Combatting Spatial Disorientation with Crash Prediction in a Spaceflight Analog Task by Deep Learning

Abstract

Spatial disorientation is a leading cause of fatal aircraft crashes and is also a significant concern for spaceflight. However, to our knowledge, no study has used deep learning to predict crashes in such conditions. To collect data, we have participants stabilize themselves while inside an unstable machine in an orientation that leads to spatial disorientation similar to what pilots and astronauts experience. Our goal is to predict the occurrence of crashes in advance and compare the performance of models based on recurrent neural networks (RNN) and other non-RNN models. We evaluate the models using metrics sensitive to the context of aviation and spaceflight and select a deep learning model based on stacked gated recurrent units (GRU) that is able to predict crash events 800ms in advance with an AUC value of 0.9925 and a 0.5289 precision at 0.95 recall. Further analysis suggests that our model can provide feedback in time to prevent many crashes in our paradigm, proving a successful and novel application of deep learning in the fields of aviation and spaceflight. Our next step will be to create a real-time alerting system incorporating our proposed approach.

1 Introduction

Spatial disorientation occurs when pilots lose sense of their orientation relative to the gravitational vertical. Studies have estimated that 90-100% of pilots experience it (Newman 2007; Gibb, Ercoline and Scharff 2011). Spatial disorientation is responsible for up to 32% of major accidents and up to 26% of those leading to death (Newman 2007). A majority of fatal aircraft accidents caused by spatial disorientation occur when pilots are unaware that they are disoriented (Braithwaite et al 1998). Astronauts will similarly be susceptible to spatial disorientation during gravitational transitions, such as when landing on the surface of a planet or the moon (Shelhamer 2018, Clement et al. 2020). Significant interest from companies such as SpaceX, Virgin Galactic, and Blue Origin has caused the space industry to grow rapidly and the problem of spatial disorientation in astronauts will increase as more people explore space. Despite improvements in technology, rates of spatial disorientation are not significantly reducing (Daiker et al. 2018). There are very few studies that have attempted to develop an alerting system that can predict crashes when a pilot is disoriented. Daiker et al. (2018) proposed a proof-of-concept idea for NASA's Cost Effective Devices for Alerting Research (CEDAR), where they use a model of the vestibular system, aircraft dynamics, and sensors to create a predictive alerting model. To our knowledge, no study has used deep learning to predict crashes in situations where participants are spatially disoriented.

Our goal is to train and compare recurrent neural networks (RNN) and non-RNN deep learning models to predict the occurrence of crashes before they happen in a disorienting aviation and spaceflight analog task. The contributions of this paper are the following:

- We present a novel data collection approach for gathering balance control data under a spatially disoriented condition. The novel dataset is available at (double-blind-safe link) https://figshare.com/s/7d935199c01c0edcafa1
- We propose a novel approach to training and selecting a model based on our aviation and spaceflight analog condition. We use area under receiver operating characteristic curve (AUC), which is a threshold-invariant metric, for convergence during training and for model selection during cross validation. We then use precision at recall for final model selection and threshold tuning.
- We demonstrate the superiority of using RNN, specifically stacked gated recurrent units (GRU), over other deep learning algorithms in the novel sequence signal classification task of aircraft and spaceflight crash prediction. The code will be publicly available at (see crash_prediction.zip in Supplementary Material).

2 Background and Related Work

2.1 Aviation and Spaceflight Analog Setting

To study spatial disorientation and resulting crashes, we collect data based on our prior work (Anonymous 2017), where human participants are secured into a multi-axis rotation system (MARS) device programmed with inverted pendulum dynamics (Figure 1). The inverted pendulum dynamics mean that the device, without any control input, is stable at the balance point (θ=0º) and any small deviation causes the device to move away from the balance point. The paradigm uses inverted pendulum dynamics because of its relevance to unstable vehicle control. Participants use a joystick to dynamically stabilize themselves around the balance point at θ=0º. They are blindfolded because the occurrence of accidents caused by spatial disorientation is significantly higher when visual information is compromised such as when flying at night or through clouds (Lyons et al. 2006; Takada et al. 2009).

In normal conditions, individuals heavily rely on gravitational cues when balancing. These gravitational cues are detected by the vestibular and somatosensory systems as participants tilt away from the gravitational vertical. To create the disorienting condition where participants cannot rely on gravitational cues, the analog paradigm orients the MARS motion in the horizontal roll plane. In this orientation, the MARS is always perpendicular to the gravitational vertical. Because the MARS does not tilt relative to the gravitational vertical, participants can no longer rely on gravitational cues to determine their angular position from the balance point. For this reason, this is also a spaceflight analog condition because astronauts cannot rely on gravitational cues when they are initially landing on a planet or the moon and they will experience similar incidents of spatial disorientation that could be potentially fatal (Anonymous 2017).

In our condition, 90% of participants report spatial disorientation and 100% of them show it in their data as a characteristic pattern of positional drifting (Anonymous 2019). Participants show minimal learning and a high rate of crashes (reaching boundaries at ± 60º). No current model explains or predicts participant control actions and perceptions.

2.2 Sequence Signal Classification

In our experiments, we cast our task as a binary classification of whether a crash will happen based on a sequence of control signals obtained from human balancing data. To enable shorter latency in future real-world implementations, where computational resources may be limited, we opt for models with lower complexity and implement several commonly used non-recurrent and recurrent deep learning models for sequential signal classification, namely, multilayer perceptron (MLP) (Rosenblatt 1961), convolutional neural network (CNN) (LeCun et al. 1990), and RNN models including single-layer long short-term memory (LSTM) (Graves 2012), GRU (Cho et al. 2014), and stacked LSTM and GRU (Pascanu et al. 2013).

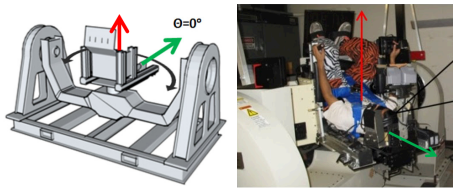

Figure 1: MARS in our spaceflight analog condition. Left: MARS in horizontal roll plane with the balance point at θ = 0º. Right: human controlling MARS, observed at a different angle.

While MLP (e.g. Ng, Feldman and Peng 2020), CNN (e.g. Gunasekaran et al. 2021; Yang et al. 2021), RNN (e.g. Jurgovsky et al. 2018; Fauvel et al. 2020), stacked RNN (e.g. Sha and Hong, 2017; Peters et al. 2018; Sun, Boukerche, and Tao 2020) have all been widely used in sequential signal classification and extensively compared in applications such as text, audio, and image (Fawaz et al. 2019), we hypothesize that in our specific task the recurrent models will achieve better evaluation metrics due to the potential long-range dependency of the signals, such as control patterns.

3 The Dynamic Stabilization Task in an Aviation and Spaceflight Analog Condition

3.1 Data Collection

We collect data using our experimental paradigm described in Section 2.1. The dynamics of our MARS device (Figure 1) is governed by the following equation:

$$\ddot{\theta} = k_p sin\theta$$

where θ is the angular deviation from the direction of balance and k_P is the pendulum constant. We use a pendulum constant of 600º/s² (≈0.52 Hz), which creates a difficult balancing task that requires learning. We program "crash" limits that restrict the angular range of the MARS to ±60º from the direction of balance. At every time step (~0.02 sec), a velocity increment proportional to the joystick deflection is added to the MARS velocity and a Runge-Kutta RK4 solver (Lambert 1973) calculates the new MARS angular position and velocity.

Human subjects are secured in the MARS, which is oriented in the horizontal roll plane, with a five-point harness, a lap belt, lateral support plates and foot straps (Figure 1). To prevent visual or auditory cues, they are blindfolded and wear earplugs and noise cancelling headphones that play white noise. A Logitech Freedom 2.4 cordless joystick is attached to the right armrest which subjects use to balance

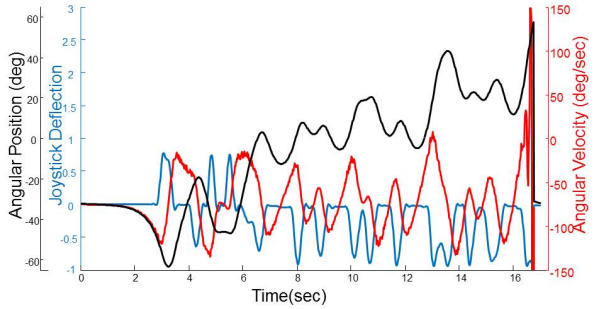

Figure 2: An episode segment of the trial data from a representative participant showing angular position (black), angular velocity (red) and joystick deflection (blue).

themselves. An auditory signal "begin" is given to subjects at the onset of each trial, and the joystick is simultaneously enabled. Whenever they reach the crash boundaries, they receive an auditory signal "lost control, resetting", and the MARS device is reset. During the reset, the joystick is disabled as the MARS automatically resets to the start position at a rate of 5º/s. Subjects participate in two sessions conducted on two consecutive days. In each session, they undergo five blocks of four trials, with each trial consisting of 100 cumulative seconds of balancing, excluding the reset times after crashes, or a total elapsed time of 150 seconds. Subjects are given no verbal feedback about their performance.

In total, 34 healthy adult subjects (18 females and 16 males, 20.4 ± 2.0 years old) who have no prior experience in our experiment gave written consent to participate in the experiment as approved by the Institutional Review Board. At a sampling rate of 50 Hz, for every trial, we obtain the angular position, angular velocity, and joystick deflections. Figure 2 provides an example of a data segment. The angular position (black) shows the characteristic pattern of positional drifting where a representative participant oscillates away from the balance point at 0º until they hit the crash boundary at 60º.

4 Experiments

4.1 Preprocessing

Our goal is to develop a model that can predict the occurrence of crashes before they happen. To generate training samples from the raw data, we slide a fixed size window from a crash event to the next one and extract each sequence segment (i.e. time series) covered by the sliding window as a sample (Figure 3). We exclude the sequence segments that overlap with the MARS resetting times (after the MARS reaches the crash boundary of 60º, it will automatically reset and the subject will have no control over the machine until the 0º point is reached.). A sample is labelled as a "crash" if

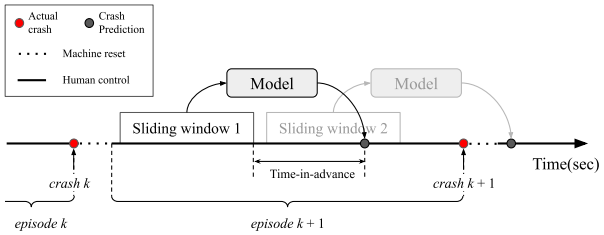

Figure 3: The crash prediction scheme for the dynamic stabilization task in the spaceflight analog condition.

a crash happens within the "time-in-advance" interval, e.g., in Figure 3, "sliding window 2" will be labelled as a crash. Otherwise, it is labelled as a "non-crash", e.g. sliding window 1 in Figure 3. Due to the rarity of the crashes, "non-crash" samples severely outnumber "crash" samples, leading to a highly imbalanced dataset. The ratios between the non-crash and crash samples range from 10.486 to 65.991, depending on the dataset configuration (see Technical Appendix A).

Previous work finds that when the MARS angular position, velocity, and joystick deflection all have the same sign (i.e. all pointed in the same direction), it will always cause the MARS to accelerate towards the crash boundaries. These are called "destabilizing joystick deflections" and are indicative of non-proficient control patterns (Anonymous 2020). To incorporate this expert knowledge, we also engineered destabilizing joystick deflection as the fourth feature for each time step in the sliding window. Destabilizing joystick deflection is a Boolean feature that is true if position, velocity, and joystick deflection all of the same sign.

4.2 Experimental Setting

Algorithms

In our experiments, we compare the following deep learning algorithms:

- MLP. Hidden layers sizes: 50, 50, 50
- CNN. Kernel size: 3, 4; filter number: 128
- LSTM. Hidden layer size: 128; dropout: 0.5; followed by a fully connected layer of size 128, a Rectified Linear Unit (ReLU) activation, and a sigmoid output layer
- GRU. Same as LSTM
- stacked LSTM. Hidden layer 1 size of 128 stacked on hidden layer 2 of size 128; rest same as LSTM
- stacked GRU. Same as stacked LSTM

The training converges on AUC with a patience of 3 epochs. All models are implemented using Keras and trained on two NVIDIA RTX 2080Ti GPU cards. For all models, we use the binary cross-entropy loss in the objective function. The Adam optimizer (Kingma and Ba 2014) is used to train the model with the settings of learning rate = 0.001, beta1 = 0.9, beta2 = 0.999, epsilon = 1e-08, and mini-batch size = 256.

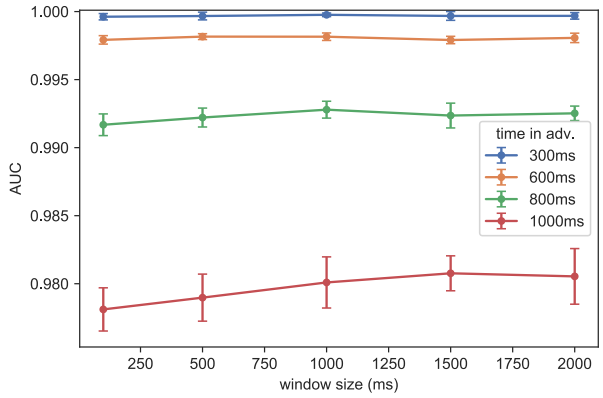

Figure 4: AUC values from stacked GRU at different time-in-advance duration and window size combinations.

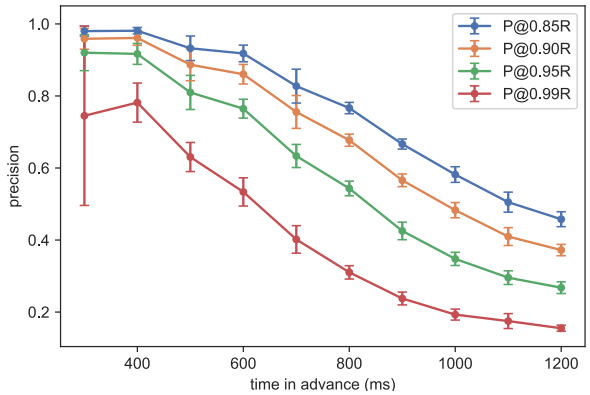

Figure 5: AUC and precision at different recall values and time-in-advance durations, for stacked GRU at 1000ms window size.

Note that the hyperparameters above are all manually tuned; the performance may be further improved upon more extensive tuning.

Classification Performance Evaluation

We evaluate the deep learning classifiers with 10-fold cross-validation (CV). To prevent data leakage, we set a project-wide random seed and group the data by episode, which is defined by the human control signals ending in either crash or trial ends. We then group-split the dataset randomly into training, validation, and test sets. This ensures that no overlapping windows will leak into the test set, causing the model to be over-confident. 10% of episodes are held out as test set to be used in data analysis.

We evaluate model performance primarily with AUC, a threshold invariant metric for model selection during 10-fold CV, and precision at 0.95 recall (P@0.95R), a metric more suitable for our application. Recall is defined by:

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}}$$

Thus, a higher recall in our application means higher percentage of all crash samples being correctly classified as crash by the model. Precision is defined by:

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}$$

In our application, a higher precision means higher percentage of the model's crash predictions are correct. The selection of the specific recall value of 0.95 is discussed in greater detail in Section 5.1.

In addition, to address the class imbalance issue, we experiment with balanced class weights when designing the loss function, which did not bring significant improvement. We also tried converging on the other main metric, P@0.95R, which consumes more training time (averaging 40 hours per 10-fold CV for stacked GRU) and does not yield any improvement.

5 Results and Discussion

5.1 Evaluation Metric Selection

We apply 10-fold cross-validation, AUC, and precision at recall to evaluate the performance of our approach. In addition to prediction accuracy, we also want to predict crashes as early as possible.

Models	Window Size	P@0.95R at Time-In-Advance (%)		
		300ms	600ms	1000ms
MLP	500ms	78.63 ± 6.01	66.9 ± 2.43	29.01 ± 1.45
	1000ms	85.98 ± 5.31	67.14 ± 2.99	30.99 ± 1.17
	1500ms	74.23 ± 14.88	65.79 ± 3.71	31.19 ± 1.67
CNN	500ms	84.01 ± 3.89	65.45 ± 3	28.72 ± 1.71
	1000ms	82.25 ± 5.34	65.76 ± 4.86	30.47 ± 1.32
	1500ms	85.25 ± 5.96	67.58 ± 3.95	32.28 ± 1.4
LSTM	500ms	92.02 ± 3.6	75.41 ± 3.91	31.71 ± 1.62
	1000ms	90.38 ± 5.99	75.16 ± 3.11	34.34 ± 2.49
	1500ms	91.08 ± 5.63	72.94 ± 2.68	34.24 ± 1.8
GRU	500ms	91.94 ± 5.73	75.05 ± 3.26	32.2 ± 2.27
	1000ms	91.44 ± 3.32	74.36 ± 3.23	34.05 ± 2.17
	1500ms	90.62 ± 5.89	71.35 ± 2.94	34.35 ± 1.61
stacked LSTM	500ms	89.8 ± 7.85	75.21 ± 2.48	32.59 ± 1.72
	1000ms	93.34 ± 3.25	76.18 ± 3.99	34.74 ± 2.36
	1500ms	92.46 ± 3.76	76.88 ± 1.32	34.93 ± 1.64
stacked GRU	500ms	89.7 ± 6.58	76.63 ± 2.63	32.83 ± 1.45
	1000ms	92.02 ± 4.98	76.47 ± 2.62	34.77 ± 1.84
	1500ms	90.42 ± 6.73	74.92 ± 1.74	35.81 ± 1.87

Table 1: precision at 0.95 recall (P@0.95R) scores averaged over 10-fold cross-validation for various model type, window size, and time-in-advance combinations.

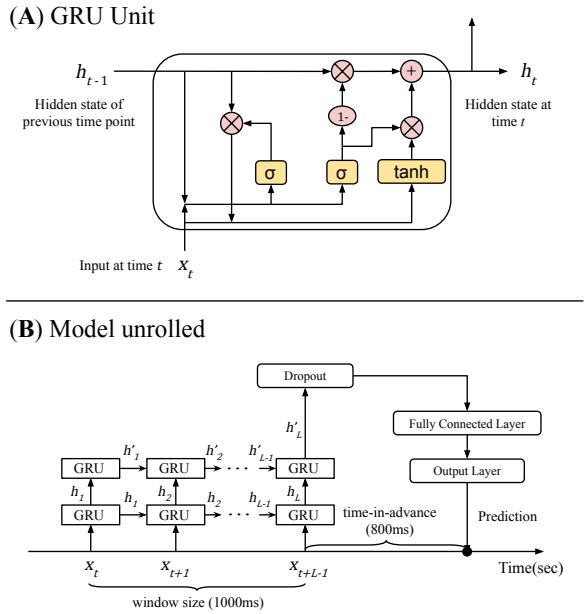

Figure 6: The final crash prediction model. (A) A closeup of the GRU cell. (B) The stacked GRU model unrolls to process an input signal sequence of 1000ms and produce a prediction of if a crash will happen in 800ms.

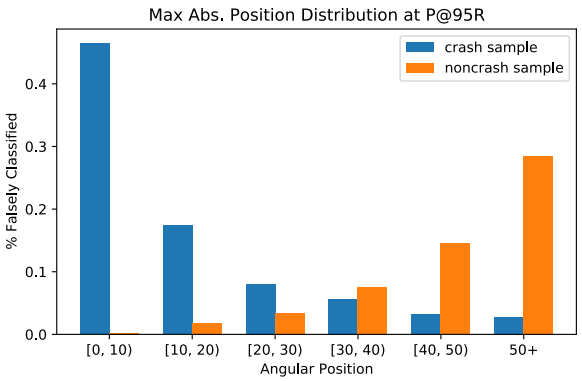

Figure 7: Samples with higher max absolute position have lower false negative rates and higher false positive rates.

We are able to predict the occurrence of crashes as early as 1000ms in advance with high AUC where the window size has minimal effect (Figure 4). However, for the longer time-in-advance duration of 1500ms, the AUC values are much lower and at that point having larger window sizes helps.

All model types for the same time-in-advance have high AUC values over 0.973 (see Technical Appendix B). While high AUC may reflect how well the model distinguishes the positive and negative labels, they do not provide a full view of how useful the model would be in our practice, where cost-driven threshold tuning is crucial. In the fields of aircraft and spacecraft control, false negatives have much higher costs than false positives. False negatives, where the model incorrectly predicts that no crash will occur, will likely lead to death. In contrast, while false positives are not ideal and may damage a pilot's trust and engagement with the system, it will not lead to death and in some situations may beneficially cause the pilot to be on alert.

Therefore, to prioritize minimizing false negatives, we prefer models with high precision values where their threshold is set to yield a high recall value. Our data suggest that there exists a tradeoff between recall and precision in all model types, where setting a higher recall results in a lower precision (e.g., stacked GRU as shown in Figure 5). Figure 5 revealed that longer time-in-advance durations result in smaller values of precision. For the rest of the analysis, we choose a high recall of 0.95 because we want to reduce missed crashes.

5.2 Dataset Configuration and Model Selection

We then determine the best dataset configuration, i.e., window size and time-in-advance, and model type for a detailed error analysis based on the P@0.95R values shown in Table 1 and participant statistics. The P@0.95R values suggest that all RNN models outperform non-RNN models, and, moreover, stacked-RNNs perform somewhat better than single-layer RNNs. We eventually choose stacked GRU for detailed examination in Section 5, because the time to train a stacked GRU (average 12.63 hours per 10-fold CV) is significantly shorter than stacked LSTM (average 15.01 hours per 10-fold CV), without degradation in evaluation metrics.

Because participants make full joystick deflection at 1-2 Hz, we choose a data window of 1000ms to capture enough joystick behavior. We want the time-in-advance duration to be as long as possible and the recall to be very high, which empirically results in a compromise on the precision value. Based on these constraints, we choose 800ms as the best time-in-advance duration. The chosen parameters result in a 10-fold CV AUC of 0.9927 ± 0.0006 and P@0.95R of 0.5432 ± 0.0203. The model takes on average 30ms to classify one data window on a 2.5GHz Intel Xeon CPU, which should impose few latency issues on real-time implementation. Figure 6 illustrates how the stacked GRU model makes prediction on the dataset with parameters of 1000ms window size and 800ms time-in-advance.

5.3 Result Analysis at High Recall

To further understand our model's capabilities, we examine those crashes that are misclassified as non-crash by the model even at a recall as high as 95%. We conduct the result analysis below on the test set, where the selected parameters described in Section 5.2 result in an AUC of 0.9918 and

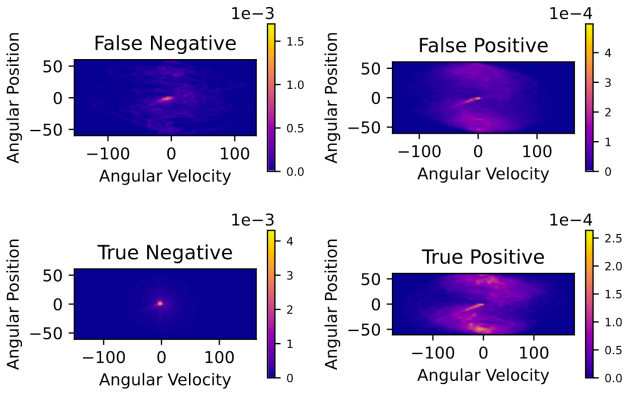

Figure 8: Density maps of the velocity and position at all time steps in time-in-advance duration.

Type of Prediction	False Negative	False Positive	True Negative	True Positive
% of DJD	67.50%	58.92%	53.94%	33.64%

Table 2: Percentage of predictions containing unexpected destabilizing joystick deflection in time-in-advance duration.

P@0.95R of 0.5040. Surprisingly, in Figure 7, we find that most of the false negative crashes happen when subjects are near the balance point, which is the furthest point from the crash boundaries and where the intrinsic acceleration is lowest (angular position is 0º).

To understand what causes a crash at such a seemingly safe location, we examine the data in the 800ms time-in-advance duration following these false negative inputs, i.e., outside of the 1000ms sliding window data available to the model. We find that subjects are making destabilizing joystick deflections in the time-in-advance duration which leads to the crashes. Table 2 shows that the percentage of destabilizing joystick deflections in the time-in-advance duration is the greatest for false negatives (Table 2), meaning that participants often make unexpected destabilizing joystick deflections in the time-in-advance portion. This suggests that the reason the model predicts no crash is because the participants are performing well near the balance point but then unexpectedly initiate a destabilizing joystick deflection because they are disoriented and do not have a clear sense of their orientation. False positives also have a large percentage of destabilizing joystick deflections in the time-in-advance duration, most likely because participants are making errors which result in high angular position and velocity, which causes the model to predict a crash will occur. True positives have the smallest percentage because critical errors have already been made that would lead to a crash.

Based on Figure 5, to have a high recall, we need to accept a low precision. However, a large portion of false positives may still prove beneficial in our situation. To understand when false positives occur, in Figure 8, we create a density map of angular position and velocity. We find that false positives have higher angular position and velocity, suggesting that the model is identifying dangerous behavior as being potential places of crashes. In the future, the false positives can be considered as a warning system as a pilot approaches the dangerous limits. We find similar results for true positives, which is expected because the greatest danger is at high values of position and velocity.

6 Conclusion and Discussion

Our objective is to create a model that can predict the occurrence of crashes in a stabilization task when participants are spatially disoriented similar to what pilots and astronauts may experience. We use the sensory readouts of angular position, velocity and joystick deflections from the stabilization task to train a stacked GRU model to predict whether a crash will occur at a certain future time point. Based on the obtained AUC, we conclude that our model performs well at predicting crashes even as early as 1500ms (Figure 5). However, for a crash avoidance system in the aviation industry, false negative errors (model incorrectly predicts no crash) will be considered much worse than false positive errors (model incorrectly predicts that a crash will occur). As a result, we set the recall very high (95%), so that the model rarely misses any crashes, which results in a low precision (50%), meaning that many non-crash events are labelled as crashes. We first explore what types of crashes the model is not able to predict even at high recall values. We are surprised to find that many of the false negative crashes start near the equilibrium point (0º) (Figure 7). We discover that the main reason why the model is unable to predict these crashes is because participants, in the time-in-advance duration, make unexpected destabilizing joystick deflections which throws the device away from the balance point (Table 2). These destabilizing joystick deflections are made because participants are disoriented and do not know where they are.

We have a relatively low precision (0.50) and find that many of false positives occur at very large angular positions and velocities (Figure 8). For this reason, these false positives could serve as a warning signal for pilots who may be disoriented and reach dangerous angular deviations and velocities. Having too high a false positive rate can result in the pilots losing trust with the model. It is not well understood what values of false positives will maintain trust between pilot and Artificial Intelligence (AI), or what level of trust in the AI is useful because case studies show that both too much and too little trust in the technology can lead to fatal accidents in ships and aircraft (Wickens 1995; Dalcher 2007; Hoff and Bashir 2015). Our future work will explore the role of trust and the rate of acceptable false positives.

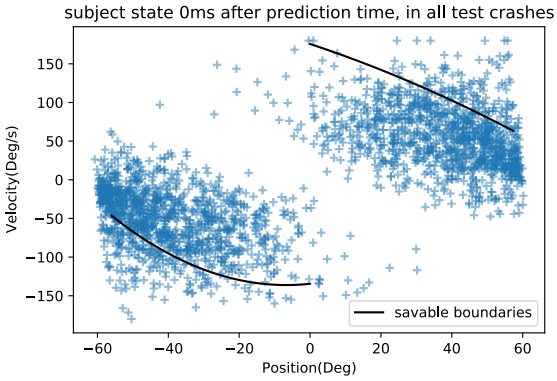

Figure 9: All crashes plotted against savable boundaries at prediction time.

Time After Window Ends	0ms	200ms	400ms	600ms	800ms
% savable	80.71%	55.42%	30.30%	8.54%	0%

Table 3: Percentage of savable crashes reduces as time elapses since prediction time.

Using the parameters that yield 0.95 recall and 0.50 precision, we wonder whether a warning signal 800ms in advance will prevent crashes. Figure 9 and Table 3 show that if immediate control is taken, 80.7% of crashes could be avoided. Humans can respond to stimuli in 250ms; however, to respond to more complex cues that require joystick deflections, participants would likely need 500-1000ms of time. Table 3 shows that with increasing reaction times, receiving a warning will not be sufficient to prevent crashes. These results suggest that imminent crashes that are detected will be best resolved if the AI system takes temporary control of the aircraft. In the case of an AI making immediate responses, we can choose shorter time-in-advance durations of 300-500ms, where precision can be much higher at a recall as high as 0.95 (Figure 5; see Technical Appendix C for detailed values).

Since a fully AI-controlled aircraft or spacecraft is not currently possible for applications, the pilot will be in charge of much of the flight. Therefore, reducing situations where a pilot enters into the danger zone will be important. In future studies, we will train the model to identify different levels of danger (as opposed to only crashes), such as patterns of behavior that lead to high angular positions and velocities. This will allow us to develop a warning system and will also reduce the percentage of false positives. Using our current work, we will develop a real time alerting system based on our novel deep learning model that will provide the warning through vibrotactile feedback which has been previously shown to be useful for spatial disorientation during airflight (Rupert 2000). This is feasible because the computations for one window take 30ms which would add minimal latency to the feedback.

Ethical impact and broad societal implications

Spatial disorientation is a significant societal problem where some studies estimate 90-100% of pilots experience it (Newman 2007; Gibb, Ercoline and Scharff 2011). Because spatial disorientation is responsible for up to 32% of major accidents and up to 26% of those lead to death (Newman 2007), developing solutions will save many lives on Earth and, in the future, for space exploration. In this study, we use a new data gathering technique to the field of AI and deep learning, where we collect data from a balancing task that reliably leads to spatial disorientation and loss of control. The deep learning and AI communities have explored problems related to crash avoidance for vehicles (Peng et al. 2019), autonomous vehicles (Perumal et al. 2021), unmanned aerial vehicles (Gandhi, Pinto and Gupta 2017), ships (Perera 2018), swarming systems (Lan, Liu and Zhao 2020), aircraft collisions with other aircraft (Julian, Kochenderfer and Owen 2019). However, no one to our knowledge has used deep learning to predict the occurrence of crashes in a novel analog condition where the operators experience disorientation similar to what pilots and astronauts will experience. We do not use any prior knowledge of the vestibular system or inverted pendulum dynamics, meaning that our implementation could be generalizable to other balance conditions such as loss of control in human postural balancing in novel force environments (Bakshi, DiZio and Lackner 2020) and with vestibular patients (Lawson, Rupert and McGrath 2016). In addition, we do not use any prior knowledge of human operators, and hence minimized potential ethical biases in choosing the deep learning models.

References

Bakshi, A.; DiZio, P.; and Lackner, J.R. 2020. The effect of hypergravity on upright balance and voluntary sway. *Journal of Neurophysiology*, *124*(6): 1986-1994. doi.org/10.1152/jn.00611.2019.

Braithwaite, M.G.; Durnford, S.J.; Crowley, J.S.; Rosado, N.R.; and Albano, J.P. 1998. Spatial disorientation in US Army rotary-wing operations. *Aviation, space, and environmental medicine*.

Cho, K.; Van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; and Bengio, Y. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Clément, G.R.; Boyle, R.D.; George, K.A.; Nelson, G.A.; Reschke, M.F.; Williams, T.J.; and Paloski, W.H. 2020. Challenges to the central nervous system during human spaceflight missions to Mars. *Journal of neurophysiology*, *123*(5): 2037-2063. doi.org/10.1152/jn.00476.2019.

Daiker, R.; Ellis, K.; Mathan, S.; and Redmond, W.A. 2018. Use of Real-Time, Predictive Human Modeling for Spatial Disorientation Detection and Mitigation. In *Modsim World 2018 Conference*.

Dalcher, D. 2007. Why the pilot cannot be blamed: a cautionary note about excessive reliance on technology. *International Journal of Risk Assessment and Management*, *7*(3): 350-366. doi.org/ 10.1504/IJRAM.2007.011988.

Fauvel, K.; Balouek-Thomert, D.; Melgar, D.; Silva, P.; Simonet, A.; Antoniu, G.; Costan, A.; Masson, V.; Parashar, M.; Rodero, I.; and Termier, A. 2020. A distributed multi-sensor machine learning approach to earthquake early warning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34: 403-411. doi.org/10.1609/aaai.v34i01.5376.

Fawaz, H.I.; Forestier, G.; Weber, J.; Idoumghar, L.; and Muller, P.A. 2019. Deep learning for time series classification: a review. *Data mining and knowledge discovery*, *33*(4), pp.917-963. doi.org/10.1007/s10618-019-00619-1.

Gandhi, D.; Pinto, L.; and Gupta, A. 2017, September. Learning to fly by crashing. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 3948-3955). IEEE. doi.org/10.1109/IROS.2017.8206247.

Gibb, R.; Ercoline, B.; and Scharff, L. 2011. Spatial disorientation: decades of pilot fatalities. *Aviation, space, and environmental medicine*, *82*(7): 717-724. doi.org/10.3357/asem.3048.2011.

Graves, A. 2012. Supervised sequence labelling. In *Supervised sequence labelling with recurrent neural networks*. Springer, Berlin, Heidelberg. doi.org/10.1007/978-3-642-24797-2.

Gunasekaran, H.; Ramalakshmi, K.; Ramanathan, S.; and Venkatesan, R., A Deep Learning CNN Model for Genome Sequence Classification. In *Intelligent Computing Applications for COVID-19*. CRC Press.

Hochreiter, S.; and Schmidhuber, J. 1997. Long short-term memory. *Neural computation*, *9*(8): 1735-1780. doi.org/10.1162/neco.1997.9.8.1735.

Hoff, K.A.; and Bashir, M. 2015. Trust in automation: Integrating empirical evidence on factors that influence trust. *Human factors*, *57*(3): 407-434. doi.org/10.1177/0018720814547570.

Julian, K.D.; Kochenderfer, M.J.; and Owen, M.P. 2019. Deep neural network compression for aircraft collision avoidance systems. *Journal of Guidance, Control, and Dynamics*, *42*(3): 598-608. doi.org/10.2514/1.G003724.

Jurgovsky, J.; Granitzer, M.; Ziegler, K.; Calabretto, S.; Portier, P.E.; He-Guelton, L.; and Caelen, O. 2018. Sequence classification for credit-card fraud detection. *Expert Systems with Applications*, *100*: 234-245. doi.org/10.1016/j.eswa.2018.01.037.

Kingma, D. P.; and Ba, J. 2014. Adam: A Method for Stochastic Optimization. arXiv preprint arXiv:1412.6980

Lambert, J.D. 1973. *Computational methods in ordinary differential equations.* volume 5. John Wiley & Sons Incorporated. doi.org/10.1002/zamm.19740540726.

Lan, X.; Liu, Y.; and Zhao, Z. 2020. Cooperative control for swarming systems based on reinforcement learning in unknown dynamic environment. *Neurocomputing*, *410*: 410-418. doi.org/ 10.1016/j.neucom.2020.06.038.

Lawson, B.D.; Rupert, A.H.; and McGrath, B.J. 2016. The neurovestibular challenges of astronauts and balance patients: some past countermeasures and two alternative approaches to elicitation, assessment and mitigation. *Frontiers in systems neuroscience*, 10:96. doi.org/10.3389/fnsys.2016.00096.

LeCun, Y.; Boser, B.; Denker, J.; Henderson, D.; Howard, R.; Hubbard, W.; and Jackel, L. 1989. Handwritten digit recognition with a back-propagation network. *Advances in neural information processing systems*, *2*.

Lyons, T.J.; Ercoline, W.; O'Toole, K.; and Grayson, K. 2006. Aircraft and related factors in crashes involving spatial disorientation: 15 years of US Air Force data. *Aviation, space, and environmental medicine*, *77*(7): 720-723.

Newman, D.G. 2007. *An overview of spatial disorientation as a factor in aviation accidents and incidents* (No. B2007/0063). Canberra City, Australia: Australian Transport Safety Bureau.

Ng, K.Y.; Feldman, A.; and Peng, J. 2020, April. Linguistic fingerprints of internet censorship: The case of sina weibo. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34: 446-453. doi.org/10.1609/aaai.v34i01.5381.

Pascanu, R.; Gulcehre, C.; Cho, K.; and Bengio, Y. 2013. How to construct deep recurrent neural networks. *arXiv preprint arXiv:1312.6026*.

Peng, L.; Sotelo, M.A.; He, Y.; Ai, Y.; and Li, Z. 2019. Rough Set Based Method for Vehicle Collision Risk Assessment Through Inferring Driver's Braking Actions in Near-Crash Situations. *IEEE Intelligent Transportation Systems Magazine*, *11*(2): 54-69.

Perera, L.P. 2018, June. Autonomous ship navigation under deep learning and the challenges in COLREGs. In *International Conference on Offshore Mechanics and Arctic Engineering*, volume 51333. American Society of Mechanical Engineers. doi.org/ 10.1115/OMAE2018-77672.

Perumal, P.S.; Sujasree, M.; Chavhan, S.; Gupta, D.; Mukthineni, V.; Shimgekar, S.R.; Khanna, A.; and Fortino, G. 2021. An insight into crash avoidance and overtaking advice systems for Autonomous Vehicles: A review, challenges and solutions. *Engineering applications of artificial intelligence*, *104*: 104406. doi.org/10.1016/j.engappai.2021.104406.

Peters, M.E.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; and Zettlemoyer, L. 2018. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.

Rosenblatt, F. 1961. *Principles of neurodynamics. perceptrons and the theory of brain mechanisms*. Cornell Aeronautical Lab Inc Buffalo NY.

Rupert, A.H. 2000. An instrumentation solution for reducing spatial disorientation mishaps. *IEEE Engineering in Medicine and Biology Magazine*, *19*(2): 71-80. doi.org/ 10.1109/51.827409.

Sha, L.; and Hong, P. 2017, September. Neural knowledge tracing. In *International conference on brain function assessment in learning*. Springer, Cham. doi.org/10.1007/978-3-319-67615-9_10.

Shelhamer, M. 2015. Trends in sensorimotor research and countermeasures for exploration-class space flights. *Frontiers in systems neuroscience*, *9*:115. doi.org/10.3389/fnsys.2015.00115.

Sun, P.; Boukerche, A.; and Tao, Y. 2020. SSGRU: A novel hybrid stacked GRU-based traffic volume prediction approach in a road network. *Computer Communications*, *160:* 502-511. doi.org/ 10.1016/j.comcom.2020.06.028.

Takada, Y.; Hisada, T.; Kuwada, N.; Sakai, M.; and Akamatsu, T. 2009. Survey of severe spatial disorientation episodes in Japan Air Self-Defense Force fighter pilots showing increased severity in night flight. *Military medicine*, *174*(6): 626-630. doi.org/10.7205/milmed-d-01-6308.

Wickens, C.D. 1995. Designing for situation awareness and trust in automation. *IFAC Proceedings Volumes*, *28*(23): 365-370. doi.org/10.1016/S1474-6670(17)46646-8.

Yang, R.; Zha, X.; Liu, K.; and Xu, S. 2021. A CNN model embedded with local feature knowledge and its application to time-varying signal classification. *Neural Networks*, *142*: 564-572. doi.org/10.1016/j.neunet.2021.07.018.