Multi-label Classification of Aircraft Heading Changes using Neural Network to Resolve Conflicts

Md Siddiqur Rahman[1,2,4][a], Laurent Lapasset[2,4] and Josiane Mothe[3,4]

[1] *IRIT UMR5505 CNRS, Univ.de Toulouse 1 Capitole, Toulouse, France*

[2] *DEVI, Ecole Nationale de l'Aviation Civile, Toulouse, France*

[3] *INSPE, IRIT, UMR5505 CNRS, Toulouse, France*

[4] *Univ.de Toulouse, Toulouse, France*

Keywords: Aircraft Conflict Resolution, Machine Learning, Neural Network, Multi-label Classification.

Abstract: An aircraft conflict occurs when two or more aircraft cross at a certain distance at the same time. Aircraft heading changes are the common resolution at the en-route level (high altitude). One or more alternative heading changes are possible to resolve a single conflict. We consider this problem as a multi-label classification problem. We developed a multi-label classification model which provides multiple heading advisories for a given conflict. This model we named CRMLnet is based on the use of a multi-layer neural network that classifies all possible heading resolution in a multi-label classification manner. When compared to other machine learning models that use multiple single-label classifiers such as SVM, K-nearest, and LR, our CRMLnet achieves the best results with an accuracy of 98.72% and ROC of 0.999. The simulated data set which consists of conflict trajectories and heading resolutions we have developed and used in our experiments is delivered to the research community on demand. It is freely accessible online at: `https://independent.academia.edu/MDSIDDIQURRAHMAN9`.

1 INTRODUCTION

The position of two or more aircraft is considered a conflict situation if they fall in a distance less than the *5 nautical miles horizontally* when crossing each other (Kuchar and Yang, 2000; Prandini et al., 2000). Once a conflict is identified, Air Traffic Control Officers (ATCOs) must make a quick decision to solve it. ATCOs consider many parameters such as the position of the aircraft (latitude, longitude, altitude), speed, destination, flight plan, as well as other elements of the environment, for instance, weather, wind direction, military zone, etc.

Although James *et al.* (Kuchar and Yang, 2000) paper is about twenty years old now, however, it provides an overview of the approaches used for conflict detection and resolution. Early solutions to solve aircraft conflicts relied on mathematical models such as probabilistic and statistic models (Prandini et al., 1999). More recently, machine learning models (Srinivasamurthy et al., 2018) including deep learning (Nanduri and Sherry, 2016; Brittain and Wei, 2018) have been used. There are three types of conflict: short-range for which an on-board automatic systems exist that automatically change the altitude/heading of the aircraft, mid-range (5-20 mn) where the ATCO solves the conflict by changing the angle of one aircraft trajectory, and long-range (20-60 mn) where the usual solution is to change the initial flight path by selecting a different way point. In this research, we consider mid-range conflicts.

Our contribution is two-folds: First, we created a dataset with multi-label annotations where for each conflict sample, the different solutions are labeled. Second, we will release this unique dataset that can be used for conflict resolution evaluation.

With regard to conflict resolution, most related work uses the current position of the aircraft from which the future position projection and the distance between the aircraft is calculated using speed, angle between them, time, and many more parameters, possibly from different sources including on-board data (Prandini et al., 1999; Prandini et al., 2000; Pham et al., 2019a; Kim et al., 2016; Pham et al., 2019b). The aircraft positions are approximate positions and this can lead to some wrong calculations. We thus

[a] https://orcid.org/0000-0002-9314-5377

Rahman, M., Lapasset, L. and Mothe, J.
Multi-label Classification of Aircraft Heading Changes using Neural Network to Resolve Conflicts.
DOI: 10.5220/0010829500003116
In *Proceedings of the 14th International Conference on Agents and Artificial Intelligence (ICAART 2022) - Volume 3*, pages 403-411
ISBN: 978-989-758-547-0; ISSN: 2184-433X

rather consider the 5-minutes trajectory of each aircraft just before the conflict would occur. Our model then learns from the aircraft movement and do not use other calculated features.

The conflict resolution is cast as a multi-label classification problem where each class corresponds to a possible angle change in trajectory. We consider a multi-layer perceptron (MLP) neural network. Such models have successfully been used mostly in complex classifications or approximation tasks (Gardner and Dorling, 1998). While in related work, the models are designed to provide a single solution to solve the conflict, our model provides multiple alternative solutions where ATCO can choose the best one considering the future possible conflicts. One solution only may be appropriate in resolving the current conflict but may create new conflicts in the future, while another solution may be both appropriate for the current conflict without producing a new conflict later.

For evaluation purposes, we have also developed other multi-label supervised models based on support vector machine, logistic regression, and K-nearest neighbors where we used separate classifiers for each output class-label solution and compare them with the neural network-based model.

This paper is organized as follows. Section 2 discusses related work. Section 3 presents the data that are used to train the model and the data sources. Section 4 discusses in detail the architecture of our neural network-based model. All the performance evaluations are detailed in Section 5. Section 6 presents the results and discusses them. Finally, Section 7 concludes this paper and provides future directions.

2 RELATED WORK

Mathematical models were widely used as the earliest solutions to conflict detection and resolution (Prandini et al., 1999; Prandini et al., 2000; Alonso-Ayuso et al., 2013; Agogino and Tumer, 2012; Bayen et al., 2005). (Paielli and Erzberger, 1997) and (Erzberger et al., 1997) applied a probability distribution method for conflict detection by projecting the nearest future of the involved aircraft. Prandini *et al.* proposed two different models for mid-range and short-range conflict detection considering speed control (Prandini et al., 1999; Prandini et al., 2000). In real conditions, an ATCO usually avoids changing the aircraft speed because this is a cause of fuel over-consumption. (Pham et al., 2019a) showed noise-free information is required for mathematical models while that is difficult to get in real-time because surveillance radar information is an approximate location of the aircraft. Thus, it is difficult to clean data while the number of aircraft increases.

Some research has proposed resolutions for free flights assuming a pilot can change his/her flight route in the mid-flight if s/he wants to. Eby and Kelly applied a distributed algorithm for free flight in (Eby and Kelly, 1999) where they assume that each aircraft can communicate with the others and change their flight plans. Alam *et al.* use a popular ensemble approach where each aircraft has an on-board system to share information with the surrounding aircraft (Alam et al., 2009). Also, (Jiang et al., 2018) used Support Vector Machine (SVM) for free flights mainly considering the current position, velocity, and predicted look-ahead time as the parameters. Although the solutions on the free flight is appealing, currently, the aircraft cannot communicate with each other.

Researchers are turning to the application of machine learning to solve conflicts with effective results (Alam et al., 2009; Kim et al., 2016; Brittain and Wei, 2018; Jiang et al., 2018; Pham et al., 2019a; Pham et al., 2019b; Wang et al., 2019; Lapasset et al., 2020; Rahman, 2020).

Recently, Rahman *et al.* (Rahman, 2020; Lapasset et al., 2020) discussed various data sources and proposed to creating a deep learning model to resolve aircraft collisions. But the concept is limited in theory, with no experimental results. (Kim et al., 2016) present a performance analysis of two separated models to solve a conflict between two airplanes: a neural network-based and a SVM-based, both supervised. The SVM model combines 9 SVM, one per class label, each one predicts different category of resolutions. Similarly, the neural network model is composed of 9 output nodes. The model output is a vector of 9 class labels that are all zero except for the most probable one which corresponds to the best predicted action. Their dataset contains category-based resolutions such as vertical, horizontal, and speed control. For example, there are two resolutions for horizontal conflicts: *Direct-to* and *Path stretch*, where *Direct-to* means the resolution maneuver is to skip some initial way-points and go direct to the targeted way-point whereas *Path stretch* is to add new way-points to make the resolution more flexible. In this case, the model only predicts these categories. The limitation here is that there is no exact heading direction to resolve the conflict. Still, ATCO needs to think about the resolutions before taking a decision.

Reinforcement learning has been used to resolve aircraft conflicts. (Brittain and Wei, 2018) applied a two-level agent-based deep reinforcement learning following a hierarchical network. In the first agent, a convolutional neural network is applied to an air traffic control video game (NASA Sector 33) image which selects the most suitable route by applying all possible initial route combinations. The second agent adjusts the speed for the route. In their case, there is no option to modify the initial route. Also, as mentioned earlier, resolving conflicts by changing the speed is avoided in practice. (Pham et al., 2019b; Pham et al., 2019a) applied a single deep reinforcement network in the specific case of two aircraft at the same altitude. Since the resolution action is not finite, an agent-based reinforcement learning resolves the conflict by applying an infinite number of actions. For each action, it gets rewards (rank) with either positive (successful) or negative (unsuccessful) feedback. From this feedback, the model fits itself. Here, the challenge is to design a reward function based on the quality of the solution. Quality comes from selecting a set of features that is a kind of rule or condition.

Although recently neural network-based reinforcement learning model is widely used, as discussed earlier, the challenge is to define a reliable reward function because this function is used to find solutions without the use of labeled input-output pairs. In our research, the most remarkable thing is that our model takes 5-minute continuous positions for each aircraft, while prior related work considers the current position only and thus needs to perform feature engineering under many conditions. Trajectory data is not noise-free (all the aircraft positions are approximated), thus, it is risky to use strict mathematical conditions such as calculating distances, angles, to create new features. We are of the opinion that a model that takes a series of positions for each aircraft to learn the conflict environment and provides multiple solutions would reduce that risk. The existing models are design to find a single solution to a conflict. We rather propose multiple solutions for a single conflict the ATCO can choose among considering the future possible conflicts. We consider multi-label (multiple output for one input) supervised models.

3 DATA

Three main sources could be use to get trajectory and ATCO's immediate order: (a) open-source data, (b) radar data from ATC station, and (c) simulated data. In this paper, we used the third one.

Original trajectory data is generally kept confidential and therefore not publicly available. No simulated data is even available publicly. The problem to use open source data such as from OpenSky Network (Schäfer et al., 2014) is to synchronize huge trajectory storage and ATC orders. Another issue is that there is no information on the heading change if it is a conflict with the aircraft or not. Since the ATCO voice command is sensitive, time-consuming, and difficult to obtain, in our paper we rather consider simulate data.

The primary components of the aircraft trajectory are latitude, longitude, and altitude. It is sometimes called the 4D trajectory where time is the 4^{th} dimension (Wandelt and Sun, 2014). An immediate order is a voice communication between a controller and a pilot to guide him/her to avoid a conflict situation. According to Pavlinović *et al.* in (Pavlinović et al., 2013), different controllers operate at different phases based on their altitude level such as pre-flight, take-off, departure, en-route, decent, approach, and landing. In our study, we consider the en-route phase (top height level) only where the altitude (height) of the aircraft usually remains unchanged. The common resolution maneuver is heading direction either turn left or right with a certain angle.

We generated the trajectory and controller's immediate order datasets using an open-source simulator named Blue Sky developed at TU Delft by Hoekstra and Ellerbroek (Hoekstra and Ellerbroek, 2016). There are many advantages to use simulators. First, it is easy to create conflict scenarios. Second, many variations can be created, which may not be possible to find in real data. We generated different conflict scenarios where a single instance contains every 10 seconds following a 5-minute window of trajectory for a pair of aircraft and the resolutions. Thus, we consider two planes in such a way they can create a conflict situation. Both aircraft's position is 20 minutes away from the conflict point. We store 5 minutes of trajectory data of them that is just before the conflict detection. Therefore, after detecting the conflict we have 15 minutes to reach the conflict point. Our model makes the resolution decision based on that 5-minute trajectory.

The parameters we stored are latitude, longitude, altitude, speed of both planes, and angle between them. Figure 1 shows a scenario with possible resolutions (range: from left 30^0 to right 30^0). At en-route level, ATCOs usually change the heading degree by a multiple of five (e.g. an immediate order could be TURN LEFT 5^0 or TURN LEFT 10^0 as shown in Figure 1). We do not consider the heading resolution in both sides (LEFT and RIGHT) at a time. We always take the heading to the side where the angle between the planes is the smallest. If there is no solution, then we look for the other side. Figure 1 shows solutions on the side of lower angle. Here, the column vector with multiple binary decisions shows an example of the left heading resolution only. We applied different

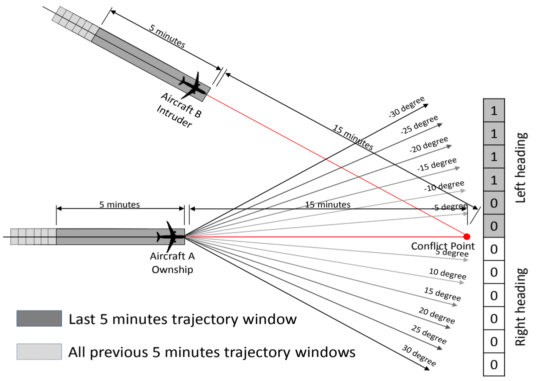


Figure 1: There are different heading changes to solve a conflict. Here, the shadow behind each plane shows the trajectory of the previous 5 minutes. Aircraft A can change its heading between left 30^0 and right 30^0 to solve the conflict while the heading of aircraft B remains unchanged. The column vector on the right shows the binary decision for this sample. Here "0" means the decision is not able to resolve the conflict whereas "1" means it can.

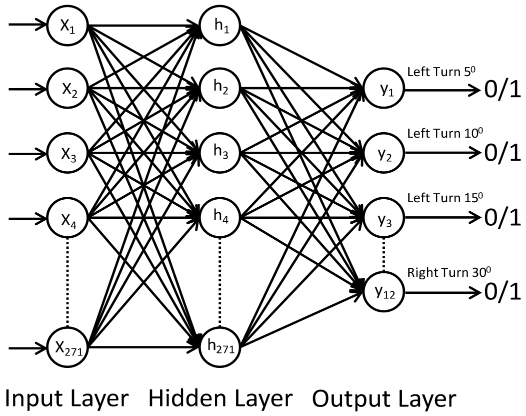


Figure 2: CRMLnet: Conflict resolution multi-label classification neural network model. There are 271 nodes in both the input layer and the hidden layer while the output layer has 12 nodes. Each output node individually provides binary output of 12 heading angles.

techniques to augment the data. For example, rotating a whole scenario does not change the decision; we also change speed considering different values to create more samples. Each scenario is split in two parts in such a way that the time slot for one is at 0 second, 10 seconds, 20 seconds, up to 5 minutes. In the same way, the other one is for 5 seconds, 15 seconds, 25 seconds, up to 5 minutes which results in a new scenario.

It is not possible to record voice commands in the simulator, we thus use text commands to simulate the ATCOs immediate orders. We have generated 1,516 sample scenarios and the corresponding valid commands to resolve them. The samples can be categorized based on the number of solutions they have: [288, 2], [288, 3], [300, 4], [372, 5], and [6, 268] where the first value of each pair is the number of samples and the second one is the number of solutions. The distribution of the samples in each category is almost balance.The complete data set is freely accessible online at: `https://independent.academia.edu/MDSIDDIQURRAHMAN9`.

4 CLASSIFICATION MODEL

The problem of aircraft conflicts can be considered as (a) a binary classification problem where the classifier decides whether the conflict is solvable or not, (b) a multi-class classification problem where the classifier selects only the best one from multiple solutions, and (c) a multi-label classification where the selection of solutions will be one or more.

In this research, we consider the conflict resolution problem as a multi-label classification because there can be more than one solution to a conflict scenario. For instance, in Figure 1, the possible solutions are 15 degrees, 20 degrees, 25 degrees, and 30 degrees in *Turn Left aircraft A*. The multi-label result is also more applicable in real life because a controller will have multiple alternative solutions in hand where it will be much easier to avoid risk. S/he can take one of the solutions thinking of the other aircraft's, which are not involved in the conflict, position to avoid additional future conflicts.

According to Tsoumakas and Katakis, it is possible to make one or more single-label classification problems from a multi-label classification problem by making some problem transformations (Tsoumakas and Katakis, 2007). An individual single-label classification can be used for every single-label. We have applied a multi-label classification using a single architecture based on a neural network as well as multiple single-label algorithms to compare with (see Section 6). There is no well-defined neural network architecture to tackle this task. Each neural network-based model can be distinguished based on a combination of hyper-parameters; it behaves differently for different data. We have tuned the hyper-parameters to properly solve the task. Section 6 discusses more about it. The following points are considered which have not previously been covered in the literature:

(a) The input layer of the model takes 5-minute of trajectory parameters of all the involved aircraft.

(b) Outputs use a separate sigmoid activation function for binary classification.

(c) This is the first model in the field of aircraft conflict resolution that provides multiple output for a single conflict.

(He and Xia, 2018) showed that a single network can perform better for multi-label classification than multiple individual networks for classifying emotions from texts. In a single network, all neurons are interconnected to each other, thus, all output decisions are based on sharing information. On the other hand, (Baker and Korhonen, 2017) mentioned two disadvantages of using separate binary classifiers for multi-label classification: first, it is assumed that class-labels are independent, although this is not happening in all cases; second, it is relatively expensive to compute because the classifiers are computing separately while using the same input. We use a multi-label classification based on neural network that we call CRMLnet. Figure 2 depicts our CRMLnet model.

Since we store 5-minutes (5×60 seconds = 300 seconds) of trajectory following a 10-second change for each aircraft, we have the same parameters at each 10-seconds but the values change with respect to time. This means we store the features repeatedly for 30 (300 seconds $\div$ 10 seconds = 30) times with different values. The angle (α) between two planes remains unchanged. Thus, we have 9 input features that are repeated 30 times every 10-seconds: *time*, *latitude (aircraft A)*, *longitude (aircraft A)*, *altitude (aircraft A)*, *heading (aircraft A)*, *latitude (aircraft B)*, *longitude (aircraft B)*, *altitude (aircraft B)*, *heading (aircraft B)*. Overall, we have 271 (1 (angle) + 9×30 (repeated parameters) = 271) total input features. For that reason, the input layer of our neural network model is composed of 271 nodes.

Additional hidden layers are needed, specifically when the problem dataset is not linearly separable. For example, Yanling *et al.* in (Yanling et al., 2002) showed that it is not possible to solve a logical XOR problem using a regular single-layer neural network. However, in CRMLnet, we limited ourselves to one hidden layer to avoid increases the loss and decreases the accuracy. Indeed, more hidden layers are more likely to increase overfitting than to increase learning ability because of the large number of neurons(Panchal et al., 2011). The number of nodes in the hidden layer is equal to the number of input layers. The output layer contains 12 nodes for 12 heading in Figure 1. We used Rectified Linear Unit (ReLU) activation function at the hidden layer to avoid the negative values and make the model training fast. Also, we used a *sigmoid* activation function for each output neuron for individual binary classification.

5 PERFORMANCE EVALUATION

We evaluated the CRMLnet model and compared it with other multi-label architectures using: Support Vector Machine (SVM), K-Nearest Neighbor Classifier (K-nearest), and Logistic Regression (LR). The evaluation is based on the simulated dataset presented in Section 3. With regard to the sampling method, we used both k-fold cross-validation ($k = 10$) (Kohavi et al., 1995) and independent test sets where the total dataset was divided into three subsets (60% for training, 20% for validation, and 20% for testing). To normalize the data before applying the machine learning model, we also perform a standard scaling method. We use usual performance metrics: accuracy (Acc), area under receiver operating characteristic curve (auROC), area under precision-recall curve (auPR), F_1 score, Sensitivity (S_n), Specificity (S_p), and Mathew's Correlation Coefficient (MCC). We focus more on F_1 (Eq. 1) score and MCC (Eq. 2). F_1-Score is the harmonic mean of precision (p) and recall (r), the latter is also known as sensitivity (S_n). MCC scores range from -1 to 1, where 1 means all the samples are correctly classified and -1 means no sample is classified correctly.

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{1}$$

$$\text{where } precision = \frac{TP}{TP+FP} \text{ and } recall = \frac{TP}{TP+FN}$$

$$\text{MCC} = \frac{(TP \times TN) - (FP \times FN)}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \tag{2}$$

Here, TP is the total number of correctly classified positive examples, TN is the total number of correctly classified negative examples, FP is for incorrectly classified positive examples, and FN for incorrectly classified negative examples.

6 RESULTS AND DISCUSSION

The performance of a machine learning model highly depends on the selection of its different hyper-parameters. On the other hand, selecting hyper-parameters is also quite difficult because of calculating the permutations of the parameters. Random Search (Bergstra and Bengio, 2012) is one of the popular and widely used algorithms to find the most influential parameters. We applied it on our CRML-net model to optimize the learning rate, the number of hidden layer, the number of nodes in each hidden layer, optimizer, etc. As we discussed earlier, in addition to the CRMLnet, we used three more binary classifier-based multi-label classification models: SVM, K-nearest, and LR, for which we have selected hyper-parameters.

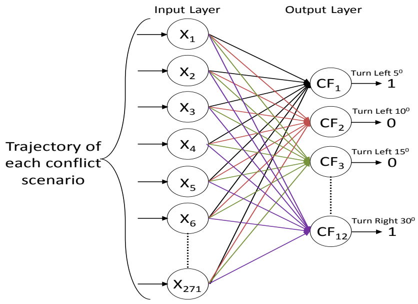


Figure 3: Multi-label classification using individual classifier for each class label. All nodes on the left are input nodes. CF_1, CF_2, ..., CF_{12} are the output nodes and these can be replaced by any binary classifier.

Figure 4 (a) and (b) plot the train and test loss and accuracy for 10-fold cross-validation while Figure 4 (c) and (d) show the train and validation loss and accuracy for independent test set of CRMLnet model. We see the CRMLnet model is the best up to 100 epochs. Here, 1 epoch means the complete forward and backward pass of input features during training. Figure 4 shows that the loss and accuracy are almost the same during training and testing. This means that up to 100 epochs our model does not overfit; no variance problem occurs either.

The average validation loss (test loss in this case) in all cases is around 0.05, which is low. The lower loss a model has, the better its performance. Accuracy of our CRMLnet model is around 98.72% for 10-fold cross-validation (designated as $CRMLnet_{cv}$, see Table 1); it is around 97.79% for the independent test set (designated as $CRMLnet_{ind}$, see Table 1). This means the performance in both cases is generalized while 10-fold cross validation is better than the independent test set. In addition to accuracy, we measured *auROC*, *auPR*, S_p, S_n, *MCC*, and F_1 score of both cross-validation and the independent test set. Table 1 shows the measurements of both 10-fold cross-validation ($CRMLnet_{cv}$) and independent test set ($CRMLnet_{ind}$).

Figure 3 shows a general architectural view of a multi-label classification model using a single-label classifier where all the $CF(CF_1, CF_2, ..., CF_{12})$ can be replaced by any of one single-label classifier (SVM, LR, or K-nearest). We designed three different architectures for SVM, LR, and K-nearest (designated as MSVM, MLR, and MK-nearest) and applied them on the same dataset using both 10-fold cross-validation and independent test set. Finally, all the results of the different models are represented in Table 1 for cross-validation sequentially as follows: $CRMLnet_{cv}$, $MSVM_{cv}$, MLR_{cv}, and MK-$nearest_{cv}$. While in the case of independent test set, it is as follows: $CRMLnet_{ind}$, $MSVM_{ind}$, MLR_{ind}, and MK-$nearest_{ind}$. The results in Table 1 show that our CRMLnet model for both cross-validation and independent test set is much better than the other models based on a single-label classifier. Although numerical results are often important, many complex things are easier to understand if they are visually presented. In Figure 5, we represent the ROC curve of individual class-label (twelve heading directions from Figure 1) for all the methods with 10-fold cross-validation: (a) Neural Network-based model CRMLnet, (b) Multiple Support Vector Machine based model MSVM, (c) Multiple K-Nearest Neighbor classifier based model MK-nearest, and (d) Multiple Logistic Regression based model MLR. We have twelve distinct class-labels (horizontal heading direction - see Figure 1) and for each class-label, we applied a single-label binary classifier (single output node for CRMLnet) to predict whether the corresponding heading change solves the conflict or not. Any of the individual classifiers use the same input features. Figure 5 shows that there are high fluctuations in ROC for the other models ($MSVM_{cv}$, MLR_{cv}, and MK-$nearest_{cv}$) while it is not the case for the $CRMLnet_{cv}$ model. We also have estimated the error of $CRMLnet_{cv}$: 0.044 and $CRMLnet_{ind}$: 0.063. We see cross-validation ($CRMLnet_{cv}$) has less error than independent test sets ($CRMLnet_{ind}$). The training of the model using cross-validation is better than independent test sets. So, in all cases, CRMLnet performs much better than other models using a separate single label classifier.

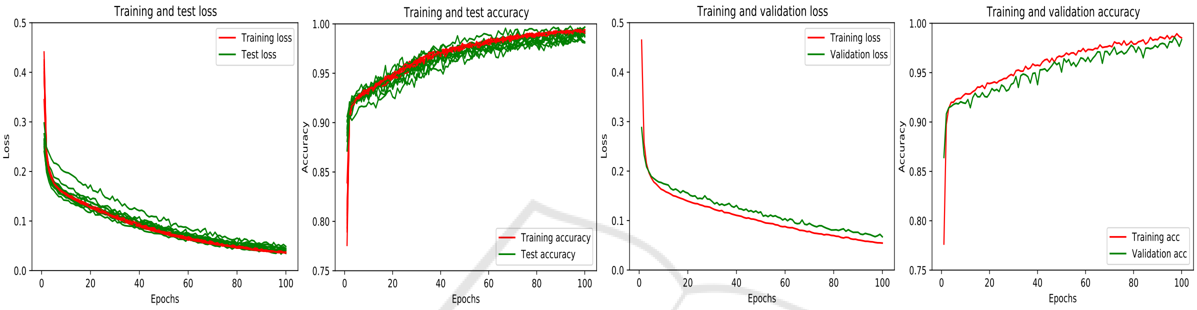
We cannot compare our model with models from the literature because the data preparation is different. The annotations for each data sample (multi-label class) are also very different from other (binary or multi-class) ones because we tackle the problem of conflict resolution in a different way. We discussed in Section 2 that the most similar work to ours is Kim *et al.*' (Kim et al., 2016) where their dataset contains category-based resolutions such as vertical, horizontal, and speed control. On contrary, our class label is heading angle modification such as right-heading or left-heading with one or more specific degree angles.

7 CONCLUSION

The purpose of this research is to develop a model that suggests different heading directions to air traffic controllers to avoid aircraft conflicts. The neural network we developed, CRMLnet, is a multi-label

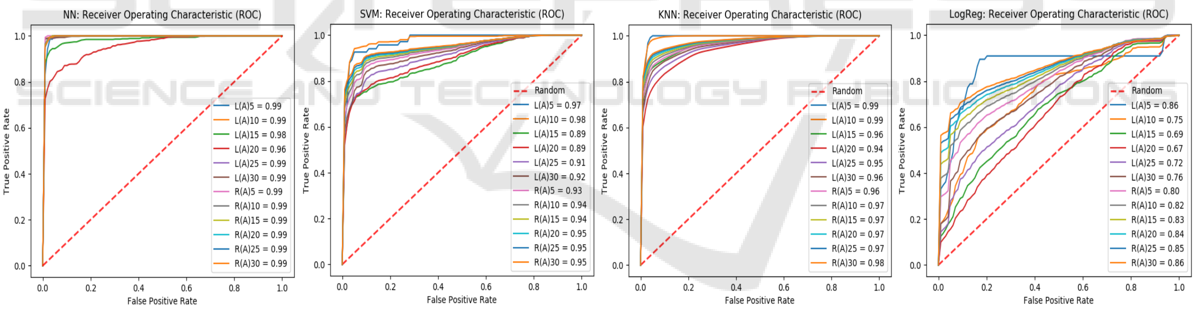
Table 1: CRMLnet is much better than the other classifiers when using cross-validation (CRMLnet$_{cv}$) and independent test set (CRMLnet$_{ind}$). Here, the 1st column is the classifier. The next columns are : Accuracy (Acc), area under receiver operating characteristic curve (auROC), area under precision-recall curve (auPR), Specificity(S$_p$), Sensitivity (S$_n$), Mathew's Correlation Coefficient (MCC), and F_1 score.

Classifiers	Acc	auROC	auPR	S_p	S_n	MCC	F_1
CRMLnet$_{cv}$	98.72%	0.999	0.998	99.11%	97.94%	0.971	0.981
MSVM$_{cv}$	91.66%	0.953	0.934	94.24%	86.54%	0.812	0.793
MK-nearest$_{cv}$	95.45%	0.979	0.958	96.68%	93.01%	0.898	0.921
MLR$_{cv}$	90.96%	0.863	0.818	93.29%	86.36%	0.797	0.785
CRMLnet$_{ind}$	97.79%	0.997	0.995	97.93%	97.36%	0.952	0.968
MSVM$_{ind}$	91.47%	0.944	0.899	94.30%	85.89%	0.808	0.768
MK-nearest$_{ind}$	93.00%	0.931	0.895	95.14%	88.78%	0.843	0.884
MLR$_{ind}$	90.97%	0.842	0.789	93.63%	85.73%	0.797	0.785



(a) CRMLnet$_{cv}$ loss (b) CRMLnet$_{cv}$ accuracy (c) CRMLnet$_{ind}$ loss (d) CRMLnet$_{ind}$ accuracy

Figure 4: Up to 100 epochs, CRMLnet does not over-fit when considering both cross-validation and independent test set. The horizontal axis represents the number epoch. The vertical axis in (a) & (c) represents the loss while in (b) & (d) for accuracy.



(a) CRMLnet (b) MSVM (c) MK-nearest (d) MLR

Figure 5: CRMLnet model is much better in terms of ROC compared to other models. Each color represents an individual heading change from Figure 1.

classification model which identifies multiple resolutions for a single conflict scenario. In addition to the classification model, we also developed a simulated dataset in a 5-minute window manner. This data set is made available to the research community. We identify as many heading directions as possible to solve a single conflict within a specific horizontal direction range (left 30^0 to right 30^0). We evaluated our model using 10-fold cross-validation (CRMLnet$_{cv}$) and independent test set validation (CRMLnet$_{ind}$). We also compare our CRMLnet model with other multi-label classification models (MSVM, MLR, and MK-nearest) and show that CRMLnet got much better performances. Our CRMLnet model obtained 98.72% of accuracy when using 10-fold cross-validation and 97.79% when using independent test set. The other models obtained the following accuracy for 10-fold cross-validation: 91.66% for MSVM$_{cv}$, 95.45% for MK-nearest$_{cv}$ and 90.96% for MLR$_{cv}$ while for independent test set: 91.47% for MSVM$_{ind}$, 93.00% for MK-nearest$_{ind}$, and 90.97% for MLR$_{ind}$. Through this research, we show that models can learn the conflict environment. Also, we show that it is possible to make conflict resolution without any prepossess (feature extraction) of this data.

REFERENCES

Agogino, A. K. and Tumer, K. (2012). A multiagent approach to managing air traffic flow. *Autonomous Agents and Multi-Agent Systems*, 24(1):1–25.

Alam, S., Shafi, K., Abbass, H. A., and Barlow, M. (2009). An ensemble approach for conflict detection in free flight by data mining. *Transportation research part C: emerging technologies*, 17(3):298–317.

Alonso-Ayuso, A., Escudero, L. F., Olaso, P., and Pizarro, C. (2013). Conflict avoidance: 0-1 linear models for conflict detection & resolution. *Top*, 21(3):485–504.

Baker, S. and Korhonen, A.-L. (2017). Initializing neural networks for hierarchical multi-label text classification. Association for Computational Linguistics.

Bayen, A., Grieder, P., Meyer, G., and Tomlin, C. J. (2005). Langrangian delay predictive model for sector-based air traffic flow. *Journal of guidance, control, and dynamics*, 28(5):1015–1026.

Bergstra, J. and Bengio, Y. (2012). Random search for hyper-parameter optimization. *The Journal of Machine Learning Research*, 13(1):281–305.

Brittain, M. and Wei, P. (2018). Autonomous aircraft sequencing and separation with hierarchical deep reinforcement learning. In *Proceedings of the International Conference for Research in Air Transportation*.

Eby, M. S. and Kelly, W. E. (1999). Free flight separation assurance using distributed algorithms. In *1999 IEEE Aerospace Conference. Proceedings (Cat. No. 99TH8403)*, volume 2, pages 429–441. IEEE.

Erzberger, H., Paielli, R. A., Isaacson, D. R., and Eshow, M. M. (1997). Conflict detection and resolution in the presence of prediction error. In *1st USA/Europe Air Traffic Management R&D Seminar, Saclay, France*, pages 17–20. Citeseer.

Gardner, M. W. and Dorling, S. (1998). Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences. *Atmospheric environment*, 32(14-15):2627–2636.

He, H. and Xia, R. (2018). Joint binary neural network for multi-label learning with applications to emotion classification. In *CCF International Conference on Natural Language Processing and Chinese Computing*, pages 250–259. Springer.

Hoekstra, J. M. and Ellerbroek, J. (2016). Bluesky atc simulator project: an open data and open source approach. In *Proceedings of the 7th International Conference on Research in Air Transportation*, pages 1–8. FAA/Eurocontrol USA/Europe.

Jiang, X.-r., Wen, X.-x., Wu, M.-g., Wang, Z.-k., and Qiu, X. (2018). A svm approach of aircraft conflict detection in free flight. *J. of Advanced Transportation*.

Kim, K., Hwang, I., and Yang, B.-J. (2016). Classification of conflict resolution methods using data-mining techniques. In *16th AIAA Aviation Technology, Integration, and Operations Conference*, page 4075.

Kohavi, R. et al. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai*, volume 14, pages 1137–1145. Montreal, Canada.

Kuchar, J. K. and Yang, L. C. (2000). A review of conflict detection and resolution modeling methods. *IEEE Transactions on intelligent transportation systems*, 1(4):179–189.

Lapasset, L., Rahman, M. S., and Mothe, J. (2020). Solving aircraft conflicts: data resources. In *1st International Conference on Cognitive Aircraft Systems (ICCAS 2020)*, page 76.

Nanduri, A. and Sherry, L. (2016). Anomaly detection in aircraft data using recurrent neural networks (rnn). In *2016 Integrated Communications Navigation and Surveillance (ICNS)*, pages 5C2–1. IEEE.

Paielli, R. A. and Erzberger, H. (1997). Conflict probability estimation for free flight. *Journal of Guidance, Control, and Dynamics*, 20(3):588–596.

Panchal, G., Ganatra, A., Kosta, Y., and Panchal, D. (2011). Behaviour analysis of multilayer perceptrons with multiple hidden neurons and hidden layers. *International Journal of Computer Theory and Engineering*, 3(2):332–337.

Pavlinović, M., Boras, D., and Francetić, I. (2013). First steps in designing air traffic control communication language technology system-compiling spoken corpus of radiotelephony communication. *International Journal of Computers and Communications*, 7(3):73.

Pham, D.-T., Tran, N. P., Alam, S., Duong, V., and Delahaye, D. (2019a). A machine learning approach for conflict resolution in dense traffic scenarios with uncertainties.

Pham, D.-T., Trant, N. P., Goh, S. K., Alam, S., and Duong, V. (2019b). Reinforcement learning for two-aircraft conflict resolution in the presence of uncertainty. In *2019 IEEE-RIVF International Conference on Computing and Communication Technologies (RIVF)*, pages 1–6. IEEE.

Prandini, M., Hu, J., Lygeros, J., and Sastry, S. (2000). A probabilistic approach to aircraft conflict detection. *IEEE Transactions on intelligent transportation systems*, 1(4):199–220.

Prandini, M., Lygeros, J., Nilim, A., and Sastry, S. (1999). A probabilistic framework for aircraft conflict detection. In *Guidance, Navigation, and Control Conference and Exhibit*, page 4144.

Rahman, M. S. (2020). Supervised machine learning model to help controllers solving aircraft conflicts. In *ADBIS, TPDL and EDA 2020 Common Workshops and Doctoral Consortium*, pages 355–361. Springer.

Schäfer, M., Strohmeier, M., Lenders, V., Martinovic, I., and Wilhelm, M. (2014). Bringing up opensky: A large-scale ads-b sensor network for research. In *Proceedings of the 13th international symposium on Information processing in sensor networks*, pages 83–94. IEEE Press.

Srinivasamurthy, A., Motlicek, P., Singh, M., Oualil, Y., Kleinert, M., Ehr, H., and Helmke, H. (2018). Iterative learning of speech recognition models for air traffic control. In *Interspeech*, pages 3519–3523.

Tsoumakas, G. and Katakis, I. (2007). Multi-label classification: An overview. *International Journal of Data Warehousing and Mining (IJDWM)*, 3(3):1–13.

Wandelt, S. and Sun, X. (2014). Efficient compression of 4d-trajectory data in air traffic management. *IEEE Transactions on Intelligent Transportation Systems*, 16(2):844–853.

Wang, Z., Li, H., Wang, J., and Shen, F. (2019). Deep reinforcement learning based conflict detection and resolution in air traffic control. *IET Intelligent Transport Systems*, 13(6):1041–1047.

Yanling, Z., Bimin, D., and Zhanrong, W. (2002). Analysis and study of perceptron to solve xor problem. In *The 2nd International Workshop on Autonomous Decentralized System, 2002.*, pages 168–173. IEEE.