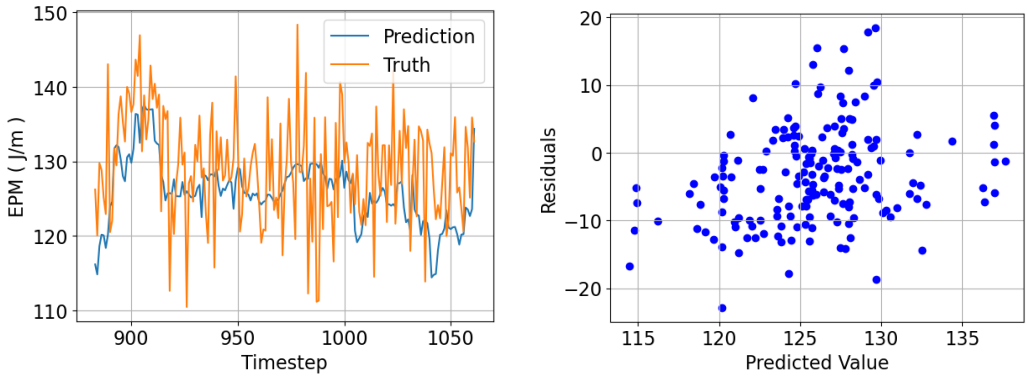
model better generalizes the data, leading to a wider range of data points equivalent to the predicted value. We observe this behavior in Fig. 5a between timesteps 1030 and 1060, where all the input features give the same predicted value. Such behavior occurs in multiple regions of Fig. 5a. Due to the generalization of the data, we see ±20 residuals for predicted values ranging from 110 to 140in Fig. 5b. The error naturally increases significantly when multiple input features equate to the same predicted value. As a result, the decision tree regressor has an MSE of 110.44 and an error variance of 40.36.

B. Results using Random Forest Regressor

Table 2 Random Forest Hyperparameters

Hyperparameter	Number of Estimators	Depth	Minimum Sample Split	Minimum Sample Leaf	Maximum Features
Value	100	99	23	1	4

The random forest regressor has many of the same hyperparameters as discussed with the decision tree. However, the significant difference is that the random forest regressor has a hyperparameter for the number of estimators to specify the number of decision trees in the forest. Each tree in the forest will predict the EPM based on the given features, and the forest will average the results from all the trees. Therefore, the trained random forest regressor can better generalize the data. Table 2 shows the tuned hyperparameter values used to train the random forest regressor.



(a) Predicted and true EPM evolutions

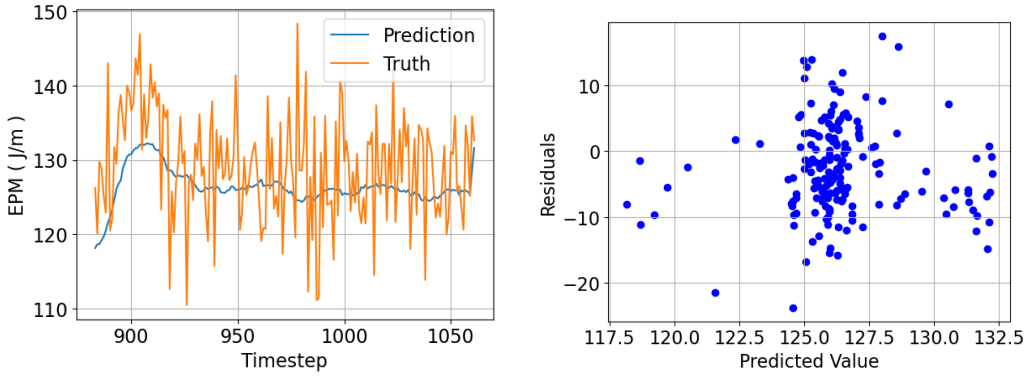
(b) Residuals vs predicted EPM values

Fig. 6 Validation results using the trained Random Forest model with the hyperparameters in Table 2.

Figure 6a shows the results of predicting the instantaneous EPM for a particular flight. We observe that the trained Random Forest model tracks the actual EPM trend more closely than the decision tree regressor results shown in Fig.5a. Random Forest regressors generally have better generalization performance than individual decision trees due to the consensus of the multiple decision trees that make up the random forest. Random forests are also less sensitive to outliers in the dataset. As a result, the trained random forest regressor has smaller resdisuals than the decision tree regressor, as seen in Fig. 6b. The trained model for the random forest regressor has an MSE of 60.56 and an error variance of 22.0.

C. Results using Linear Regression Regressor

The Linear Regression Regressor looks to minimize the residual sum between the actual and predicted values by fitting a linear model to the training data. Unlike the Decision Tree or Random Forest regressors, the linear regressor in the scikit-learn library [12] has no hyperparameters for tuning. Figure 7 shows the results of predicting the instantaneous EPM for the linear regressor. The trained linear regressor model in Fig. 7a follows the general trend of the actual EPM signal with fewer oscillations compared to the Decision Tree and Random Forest regressors. However, Fig. 7b shows that the linear regressor has minimal error overall. The trained model can minimize the error by nearly bisecting through the high oscillation portions of the actual EPM signal, for example, between timesteps 950 and 1,000 in Fig. 7a. The



(a) Predicted and true EPM evolutions (b) Residuals vs predicted EPM values

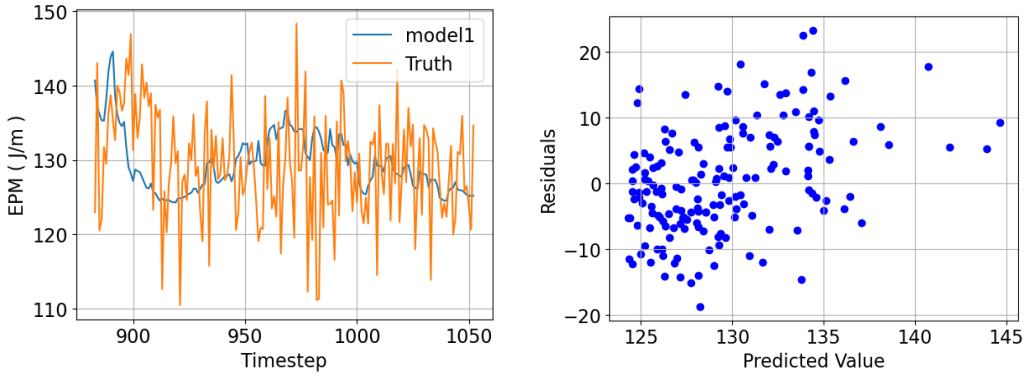
Fig. 7 Validation results using the trained Linear Regression model.

predicted EPM signal behaves as an average between the oscillations, reducing the overall error for the flight interval. The predicted EPM for the linear regression model produces an MSE of 54.98 and an error variance of 18.64.

D. Results using RNN

Table 3 Recurrent Neural Network Parameters

Parmeters	Learining Rate	Loopback	Epochs
Values	0.001	5	10,000



(a) Predicted and true EPM evolutions (b) Residuals vs predicted EPM values

Fig. 8 Validation results using the trained RNN model.

The results using the Decision Tree, Random Forest, and Linear Regression regressors are to be compared with RNN [13] [Add citation], which was applied in our previous work for EPM prediction. RNN can handle sequential data and inputs of varying lengths. RNNs can store information from previous inputs by maintaining crucial information to generate a sequence of outputs. Table 3 shows the training setup for the RNN. We trained the neural network with the stochastic gradient descent method, Adam, with a learning of 0.001. The learning rate is critical as it determines the step size taken toward the global/local minimum. Choosing a small learning rate increases the risk of falling into a local minimum, while a more significant learning rate increases the risk of overshooting the global minimum. The neural network also utilizes the sigmoid activation function for each perceptron, which outputs values between zero and one. We choose the sigmoid activation function as EPM can only be positive values. Another important hyperparameter is

the lookback, which determines the number of previous time steps to predict the next time step. We trained the neural network with a lookback of fifteen to provide the neural network with sufficient data to observe a pattern and predict the EPM for the next time step.

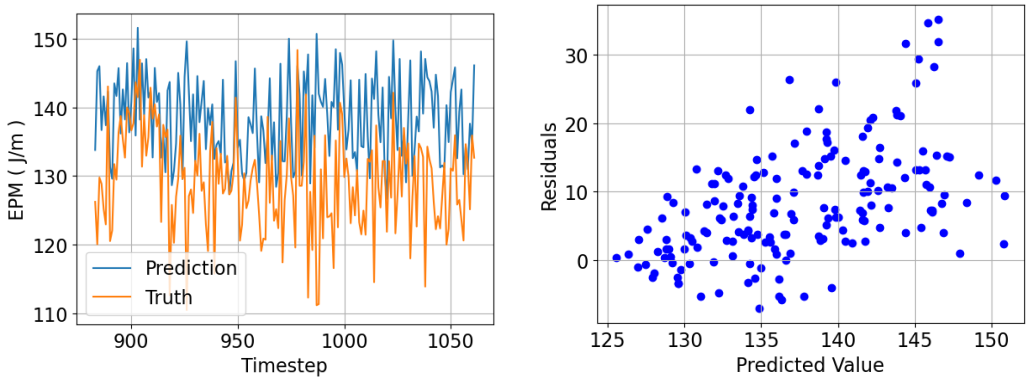
Figure 8 shows the training results for the RNN, which was trained for 1,500 epochs. The results in Fig. 8a show that the trained RNN model tracks the overall trend of the actual EPM signal. The errors are reflected in Fig. 8b. The errors is comparable to the results using the Random Forest regressor. The trained RNN model had an MSE of 65.22 with an error variance of 22.9.

E. Results using FFNN

We applied a feed-forward neural network (FFNN) to predict EPM [Add citation]. An FFNN is a biologically inspired network of perceptron that sends data from the input layer through multiple nodes in the hidden layer until it reaches the output node. Every node in the hidden and output layers calculates a weighted sum of all the inputs obtained from the node(s) in the previous layer. The learned weight will influence the node's consideration of the associated input. FNNs are designed to identify patterns in data and have the ability to handle noisy data.

Table 4 Feed Forward Neural Network Parameters

Parmeters	Learining Rate	L1 Regularization	L2 Regularization	Epochs
Values	$1 * 10^{-7}$	0.01	0.01	20,000



(a) Predicted and true EPM evolutions (b) Residuals vs predicted EPM values

Fig. 9 Validation results using the trained FFNN model.

Table 4 shows the parameters for the FFNN. The neural network was trained with the stochastic gradient descent method, Adam, with the learning of 1e-7. The neural network also utilizes the Relu activation function for each perceptron, which outputs values between zero and one. Another important hyperparameter is kernel regularization, which allows the model to converge optimally by applying penalties on the layer's bais, kernel, and output parameters. The neural network was trained for a total of 20,000 epochs. Since we wish to minimize the difference between the actual and predicted values, we train the network to minimize the mean squared logarithmic error cost function.

Figure 9 shows the results for the FFNN. The results show that the neural network can train a model to predict the EPM. Fig. 9a shows the predicted instantaneous EPM compared to the actual instantaneous EPM for the flight interval. It can be seen that the predicted EPM follows the trend of the true EPM but with much larger oscillations compared to other applied ML techniques. The residuals in Fig. 9b reflect the error for each predicted value. The residuals range from -10 to 30, approximately, reflecting the largest residual distribution among all proposed ML techniques. We calculate an MSE of 70.22 with an error variance of 53.24 for the FFNN results.

F. Comparsion

Table 5 shows the results for each machine learning technique, with the MSE and error variances plotted in Fig. 10. Although all the proposed algorithms can track the overall trend of the true EPM signal, the Linear Regression regressor

Table 5 Results of the Trained Models

Model	MSE	Error Variance	Training Time (Sec)
Decision Tree Regressor	110.44	40.36	6.1 sec
Random Forest Regressor	60.56	22.0	28.1 sec
Linear Regression Regressor	54.98	18.64	1.8 sec
Recurrent Neural Network	65.22	22.9	12 hr. 2 min. 8 sec.
Feed Forward Neural Network	70.22	53.24	6 hr. 13 min. 20 sec.

outperforms other techniques with the least MSE and error variance. The linear regression regressor had the overall lowest MSE and error variance. However, the random forest regressor tracked the variations within the actual EPM signal better than any other trained model. The better signal tracking in the random forest regressor came at the cost of a significantly higher training time. The linear regression regressor captured the general trend in far less training time.

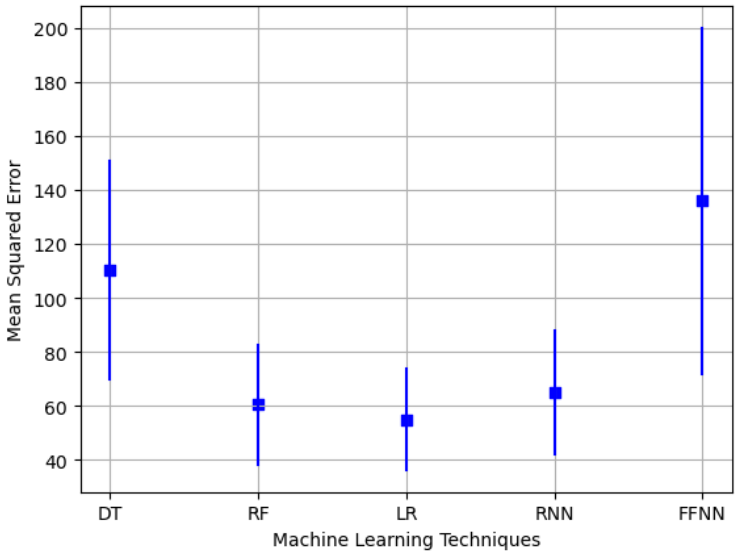


Fig. 10 MSE and error variance for the trained models using Decision Tree (DT), Random Forest (RF), Linear Regression (LR), Recurrent Neural Network (RNN), and Feed-Forward Neural Network (FFNN). Each point represents the MSE for the respected machine learning technique, and the lines extending the points represent the error variance.

V. Conclusion

In this paper, we implemented and compared various state-of-the-art machine learning (ML) techniques to predict the flight energy consumption of an Uncrewed Aerial Vehicle (UAV). These ML algorithms are the Decision Tree regressor, Random Forest regressor, Linear Regression regressor, Recurrent Neural Network, and Feed-Forward Neural Network. The inputs of the model are selected as wind speed, wind angle, UAV heading (yaw) angle, and UAV ground speed. A real-world flight test data set is used to train and validate the ML models. The results generated by all applied ML algorithms can follow the trend of the true EPM signal. The comparison results reflect that the Linear Regression

method outperforms all other algorithms on the basis of the Mean Squared Error (MSE) and the error variance.

We look to extend the work by fine-tuning the hyperparameters for each proposed ML algorithm. This work looked at flight intervals with constant velocity near four m/s. We will use data of a broader and varying velocity range for training and testing.

References

[1] Federal Aviation Administration, "Working to Build a Net-Zero Sustainable Aviation System by 2050," https://www.faa.gov/sustainability#:~:text=2021%2C%20U.S.%20Transportation%20Sec.,large%20part%20of%20the%20solution., accessed in Dec 2023.

[2] Federal Aviation Administration, "Advanced Air Mobility Implementation Plan," https://www.faa.gov/air-taxis/implementation-plan, accessed in Dec 2023.

[3] Johnson, W., and Silva, C., "NASA concept vehicles and the engineering of advanced air mobility aircraft," *The Aeronautical Journal*, Vol. 126, No. 1295, 2022, p. 59–91. https://doi.org/10.1017/aer.2021.92.

[4] Goyal, R., Reiche, C., Fernando, C., and Cohen, A., "Advanced air mobility: Demand analysis and market potential of the airport shuttle and air taxi markets," *Sustainability*, Vol. 13, No. 13, 2021, p. 7421.

[5] Cohen, A. P., Shaheen, S. A., and Farrar, E. M., "Urban air mobility: History, ecosystem, market potential, and challenges," *IEEE Transactions on Intelligent Transportation Systems*, Vol. 22, No. 9, 2021, pp. 6074–6087.

[6] Masuch, N., Keiser, J., Lützenberger, M., and Albayrak, S., "Wind power-aware vehicle-to-grid algorithms for sustainable EV energy management systems," *2012 IEEE International Electric Vehicle Conference*, IEEE, 2012, pp. 1–7.

[7] Zhang, J., Campbell, J. F., Sweeney II, D. C., and Hupman, A. C., "Energy consumption models for delivery drones: A comparison and assessment," *Transportation Research Part D: Transport and Environment*, Vol. 90, 2021, p. 102668.

[8] Dorling, K., Heinrichs, J., Messier, G. G., and Magierowski, S., "Vehicle Routing Problems for Drone Delivery," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, Vol. 47, No. 1, 2017, pp. 70–85. https://doi.org/10.1109/TSMC.2016.2582745.

[9] Shavarani, S. M., Nejad, M. G., Rismanchian, F., and Izbirak, G., "Application of hierarchical facility location problem for optimization of a drone delivery system: a case study of Amazon prime air in the city of San Francisco," *The International Journal of Advanced Manufacturing Technology*, Vol. 95, 2018, pp. 3141–3153.

[10] Rodrigues, T. A., Patrikar, J., Choudhry, A., Feldgoise, J., Arcot, V., Gahlaut, A., Lau, S., Moon, B., Wagner, B., Matthews, H. S., et al., "In-flight positional and energy use data set of a DJI Matrice 100 quadcopter for small package delivery," *Scientific Data*, Vol. 8, No. 1, 2021, pp. 1–8.

[11] Raschka, S., and Mirjalili, V., *Python machine learning: Machine learning and deep learning with Python, scikit-learn, and TensorFlow 2*, Packt Publishing Ltd, 2019.

[12] "scikit-learn: machine learning in Python," https://scikit-learn.org/stable/index.html, accessed in Dec 2023.

[13] Samiei, A., Selje, R. A., and Sun, L., "Distributed Limited Resource Allocation and Energy-Expenditure Learning for Advanced Air Mobility," *AIAA SCITECH 2023 Forum*, 2023, p. 1471.

[14] "Keras: Deep Learning for Human," https://keras.io/, accessed in Dec 2023.