Classifying Cognitive States of Pilots In-Flight

Spencer Solomon

1. Background and Introduction

In the past few years, there has been a startling increase in incidents in the US aviation industry, particularly with runway incursions [6]. While none of these incidents has resulted in an accident, the sharp uptick in frequency is very alarming and has caused many to question the training and fitness of American pilots and Air Traffic Controllers. The causes of these incidents are extremely varied but a consistent theme through all of them is some degree of pilot error. As a result of the post-COVID pilot shortage, pilot's are in increasingly large demand resulting in more pressure from the airlines for pilots to work long hours and ensure flights are completed on-time so as to not jeopardize other flights. The FAA, the government body which regulates airplanes and airlines, has strict laws around pilot rest time and the amount of flight hours a pilot can clock within certain time intervals. These time limitations are especially strict if the pilot is operating through their window of circadian low, typically around 3AM in the pilot's local time [6]. In addition to these time limitations, pilots must fill out self reported surveys before each flight, detailing how they are feeling mentally and physically. These surveys must be reviewed and approved by their co-pilot and dispatcher before flying.

While these strategies have been very successful in reducing accidents, the recent string of near-misses may indicate that they are not stringent enough to prevent pilot fatigue from becoming a factor in a future commercial aviation accident. In addition, both of these mitigation techniques are preventative and can only catch dangerous situations on the ground, during the pre-flight preparations. For this reason, much research and development has been put towards developing a system for detecting changes in a pilot's mental state inflight, allowing pilots to be alerted about potentially dangerous mental situations, just like they would be alerted about any mechanical issues with the airplane. The goal of this project is to develop a Machine Learning model which uses physiological data obtained from pilots in-flight to predict the mental state of pilots in real time.

2. Data and Preprocessing

The data used for this project was a Kaggle dataset containing 4.9M timestamps of training data, consisting of 26 relevant feature columns and one target column. In order to obtain this data, six crews of two pilots, a pilot and copilot, were hooked up to physiological measurement devices and subjected to three scenarios in a realistic flight simulator. There were four possible mental states for the pilot, which were recorded at each timestamp. The first state is Baseline (A), where the flight is going as planned and the pilot's attention is properly focused. Next, is Channelized Attention (B), where pilots focus too much on one specific issue and ignore other responsibilities. The final two states are Distracted Attention (C), where pilots stop paying attention to one task to think about

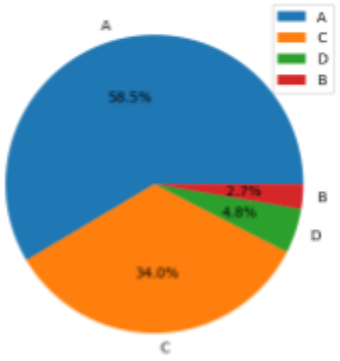

Figure 2.1 - Unbalanced Data

another and Surprised/Startled (D), which occurs when pilots suddenly find themselves in an unexpected situation.

The physiological data used for training was primarily obtained using electroencephalogram (EEG) and electrocardiogram (EKG) readings, as well as Respiration and Galvanic Skin Response monitors. Most of the relevant data comes from the EEG sensors, which measure electrical activity in the brain. In order to utilize this data for classification, it had to first be cleaned and preprocessed. After performing some basic operations on the dataset, the distributions of each data column were plotted, which showed a lot of outliers and noise in the data. In order to combat this, mean imputation was performed on the maximum and minimum 5% of each column, reducing their values to the mean value for the column. This

	Accuray	F1-Score
Balanced Data	.3377	.3333
Unbalanced Data	.5756	.4488

Figure 2.2

technique allows us to get rid of outliers and attain cleaner data distributions for the features, without throwing away a lot of data. Another issue was that the data was extremely unbalanced, with most of the training timestamps being of the pilots baseline state (Figure 2.1). Since the dataset was already so large, this issue was combatted using undersampling, leaving the dataset perfectly balanced across all four classes. This was tested empirically by training a logistic regression classifier on the balanced and unbalanced data (Figure 2.2). While the Accuracy and F1 Score were higher for the unbalanced dataset than the balanced one, looking at the accuracy class by class reveals a different picture (Figure 2.3). When looking at these values, we can see that the classifier trained on unbalanced data does very well at predicting the majority class, but extremely poorly when predicting the minority classes. In our use case, it is much more important to classify

Accuracy By Class	A	B	C	D
Balanced Data	.4452	.2195	.3414	.3248
Unbalanced Data	.9559	0	.1000	0

Figure 2.3

the minority states, so the balanced training set was used moving forward.

The final aspect of preprocessing explored was the potential for reducing the dimensionality of the data by utilizing PCA. The principal components of the data were calculated and their magnitudes graphed and we found that almost all of the variance in the target data could be explained by the first principal component. Once again, this change was tested empirically on a Logistic Regression model, comparing the original data, the top 15 principal components and the top 1 principal component (Figure 2.4). We found that the original data distribution produced the best results and decided to use this data moving forward, as the number of features in the data was small enough to train on the original data points.

	Accuracy	F1-Score
Original Data	.3377	.3333
Top 15 Components	.3118	.2990
Top 1 Components	.2920	.2002

Figure 2.4

3. Approach and Implementation

While performing the analysis on the data, a Logistic Regression model was used

as a baseline in order to compare results and iteratively improve the data preprocessing techniques. However, this type of model does not take the time dimension of the data into account, and instead treats each timestep as an individual data point. In order to capture this modality of the data, we trained models which train and predict on time series of data, instead of singular data points. These models capture information about how previous problem states impact future ones, and make predictions based on a series of readings, hopefully increasing their efficacy on the problem. Specifically, a Gaussian Process model and a Recurrent Neural Network were trained on the full time series data for each of the test flights.

The first model utilized was the Gaussian Process model. The Gaussian Process model assumes that each of the given features can be modeled as a Gaussian Distribution and attempts to model the Multivariate Gaussian Distribution of the problem space [2]. This distribution is defined by its mean, μ, and its variance, σ^2. These values are described by the following equations:

$$\mu_{out} = \mu_{test} + (K_{train-train})^{-1}(y_{train} - \mu_{train})$$

$$\sigma^2_{out} = K_{test-test} - K_{test-train}(K_{train-train})^{-1}K_{train-test}$$

where K is the covariance matrix between two sets of feature data. These covariance matrices encode the "distance" between the two datasets and are calculated using a kernel function, which is a hyperparameter of the algorithm. For this project, the Radial Basis Function Kernel and the Rational Quadratic Kernel were tested and compared. The RBF kernel was found to be more effective and was used for evaluation of the model. Unfortunately, while training the model, it was discovered that the full dataset was much too large to hold the precalculated covariance matrices in memory. As such, 2000 samples were drawn from the balanced dataset on which the models were trained. This likely had a large impact on the models performance but goes to show that non-parametric, statistical solutions, such as a Gaussian Process, often do not scale well to very large datasets.

The second model trained was a Recurrent Neural Network, consisting of four Long-Short Term Memory (LSTM) layers. LSTM layers are used instead of standard RNN layers to reduce the issue of vanishing gradients. In standard RNN classifiers, calculating the gradient of a long chain of events precisely becomes increasingly difficult the longer the time series is. As such, the impacts of events far in the past tend to vanish from the model over time. To combat this issue, LSTM layers attempt to learn what aspects of the data are important and which are not, so that the model can "remember" the important information from previous observations and "forget" the useless aspects. In addition to the LSTM layers, Dropout layers are used for regularization, ensuring that the model does not overfit to the training data by randomly removing some of the training sample after each of the layers of the model.

Finally, a Deep Neural Network was trained on the data, to evaluate whether training on the time component of the data improved or worsened model results. In order to test this accurately, the model's architecture was designed to match the architecture of the Recurrent Neural Network but instead used Dense layers with the Rectified Linear activation function, as opposed to the LSTM layers used in the RNN. The model retained the same Dropout layers for regularization as the RNN model. Both models were trained for 50 epochs. By only changing the recurrent layers of the model, we can ensure that any performance differences between the models can be attributed to the merit, or lack thereof, of

using time series based models for the problem specification.

4. Results & Evaluation

After each of the models was trained, its performance was evaluated on a blind test set, and metrics were calculated in order to compare the model performance. Specifically, the metrics used for comparison were Accuracy, F1-Score and the Class-by-Class Accuracies. The Accuracy allows us to get a quick glimpse at the model's performance across all of the data samples. However, this score can be very negatively influenced by unbalanced datasets, allowing models which just predict one class for every record to have a high score. In order to mitigate this, F1 Score was used as the primary decision metric for comparing models. F1 Score balances precision and recall, allowing it to weed out models which have overfit or underfit to the training data. Additionally, F1 Score does much better at representing the model's performance on unbalanced datasets. As a final sanity check, the Accuracy for each class was calculated separately and compared. This allowed us to see when models were getting high scores based only on their ability to predict one class, and allowed us to see some very interesting trends.

	Accuracy	F1 Score	A	B	C	D
Logistic Regression	.34	.33	.45	.22	.34	.32
Gaussian Process	.94	.92	.99	.01	0	0
Recurrent Neural Network	.29	.15	.97	.04	.02	.00
Deep Neural Network	.81	.80	.56	.96	.88	.89

Figure 4.1

The results of this testing (Figure 4.1), lead to some very interesting conclusions about the data and the models used on it. The first thing that jumps out from the results is the performance of the Gaussian Process model. While the model had the highest Accuracy and F1 Scores, it performed extremely poorly on the minority classes. This points to an issue in how the data was being sampled for the training and means that despite its seemingly high metrics, this model actually performed very poorly on the problem. The fact that the Gaussian Process model scored so highly on the F1 Score also points to some issues with using this metric as the primary metric for comparing the remaining models.

Perhaps the most interesting conclusion came from comparing the RNN to the similarly designed DNN. Interestingly, the RNN performed much worse on the test data than the DNN. These were not the results we expected going into the project, as we expected the algorithms which could model the full dimensionality of the problem space would be more successful at predicting the solution. However, it seems that on this problem, attempting to encode the transitions between the data points actually adds more noise and complexity to the models and makes it more difficult to truly learn how the features impact the target at each timestep. Since the DNN performed so well on the test data, we decided to do a more thorough search of the hyperparameter space for the DNN. We trained models with one Fully Connected Layer, a Dropout Layer and an Output layer, and tested a range of sizes for the fully connected layer, activation functions, dropout ratios and learning rates. In the end, we determined that the optimal hyperparameters were 128 units, sigmoid activation function, .01 dropout ratio and .01 learning rate. This model achieved results which slightly exceeded those achieved by the original DNN, but required a much less complex model to achieve them (Figure 4.2).

	Accuracy	F1 Score	A	B	C	D
Original DNN	.81	.80	.56	.96	.88	.89
Optimal DNN	.82	.82	.67	.95	.86	.84

Figure 4.2

5. Conclusion

While the results achieved through this project were not inline with what was expected going into it, there is still much to be learned and much future work to be done. While the results attained by the Deep Neural Network indicate that our models may soon be able to effectively predict the pilots mental states, there is still a lot of work to be done before this technology is ever used in commercial cockpits. Primarily, the current sensing equipment necessary to obtain the physiological data in real time is prohibitively invasive for use in the cockpit everyday. Additionally, there is much research and development which must be done in order to determine how this information should best be presented in the cockpit, allowing pilots to easily interpret the alert, without it distracting or confusing them further. Despite these issues, it seems increasingly likely that these technologies will be developed and implemented in the near future, allowing for these models to be integrated into plane's computer systems and making the skies a little bit safer for us all.

Works Cited

[1] Advisory circular 117-3 - federal aviation administration, https://www.faa.gov/documentLibrary/media/Advisory_Circular/AC%20117-3.pdf (accessed Dec. 3, 2023).

[2] "How does the gaussian process regression work?," Quora, https://www.quora.com/How-does-the-Gaussian-process-regression-work (accessed Dec. 3, 2023).

[3] L. Astolfi and R. W. Backs, "Measuring neurophysiological signals in aircraft pilots and car drivers for the assessment of mental workload, fatigue and drowsiness," Neuroscience & Biobehavioral Reviews, https://www.sciencedirect.com/science/article/abs/pii/S0149763412001704 (accessed Dec. 2, 2023).

[4] Toward the "Cognitive Cockpit": Flight Test platforms and methods for ..., https://percept.eecs.yorku.ca/papers/0830best_Schnell_paper.pdf (accessed Dec. 2, 2023).

[5] "Reducing commercial aviation fatalities," Kaggle, https://www.kaggle.com/competitions/reducing-commercial-aviation-fatalities/overview (accessed Dec. 2, 2023).

[6] M. O'Brien, K. Cuevas, and G. Bennett, "What's behind the alarming rise in near-collisions of commercial airplanes," PBS, https://www.pbs.org/newshour/show/whats-behind-the-alarming-rise-in-near-collisions-of-commercial-airplanes (accessed Dec. 2, 2023).