Real-Time Analytics to Support Operational Decision Making (Predictive Modeling)

Dejan Neskovic
Booz Allen Hamilton
McLean, Virginia
neskovic_dejan@bah.com

Jerry Scott Sheehan
Booz Allen Hamilton
El Segundo, California
sheehan_jerry@bah.com

Alec Gray Jr.
Booz Allen Hamilton
Arlington, Virginia
grayjr_alec@bah.com

ABSTRACT

In the current global threat environment, homeland security depends on both domain and situational awareness. The probability of a secure homeland is based on conditional probabilities describing domain and situational awareness. Data-driven models that make sense of large volumes of information available increase probability estimates. With data continuing to grow in scope and complexity, the Department of Homeland Security (DHS) Science and Technology (S&T) and DHS Customs and Border Protection (CBP) Air and Marine Operations Center (AMOC) for National Air Domain Security require innovative strategies, services, and technologies to unlock the value of their data analytics potential. This paper describes an approach for doing predictive threat models (PTM) where multiple classification and deep learning models are tested and evaluated. Some of the models used include multilayer perceptron (MLP) classification, adaptive boosting, and artificial neural networks (ANN). The top performers are then selected and deployed using modern machine learning operations (MLOps) best practices such as automated pipelines, continuous integration/continuous deployment (CI/CD), and model performance evaluation. Doing so allows the models to easily adapt to and accommodate changes in mission priorities. This helps DHS fulfill various missions such as providing real-time predictive analytics to operators for faster and better decision making. The models can also be used as stand-alone tools to predict future trends/events and provide support for tactical resource-allocation decisions.

ABOUT THE AUTHORS

Dejan Neskovic is an aviation subject matter expert (SME) and Senior Lead Scientist/Task and Modeling Lead for Booz Allen Hamilton supporting the Director of Modeling and Simulation Technology Center (MS-TC). He provides strategic engagement advice and guidance to MS-TC on coordination with internal DHS components and other Department of Defense and civilian agencies on projects and technology development opportunities. His current focus is on practical application of predictive modeling such as PTM. Mr. Neskovic holds a Bachelor of Science in Electrical Engineering from the University of Tennessee, Knoxville.

Scott Sheehan received his bachelor's degree in Applied Mathematics and Statistics from Johns Hopkins University. He currently works as an Advanced Lead Data Scientist at Booz Allen Hamilton where he pairs his mathematics background with eight years of combined industry and government experience performing professional data and business analytics to support a variety of government projects. He is the Technical Lead for the PTM contract where he assists the decision-making process at AMOC through innovative, quality, and timely modeling solutions.

Alec "AJ" Gray holds a master's degree in Data Analytics Engineering from George Mason University. He is currently an experienced Machine Learning Engineer and Python SME at Booz Allen Hamilton who leads the Data Science workflow under Scott Sheehan in support of the PTM contract. Applying advanced expertise in machine learning and deep learning, he manages and optimizes a deep ANN and other modern predictive models, analyzing model predictions with interactive dashboards to continue model improvements.

Real-Time Analytics to Support Operational Decision Making (Predictive Modeling)

Dejan Neskovic
Booz Allen Hamilton
McLean, Virginia
neskovic_dejan@bah.com

Jerry Scott Sheehan
Booz Allen Hamilton
El Segundo, California
sheehan_jerry@bah.com

Alec Gray Jr.
Booz Allen Hamilton
Arlington, Virginia
grayjr_alec@bah.com

INTRODUCTION

Background

The evolving threat of illegal smuggling and entry along the U.S. southern border requires efficient threat classification and resource allocation (Rosenblum et al., 2013). The U.S. southern border remains a primary entry point for illegal drugs into the United States. Transnational criminal organizations, particularly drug cartels, employ sophisticated methods to smuggle narcotics such as cocaine, methamphetamine, heroin, and fentanyl across the border using means such as vehicles, tunnels, and aircraft. Ninety percent of fentanyl flows through our southern border. U.S.-Mexican cooperation on drug trafficking has faced significant challenges over the past decade, and between fiscal years 2017 and 2021, fentanyl trafficking offenses increased 950% (U.S. Sentencing Commission, 2021).

The CBP AMOC Detection Enforcement Officers (DEO) are tasked with diminishing drug trafficking through aviation. DEOs do so through threat classification where they determine which flights are likely to pose a risk to the country, establishing suspicious flight profiles, and assessing the risk level they pose to national security. Threat classification helps prioritize and allocate resources to each threat effectively. DEOs classify flights following patterns seen previously that led to illegal smuggling of drugs or people such as unauthorized intrusions into U.S. airspace and suspicious aircraft behaviors. Once a flight is classified as a high risk to the country, DEOs determine the destination of the flight; this way they can intercept the flight and prevent drugs from entering and being distributed in the U.S.

The job of a DEO is complex. Thousands of flights enter the U.S. each day, and the classification, resource allocation, and final destination must all be determined in a short window. Handling so many flights at once and making complex decisions in real time poses challenges as research has shown that individuals can effectively handle only up to five tasks at a time (Cowan, 2010). Considering the combination of temporal and geospatial aspects, the complexity of data availability, format, and model input variables, this task becomes increasingly difficult, even for experts in the field. Additionally, DEOs heavily rely on historical information and personal experience to make decisions. However, the high turnover rate among staff leads to a loss of valuable knowledge when employees leave their positions.

Use of Big Data and Machine Learning

Big data and machine learning (ML) have the potential to significantly enhance the support provided to DEOs in threat classification and resource allocation activities. Currently, DEOs must rely on their own knowledge, remembering previous patterns they have seen and applying those that are applicable to the current tracked flights. The number of previous patterns seen is more expansive for some DEOs than others but is still limited to the knowledge of one officer. By building an ML model that can view all historical flight data, it is able to learn the pattern of every flight quicker than a human and remember them all at once. Using this knowledge, the ML model acts as a force multiplier, joining the experience and minds of all current and previous enforcement officers to make these classifications. By using these models, each DEO will have that same expansive knowledge allowing them to make more informed and quicker decisions on which flights to intercept, resulting in more flights being intercepted, and fewer resources being used. This, in turn, increases the utilization of available assets and enables leadership to make more strategic decisions, optimizing the allocation of human resources and assets while developing better data-driven acquisition strategies.

A key aspect of using big data and building ML models as described above is the ability to rapidly develop, scale, and deploy predictive solutions at both the department and component level. This includes using available data sources,

processing and integrating the data, and applying predictive modeling analytics to support various core mission areas of the DHS. Currently, many components lack the comprehensive data architecture, collections, and storage mechanisms necessary for developing more comprehensive predictive analytics. By addressing these limitations, the DHS can harness the power of big data and ML to support mission-critical operations. Doing so allows the ML models discussed above to be integrated in real time into DEOs' decision-making processes to swiftly classify flights, establish suspect flight profiles, and assess the risk they pose to national security.

Models Used

To assist DEOs with making quicker and better-informed threat classification and landing location decisions, two types of ML models were implemented: classification models and a deep learning model. These models were created by training on historical data of flights entering U.S. airspace. Classification models were used to predict the (a) level of risk a flight poses to the nation, (b) type of mission it is on, and (c) possible landing area. The classification models used for this research included decision trees, Naive Bayes, logistic regression, support vector machines (SVM), MLP classification, and adaptive boosting. All these models were well suited for use cases a, b, and c, respectively. After the models were trained, we compared the accuracy of each and then selected the top performer using a predefined set of criteria. Each of our models, a, b, and c, could have a different ML model technique based on their performance for the particular need. After the classification model deems a flight a high enough risk to require interdiction, the deep learning model will determine the potential landing area by predicting coordinates along with the surrounding area of uncertainty that could then be used to pick out the most likely airstrip destination.

METHOD

Model Development

The predictive models were developed using six-steps that follow a similar approach to those performed in previous studies (e.g., Wei et al., 2019): (1) Data capture, (2) Extract, Transform, Load Phase (ETL) and Feature Engineering, (3) Feature Selection, (4) Building and training models, (5) Selecting the best models, and (6) Deploying the models.

Step 1: Capture Data:
Collecting and capturing data in a secure manner is a crucial foundational step in the process of building a reliable and effective model. It involves gathering the necessary data from various sources and helping to ensure its integrity, quality, and confidentiality. For this study, Amazon Web Services (AWS) GovCloud was used to capture and consolidate government data of historical flights of interest collected by radar detection. AWS GovCloud is a specialized cloud computing environment designed to meet specific regulatory requirements and security standards for government agencies and other highly regulated industries. AWS GovCloud helps ensure the data collection process adheres to the stringent security measures and compliance standards necessary to handle sensitive information.

Step 2: ETL and Feature Engineering:
The next step is the ETL phase and feature engineering. To achieve this, a custom automated ETL pipeline was developed to streamline the collection, processing, and transformation of the data.

The pipeline retrieves the relevant data from AWS GovCloud and then cleans the data by handling missing values, outliers, and inconsistencies. Additionally, data normalization, standardization, and scaling techniques are applied to bring the data to a common scale or range, enabling fair comparisons between different features.

Feature engineering was then performed to create enhanced training features not available in the raw data with high predictive value. This allowed for the creation of model training data that has greater predictive value than the sum of its parts and the ability to uncover latent patterns and capture complex relationships that may not be readily apparent in the raw data alone. By enriching the training data with these valuable features, it enhanced the modeling process and enabled the models to make more accurate predictions.

Step 3: Feature Selection

After the data was prepared and new features were created, feature selection was performed. Feature selection is a technique used to identify the most relevant features or variables from a dataset that contributes significantly to the variability of the target variable. It helps in reducing dimensionality by selecting a subset of features that have a strong association with the target variable and discarding those features that are irrelevant or redundant. In doing so, it enhances model performance, reduces computational complexity, and improves interpretability.

For this study, Analysis of Variance (ANOVA) and Spearman were engaged to perform feature selection. ANOVA can be used when both the target variable and the features are numeric, to assess the relationship between the target variable and each individual feature. It examines whether the variation in the target variable can be explained by the variability in the numeric features. Since not all our features are numeric, one-hot encoding was required in the ETL pipeline to transform any categorical features into numeric. Once that was complete, ANOVA was used to calculate the F statistic and associated p-value to determine the statistical significance of the relationship between the target variable and the features.

Whereas ANOVA performs well for linear relationships, Spearman performs well for nonlinear and nonnormally distributed data and is therefore a good comparison for our models. To perform this analysis, Spearman uses statistical measures to assess the strength and direction of the monotonic relationship between two variables. It is therefore able to compare the target variable to each feature and determine those most strongly associated with each other.

Both feature selection processes were performed in this study. After collecting the features that met a given threshold and comparing the performance of both, the better performing features were identified.

Step 4: Building the Predictive Threat Models:
The PTM models were split into two parts: classification and deep learning.

Classification Models
Classification models provide valuable insights by categorizing data into distinct classes or categories (Nikam, 2015). The primary objective of classification models is to identify patterns and relationships within datasets, enabling the prediction of class labels for unlabeled instances based on their features. The power of classification models lies in their ability to convert raw data into actionable insights. These models enable decision makers to understand and interpret large volumes of information, leading to informed decision making and improved outcomes (James et al., 2013). Classification models are commonly used for multiple forms of analysis such as email spam detection where it is used to classify an email as either “Spam” or “Not Spam.”

As discussed previously, the classification models used for this study predict the risk level of a flight, the particular mission it is on, and bucketed values for latitude and longitude. The risk classification involves binary categorization into high or low risk, while the mission classification aims to identify the purpose of the flight. The bucketed latitude and longitude values provide a range for each coordinate, so rather than predicting 31.54° latitude, it would predict 30–32° degrees, for example.

ML classification models have proven superior performance compared with traditional statistical methods. Previous studies have identified Naive Bayes, logistic regression, SVMs, adaptive boosting, and random forest as some of the best-performing models for classification problems (Chilyabanyama, et al. 2022), along with MLPs. All of these models are described in Table 1.

Table 1. Classification Models

Model Type	Description
Naive Bayes	Naive Bayes is advantageous due to its lack of presumptions about relationships among data elements (Domingos & Pazzani, 1997). Bayesian modeling is part of a class of modeling methodologies called adaptive systems that are continuing to learn to improve decision making. Organically built data-driven models offer much greater chance of success because this approach

	minimizes the expert's subjectivity of preconceived notions about how the variables relate and impact each other.
Logistic Regression	Logistic Regression is a commonly used method for binary classification, estimating the probability of the outcome based on input variables.
SVM	SVM creates a linear model to separate data into classes using a line or hyperplane in a high-dimensional space.
Random Forest	Random forest constructs multiple decision trees through bagging, reducing variance and improving prediction accuracy.
Adaptive Boosting	Adaptive Boosting is developed for problems that require binary classification and can be used to improve the efficiency of decision trees. It iteratively builds weak classifiers or decision stumps and weights them based on their performance to create a stronger overall classifier. Adaptive Boosting is generally considered to be more accurate than random forest but can also be more sensitive to overfitting. It does well for imbalanced data which can occur for our use case.
MLP Classification	MLP Classification is a type of neural network model that consists of multiple layers of interconnected nodes, enabling it to learn complex patterns and relationships in data. MLPs are particularly effective in solving classification problems where the relationships between input features and output classes are nonlinear. They have the ability to capture intricate decision boundaries and can handle a wide range of input data types, making them a powerful tool for various applications such as image recognition, natural language processing, and pattern recognition.

By comparing the results of multiple models, researchers and practitioners can make informed decisions and ensure the deployment of the most effective predictive model. For this reason, we use all models listed in Table 1 and select the best performing for deployment.

Deep Learning Models

Although classification models perform well for binary target variables such as risk (High, Low) and categorical data such as the bucketed latitude, they cannot predict the exact coordinates at which a flight would land. For this, a deep learning model was used. Deep learning models recently gained prominence due to their ability to automatically learn hierarchical representations of data through multiple layers of interconnected neurons. This hierarchical representation enables the models to capture intricate and nonlinear relationships that may be challenging to capture using traditional ML techniques (LeCun et al., 2015). Deep learning models have achieved groundbreaking results across various domains, including computer vision (Krizhevsky et al., 2012), natural language processing (Mikolov et al., 2013), recommendation systems (He et al., 2017), and time series analysis (Lipton et al., 2015).

Deep learning models can also generalize well to unseen data. By scanning large amounts of training data and employing techniques such as regularization and dropout, deep learning models can effectively combat overfitting, leading to improved generalization performance (Goodfellow et al., 2016). This is especially valuable when the availability of extensive labeled data is limited, as deep learning models can learn meaningful representations through the inherent structure of the data.

Furthermore, deep learning models can use parallel computing architectures, such as graphics processing units (GPU) and specialized hardware like tensor processing units (TPU), to accelerate model training and prediction. The computational efficiency and scalability of deep learning models enable the processing of large datasets and the deployment of models in real-time applications (Abadi et al., 2016).

The application of deep learning models for similar purposes has been explored in previous studies. Liu et al. (2017) presented a deep learning approach for airline arrival time prediction, demonstrating superior prediction accuracy compared with traditional models.

The deep learning model used for this study incorporates various processors for tasks such as imputation, one-hot encoding, data cleaning, and principal component analysis (PCA). The inclusion of a residual block allowed information to flow from the first layer to the last, facilitating more efficient learning. PyTorch's nn library, coupled with rectified linear units, was used to compile the model sequentially. The training data was then processed through the pipeline to create the model, which was subsequently used to process all relevant data.

In addition to the feature selection process in step 3, the deep learning model applies PCA to reduce the dimensionality of the dataset. PCA is a technique that creates new uncorrelated variables, maximizing variance and improving interpretability while minimizing information loss. By incorporating these techniques, we aim to enhance the efficiency and accuracy of the deep learning model in determining the landing coordinates for flights.

Step 5: Select the Best Models:
Validation in predictive modeling can be challenging when dealing with distributions rather than point estimates as predicted results. Additionally, the presence of multiple potential outcomes and mission suggestions further complicates the validation process. To address this, cross-validation has been widely used as a standard method to obtain unbiased estimates of a model's goodness of fit (Buda et al., 2018). By comparing various learning strategies, including different combinations of algorithms, fitting techniques, and parameters, researchers can select the best model for the latest dataset based on transparent and quantifiable metrics.

The weighted F_1-score is then used to determine which model performed the best out of those listed in Table 1 for risk and mission. The F_1-score is a metric that combines both precision (measures the ability of a model to correctly identify only the truly positive instances among all instances predicted as positive) and recall (sensitivity: measures the ability of a model to correctly identify all positive instances from the total actual positive instances in the dataset), equally into a single metric. By combining the two metrics, it provides a balanced metric of both precision and recall, which performs well with our imbalanced data. Imbalanced data is data in which there is a significant difference in the number of instances between classes (values of variables we are trying to predict) such as many more High than Low risk flights. Utilizing the weighted F_1-score further helps with handling the imbalanced data. It is calculated by taking the average between recall and precision for each class and assigning a weight based on the frequency of the class. A higher weight is given to a less frequent value, therefore ensuring equal importance is given to all classes. Doing so avoids being overly biased to a majority class.

To determine the performance of the bucketed latitude and bucketed longitude models, weighted accuracy was used. The accuracy shows how often the predictions were correct, meaning if the actual latitude is 3.4° and the model predicted the plane to land in the 2–4° box, this was considered a correct prediction. Accuracy represents model performance better, especially due to this data being less imbalanced, and allows for it to be compared to the deep model performance.

The deep model predicted both the latitude and longitude and provided the actual coordinates. To select the best deep learning model, the distance loss was used, which is a form of the mean squared error (MSE). The formula used to calculate it is shown in equation (1).

$$\text{Distance loss} = \text{avg}(\text{square}(\text{latitude prediction} - \text{actual latitude})) + \text{avg}(\text{square}(\text{longitude prediction} - \text{actual longitude}) \quad (1)$$

This formula was used to calculate the distance between each of the longitude and latitude predictions from their actual value. Using distance loss allowed the model with the closest predictions to the actual to be selected. Therefore, the smaller the distance loss, the better the model. To then compare the deep model performance to the latitude and longitude bucketed models, a separate metric was created for each to determine the accuracy within 2°. By doing so, the deep model performance could be compared to the latitude and longitude bucketed models, respectively. The deep model predicted a separate latitude and longitude, rounded each prediction into a 2° bucket, and then determined if that bucket included the correct prediction.

Step 6: Deploying the Models:
The highly customizable ML training pipeline described above is aimed at creating a hunger games system for training, validating, and selecting the best-performing predictive models. The next step is deploying the selected models via a CI/CD pipeline. This pipeline was designed to ingest, process, and run predictions on millions of records in real time and send operators critical decision-making/supporting information from the predictive model.

The pipeline developed for deployment is an open source, automated, predictive modeling pipeline that trains, optimizes, validates, containerizes, and selects the best performing models for operational testing and deployment in support. The pipeline was developed and deployed in an open-source event-driven microservices architecture for quick and recurrent operational implementation of the containerized predictive models so that they are scalable, portable, highly available, and secure.

RESULTS

Due to our results being law enforcement sensitive, we are only able to share the results from training on sanitized data. The models performed similarly in comparison with each other when using actual data, but the values themselves are different.

Six different classification models (adaptive boosting [ada], random forest [rf], logistic regression [lr], Naive Bayes [bayes], SVM [svm], and MLP [nn]) were run for the risk, mission, and the bucketed landing location latitude and longitude values. Various parameters were attempted for each to find the optimal model performance. Feature selection was used as discussed previously to include Spearman and ANOVA. Different thresholds were then used to determine the level of correlation needed for a feature to be included. For example, the model risk_Spearman_0.6 used a Spearman correlation to find all features that had a correlation coefficient, *r*, of 0.6. The runs performed were each a combination of the following:

- Spearman correlation with a threshold of 0.9, 0.06, 0.3, and 0.1
- ANOVA correlation with a threshold of 0.95, 0.9, 0.6, 0.3, and 0.1
- All features

These combinations were then each run for the six classifiers, resulting in 60 runs total.
Note: ANOVA included 0.95 but Spearman did not as we found that high of a threshold for Spearman eliminated almost all features from being selected. It was therefore determined this would not be an applicable method to use.

Table 2 looks into the performance of the binary classification risk model and shows the top 10 runs. The top performing model was the MLP with a weighted F_1 score of 86.73%. This model was then selected and deployed to predict the outcome risk levels, High or Low, which informs DEOs whether they should pursue a flight and coordinate with local enforcements for their interdiction. As the models continue to operationalize, performance is expected to increase resulting in a better high-value target hit rate after their implementation.

Table 2. Risk Model Runs

Model	Classifier	Weighted F_1-Score	Weighted Precision	Weighted Accuracy
risk_Spearman_0.6	nn	**86.73%**	86.50%	87.36%
risk_ANOVA_0.3	lr	84.95%	85.67%	86.81%
risk_Spearman_0.6	ada	84.87%	84.64%	85.16%
risk_Spearman_0.9	nn	84.34%	84.18%	85.71%
risk_Spearman_0.1	lr	84.16%	84.26%	84.07%
risk_Spearman_0.3	bayes	83.89%	83.58%	85.16%
risk_Spearman_0.3	rf	83.81%	87.22%	86.81%
risk_Spearman_0.9	lr	83.72%	83.28%	84.62%
risk_Spearman_0.9	bayes	83.72%	83.28%	84.62%
risk_Spearman_0.6	bayes	83.72%	83.28%	84.62%

For nonbinary classification target variables, SVM was the top-performing model, as shown in Table 3, with a weighted F_1-score of 53.32%. This model was then selected and deployed to provide DEOs with additional information on the type of mission flights the monitored aircraft are on, therefore providing additional information on whether they should interdict the plane.

Table 3. Mission Model Runs

Model	Classifier	Weighted F_1-score	Weighted Precision	Weighted Recall
mission_Spearman_0.3	svm	**53.32%**	54.25%	47.46%
mission_Spearman_0.3	lr	50.54%	50.43%	46.89%
mission_Spearman_0.6	ada	48.95%	38.01%	27.47%
mission_Spearman_0.6	nn	48.61%	35.67%	40.11%
mission_Spearman_0.6	rf	48.47%	38.59%	33.52%
mission_Spearman_0.6	lr	48.19%	36.04%	26.92%
mission_Spearman_0.6	bayes	48.12%	1.21%	8.79%
mission_Spearman_0.1	svm	48.06%	31.79%	28.02%
mission_Spearman_0.1	ada	47.57%	38.58%	27.47%
mission_Spearman_0.1	nn	47.55%	8.16%	28.57%

Predicting the landing location proved to be more challenging. Table 4 shows the results for the bucketed latitude model where the weighted accuracy was used to find the performance of the model in predicting the flight within a 2° latitude range. Adaptive boosting was the top performing model, correctly predicting the latitude bucket 48.59% of the time.

Table 4. Classification Latitude Model

Model	Classifier	Weighted Accuracy	Weighted Precision	Weighted F_1-score
lat2_ANOVA_0.3	ada	**48.59%**	55.53%	46.67%
lat2_ ANOVA _0.3	lr	47.46%	48.06%	45.35%
lat2_ANOVA _0.3	bayes	46.89%	50.23%	44.78%
lat2_ ANOVA _0.1	rf	46.33%	52.88%	44.73%
lat2_ANOVA _0.1	bayes	46.33%	51.80%	43.53%
lat2_ANOVA _0.3	rf	46.33%	48.87%	44.27%
lat2_ANOVA_0.3	bayes	46.33%	54.78%	44.44%
lat2_ANOVA _0.6	rf	46.33%	48.73%	44.26%
lat2_ANOVA _0.6	bayes	45.76%	46.94%	43.71%
lat2_ANOVA _0.9	rf	45.76%	53.09%	44.27%

Table 5 shows the results for the bucketed longitude model where the accuracy was used to find the performance of the model in predicting the flight within a 2° longitude range. Random forest was the top performing model predicting the correct longitude bucket 48.02% of the time.

Table 5. Classification Longitude Model

Model	Classifier	Weighted Accuracy	Weighted Precision	Weighted F_1-score
lon2_Spearman_0.3	rf	**48.02%**	52.82%	46.56%
lon2_Spearman_0.6	lr	45.20%	50.84%	44.28%
lon2_Spearman_0.9	rf	44.07%	46.44%	43.16%
lon2_Spearman_0.1	rf	44.07%	45.17%	40.90%

lon2_Spearman_0.6	ada	43.50%	47.58%	43.25%
lon2 _Spearman_0.95	rf	43.50%	50.03%	42.73%
lon2 _Spearman_0.3	rf	43.50%	50.99%	42.47%
lon2 _Spearman_0.3	lr	42.94%	49.59%	42.87%
lon2_ANOVA_0.95	lr	42.94%	49.59%	42.87%
lon2_All_Fields	rf	42.94%	46.52%	40.13%

Table 6 displays the deep learning model results, which show the performance of the model in predicting the actual coordinates of the flight. The distance loss was used to determine the best performing model. Multiple parameters were tested to compare the model performance including the train_test_split function, which determines the percentage of data to train on, validate on, and test on. 80/15/5, 90/5/5, and 95/3/2 were all tested and are represented in the table as 80_15_5, 90_5_5, and 95_3_2. Additionally, the batch sizes of 128 and 256 were compared. The batch size represents the number of data samples processed for each step during each epoch-iteration of the datasets while the model is learning and updating. Based on the results, 80_15_5_b128, performed best, which represents 80% given to training, 15% given to validating, and 5% of the data set aside for testing, with a batch size of 128.

Table 6. Deep Learning Latitude and Longitude Models

Deep Model Run	**Distance Loss**
80_15_5_b128	46.20
95_3_2_b128	46.77
95_3_2_b256	48.86
90_5_5_b128	50.46
80_15_5_b256	62.96
90_5_5_b256	64.93

When comparing the performance of the deep learning model to predict the landing location of flights with the other bucketed methods, the accuracy of each was compared. Table 7 compares the best performing model from Table 4 to the best performing model in Table 6 by looking at their accuracy to predict flight landing locations within 2° latitude buckets. It then also compares the best performing model from Table 5 to the best performing model in Table 6 by looking at their accuracy to predict flight landing locations within 2° longitude buckets. The table below shows the deep model outperformed the latitude bucketed model 56% to 48.59%. For longitude, the two models performed nearly identically. We therefore deployed the deep learning model as it predicted as well as or better than the classification models were able to do while providing more specific coordinates on the flight's final destination.

Table 7. Landing Location Prediction Comparison

Model	**2° Latitude Accuracy**	**2° Longitude Accuracy**	**2° Latitude and Longitude Box**
80_15_5_b128_deep_model	56%	48%	33%
lat2_ANOVA_0.3	48.59%	Not applicable (NA)	NA
lon2_Spearman_0.3	NA	48.02	NA

IMPACT, BENEFITS, AND OTHER APPLICATIONS

Predictive analytics play a crucial role in supporting faster and better data-driven decisions across the DHS enterprise. By transcending agency boundaries, these analytics enable operators and leadership to proactively mitigate threats to the homeland across all domains. The predictive threat models discussed above are a great use case of this. With the success of the risk classification models, DEOs can focus their attention on flights that pose a threat to our nation with a high level of certainty. They can then pair that knowledge with the mission model to have an idea of the type of behavior a flight will perform, giving them additional insights into the level of attention the flight needs. Pairing the models results with their own expertise gives DEOs the information they need to classify a flight as a threat to our

nation. Once this is determined, the deep learning model's location prediction provides them with a better idea of where this flight is going to land, allowing them a head start to coordinate interdicting it with law enforcement.

Though there is room for improvement for each model, they perform at a high enough level to provide additional information that leads to improved operations by fostering more efficient, proactive, and data-driven decision-making processes. This improvement is significant regardless of an individual's experience or the availability of local technical or human resources. By implementing the vendor-agnostic open-source solutions described in this paper, the barriers to development are lowered, allowing for seamless cross-systems integration. Furthermore, these open-source tools enable clients to harness the latest, most advanced, and secure technologies available.

The research also aims to bring the full MLOps lifecycle to the DHS, with the potential for rapid expansion into other mission areas and domains within the DHS enterprise. This comprehensive approach enables the predictive modeling pipeline to be low maintenance, highly scalable, flexible, and always available. Continuous integration practices guarantee that all aspects of the pipeline incorporate the latest evidence-driven and battle-tested methodologies. Moreover, these practices help ensure that the pipeline remains secure and free from threats posed by bad actors and common software development issues.

DISCUSSION

The predictions are currently being implemented into the DEOs' decision-making process. It is too early to determine the changes in performance, but the ability to apply modern ML algorithms as discussed in this paper will become a crucial component in helping enforcement officers detect suspect aircraft behavior and mitigate risk in and around U.S. airspace, whether using the models discussed in this study or others. The architecture of more advanced applications of ML, such as the deep learning models, is very flexible, lending to highly customizable models. This by consequence adds to the challenge of achieving model optimality for niche problems. However, this is also a benefit of these models—the ability to continuously execute various combinations of hyperparameters, regularization techniques, activation functions, and more to reach optimality. Because of this flexibility, the models can be adjusted to solve other use cases.

Other ML models such as Long Short-Term Memory (LSTM) could be a good method to try in the future as they perform well for time series data such as that which is collected on each flight. After determining the impact of the models created in this study for DEOs in practice, a follow-on study could involve comparing the performance of DEOs in predicting the landing location of a flight using the methods described above with those predicted by a trained LSTM model. In fact, we have begun developing such an LSTM model, which we believe could eventually be used or even paired with the current deep learning model to achieve significantly improved results.

ACKNOWLEDGMENTS

The research in this document was conducted with the U.S. Department of Homeland Security (DHS) Science and Technology Directorate (S&T) under contract 47QFCA22F0047. Any opinions contained herein are those of the authors and do not necessarily reflect those of DHS S&T.

REFERENCES

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., ... & Zheng, X. (2016). Tensorflow: Large-scale machine learning on heterogeneous distributed systems. arXiv preprint arXiv:1603.04467.

Buda, M., Maki, A., & Mazurowski, M. A. (2018). A systematic study of the class imbalance problem in convolutional neural networks. Neural networks, 106, 249-259.

Chilyabanyama, O. N., Chilengi, R., Simuyandi, M., Chisenga, C. C., Chirwa, M., Hamusonde, K., ... & Bosomprah, S. (2022). Performance of Machine Learning Classifiers in Classifying Stunting Among Under-Five Children in Zambia. *Children, 9*, 1082.

Cowan, N. (2010). The Magical Mystery Four: How is Working Memory Capacity Limited, and Why? *Current Directions in Psychological Science, 19*, 51–57.

Domingos, P., & Pazzani, M. (1997). On the Optimality of the Simple Bayesian Classifier Under Zero-One Loss. *Machine Learning, 29*, 103–130.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. Cambridge, MA: MIT Press.

He, X., Liao, L., Zhang, H., Nie, L., Hu, X., & Chua, T. S. (2017). Neural Collaborative Filtering. In: *Proceedings of the 26th International Conference on World Wide Web* (pp. 173–182).

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An Introduction to Statistical Learning* (Vol. 112, p. 18). New York: Springer.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. In: *Proceedings of the 25th International Conference on Neural Information Processing Systems (NIPS)* (pp. 1097–1105).

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep Learning. *Nature*, *521*, 436–444.

Lipton, Z. C., Berkowitz, J., & Elkan, C. (2015). A Critical Review of Recurrent Neural Networks for Sequence Learning. *arXiv* preprint arXiv:1506.00019.

Liu, L., Chen, J., & Yang, Z. (2017). Deep Learning for Airline Arrival Time Prediction. *Journal of Air Transport Management, 62*, 104–115.

Nikam, S. S. (2015). A Comparative Study of Classification Techniques in Data Mining Algorithms. *Oriental Journal of Computer Science and Technology*, *8*, 13–19.

Rosenblum, M. R., Bjelopera, J. P., & Finklea, K. M. (2013). *Border Security: Understanding Threats at U.S. Borders.* Washington, DC: Congressional Research Service.

U.S. Sentencing Commission. (2021). Quick Facts: Fentanyl (FY21) (No. QA0114R1). U.S. Sentencing Commission. https://www.ussc.gov/sites/default/files/pdf/research-and-publications/quick-facts/Fentanyl_FY21.pdf

Wei, J., Chu, X., Sun, X. Y., Xu, K., Deng, H. X., Chen, J., ... & Lei, M. (2019). Machine Learning in Materials Science. *InfoMat 1*, 338–358.