Knowledge Acquisition in the Cockpit Using One-Shot Learning

Evana Gizzi*, Lisa Le Vie†, Matthias Scheutz*, Vasanth Sarathy*, Jivko Sinapov*

*Department of Computer Science
Tufts University
Medford, MA
Evana.Gizzi@tufts.edu

†Crew Systems and Aviation Operations Branch
NASA Langley Research Center,
Hampton, VA

***Abstract*—Intelligent systems for aviation need to be capable of understanding and representing anomalous events as they happen in real-time. We explore this problem with a proof of concept framework based on contextual one-shot learning, run on a human-in-the-loop flight simulator. We ran a total of 24 trials, with variations in training, fliers, and set values within the framework, and found that our framework was able to detect and reason about anomalies in all trials. In future work, we would like to explore different heuristics for anomaly reasoning, including nonlinear interactions of cockpit data, and feedback from the flight crew through psychophysiology sensors or natural language interactions.**

***Index Terms*—symbolic, aviation, knowledge, representation**

I. Introduction

Intelligent systems are becoming increasingly present in aviation, with next generation systems like the NASA's Intelligent Flight Control System (IFCS) and the Federal Aviation Administrations Decision Support System (DDS), both of which have helped to optimize aircraft and pilot performance through the use of data driven approaches. While these systems have been shown to increase safety and efficacy of missions, they remain limited in their data-driven approach to synthesizing intelligence, which employs black-boxed methods that are not conducive to acquiring new knowledge representations, or handling anomalous circumstances. There have been some efforts to resolve this problem, both in aviation and non aviation contexts [5]. The general problem of augmenting data driven paradigms with symbolic reasoning systems to resolve the knowledge acquisition issue has been explored in Neural Symbolic Learning and Reasoning. However, to the best of our knowledge, there have been no frameworks implemented within the aviation domain which employ symbolic reasoning measures to acquire new knowledge representations. We explore this problem specifically by looking at the anomaly detection and reasoning domain [1], [2], [3]. This area of research looks mostly at domain specific applications [4], and focuses almost fully on the anomaly detection portion of research. In our research, we implement a proof of concept framework for anomaly detection and reasoning in a real-time, human in the loop aviation setting. The framework is generalizable across various aviations contexts, and shows promise for the general problem of knowledge acquisition in the cockpit setting [18].

This work was motivated by a vision to build a decision support system for the aviation domain. The system would be able to support the pilot at a level that is comparable or better than that which is currently provided by the copilot. The system would be equipped with the ability to reason like a human, while also leveraging the computational power of a machine. The implementation of this system is less trivial than it may seem, as it would require both statistical and non statistical paradigms to capture the human and machine capabilities. Let us consider the necessary components for this system:

1) **Subject matter expert (SME):** The system would need to be able to answer any subject matter specific questions that the pilot may have. This knowledge base would require all information available in commercial airline flight manuals, including algorithmic maneuvers or situational instructions. The SME capability could be engineered using data driven statistical means, leveraging technologies such as IBM Watson.
2) **Reasoning based on regularities of flight:** This capability would aid in real-time reasoning by using past flight data as a means for reasoning and prediction. This portion could be implemented using data driven statistical means which leverage pattern recognition methods for predictive capabilities.
3) **Reasoning in anomalous circumstances:** The anomaly handling capability would aid the pilot in circumstances that are fully or partially novel. In these circumstances, there is no past data or information in flight manuals available to support reasoning. Quite trivially, this portion of the system could not be implemented using a data driven statistical approach.

Our research focuses on developing a framework and proof of concept for the third portion of the system. We first develop a methodology for detecting anomalies and reasoning about why they occurred. Once a sufficient explanation for their occurrence is developed, the system creates a knowledge representation of the anomalous circumstance in order to be able to handle any similar future occurrences of the anomaly.

II. Preliminaries

To begin, we review a set of preliminary concepts used to construct our framework and proof of concept implementation.

A. One-Shot Learning

One shot learning is the process of learning something from one exemplar. In most instances of one-shot learning applications, the concept or data point to be learned is non trivial, in that a rich knowledge representation is required for success [10]. For example, in image processing, one-shot learning would exist as learning an image classifier using just one training sample [11]. One-Shot Learning is challenging to accomplish using data driven paradigms, which rely on data to inform output [16], [17]. Typically, these paradigms use techniques such as pattern recognition or the training of classifiers as ways to generate accurate output. One-shot learning is well suited for instances which rely on little to no data, or sparse/unreliable data sets, In these cases, knowledge representations are instead derived by using context, relevancy, and other situational data.

B. Anomalies vs Outliers

We define *abnormalities* as points which deviate from what is expected. Within the context of our framework, an abnormality would exist as any instance of a scenario in flight where an observed action is at odds with a predicted action. We further classify points as either *outliers* or *anomalies*. *Outliers* are irregularities which are not important, but rather present themselves in a way that is similar in nature to *noise*, or *statistical outliers*. In this way, outliers provide no interest to the analyst, but rather acts as a hindrance to analysis. *Anomalies* are those points which deviate from an expected outcome in a way that is of interest to the analyst. These irregularities are of particular importance, usually calling for an actionable outcome. Within the context of our framework, this actionable outcome is discovering what saliency point is responsible for the anomaly, and creating a new context with associated rule representations to characterize and handle that novel scenario.

III. Aviation Example

Here, we review a basic aviation example to describe how the preliminary concepts will be used in our framework. Our example will also prime our review of the Knowledge Acquisition Framework in the following section.

A. The "Go-Around" Scenario

Suppose it is the case that we have a context *C*, where an aircraft is unstable below 1000ft (304.8 m) on its approach. Formally, *C* = $\{$*UNSTABLE, SUB1000FT*$\}$. Furthermore, it has been the case that in 99 instances, $C \implies GA$ where *GA* is the outcome that the pilot has executed a Go-Around maneuver. However, in 1 instance, it has been the case that $C \implies L$ where *L* is the outcome that the pilot has executed a landing. This set of implications are problematic in that in the case of *C*, it is impossible for both *GA* and *L* to hold, and thus, we must develop a means for deterministically deciding when each rule should apply [13], [15]. Using frequency values alone, we generate the following:

$$\begin{aligned} C &\implies GA_{0.99} \\ C &\implies L_{0.01} \end{aligned} \tag{1}$$

the case of $C \implies L_{0.01}$, our interpretation says "In the context *C*, *L* should hold 1% of the time." However, if it is the case that in the 1 instance, an anomalous circumstance took place in which executing a landing was in fact very appropriate (for example, emergency landing due to some unprecedented circumstances), then we know that there is a much greater than 1% chance that *L* should hold. So although we know $C \implies L$ only held 1% of the time, we are uncertain of this prediction probability. Thus, using probability values alone is problematic in that it conflates frequency of occurrence with confidence in whether a rule should hold.

To resolve this issue, we generate rules in the following form:

$$\begin{aligned} C &\implies GA_{[0.89,0.99]} \\ C &\implies L_{[0.01,0.11]} \end{aligned} \tag{2}$$

Each rule has an associated interval, wherein lies the true probability for the hypothesis. The width of the interval around this probability value expresses the certainty on the probability value. A larger interval corresponds to a greater uncertainty whereas smaller interval corresponds to a smaller uncertainty. The intervals associated with each rule are contained by the interval [0,1]. We call these intervals the Dempster-Shafer (DS) intervals. In the next section, we describe how such intervals are generated.

B. Rule Learning with Dempster-Shafer Theoretic Framework

In order to characterize the predicted outcomes of being in a context, our framework relies on Dempster-Shafer Theory (DST). DST is a generalization of the Bayesian uncertainty framework, that allows for processing of uncertainty and ignorance on pieces of evidence supporting a claim [6], to produce a degree of belief of the existence of the claim [12]. DST is useful in cases where there is a lack of data and/or distributional information about the data to inform the existence of claims, which is typically needed in a probabilistic paradigm [7]. In our framework, we are able to leverage DST to translate our cockpit data (which we call *evidence* *measures*) into a flight-based prediction. More specifically, we use cockpit data to build contextual representations, and generate rules based on those contexts which are quantified using DST .

DST requires a scenario which contains a set of mutually exclusive hypotheses $h_1, h_2, \ldots, h_n$ which collectively are referred to as the Frame of Discernment (*FoD*), denoted by Θ, representing all possible states of the system, and pieces of evidence $e_1, e_2, \ldots, e_n$ to support those hypotheses. DST assigns a mass value to each member of the power set of Θ, denoted 2^Θ. The mapping m : $2^\Theta \implies [0,1]$ from combinations of hypotheses to mass values using pieces of evidence is called the *basic belief assignment (BBA)* where $m(\emptyset) = 0$ and $\Sigma_{A \subset \Theta} m(A) = 1$. The *BBA* is responsible for distinguishing probabilities of the occurrence of a hypothesis from the evidence measures available [8]. The elements of *A* with non zero mass are called the *focal elements* (F_Θ), and the triple $\varepsilon = \Theta, F_\Theta, m_\Theta(\cdot)$ is called the *Body of Evidence (BoE)*. Collectively, the mass values generate a lower bound called the *belief (Bel)*, and an upper bound called the *plausibility (Pl)*, on the probability of a set in 2^Θ occurring:

$$Bel(A) = \Sigma_{B \subset A} m_\Theta(B) \tag{3}$$

$$Pl(A) = \Sigma_{B \cap A} m_\Theta(B) \tag{4}$$

where $A \subset 2^\Theta$. The belief is representative of the amount of justifiable support given to A, where the plausibility can be thought of as the maximum amount of specific support that could be given by A if further justified [8]. The interval $[Bel(A), Pl(A)]$ is defined as the *evidential interval range* of *A* and the value $Pl(A) - Bel(A)$ as the *uncertainty* associated with A ($Un(A)$). Each piece of evidence contributes to the mass values of one or all hypotheses in Θ, and are combined to formulate the collective:

$$m(h) = \frac{\Sigma_{A \cap B = H \neq \phi} m(A) \cdot m(B)}{1 - \Sigma_{A \cap B \neq \phi} m(A) \cdot m(B)} \tag{5}$$

for all $h, A, B \subset \Theta$. We call this the *Dempster-Shafer Rule of Combination (DRC)*, which states that for any hypothesis H, we combine the evidence which informed A with that which informed B.

1) Evidence Updating Strategy: We replace DRC with an evidence filtering strategy, which was developed as an upgrade to DRC to address some of its shortcomings with conflicting pieces of evidence [9]. This strategy is better suited for handling the inertia of available evidence as it becomes available, and its use of conditionals handles the combination of partial or incomplete information well. Specifically, given $BoE_1 = \{\Theta, F_1, m_1\}$ and $BoE_2 = \{\Theta, F_2, m_2\}$, and some set $A \in F_2$, the *updated belief* $Bel_{k+1} : 2^\Theta \mapsto [0,1]$, and the *updated plausibility* $Pl_{k+1} : 2^\Theta \mapsto [0,1]$ of an arbitrary proposition $B \subseteq \Theta$ are:

$$Bel(B)(k+1) = \alpha_k Bel(B)(k) + \beta_k Bel(B|A)(k) \tag{6}$$

$$Pl(B)(k+1) = \alpha_k Pl(B)(k) + \beta_k Pl(B|A)(k) \tag{7}$$

where $\alpha_K, \beta_k \geq 0$ and $\alpha_k + \beta_k = 1$. The conditional used above is the Fagin-Halpern conditionals which can be considered an extension of Bayesian conditional notions [14]. Given some $BoE = \{\Theta, F, m\}$, $A \subseteq \Theta$ s.t. $Bel(A) > 0$ and an arbitrary $B \subseteq \Theta$, the *conditional belief* $Bel(B|A) : 2^\Theta \mapsto [0,1]$ and *conditional plausibility* $Pl(B|A) : 2^\Theta \mapsto [0,1]$ of *B* given *A* are:

$$\begin{aligned} Bel(B|A) &= \frac{Bel(A \cap B)}{[Bel(A \cap B) + Pl(A - B)]} \\ Pl(B|A) &= \frac{Pl(A \cap B)}{[Pl(A \cap B) + Bel(A - B)]} \end{aligned} \tag{8}$$

IV. Knowledge Acquisition Framework

Next, we review the knowledge acquisition algorithm in detail.

A. Central Elements

Our framework examines three *central elements (CE)* in order to detect abnormality points, and to further refine them as either outliers or anomalies. The central elements include the assumed *context (C)*, the *predicted outcome (P)* of being in *C*, and the currently *observed outcome (O)* that is happening while in *C* (*CE* = $\{$*C, P, O*$\}$). We define CE_{conf} as the set of confidence values for the elements of CE, characterizing the degree of certainty that a given element holds (CE_{conf} = $\{C_{conf}, P_{conf}, O_{conf}\} \in [0,1]$). We define a threshold value $T \in [0,1]$, such that

$$\begin{aligned} e < T &\implies e == LOW \\ e \geq T &\implies e == HIGH \end{aligned} \tag{9}$$

where $e \in CE_{conf}$. Given these preliminaries, we describe abnormality, outlier, and anomaly detection in the following way:

- **Abnormality:** $P \neq O$
- **Outlier:** $P \neq O$
- **Anomaly:** $P \neq O \wedge (C_{conf} == HIGH) \wedge (P_{conf} == HIGH) \wedge (O_{conf} == HIGH)$

Here, any abnormality point which is not an anomaly is an outlier. In our aviation example, this implies that *C* = $\{$*UNSTABLE, BELOW1000FT*$\}$ $\implies$ L is only anomalous if there is a high confidence that C holds, that GA should hold as a consequence of being in C, and that L is being observed instead. With a weak confidence on any of these three values in the $P \neq O$ scenario (or in this case, $GA \neq L$), the framework instead flags an outlier.

B. Rule Refinement

Once an anomaly point is detected, the algorithm makes the assumptions that there is a salient feature that is occurring in the context of the anomaly, that was not otherwise present in the assumed context. That is, the anomaly posits that some additional factor F has caused the discrepancy in P and O. Using our aviation example, the algorithm further refines the rules as follows:

Fig. 1. VISTAS simulator, NASA Langley Research Center, Hampton VA

Fig. 2. Real-time simulation running alongside Knowledge Acquisition Framework

$$\begin{aligned} C \wedge \neg F &\implies GA \\ C \wedge F &\implies L \end{aligned} \tag{10}$$

The framework generates a set of basic heuristics to identify F using cockpit data. Specifically, a basic feasibility region is generated using past flight data, extracting the minimum and maximum values of all cockpit data fields over every past flight. For example, it may pull the highest altitude that the aircraft has ever reached, or the lowest wind speeds that the aircraft has ever encountered. We call these the *critMin* and *critMax* values. Likewise, we calculate *deltaMin* and *deltaMax* values, which record the largest change in a data field over a set window of time. For example, the biggest change of speed of the aircraft over a window of 50 Hz. The moment an anomaly is incurred, the system employs one-shot learning by comparing the cockpit data at the time of occurrence against a feasibility region, to figure out which features exceed the bounds of the feasibility region.

V. Proof of Concept

A. Scenario

We validated our Knowledge Acquisition framework with a proof of concept model run in a rapid prototyping simulator at NASA Langley Research Center in Hampton VA. The experiment was run in a real-time, human in the loop setting. The selected scenario involved the landing phase of flight of a Boeing 757 descending into the Reno-Tahoe International Airport, which sits at 4450 ft above sea level. The trails started at 8000 ft (2438.4 m) altitude, and each participant was asked to force a landing in an unstable configuration. The simulator ran at 50 Hz, and the Knowledge Acquisition algorithm ran at 1 Hz (each *time step* = 1Hz), pulling cockpit data at each time step. The trials consisted of two phases; A training phase (Phase 1), and a knowledge acquisition phase (Phase 2), with a total of 24 trials across 3 participants. *participant_1* flew trials 1-12, *participant_2* flew trials 13-17, and *participant_3* flew trial 18-24. Trials 13, 14, 15, and 21 were removed due to experimental error.

TABLE I
Training Schematic

Trial Number	Training Datasets	Iterations	Schematic
1-6	$dataset\_A$	1	single
7-9	$dataset\_A$	2	single
10-12	$dataset\_A$	3	single
13-24	$dataset\_A$ $dataset\_B$ $dataset\_C$	1	multi

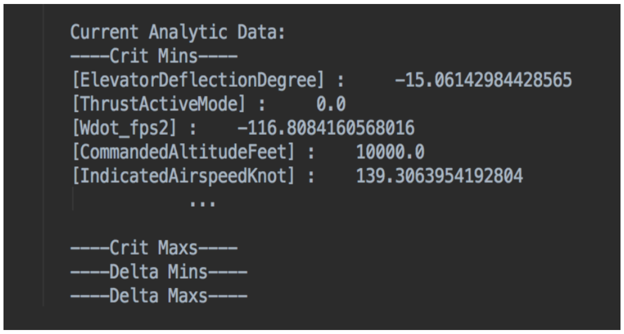
B. Training

The system was first trained with past flight records of Go-Around scenarios conducted in unstable configurations. This training resulted in an initial representation of context and corresponding rules of that context. There were a total of 3 flights used for training the initial context, flown by 3 fliers. We refer to these trails as $dataset\_A$, $dataset\_B$, and $dataset\_C$. Using these datasets, we primed each execution of the Knowledge Acquisition framework with a training schematic. We used either *single* or *multi* training methods, where in *single* cases, our system was trained with only one Go-Around data sample ($dataset\_A$), and in *multi* cases, our system was trained with 3 different data samples ($dataset\_A$, $dataset\_B$, and $dataset\_C$).

Training resulted in an initial representation of the context ”unstable below 1000ft on approach”, including a feasibility region for the 130 data fields, a set of discrete values, and a set of trained DST based rules. Formally, resulting context $C_* = UNSTABLE, BELOW1000FT$ predicts GA as an outcome.

C. Calculation of Confidence Values

At each iteration of the algorithm, the confidence values of the central elements are calculated (CE_{conf}). The confidence value on the context is calculated by using cockpit data to determine a value between 0 and 1 representing how likely it is, based on cockpit evidence, that the discrete features of the assumed context currently hold. For example, using altitude data and glide slope data coming in from the cockpit, a value for each discrete feature is generated, and the mean of the confidence of all discrete features in the assumed context holding true is generated. A similar process holds for the calculation



```
Current Analytic Data:
----Crit Mins----
[ElevatorDeflectionDegree] :     -15.06142984428565
[ThrustActiveMode] :     0.0
[Wdot_fps2] :    -116.8084160568016
[CommandedAltitudeFeet] :    10000.0
[IndicatedAirspeedKnot] :    139.3063954192804
            ...

----Crit Maxs----
----Delta Mins----
----Delta Maxs----
```

Fig. 3. Feasibility Region associated with a given context

of the observed outcome. However, for the calculation of the confidence of a prediction value, the algorithm instead uses DST intervals to extract out a confidence measure.

Example Suppose we have a context $C_* = \{UNSTABLE, BELOW1000FT\}$ with associated DST rules $C_* \implies GA_{[0.7,0.9]}$ and $C_* \implies L_{[0.1,0.3]}$. Suppose we also observed that the aircraft is at altitude *alt* and descending. We proceed to calculate the confidence values for the central elements of C_* in the following manner:

$$C_{*conf} = \frac{UNSTABLE_{conf} + BELOW1000FT_{conf}}{2} \tag{11}$$

$$P_{*conf} = max(\frac{Bel(GA) + Pl(GA)}{2 \cdot Un(GA)}, \frac{Bel(L) + Pl(L)}{2 \cdot Un(L)}) \tag{12}$$

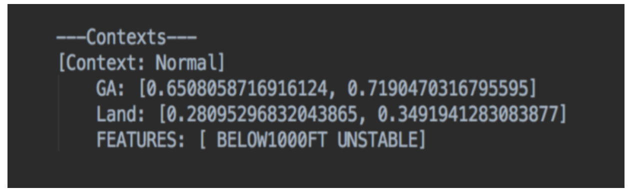
$$O_{*conf} = \begin{cases} \frac{|alt-1000|}{100} & 900 \leq alt \leq 1100 \\ 0 & alt \geq 1100 \\ 1 & alt \leq 900 \end{cases} \tag{13}$$

where $UNSTABLE_{conf}$ and $BELOW1000FT_{conf}$ are extracted from altitude and glide slope data, and *alt* is the altitude of the aircraft at the time of calculation of the confidence values.

D. Data Representations

Data Point At the most primitive level, we represent each *raw data point* as a list of 130 data fields corresponding to the information available in the cockpit. These data points are used by the framework to build representation of context, and to calculate confidence values of the central elements. Data included information on aircraft state, the state of the devices within the aircraft, weather information, etc.

Context We represent a given context *C* with three parts; *rules*, *discrete data*, and *continuous data*. The rules are a set of DST hypotheses in the form of implications which express the outcomes of being in a context. For example, $C_* \implies GA_{[0.7,0.9]}$ and $C_* \implies L_{[0.1,0.3]}$. The DST intervals are generated in the initial training step. The discrete data within a context consists of a set of features, predefined by subject matter experts, that contains a subset of the elements $\{ABOVE1000FT$, $BELOW1000FT$, $UNSTABLE$, $STABLE\}$. The selection of discrete features happens through analysis of aircraft altitude and glide slope. *ABOVE1000FT*, *BELOW1000FT* are mutually exclusive, and *UNSTABLE*, *STABLE* are mutually exclusive. Additionally, the heuristics used by the framework to find salient features at the time of anomaly occurrence are able to discover discrete data that is not predefined, but rather found through analysis of cockpit data. Lastly, the continuous data of a context is generated by populating the context with a feasibility region, which is learned through training on past flight data.



```
---Contexts---
[Context: Normal]
    GA: [0.6508058716916124, 0.7190470316795595]
    Land: [0.28095296832043865, 0.3491941283083877]
    FEATURES: [ BELOW1000FT UNSTABLE]
```

Fig. 4. Normal Context with associated rules

VI. Results

We tested our system for its ability to successfully detect an anomalous context at the point in time where it occurred, and for its accuracy in explaining the reason for the occurrence. In our trial runs, we adjust the fuel weight of the system to be at a lowered level to simulate a condition for an emergency landing. In order to quantify "explainability," we looked at the number of anomalous factors that were detected at the time of the anomaly, and the accuracy of the factors. A perfect set of explanatory factors was 4 (*BELOW1000FT*, *UNSTABLE*, *FUELWEIGHTLB_BELOW_CRIT*, *UPDATECOUNTER_ABOVE_CRIT*), but a successful run is only required to have at least these 4 factors. That is, it was successful in the case that it had more than 4 features, potentially erroneous, as long as the set of four were detected. This characterizes systems ability to detect the original discrete contextual features ($BELOW1000FT$, $UNSTABLE$) to ensure that it was able to accurately detect context, along with the additional features discovered through analysis of the feasibility region ($FUELWEIGHTLB\_BELOW\_CRIT$, $UPDATECOUNTER\_ABOVE\_CRIT$).

A. Variance in Timing of the Anomaly

Each time step ranged from 744 ms to 766 ms with an average of 747.7 ms per time step. The first 12 simulations were run at 100 time steps per run, and the last 10 were run at 120 time steps per run (overall average of 109.1 time steps, or 81567.3 ms). The extended time was to allow for the fliers to take their time when dropping below the 1000ft marker, to see how the timing of the anomaly affected results. We found that the timing of the anomaly had no effect on the system's ability to successfully detect anomalies, or on its ability to explain the anomaly (the systems' ability to detect the low fuel level).

```
---Contexts---
[Context: Normal]
    GA: [0.6590191350054148, 0.7170576490424871]
    Land: [0.2829423509575121, 0.34098086499458424]
    FEATURES: [ BELOW1000FT UNSTABLE]
[Context: Anomaly_82]
    GA: [0.0, 0.13157894736842102]
    Land: [0.8684210526315785, 1.0]
    FEATURES: [ BELOW1000FT
                UNSTABLE
                FUELWEIGHTLB_BELOW_CRIT_10133
                UPDATECOUNTER_ABOVE_CRIT_810992]
```

Fig. 5. Contextual representations with corresponding rules

B. Variance in Training

We tested our system against variance in training data. We used either *single* or *multi* training methods, where in *single* cases, our system was trained with only one Go-Around data sample, and in *multi* cases, our system was trained with 3 different data samples. We tested 4 different single case trials, and found that in 3 of the trials, the system had a perfect explanatory detection, and in 1 trial, the system picked up 6 additional factors. We attribute this to the variance in flight trajectory presented in the the 3 successful trials, which included ”rolls” and ”zig zags.” For every trial flown, we tested the system with both *single* and *multi* schemes. We found that the *single* training cases found an average of 11.31 newly discovered discrete features (explanatory factors), whereas *multi* training schemes had an average of 5.23 newly discovered discrete features, with an overall of 2.35 times more features found in the *single* training case versus *multi*. By adding two additional training samples, the system's accuracy was increased by 116%.

C. Variance in Fliers

Our 24 trials were flown with three participants, none of whom had past flight experience. The three participants had 3 levels of flying experience with the VISTAS simulator. Within these trials, we varied flight trajectory, with some flights flying in an unstable configuration below the glide slope, and others flying above the glide slope. Despite these differences, the system was able to detect the anomaly at the time of its occurrence in every trial.

VII. Conclusion and Future Work

We created a system that was able to detect and reason about anomalies in an aviation context in a real-time, human-in-the-loop proof of concept implementation. The success of the system was invariant across changes in participants, training, time of the anomaly occurrence, and flight trajectory or training and system testing. The main limitation of our framework is that the feasibility region heuristic for anomaly detection is limited, and its single dimensionality may not be sufficient for all anomaly cases. Some anomalies may only be realized when considering a multifaceted combination of factors. These cases were not considered in our proof of concept. In future work, we plan on extending our framework to handle more complex aviation domain applications. We would also like to explore the potential for integrating human factors heuristics for discovering the explanatory features of the anomalous context, like psycho-physiological data measures from the pilot (ex. eye trackers), or natural language input from the flight crew.

Acknowledgment

This work was funded by the National Institute of Aerospace through Crew Systems and Aviation Operations branch at NASA Langley Research Center, with additional contribution from the team members of the Big Data and Machine Intelligence Capability at NASA Langley Research Center.

References

[1] T. Shon, and J. Moon, ”A hybrid machine learning approach to network anomaly detection.” Information Sciences, vol 177, no. 18, 2007, pp.3799-3821.

[2] A. Srivastava, A. Kundu, S. Sural, and A. Majumdar, ”Credit card fraud detection using hidden Markov model.” IEEE Transactions on dependable and secure computing, vol. 5, no. 1, 2008, pp.37–48.

[3] V. Mahadevan, W. Li, V. Bhalodia, and N. Vasconcelos, ”Anomaly detection in crowded scenes.” IEEE Conference on Computer Vision and Pattern Recognition, June 2010, pp. 1975–1981. IEEE.

[4] V. Chandola, A. Banerjee, and V. Kumar, ”Anomaly detection: A survey.” ACM computing surveys (CSUR), vol. 41, no. 3, 2009, p.15.

[5] M. Sharma, K. Das, M. Bilgic, B. Matthews, D. Nielsen, and N. Oza, Active learning with rationales for identifying operationally significant anomalies in aviation.” In Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Sept 2016, pp. 209–225. Springer, Cham.

[6] G. Shafer, ”A mathematical theory of evidence” vol. 42, 1976. Princeton university press.

[7] T. Williams, and M. Scheutz, ”Resolution of referential ambiguity using Dempster-Shafer theoretic pragmatics.” In AAAI Fall Symposium on AI and HRI, Sept 2016.

[8] U. K. Rakowsky, ”Fundamentals of the Dempster-Shafer theory and its applications to reliability modeling.” International Journal of Reliability, Quality and Safety Engineering, vol. 14, no. 06, 2007, pp.579–601.

[9] D. A. Dewasurendra, P. H. Bauer, & K. Premaratne, ”Evidence filtering.” IEEE Transactions on Signal Processing, vol. 55, no. 12, 2007, pp.5796–5805.

[10] E. A. Krause, M. Zillich, T. E. Williams, and M. Scheutz, ”Learning to Recognize Novel Objects in One Shot through Human-Robot Interactions in Natural Language Dialogues.” In AAAI, July 2014, pp.2796–2802.

[11] L. Fe-Fei, ”A Bayesian approach to unsupervised one-shot learning of object categories.” Proceedings of the Ninth IEEE International Conference on Computer Vision, October 2003, pp. 1134–1141. IEEE.

[12] N. J. Napoli, L. E. Barnes, and K. Premaratne, ”Correlation coefficient based template matching: Accounting for uncertainty in selecting the winner.” In 18th International Conference on Information Fusion, July 2015, pp.311–318. IEEE.

[13] R. Agrawal, T. Imieliski, and A. Swami, ”Mining association rules between sets of items in large databases.” In Acm sigmod record, vol. 22, no. 2, June 1993, pp.207–216. ACM.

[14] R. Fagin, and J. Y. Halpern, ”A new approach to updating beliefs.” arXiv preprint arXiv:1304.1119. 2013.

[15] C. M. Kuok, A. Fu, and M. H. Wong, ”Mining fuzzy association rules in databases.” ACM Sigmod Record, vol. 27, no. 1, 1998, pp.41–46.

[16] A. L. Buczak, and E. Guven, ”A survey of data mining and machine learning methods for cyber security intrusion detection.” IEEE Communications Surveys & Tutorials, vol. 18, no. 2, 2016, pp.1153–1176.

[17] S. Agrawal, and J. Agrawal, ”Survey on anomaly detection using data mining techniques.” Procedia Computer Science, vol. 60, 2015, pp.708–713.

[18] E. Gizzi, L. Le Vie, M. Scheutz, V. Sarathy, and J. Sinapov, ”A Generalized Framework for Detecting Anomalies in Real-Time using Contextual Information.” Tenth International Workshop for Modelling and Reasoning in Context, 2018.